Cross-view Learning with Limited Supervision

Yao-Hung Hubert Tsai

November 2021
CMU-ML-21-113

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Ruslan Salakhutdinov, Co-Chair
Louis-Philippe Morency, Co-Chair
Barnabás Póczos
Jimmy Ba (University of Toronto)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was supported by: Air Force Research Laboratory award FA875016C0130001, Defense Advanced Research Projects Agency award FA875018C0150, Department of the Interior award D17AP00001, Office of Naval Research awards N000141613074 and N000141812861, and a graduate fellowship from Facebook.

Keywords: Cross-view Learning, Multi-view Learning, Self-supervised Learning, Unsupervised Learning, Weakly-supervised Learning, Mutual Information Estimation, Conditional Mutual Information Estimation, Relationship Modeling

To my family

Abstract

Real-world data is often multi-view, with each view representing a different perspective of the data. These views can be different modalities, different sets of features or different viewpoints. For instance, human communication contains heterogeneous sources of information (views as different modalities) spanning tones of voice, facial gestures and spoken word. As another example, autonomous systems collect features from various sensors, such as LiDAR, RADAR and RGB signals (views as different sets of features). As the third example, surveillance cameras record scenes from multiple angles (views as different viewpoints). Learning representations from multi-view data, dubbed *cross-view learning*, requires modeling the complementarity within and understanding the relationships across views, such as knowing the information shared among different views and the information in a particular view. This process is challenging due to the heterogeneity of data and complex structures that link the different views (e.g., asynchrony between views). In this thesis, we study cross-view learning in scenarios where label supervision is not available for downstream tasks, but we have pairing information between views (i.e., limited supervision). We focus on these scenarios since they are close to reality in many fields, where collecting a large number of labels tends to be expensive, both computationally and effort-wise. To address this significant challenge of cross-view learning with limited supervision, we scaffold it in three core technical challenges.

The first challenge, which we refer to as *cross-view heterogeneous structures*, focuses on learning to align and synchronize different views and disentangling complementary factors from multi-view data. For the sake of simplicity, the first challenge is made under a fully supervised setup. Then, we note another important aspect of modeling the complementarity among views is quantifying the cross-view relationships within the views. This leads us to discuss the second challenge: *relationship quantification*. We focus on quantifying the relationship via mutual information, studying tractable and scalable estimators for it. Last, we discuss the third challenge: *learning with limited supervision*. We transit from the supervised to the unsupervised setting, where the only information comes from pairs between views, but without labels for the downstream task. We present how to learn good representations from multi-view data by considering the complementarity across views, when labels or downstream supervision is not available. Within the learning with limited supervision challenge, we may sometimes have access to additional information, more than just the data itself. The additional information can be auxiliary or undesirable information of data. For instance, the auxiliary information can be the hashtags for Instagram images, and the undesirable information can be the personal information from physiological data. We show how to either leverage the auxiliary information to learn better representations, or remove the undesirable information in the representations. The thesis discussed our contributions to all three challenges.

This thesis opens up many avenues for future research directions. One of these directions is to scale multi-view representation learning methods up to plenty of views, such as learning representations from signals for aircraft sensors that track oil temperature, fuel pressure, airspeed measurement, lightning detection, vibration detection, etc. Next, since most theoretical analyses on self-supervised learning lie mainly within visual modality, another direction is establishing theoretical bases for self-supervised learning beyond the visual modality, such as the textual and acoustic modality. Last, most existing multi-view learning literature focuses primarily on perception and less on action generation (e.g., action generation for navigation). Hence, a future direction is multi-view representation learning for action generation.

Acknowledgements

I am immensely grateful to my advisors, Ruslan Salakhutdinov and Louis-Philippe Morency. Russ is extremely easy to work with, offers me complete freedom for my research direction, and provides unconditional support for anything I need during my Ph.D. LP is always excited about my research, offers detailed guidance throughout each of my projects, and helps me overcome the challenges I have encountered. Russ and LP both have remarkable qualities that I have learned from, which made me realize how fortunate I can have them to be my advisors!

I am also grateful to my internship hosts, Makoto Yamada at Kokuritsu Kenkyū Kaihatsu Hōjin Rikagaku Kenkyūsho for Advanced Intelligence Project (RIKEN AIP), Nebojsa at Microsoft Research (MSR), Santosh Kumar Divvala and Ali Farhadi at Allen Institute for Artificial Intelligence (AI2), Nitish Srivastava at Apple AI Research (AIR), and Abdelrahman Mohamed at Facebook AI Research (FAIR). Makoto is always passionate about research and is my very good friend. I remember that we even discussed a research problem on a tissue paper in a ramen bar! Nebojsa has a happy life and family, and I learned a lot from him about the importance of maintaining a good work-life balance. Santosh and Ali are such great mentors who create a fun research environment so that everyone feels comfortable discussing research topics. This environment stimulates highly efficient discussions and leads to fruitful results of my intern project. Nitish is very knowledgeable and always provides insightful comments during our discussions. He has provided me the clear guidance that I need during the internship. Abdo is a perfect mentor that sets up rigorous meeting schedules, encourages plentiful discussions between various researchers and engineers, and offers me lots of valuable suggestions on job markets. Without these amazing researchers, I would definitely not able to accomplish my achievements!

I also like to thank Barnabas Poczos and Jimmy Ba for being my thesis committee members. Feedbacks from and discussions with Barnabas and Jimmy let me retrospect my research so that I can find insufficiency about it and make appropriate adjustments. Without a doubt, the help from Barnabas and Jimmy has significantly influenced how I think about and shape my research.

In addition to my advisors, internship hosts, and my committee members, I am also lucky to have the chance to work with excellent collaborators inside and outside CMU: Han Zhao, Shaojie Bai, Devendra Singh Chaplot, Paul Pu Liang, Zhilin Yang, Jian Zhang, Hanlin Goh, Charlie Tang, Liyuan Lucas Liu, Wei-Ning Hsu, Benjamin Bolte, Amir Zadeh, Liang-Kang Huang, Denny Wu, Ziyin Liu, Muqiao Yang, Martin Q. Ma, Tianqin Li, Yue Wu, Shangda Li, Zico Kolter, Geoff Gordon, Kenji Fukumizu, and Masashi Sugiyama. The dissertation results from a series of help and guidance throughout various research projects, and many of these collaborators are my close friends, which makes me enjoy much when working on research.

The Machine Learning Department at CMU is the best place I can think of for pursuing a Ph.D. degree. The very first person I have to thank is Diane. Diane has answered lots of questions for me, and I really appreciate Diane for her patience and generous help. I am also grateful to meet the following awesome colleagues: Jason, Fan, Chenghui, Yifan, Xun, Tom, Emilio, Ojash, Robin, Lisa, Mu-chu, Jennifer, Simon, Yangyi, Arun, Ben, Adarsh, Will, Leqi, Sebastian, Sid, Amanda, Biswa, Nicholay, Kartik, Tiffany, Shrimai, Chirag, Theo, Po-Wei, Haitian, Chieh, Fish, Chun-Liang, Bingbin, Tim, Vivek, Ezra, Quanbin, Jing, Hengyuan, Rui, Yijie, Chris, Ean, Ritesh, Otilia, Zhiting, Adams, Yichong, Anthony, Wei, Manzil, Chaitanya, Volkan, Zhun, and Ying. The time I spent with these amazing people makes all these years more enjoyable!

Last, I would like to thank my parents and sisters for their endless love, support, and encouragement. I am also grateful to who I've loved or loved me. Because of them, I have an excited and memorable journey during my Ph.D. study, and the journey makes me grow stronger into a more reliable person.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In real world scenario, data contains multiple views, with each view representing a different perspective of the data. These views can be different modalities, different sets of features or different viewpoints. As a first example seeing views as different modalities, human communication contains heterogeneous sources of modalities spanning tones of voice, facial gestures and spoken words. As a second example seeing views as different sets of features, autonomous systems collect features from various sensors including LiDAR, RADAR and RGB signals. As a third example seeing views as different viewpoints, surveillance cameras take photos from multiple angles of a scene. The fact that the data contains multiple views is dubbed the multi-view property of the data, and the data is known as the multi-view data. Although at a first glance, more views provide more information of the data, yet the *heterogeneity* exists across views pose the difficulty to study multi-view data. As an example, the heterogeneity can come from variable sampling rates, different information present or disparate modalities of distinct views. Modeling the heterogeneity across views is a fundamental problem to better understanding the multi-view data.

In this thesis we study *cross-view learning*, a computational process to analyze and integrate high-dimensional and heterogeneously-structured multi-view data into intermediate representations. These representations are essential building blocks of modern computational models. At the core of cross-view learning is the challenge of modeling the internal structure of each view while understanding the relationship across views, specifically the information shared among different views and the information unique to a particular view. An illustrative example of cross-view learning could be sarcasm prediction. Sarcasm is often expressing through multiple modalities, including facial gestures, spoken words and acoustic modality tones of voice. Sarcasm can sometime be expressed when the tones of voice and the facial gestures convey similar messages (messages that imply a negative sentiment) while the spoken words convey opposite messages (messages that imply a positive sentiment). Sarcasm exemplify the diverse and complex relationships that can exist between the heterogeneous views.

Learning representations that discover the full potential from the multi-view data depends very often on the quantity and quality of labels used to train the model. As we know, it is often computationally expensive or sometimes infeasible to collect a large amount of labeled data, especially in real-world scenarios. A key aspect of this thesis is that we are studying cross-view learning in the context of having only limited supervision from data. Precisely, in the traditional supervised learning setup, the downstream labels act as the supervision signals to learn representations from our computational models. In contrast, this thesis tackles the problem of learning with limited supervision, when downstream labels are not readily available. Before studying these limited supervision scenarios, we also study some fundamental aspects of multi-view learning in the supervised setting (more details below). We discuss three important types of problem: learning with only the information of pairing between views (e.g., audio and video streams temporally

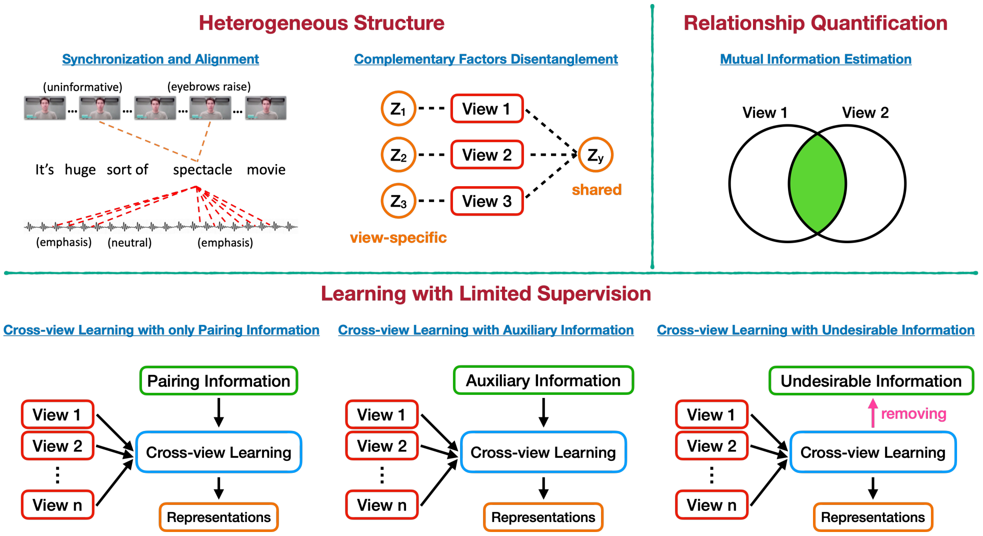


Figure 1.1: Challenges and the corresponding sub-challenges studied in this thesis.

synchronized), learning with auxiliary information (e.g., the hashtags for Instagram images), and learning with the presence of undesirable information (e.g., the privacy-related information such as the personal information of physiological data).

1.1 Challenges

We scaffold the problem of **Cross-view Learning with Limited Supervision** into three challenges: **Heterogeneous Structure**, **Relationship Quantification** and **Learning with Limited Supervision**. We first discuss the challenge of the heterogeneous structure across views, by focusing on disentangling complementary factors from multi-view data and learning to synchronize and align different views. For the sake of simplicity, the first challenge will be made under the traditional supervised setting. Next, since the cross-view learning includes the crucial step of understanding the relationship across views, we also want to know how to quantify this relationship. This leads us to discuss the challenge of relationship quantification, and we focus on quantifying the mutual information (i.e., the statistical relationship) across views via tractable and scalable methods. Lastly, we transit to the setup when having no access to the downstream labels, aka limited supervision. Our goal is to still learn good representations from multi-view data under this challenging scenario.

Heterogeneous Structure. Understanding how to computationally model the heterogeneous structures across views is the first step of learning good representations from multi-view data. The heterogeneity in multi-view data often comes from different formats or patterns across views. An important hypothesis in many multi-view problem is that there exist complementary information across views. Let's take the example of multimodal sentiment, where the views come from multiple modalities that include information such as tones of voice in the acoustic modality, facial attributes in the visual modality and spoken words in the language modality. First, we see that signals from different views may be unaligned due to variable receiving frequency of the receptor from each modality (e.g., the audio signal is captured at 100Hz and the video frame rate is 60Hz), which is referred to as the heterogeneous patterns across views. Additionally, a frowning face (from visual modality) may relate to a pessimistic words (from textual modality) spoken earlier. Hence, the multi-view data often are unaligned and require inferring long-term dependencies across views. We identify this as the **Synchronization and Alignment** sub-challenge.

Next, we can understand the complementary information across views by studying different explanatory factors that are unique to a view or shared across views. If looking at a multi-view example that expresses positive sentiment with a happy face (the first view) and a sentence "this is an awesome movie" (the second view), both views contain the factors for inferring the positive sentiment. Then, these factors are shared across views and represent the information jointly exist among views (i.e., multi-view factors). However, each view contains some information that is unique to it (i.e., view-specific factors). On the same example, by altering view-specific factors, we can alter each view while maintaining the same information that infers the positive sentiment, i.e., changing the view-specific factors for text will make the words "this is an awesome movie" to "the movie is actually quite good". Disentangling explanatory factors from the multi-view data enables us to understand the complementarity across views and hence can learn better representations. We identify this as the **Complementary Factors Disentanglement** sub-challenge.

Relationship Quantification. As pointed out in the prior challenge, cross-view learning requires modeling the heterogeneous structures across views, where distinct view has different information. Hence, we like to know how to quantify the information across views, so that we can get a better understanding of the cross-view relationships. For instance, for the human multi-modal language, if the tones of voice and the facial gestures from human communications are highly correlated (e.g., express either positive or negative sentiment simultaneously), then the quantified relationship is high; if they are weakly correlated (e.g., the sentiments expressed from the two views have low coincidence), then their quantified relationship is low. In this thesis we study the relationship quantification problem by measuring mutual information, which is a well known concept that represents the statistical relationship between two entities [Cover, 1999]. Nonetheless, estimating the mutual information is notoriously hard, especially when we want to perform the estimation on high-dimensional continuous data, such as images, text, audio streams, etc. In other words, developing tractable (i.e., efficient) and scalable (i.e., can work on high-dimensional data) mutual information estimators is the key to good relationship quantification. We identify this as the **Mutual Information Estimation** sub-challenge.

Learning with Limited Supervision. After discussing the challenges of the heterogeneous structure and the relationship quantification in cross-view learning, we are ready to discuss the challenge of learning with limited supervision. In particular, we want to study representation learning from multi-view data without using the labels of downstream tasks, but leveraging only the pairing information between views. Compared to the traditional supervised learning that requires both high-quantity and high-quality downstream labels, our setup is more flexible since the pairing information is more easily accessible. In the context of cross-view learning, we identify this as a the **Cross-view Learning with only Pairing Information** sub-challenge. While downstream task labels are expensive, in some cases, data comes with *weak* supervision signals such as the grouping or clustering (potentially hierarchically) of the data. For example, images on Instagram come with the hashtags, and these hashtags can be seen as a form of the weak supervision. These weak supervision signals may not directly related to the downstream labels, yet it is possible that they may help us learn better representations. The weak supervision signals can be seen as the auxiliary information of the data, and we identify this research topic as the **Cross-view Learning with Auxiliary Information** sub-challenge. Last, we note that data contains sometimes information that may be undesirable for downstream tasks. For instance, gender information may lead to biased decisions on many gender-agnostic tasks. We want to remove undesirable information in our learned representations, and ensure the representations could still perform well on the downstream tasks. We identify this as the **Cross-view Learning with Undesirable Information** sub-challenge.

1.2 Contributions

This thesis addressed the challenges and sub-challenges in the previous section. In this section, we provide a highlight of our main thesis contributions and how they relate to the challenges and sub-challenges for cross-view learning with limited supervision.

1. **Heterogeneous Structure - Synchronization and Alignment (Chapter 3)**
   (a) **(Synchronization and Alignment)** We introduce the Multimodal Transformer [Tsai et al., 2019a] to generically address the issues of cross-view alignment and long-range dependency. This approach has the advantage that the model can be trained in an end-to-end manner without aligning the data in advance.

   (b) **(Latent Correlation)** At the heart of our Multimodal Transformer is the directional pairwise cross-view attention, which attends to interactions between views across distinct time steps and latently correlates the cross-view signals. The latent correlation implicitly relates signals across views, such as relates a frowning face to pessimistic voice.

2. **Heterogeneous Structure - Complementary Factors Disentanglement (Chapter 4)**
   (a) **(Factors Disentanglement)** We propose a method to disentangle independent factors of variation in multi-view data, via a hybrid generative-discriminative model [Tsai et al., 2019d]. The factors are multi-view discriminative factors and view-specific generative factors. Multi-view discriminative factors are shared across all views and contain joint view features required for discriminative tasks such as prediction and regression. View-specific generative factors are unique for each view and contain the information required for generating data.

   (b) **(Generation)** Our model demonstrates flexible generative capabilities by conditioning on independent factors and can reconstruct missing modalities without significantly impacting performance.

   (c) **(Interpretation)** We also devise methods to interpret these independent factors from the multi-view data that influence the dynamics of multi-view prediction and generation. The devised interpretation methods represent both overall trends (aka global interpretation) and fine-grained analysis (aka local interpretation) on understanding multi-view representation learning.

3. **Relationship Quantification - Mutual Information Estimation (Chapter 5)**
   (a) **(Tractable and Scalable Estimators)** We propose efficient estimators for mutual information on high-dimensional data, using neural networks via gradient descent optimization [Tsai et al., 2020d]. One of the proposed estimators casts the mutual information estimation problem into class-posterior classification problem, which can be efficiently optimized using existing deep learning optimization tools. The other proposed estimator contains no logarithm or exponential during optimization and has good numerical stability in practice.

   (b) **(Plugging-in Estimation)** Instead of directly optimizing mutual information bounds, we suggest to first estimate the point-wise dependency and then plugging-in the estimated point-wise dependency to estimate the mutual information. Empirically, we show this mutual information estimation has low bias and variance. Theoretically, the estimated mutual information converges to the true mutual information at rate $\sqrt{\frac{1}{n}}$ with n being the number of samples.

   (c) **(Instance and Population Level)** We study both instance- and population-level estimation of the mutual information, which gives us both fine-grained and average understanding of the dependencies between views.

4. **Learning with Limited Supervision - Cross-view Learning with only Pairing Information (Chapter 6)**

   (a) **(Complementarity Modeling)** To learn representations from multi-view data without downstream supervision, we propose methods that leverage the complemantarity and the statistical relationships among multi-view data [Tsai et al., 2020c]. These methods include a wide range of prior work in unsupervised representation learning and pave a large space of composing unsupervised representation learning objectives.

   (b) **(Goodness of the Representations)** Under an information-theoretical perspective, we show that, under the assumption that view-specific information contains less information about the downstream task, the presented methods are able to learn representations that are almost as good as the supervised learned representations.

   (c) **(Generalization Error)** We provide both theoretical and empirical supports of the above claim in terms of the generalization error, such as Bayes error rates and test generalization error, on the unsupervised learned presentations.

5. **Learning with Limited Supervision - Cross-view Learning with Auxiliary Information (Chapter 7)**

   (a) **(Auxiliary Information Integration)** We show how to integrate the auxiliary information (e.g., additional attributes for data such as the hashtags for Instagram images) in a self-supervised learning process [Tsai et al., 2021b]. Specifically, we introduce the Clustering InfoNCE (Cl-InfoNCE) objective that learns similar representations for data sharing similar auxiliary information and vice versa.

   (b) **(Structural Information Modeling)** The core of the presented Cl-InfoNCE method is its ability to leverage the data structural information. In particular, under the weakly-supervised setting, Cl-InfoNCE uses the structural information suggested by the auxiliary information. We show that Cl-InfoNCE can also work under the unsupervised setting, where Cl-InfoNCE uses the unsupervised constructed clusters (e.g., k-means clusters).

   (c) **(Goodness of Representations)** We connect the goodness of the learned representations with the statistical relationships: i) the mutual information between the labels and the data structures used in Cl-InfoNCE and ii) the conditional entropy of the data structures given the labels.

6. **Learning with Limited Supervision - Cross-view Learning with Undesirable Information (Chapter 8)**

   (a) **(Undesirable Information Removal)** We show how to remove the undesirable information (e.g., the gender information for gender-irrelevant tasks) in the self-supervised learning process [Tsai et al., 2021d]. In particular, we introduce Conditional InfoNCE (C-InfoNCE) and Weak-Conditional InfoNCE (WeaC-InfoNCE) that remove the effect of variations of the undesirable variable by conditioning on its values. Since the variations are fixed, the effect of the variable will not be accounted for in the learned representations.

   (b) **(Conditional Contrastive Learning)** C-InfoNCE and WeaC-InfoNCE belong to the family of conditional contrastive learning approaches that learn similar representations for conditionally-correlated data pairs and dissimilar representations for conditionally-unrelated data pairs. WeaC-InfoNCE is a more computationally efficient variant of C-InfoNCE.

   (c) **(Conditional Mutual Information Estimation)** We also show that WeaC-InfoNCE and C-InfoNCE are lower bounds of the conditional mutual information, and hence both of the apporaches can be used to estimate the conditional mutual information between two variables.

1.3 Other Contributions

In this section, we highlight other contributions that happened during the graduate study but are not clearly introduced in the thesis.

1. **Learning Visual-Semantic Representations** [Tsai and Salakhutdinov, 2017, Tsai et al., 2017a]
   (a) **(Robustness to Label Supervision)** We present to combine supervised and unsupervised learning techniques when learning representations from visual-textual data [Tsai et al., 2017a]. The representations can benefit from using unlabeled data and are robust even when having only a small number of labeled data.

   (b) **(Zero/Few-Shot Learning)** The applications are mainly on zero-shot, one-shot and few-shot visual-textual representation learning [Tsai and Salakhutdinov, 2017, Tsai et al., 2017a], from inductive to transductive setting.

   (c) **(Visual-Textual Domain Minimization Representations)** We find minimizing the distribution divergence between visual and textual domain [Tsai et al., 2017a] enables us to learn better representations.

   (d) **(Visual-Textual Dependency Maximization Representations)** We find maximizing the dependency between visual and textual features [Tsai and Salakhutdinov, 2017] also helps learn better representations.

2. **Temporal Order Discovery** [Tsai et al., 2017b]
   (a) **(Unsupervised Order Discovery)** We present to extract the order of data instances in an unsupervised way [Tsai et al., 2017b]. We assume the instances are sampled from a Markov chain, and we present to learn the transitional operator of the underlying Markov chain, as well as the order by maximizing the generation probability under all possible data permutations.

   (b) **(Space Complexity Amortization)** We use neural network as a compact and soft lookup table to approximate the possibly huge, but discrete transition matrix in the Markov chain [Tsai et al., 2017b]. This strategy allows us to amortize the space complexity with a single model.

   (c) **(Linear-time Approximation)** We propose a greedy batch-wise permutation scheme ($O(n)$ time complexity with n being the number of samples) to approximate the full permutation ($O(n!)$ time complexity) [Tsai et al., 2017b].

3. **Video Common Sense Reasoning** [Tsai et al., 2019c]
   (a) **(Video Relationship Modeling)** We study the visual relationships between object, predicate and subject in videos [Tsai et al., 2019c]. We design models to study relational entities spatially and temporally.

   (b) **(Fully-connected Spatio-temporal Graph)** We construct a Conditional Random Field on a fully-connected spatio-temporal graph [Tsai et al., 2019c] that exploits the statistical dependency between relational entities in videos.

   (c) **(Observation-adaptive Relation)** We parametrize the pair-wise energy function in the fully-connected graph with the parametrization conditioned on visual observations [Tsai et al., 2019c]. Then, the relations among entities are adaptive to visual observations.

4. **Attention Mechanism** [Tsai et al., 2019b]

    (a) **(Kernel Formulation)** We present a new formulation of attentional mechanism in Transformer via the lens of the kernel [Tsai et al., 2019b]. This new formulation gives us a better way to understand individual components of the Transformer's attention, such as the better way to integrate the positional embedding.

    (b) **(Larger Space of Composing Attention)** We also pave the way to a larger space of composing Transformer's attention [Tsai et al., 2019b]. For example, we propose a new variant of Transformer's attention which models the input as a product of symmetric kernels. This approach achieves competitive performance to the current state of the art model with less computation.

5. **Deep Neural Networks Regularization** [Tsai et al., 2019e]

    (a) **(Approximate Empirical Bayes Regularization)** We propose an adaptive and data-dependent regularization on deep neural networks [Tsai et al., 2019e] motivated by the empirical Bayes method.

    (b) **(Neurons Statistical Correlation)** We propose a data-dependent prior on weights, which captures the correlations in neurons through back-propagation [Tsai et al., 2019e]. The prior encourages neurons to borrow statistical strength from one another.

    (c) **(Robustness to Number of Training Samples)** We show we can learn good representations using the proposed data-dependent regularization even with only a small number of training data [Tsai et al., 2019e].

6. **Routing Mechanism** [Tsai et al., 2020a,b]

    (a) **(Scalability of Capsule Networks)** We introduce a new routing algorithm for Capsule Networks [Tsai et al., 2020b], where the new routing algorithm scales up the usage of Capsule networks to complex real-world datasets. In particular, the performance is at-par with powerful CNNs with much fewer parameters.

    (b) **(Interpretation)** We propose Multimodal Routing [Tsai et al., 2020a], which dynamically adjusts weights between input modalities and output representations differently for each input sample. Multimodal routing can identify relative importance of both individual modalities and cross-modality features. Hence, the weight assignment by routing allows us to interpret modality-prediction relationships not only globally (i.e. general trends over the whole dataset), but also locally for each single input sample, meanwhile keeping competitive performance compared to state-of-the-art methods.

7. **Robust Self-supervised Representation Learning** [Tsai et al., 2021c]

    (a) **(Challenges for Contrastive Learning Objectives)** We identify the three challenges when modeling the contrastive learning objectives: training stability, sensitivity to minibatch size, and downstream task performance. Then, we propose Relative Predictive Coding (RPC) [Tsai et al., 2021c], that achieves a good balance among the three challenges.

    (b) **(Robust Contrastive Learning Objective)** The presented RPC introduces the relative parameters to regularize the objective for boundedness and low variance. Additionally, RPC contains no logarithm and exponential functions, which are the main cause of training instability in prior contrastive objectives.

8. **Negative-samples-free Self-supervised Rerpesentation Learning [Tsai et al., 2021a]**

    (a) **(Negative-samples-free Contrastive Objectives)** We show that the Barlow Twin's method [Zbontar et al., 2021], a recent self-supervised learning method, is an instance of contrastive learning approach that requires no construction of negatively-paired samples. We further manifest that avoiding the need to construct the negative samples improves the training stability of the approach, getting rid of the special cares of the network designs, and increases the robustness to the training batch size.

(b) **(Hilbert-Schmidt Independence Criterion for Self-supervised Learning)** We present a new self-supervised learning objective, named HSIC_SSL [Tsai et al., 2021a], which is inspired by Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2005b]. The presented approach is also an instance of negative-samples-free contrastive objectives.

1.4 Thesis Outline

In this section, we provide an outline of our thesis.

Chapter 2 discusses the technical backgrounds of our thesis. Our discussion focuses on three topics: *multi-view representation learning*, *relationship quantification*, and *self-supervised learning*.

Chapter 3 discusses the sub-challenge *Synchronization and Alignment* within the challenge *Heterogeneous Structure*.

Chapter 4 discusses the sub-challenge *Complementary Factors Disentanglement* within the challenge *Heterogeneous Structure*.

Chapter 5 discusses the sub-challenge *Mutual Information Estimation* within the challenge *Relationship Quantification*.

Chapter 6 discusses the sub-challenge *Cross-view Learning with only Pairing Information* within the challenge *Learning with Limited Supervision*.

Chapter 7 discusses the sub-challenge *Cross-view Learning with Auxiliary Information* within the challenge *Learning with Limited Supervision*.

Chapter 8 discusses the sub-challenge *Cross-view Learning with Undesirable Information* within the challenge *Learning with Limited Supervision*.

Chapter 9 draws conclusion and discusses potential limitations of the thesis. This chapter also delineates future directions for *cross-view learning with limited supervision*.

Chapter 2

Technical Background

In this chapter, we present background information about technical concepts related to the main topics of this thesis, which centered around *Cross-view Learning with Limited Supervision*. We focus on our background discussion on three topics: *multi-view representation learning*, *relationship quantification*, and *self-supervised learning*.

2.1 Multi-view Representation Learning

The first related topic to our thesis is multi-view representation learning (MRL), which aims to learn representations from multi-view data. MRL is a fundamental research problem, as most of the real-world data naturally come with multiple views. As an example, human perception contains visual (i.e., seeing objects), auditory (i.e., hearing sounds), tactile (i.e., feeling texture), gustatory (i.e., tasting flavors), and olfactory (i.e., smelling odors) senses. Different views of the data may convey the same or distinct messages, and hence learning representations from the multi-view data requires exploiting the complementarity and redundancy among modalities, which is particularly challenging. In the following, we provide an overview of multi-view representation learning by studying its challenges. As outlined by prior work [Baltrušaitis et al., 2019, Li et al., 2018, Xu et al., 2013], we study the following five challenges: *representation*, *translation*, *alignment*, *fusion*, and *co-learning*.

Representation. The representation challenge is to study how different forms of representations can be used for multi-view representation learning. In specific, there are two different forms: the joint and the coordinated representations. The joint representation summarizes the information from different views of the data into a single representation space. For instance, Multi-modal Deep Boltzmann machine (Multi-modal DBM) [Srivastava and Salakhutdinov, 2012] is an undirected graphical model with bipartite connections between adjacent layers of hidden units, and it learns to model the joint density over the space of multi-modal inputs. The hidden units in Multi-modal DBM are the form of the joint representation. Instead of summarizing the multi-view information into a single representation space, the coordinate representation separate representations for each modality but coordinate them through a constraint. For example, Canonical Correlation Analysis (CCA) [Thompson, 1984] first applies projection from individual views of the data and then maximizes the correlation (as the constraint) between the projections. Since the joint representation projects multi-view data into a common space, hence it is best suited when all of the views are present during inference. On the other hand, the coordinated representation projects each view of the data into a separate but coordinated space, making it suitable for the scenarios when only a single view is present during inference. Nonetheless, a downside of the coordinated representation is that the coordinating constraint is hard to extend beyond two views. Hence, most of the coordinated representations are limited to performing pairwise, but not higher-order, contextualization.

Translation. The translation challenge within multi-view representation learning addresses how we can translate one view of the data to another. An example of the translation is automatic speech recognition [Yu and Deng, 2016], which aims to translate human speech to text. Another example is image captioning [Xu et al., 2015], which aims to translate the visual view of the data (i.e., the image) to the textual view (i.e., the captions). Therefore, the translation can be understood as an encoding-decoding process, where we first encode the source view and then decode the encoded information to the target view. This encoding-decoding process is particularly challenging, as we often need to deal with very high-dimensional source and target view. In the example of the image captioning, both images and captions are very high-dimensional, hence conventional machine learning techniques may not be able to handle the translation process well. Fortunately, recent advances of deep learning lead to the breakthrough for the translation challenge. In particular, the increasing computational powers (e.g., GPUs and cloud computing resources) and the introduction of large and complex network architectures [Devlin et al., 2018, He et al., 2016] ease the effort of dealing with the high-dimensional source and target views. Nonetheless, a drawback is that training these deep neural networks often requires a large number of the source-target-view-paired data.

Alignment. The alignment challenge within multi-view representation learning identifies the relations between elements among different views of data. For instance, we like to associate elements from the script of a movie (i.e., the textual view) to scenes of its video (i.e., the visual view). Modeling the association requires measuring the similarity between different views (e.g., the semantic similarity between the script and the video) as well as finding the long-range cross-view dependencies (e.g., a frowning face later in the video may relate to the monologue earlier in the script). We can categorize the alignment into two types - implicit and explicit. The implicit alignment is used as an intermediate step for another task, such as the Cross-modal Attentional mechanism in Multi-modal Transformer [Tsai et al., 2019a]. The cross-modal attentional mechanism attends to interactions between multi-modal sequences (e.g., textual and visual view of the human speech) across distinct time steps and latently adapts streams from one view to another. Note that this latent alignment will not be directly used in the downstream tasks (e.g., Multi-modal Transformer considers the sentiment analysis and emotion recognition as the downstream tasks). On the other hand, the explicit alignment refers to the case that the main objective is aligning sub-components of instances from different views, such as aligning recipes to cooking videos. The difficulties within the alignment challenge are 1) alignment between views is expensive to annotate; 2) similarity metrics between views are hard to design; and 3) there may exist multiple possible alignments and not all elements in one view are associated to another view.

Fusion. The fusion challenge within multi-view representation learning defines the process of joining information from multiple views of the data to perform a prediction. The fusion types are model-agnostic and model-based approaches. The model-agnostic approaches are independent of the machine learning algorithms or systems for processing each view of the data, with examples being early- and late-fusion methods. The early-fusion method integrates features immediately after they are extracted, often by simply concatenating the features; the late-fusion method instead ignores the low-level interaction among views and integrates the decisions made by each view, such as weighted-averaging the decisions. The advantage of model agnostic approaches is that it enjoys a simple training pipeline, and can be used for almost any data types. On the other hand, the model-based approaches are designed to cope with multi-view data directly, which address the fusion by the construction of data. Examples are Multiple Kernel Learning

(MKL) [Gönen and Alpaydın, 2011] and multi-view LSTM [Rajagopalan et al., 2016]. MKL extends conventional kernel support vector machines [Schölkopf et al., 2002] by making use of different kernels for different views of the data. Since distinct kernel is used for each view (i.e., view-specific kernel), MKL allows better fusion of data with heterogeneous views (e.g., data with textual and visual view). Multi-view LSTM extends conventional LSTM [Hochreiter and Schmidhuber, 1997] to multi-view setting by explicitly modeling the view-specific and cross-view interactions over time. To conclude, the fusion challenge has been a long-standing research topic in multi-view representation learning, with each method (e.g., model-agnostic or model-based approaches) having its own strengths and weaknesses.

Co-learning. The co-learning challenge within multi-view representation learning studies how we can aid the modeling of one view (usually resource poor) of data by exploiting the knowledge from another view (usually resource rich) of data. An example of the co-learning is Heterogeneous Domain Adaptation [Tsai et al., 2016], which associates the learning tasks (e.g., classification) across different views of data with each view having different types of features (e.g., textual and visual views). In particular, it considers the setting that the source view (e.g., images) has plenty of labeled data, while the target view (e.g., image captions) has only a limited number of labeled data. Then, it hopes to leverage the source information to help better classify the target data. There are two types of co-learning approaches based on their training resources: parallel co-learning and non-parallel co-learning. The parallel co-learning approaches require the pairing between views. An example is Co-training [Blum and Mitchell, 1998] algorithm, which creates more labeled training samples when we have only few labeled samples in a multi-view setting. In particular, it builds weak classifiers for each view to bootstrap each other with labels for the unlabeled instances. By construction, Co-training requires the pairing between views. On the other hand, the non-parallel co-learning approaches do not require the pairing between views. The aforementioned Heterogeneous Domain Adaptation belongs to this co-learning type. To conclude, co-learning defines the process of how one view influence the training of another view.

Connection to Our Contributions. We discussed the five challenges within multi-view representation learning - the *representation*, *translation*, *alignment*, *fusion*, and *co-learning* challenges. Multi-view representation learning is a multi-disciplinary field, and hence we often study multiple challenges at the same time. Many of these challenges are studied in this thesis. In Chapter 3, we study the synchronization and alignment of multi-view data, which connects to the *representation*, *alignment*, and *fusion* challenges. In Chapter 4, we study the complementary factors disentanglement in multi-view data, which connects to the *representation*, *translation*, *fusion*, and *co-learning* challenges. In Chapter 5, we present to associate different views of the same data, which connects to the translation challenges. In Chapters 6, 7, and 8, we study how we can learn better representations from multi-view data by giving only the association between views, which connects to the *representation* and *translation* challenges.

2.2 Relationship Quantification

The second related topic to our thesis is relationship quantification, which aims to quantify the relationships between different views in the multi-view data. In particular, the relationship quantification measures the association between two views, and hence we can understand how one view affect the change to the other view. For example, human multi-modal utterance contains the visual (e.g., facial attributes), acoustic (e.g., tones of the voice), and textual views (e.g., transcribed text), and the relationship quantification allows us to analyze the relationships between different views for interpreting human behaviors. Here, we present an overview of different relationship quantification approaches, by categorizing them into *linear relationship quantification* and *non-linear relationship quantification*.

Linear Relationship Quantification. As the name implies, the linear relationship denotes the relationship in its linear form. Although in most cases, the relationship between two quantitative variables is more complex than the linear form, linear relationship quantification is simple and is easy to interpret. The most used measurement for the linear relationship is Pearson's correlation coefficient, and we will discuss it in the following.

Pearson's Correlation Coefficient. Pearson's correlation coefficient [Hogg et al., 2005], abbreviated as correlation, measures the linear association between two sets of data. With random variables X and Y, the correlation has the form

$$\rho_{XY} = \frac{\mathbb{E}_{XY}[(X-\bar{X})(Y-\bar{Y})]}{\sqrt{\mathbb{E}_X[(X-\bar{X})^2]}\sqrt{\mathbb{E}_Y[(Y-\bar{Y})^2]}} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}} = \frac{\mathrm{Cov}(X,Y)}{\sigma_x\sigma_y},$$

where ρ_{XY} is the correlation, $\bar{X}$ is the mean of the random variable X, $\mathrm{Cov}(X,Y)$ is the covariance between X and Y, $\mathrm{Var}(X)$ is the variance of X, and $\sigma_x = \sqrt{\mathrm{Var}(X)}$ is the standard deviation of X. Now, we discuss several properties of the correlation. First, ρ_{XY} ranges between -1 and $+1$. $\rho_{XY} < 0$ means the two variables are negatively correlated, $\rho_{XY} = 0$ means the two variables are uncorrelated, and $\rho_{XY} > 0$ means the two variables are positively correlated. Second, independence implies $\rho_{XY} = 0$, but $\rho_{XY} = 0$ does not imply independence. It is because the correlation captures only the linear relationship between two variables, and hence even the two variables are dependent, they can still have zero correlation. Third, the correlation only measures the relationships between uni-variate variables. This limitation hinders the usage of the correlation to multi-variate data, such as images, audio signals, and texts. To conclude, the correlation is easy to compute and has been widely used to statistically interpret and analyze the relationships between variables, with applications in linear regression analysis [Seber and Lee, 2012], hypothesis testing [Wasserman, 2013], algorithmic prediction interpretability [Molnar, 2020].

Non-linear Relationship Quantification. In the real world, relationships between variables may be highly non-linear. Moreover, most of the real-world data are multi-variate, and the relationships between multi-variate data cannot be directly measured via linear relationship quantification (the reason is that the correlation is a scalar and can only capture the linear relationship between uni-variate random variables). Hence, we require tools or statistical measurements to quantify the non-linear relationships. In the following, we discuss two popular measurements for non-linear relationships: *Mutual Information (MI)* and *Hilbert-Schmidt Independence Criterion (HSIC)*.

Mutual Information (MI). Mutual Information (MI) Cover [1999] is used to measure the mutual dependency between two variables. In specific, MI quantifies the amount of the information obtained about one random variable through observing the other random variable. Between the two random variables X and Y, MI has the formulation:

$$\mathrm{MI}(X;Y) = D_{\mathrm{KL}}(P_{XY} \,\|\, P_X P_Y) = \mathbb{E}_{XY}[\log \frac{p(x,y)}{p(x)p(y)}],$$

where $\mathrm{MI}(X;Y)$ is the mutual information and D_{KL} is the KL-divergence. MI has the following properties. First, $\mathrm{MI}(X;Y) \geq 0$. A large $\mathrm{MI}(X;Y)$ means a high dependency between X and Y, and zero $\mathrm{MI}(X;Y)$ means X and Y are independent. Second, $\mathrm{MI}(X;Y)$ work for both uni-variate and multi-variate variables. In other words, we can compute the mutual information between two sets of images and even two sets of videos. In short, the applications for MI are similar to the applications for Pearson's correlation coefficient, and MI has further benefits of working on multi-variate variables and captures non-linear relationships.

The disadvantage of quantifying the relationships using MI is that MI estimation is notoriously difficult [McAllester and Stratos, 2020, Moddemeijer, 1989, Song and Ermon, 2019]. Prior approaches leverage counting-based [Bouma, 2009, Church and Hanks, 1990, Levy and Goldberg, 2014] methods for estimating MI, which approximates the joint density by counting the occurrence of the pair (i.e., (x, y)) and the marginal density by counting the presence of the individual outcome (i.e., x or y). Unfortunately, counting based approaches can only work on discrete data and may be unrealistic when the data is sparse. Recent approaches [Belghazi et al., 2018, Poole et al., 2019] present neural methods that estimate MI via its variational bounds. They consider MI 1) lower bounds such as Donsker-Varadhan bound [Donsker and Varadhan, 1983] and Nguyen-Wainwright-Jordan bound [Nguyen et al., 2010]; and 2) upper bound such as Barber-Agakov bound [Barber and Agakov, 2003]. Although the neural methods can work on continuous data, the variational bounds exhibit inevitable large variance [Song and Ermon, 2019], which leads to severe training instability in practice [Poole et al., 2019, Tschannen et al., 2019].

Hilbert-Schmidt Independence Criterion (HSIC). Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2005a] is another measurement of the mutual dependency between two variables. HSIC is a kernel-based approach with the formulation

$$\mathrm{HSIC}(X;Y) = D_{\mathrm{MMD}}(P_{XY} \,\|\, P_X P_Y) = \|C_{XY}\|^2_{\mathrm{HS}},$$

where $\mathrm{HSIC}(X;Y)$ is the HSIC between random variables X and Y, D_{MMD} is the maximum mean discrepancy [Gretton et al., 2012], C_{XY} is the cross-covariance operators between the Reproducing Kernel Hilbert Spaces (RKHSs) of X and Y, and $\|\cdot\|^2_{HS}$ is the Hilbert-Schmidt norm. Similar to the mutual information, $\mathrm{HSIC}(X;Y) \geq 0$. $\mathrm{HSIC}(X;Y) = 0$ means the two random variables are independent, and large $\mathrm{HSIC}(X;Y)$ means the high dependency between X and Y. Last, although HSIC can work for multi-variate data, since HSIC is studied in the context of kernel-based methods, which can make it difficult to apply in practice when data is high-dimensional and complex-structured [Gretton et al., 2005a].

Connection to Our Contributions. In our thesis, we contribute to improve the estimation for the non-linear relationship - mutual information (MI). As discussed above, estimating MI contains severe training instability, which leads to either large variance or large bias in practice [Poole et al., 2019]. The reason is that prior approaches [Barber and Agakov, 2003, Belghazi et al., 2018, Donsker and Varadhan, 1983, Nguyen et al., 2010, Oord et al., 2018, Poole et al., 2019] consider estimating MI via its variational bounds, and computing these bounds exhibits inevitable large variance and bias [McAllester and Stratos, 2020, Song and Ermon, 2019]. Rather than considering the variational bounds, we present to estimate MI by plugging-in estimated point-wise mutual information (a fine-grained dependency measurement, see Chapter 5). Our presented methods utilize neural networks, and hence they can work for high-dimensional real-world data. Also, our methods contain no logarithm and exponentiation, which avoid the optimization instability.

2.3 Self-supervised Learning

The third related topic to our thesis is self-supervised learning (SSL), which performs representation learning by leveraging the supervision from the data itself, but not downstream task labels. Hence, SSL provides us a way to leverage a large amount of unlabeled data to learn good representations. To provide an overview of SSL methods, we group them for research subjects, spanning *Computer Vision*, *Natural Language Processing*, *Speech Processing*, *Theoretical Foundations*, and *Others*.

SSL in Computer Vision. The beginning of SSL in computer vision starts with learning representations for context prediction. For instance, Doersch et al. [2015] presents an approach to learn the representations for predicting the spatial context, such as predicting the relative positions of two image patches. Lee et al. [2017] presents an approach to learn the representations from shuffled frames in an video such that the temporal coherence can be recovered. In short, these context-prediction approaches design the SSL objectives to solve the tasks that require high-level semantic understanding of data, and these tasks are often referred to as pretext tasks [Gidaris et al., 2018, Noroozi and Favaro, 2016, Noroozi et al., 2017, Zhang et al., 2016].

The next type of SSL in computer vision is learning representations for instance discrimination, or called contrastive learning [Chen et al., 2020a, He et al., 2019, Oord et al., 2018, Wu et al., 2018b]. These methods (e.g., the simple contrastive learning (SimCLR) [Chen et al., 2020a] and the momentum contrastive learning (MoCo) [He et al., 2019]) consider learning similar representations for the augmented variants (by applying different image augmentations) of the same image and dissimilar representations for different images. The contrastive approaches are shown to learn the representations that can perform as well as the supervised learned representations on downstream tasks [Arora et al., 2019]. Nonetheless, these methods often require large training batch sizes and large networks [Chen et al., 2020a], and hence they are computationally much more expensive than the supervised approaches.

The third type of SSL in computer vision is learning invariant representations with respect to image augmentations but without contrasting representations between different images [Caron et al., 2020, Grill et al., 2020, Zbontar et al., 2021]. In particular, similar to the contrastive approaches (the second phase of SSL in computer vision), the new methods (e.g., the bootstrap your own latents method (BYOL) [Grill et al., 2020] and the Barlow Twins' method [Zbontar et al., 2021]) consider learning similar representations for the augmented variants of the same image. Nonetheless, different from the contrastive approaches, these new methods do not force the representations to be dissimilar between different images. To conclude, these methods are shown to learn the representations that perform as well as the contrastive methods on downstream tasks, and they have further benefits of enjoying better robustness to the training batch sizes, resulting in higher computational efficiency.

SSL in Natural Language Processing. The first type of SSL in natural language processing is the development of word embeddings (i.e., Word2Vec [Mikolov et al., 2013] and Glove [Pennington et al., 2014]). The word embeddings methods present center or neighborhood entities prediction, such as predicting the center word given the nearby words in the Word2Vec [Mikolov et al., 2013] method. These methods have been immensely influential, since they map less-expressive, high-dimensional, sparse, and discrete words into more-expressive, low-dimensional, dense, and continuous representations. Nonetheless, the major limitation is that the learned word representation is relatively stable across sentences, which means the representation will not change when having different contexts. In other words, these methods are learning non-contextualized representations.

The second type of SSL in natural language processing is the development of learning contextualized representations [Devlin et al., 2018, Lewis et al., 2019, Peters et al., 2018, Radford et al., 2018]. In particular, the contextualized representation learning methods learn word representations by taking account of the context of a word, and hence the same word under different contexts would result in distinct representations. For instance, "Apple" can be a fruit or a company, and their representations should be different. We can understand the contextualized approaches by discussing their network architectures and objectives. First, these methods consider neural network sequence models as the network architectures, such as LSTMs [Hochreiter and Schmidhuber, 1997] in ELMO [Peters et al., 2018] and Transformers [Vaswani et al., 2017] in BERT [Devlin et al., 2018], GPT [Radford et al., 2018], and BART [Lewis et al., 2019]. Second, these methods consider the objectives that require high-level semantic understanding of the training text corpora, such as predicting next words in ELMO [Peters et al., 2018] and GPT [Radford et al., 2018], predicting masked words from non-masked words [Devlin et al., 2018], and recovering order of words in permuted sentences in BART [Lewis et al., 2019].

SSL in Speech Processing. The development of SSL in speech processing follows by the development of SSL in natural language processing. The procedure of SSL in both domains are nearly identical, with the main difference that the speech data is continuous and the text data is discrete. In specific, Wav2Vec [Schneider et al., 2019] and APC [Chung and Glass, 2020] share similar training paradigm with the auto-regressive language models like ELMO [Peters et al., 2018] and GPT [Radford et al., 2018]. Wav2vec 2.0 [Baevski et al., 2020] shares similar training paradigm with the masked language models like BERT [Devlin et al., 2018] and XL-Net [Yang et al., 2019].

Theoretical Foundations for SSL. While SSL approaches work well empirically, we are interested in understanding the theoretical foundations behind them. At a high level, all these methods focus on showing that the SSL approaches can provably learn the representations that perform well on downstream tasks even without access to downstream supervision. The very first study is presented by [Arora et al., 2019], which studied the efficacy of a popular family of SSL approaches, the contrastive approaches [Chen et al., 2020a, He et al., 2019]. Tosh et al. [2020] extended the study for contrastive approaches, from a multi-view perspective. Then, Lee et al. [2020] studied the efficacy of another popular family of SSL approaches, the predictive learning approaches [Devlin et al., 2018, Zhang et al., 2016]. Recently, Teng and Huang [2021] provided the study on the efficacy of the SSL approaches that perform context prediction, in particular the tasks that require high-level semantic understanding of data [Gidaris et al., 2018, Noroozi and Favaro, 2016, Noroozi et al., 2017]. As a summary, although the theoretical foundations of SSL are not yet complete, building these foundations can potentially encourage better designs of SSL methods.

SSL in Other Domains. SSL also emerges in lots of different domains. It appears in cross-modality learning (e.g., audio-visual learning [Arandjelovic and Zisserman, 2018, Owens and Efros, 2018, Zhao et al., 2018] and visual-textual [Radford et al., 2021]), robotics (e.g., the Curious Robot [Pinto et al., 2016] and the Visual Pushing for Grasping [Zeng et al., 2018]), reinforcement learning (e.g., the Curiosity-driven Learning [Burda et al., 2018, Pathak et al., 2017]), and graph representation learning (e.g., the Deep Graph Infomax [Veličković et al., 2018] and the Graph Contrastive Approach [Hassani and Khasahmadi, 2020]).

Connection to Our Contributions. Our contributions for SSL are four folds. First, we attempt to understand SSL from a multi-view perspective. In Chapter 6, we provide information-theoretical analysis on self-supervised learned representations, explaining why the representations can perform well on downstream tasks even without access to downstream supervision, and connecting two families of SSL methods (the contrastive [Chen et al., 2020a, He et al., 2019] and the predictive learning methods [Devlin et al., 2018, Zhang et al., 2016]) together. Second, we show in Chapter 5, existing SSL objectives relate to the mutual information estimation and maximization. The relatedness inspires better optimization process for SSL objectives, leading to better downstream performance. Third, we present to include or exclude external information from the self-supervised learned representations. In Chapter 7, we discuss methods for including auxiliary information of data (e.g., hashtags for Instagram images) into the self-supervised representation learning process. Fourth, in Chapter 8, we present methods for excluding undesirable information (e.g., the gender information for gender-irrelevant tasks) from data.

Chapter 3

Heterogeneous Structure - Synchronization and Alignment

In this chapter, we study human communication as the cross-view data and discuss the sub-challenge of synchronization and alignment within the challenge of heterogeneous structure. Human language possesses not only spoken words but also nonverbal behaviors from vision (facial attributes) and acoustic (tone of voice) modalities [Gibson et al., 1994]. This rich information provides us the benefit of understanding human behaviors and intents [Manning et al., 2014]. Nevertheless, the heterogeneities across modalities often increase the difficulty of analyzing human language. For example, the receptors for audio and vision streams may vary with variable receiving frequency, and hence we may not obtain optimal mapping between them. A frowning face may relate to a pessimistically word spoken in the past. That is to say, multimodal language sequences often exhibit "unaligned" nature and require inferring long term dependencies across modalities, which raises a question on performing efficient multimodal fusion.

To address the above issues, in this thesis we propose the Multimodal Transformer (MulT), an end-to-end model that extends the standard Transformer network [Vaswani et al., 2017] to learn representations directly from unaligned multimodal streams. At the heart of our model is the crossmodal attention module, which attends to the crossmodal interactions at the scale of the entire utterances. This module latently adapts streams from one modality to another (e.g., vision → language) by repeated reinforcing one modality's features with those from the other modalities, regardless of the need for alignment. In comparison, one common way of tackling unaligned multimodal sequence is by forced word-aligning before training [Gu et al., 2018, Pham et al., 2019, Poria et al., 2017b, Tsai et al., 2018, Zadeh et al., 2018a,c]: manually preprocess the visual and acoustic features by aligning them to the resolution of words. These approaches would then model the multimodal interactions on the (already) aligned time steps and thus do not directly consider long-range crossmodal contingencies of the original features. We note that such word-alignment not only requires feature engineering that involves domain knowledge; but in practice, it may also not always be feasible, as it entails extra meta-information about the datasets (e.g., the exact time ranges of words or speech utterances). We illustrate the difference between the word-alignment and the crossmodal attention inferred by our model in Figure 3.1.

For evaluation, we perform a comprehensive set of experiments on three human multimodal language benchmarks: CMU-MOSI [Zadeh et al., 2016], CMU-MOSEI [Zadeh et al., 2018c], and IEMOCAP [Busso et al., 2008b]. Our experiments show that MulT achieves the state-of-the-art (SOTA) results in not only the commonly evaluated word-aligned setting but also the more challenging unaligned scenario, outperforming prior approaches by a margin of 5%-15% on most of the metrics. In addition, empirical qualitative analysis further suggests that the crossmodal attention used by MulT is capable of capturing correlated signals

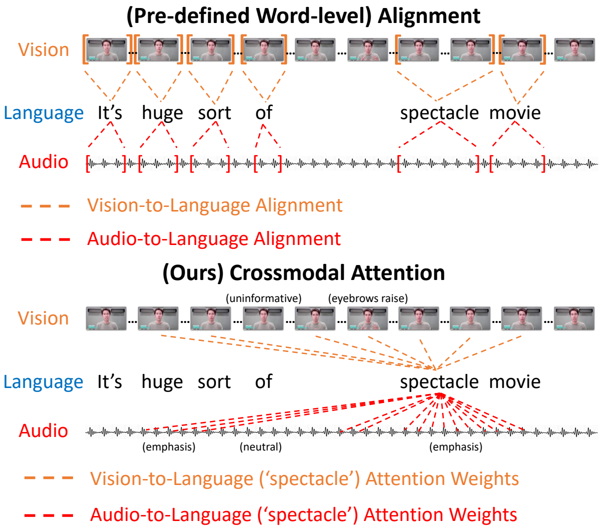

(Pre-defined Word-level) Alignment
Vision
Language
It's
huge
sort
of
spectacle
movie
Audio
Vision-to-Language Alignment
Audio-to-Language Alignment
(Ours) Crossmodal Attention
(uninformative)
(eyebrows raise)
Vision
Language
It's
huge
sort
of
spectacle
movie
Audio
(emphasis)
(neutral)
(emphasis)
Vision-to-Language ('spectacle') Attention Weights
Audio-to-Language ('spectacle') Attention Weights

Figure 3.1: Example video clip from movie reviews. [Top]: Illustration of word-level alignment where video and audio features are averaged across the time interval of each spoken word. [Bottom] Illustration of crossmodal attention weights between text ("spectacle") and vision/audio.

across asynchronous modalities.

3.1 Related Work

Human Multimodal Language Analysis. Prior work for analyzing human multimodal language lies in the domain of inferring representations from multimodal sequences spanning language, vision, and acoustic modalities. Unlike learning multimodal representations from static domains such as image and textual attributes [Ngiam et al., 2011, Srivastava and Salakhutdinov, 2012], human language contains time-series and thus requires fusing time-varying signals [Liang et al., 2018a, Tsai et al., 2018]. Earlier work used early fusion approach to concatenate input features from different modalities [Lazaridou et al., 2015, Ngiam et al., 2011] and showed improved performance as compared to learning from a single modality. More recently, more advanced models were proposed to learn representations of human multimodal language. For example, Gu et al. [2018] used hierarchical attention strategies to learn multimodal representations, Wang et al. [2019] adjusted the word representations using accompanying non-verbal behaviors, Pham et al. [2019] learned robust multimodal representations using a cyclic translation objective, and Dumpala et al. [2019] explored cross-modal autoencoders for audio-visual alignment. These previous approaches relied on the assumption that multimodal language sequences are already aligned in the resolution of words and considered only short-term multimodal interactions. In contrast, our proposed method requires no alignment assumption and defines crossmodal interactions at the scale of the entire sequences.

Transformer Network. Transformer network [Vaswani et al., 2017] was first introduced for neural machine translation (NMT) tasks, where the encoder and decoder side each leverages a *self-attention* [Lin et al., 2017, Parikh et al., 2016, Vaswani et al., 2017] transformer. After each layer of the self-attention, the encoder and decoder are connected by an additional decoder sublayer where the decoder attends to each element of the source text for each element of the target text. We refer the reader to prior work [Vaswani et al., 2017] for a more detailed explanation of the model. In addition to NMT, transformer networks have

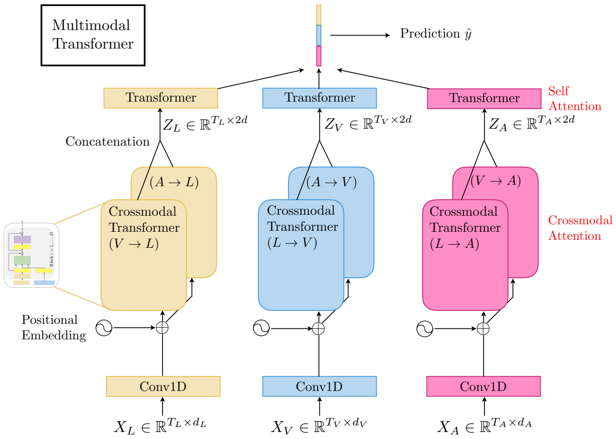

Multimodal Transformer
Prediction $\hat{y}$
Transformer
Transformer
Transformer
Self Attention
$Z_L \in \mathbb{R}^{T_L \times 2d}$
$Z_V \in \mathbb{R}^{T_V \times 2d}$
$Z_A \in \mathbb{R}^{T_A \times 2d}$
Concatenation
$(A \rightarrow L)$
$(A \rightarrow V)$
$(V \rightarrow A)$
Crossmodal Transformer $(V \rightarrow L)$
Crossmodal Transformer $(L \rightarrow V)$
Crossmodal Transformer $(L \rightarrow A)$
Crossmodal Attention
Positional Embedding
Conv1D
Conv1D
Conv1D
$X_L \in \mathbb{R}^{T_L \times d_L}$
$X_V \in \mathbb{R}^{T_V \times d_V}$
$X_A \in \mathbb{R}^{T_A \times d_A}$

Figure 3.2: Overall architecture for MulT on modalities (L, V, A). The crossmodal transformers, which suggests latent crossmodal adaptations, are the core components of MulT for multimodal fusion.

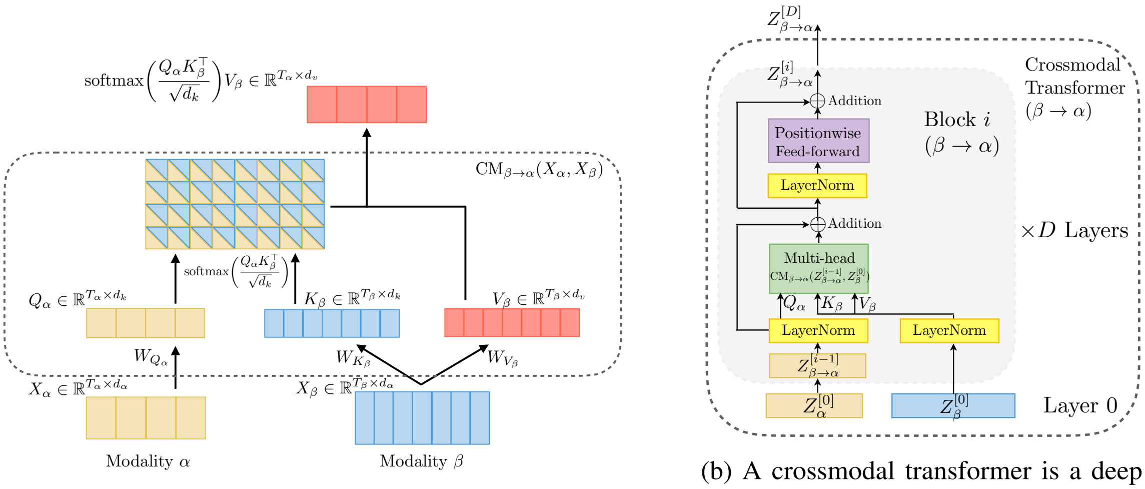
also been successfully applied to other tasks, including language modeling [Baevski and Auli, 2019, Dai et al., 2018], semantic role labeling [Strubell et al., 2018], word sense disambiguation [Tang et al., 2018], learning sentence representations [Devlin et al., 2018], and video activity recognition [Wang et al., 2018].

This thesis absorbs a strong inspiration from the NMT transformer to extend to a multimodal setting. Whereas the NMT transformer focuses on unidirectional *translation* from source to target texts, human multimodal language time-series are neither as well-represented nor discrete as word embeddings, with sequences of each modality having vastly different frequencies. Therefore, we propose not to explicitly translate from one modality to the others (which could be extremely challenging), but to *latently* adapt elements across modalities via the attention. Our model (MulT) therefore has no encoder-decoder structure, but it is built up from multiple stacks of pairwise and bidirectional crossmodal attention blocks that directly attend to low-level features (while removing the self-attention). Empirically, we show that our proposed approach improves beyond standard transformer on various human multimodal language tasks.

3.2 Proposed Method

In this chapter, we describe our proposed Multimodal Transformer (MulT) (Figure 3.2) for modeling unaligned multimodal language sequences. At the high level, MulT merges multimodal time-series via a feed-forward fusion process from multiple directional pairwise crossmodal transformers. Specifically, each crossmodal transformer (introduced in Chapter 3.2.2) serves to repeatedly reinforce a *target modality* with the low-level features from another *source modality* by learning the attention across the two modalities' features. A MulT architecture hence models all pairs of modalities with such crossmodal transformers, followed by sequence models (e.g., self-attention transformer) that predicts using the fused features.

The core of our proposed model is crossmodal attention module, which we first introduce in Chapter 3.2.1. Then, in Chapter 3.2.2 and 3.2.3, we present in details the various ingredients of the MulT architecture (see Figure 3.2) and discuss the difference between crossmodal attention and classical multimodal alignment.


softmax$\left(\frac{Q_\alpha K_\beta^\top}{\sqrt{d_k}}\right)V_\beta \in \mathbb{R}^{T_\alpha \times d_v}$
$\mathrm{CM}_{\beta\to\alpha}(X_\alpha, X_\beta)$
softmax$\left(\frac{Q_\alpha K_\beta^\top}{\sqrt{d_k}}\right)$
$Q_\alpha \in \mathbb{R}^{T_\alpha \times d_k}$
$K_\beta \in \mathbb{R}^{T_\beta \times d_k}$
$V_\beta \in \mathbb{R}^{T_\beta \times d_v}$
W_{Q_α}
W_{K_β}
W_{V_β}
$X_\alpha \in \mathbb{R}^{T_\alpha \times d_\alpha}$
$X_\beta \in \mathbb{R}^{T_\beta \times d_\alpha}$
Modality α
Modality β
$Z^{[D]}_{\beta\to\alpha}$
$Z^{[i]}_{\beta\to\alpha}$
Addition
Positionwise Feed-forward
LayerNorm
Addition
Multi-head $\mathrm{CM}_{\beta\to\alpha}(Z^{[i-1]}_{\beta\to\alpha}, Z^{[0]}_\beta)$
Q_α K_β V_β
LayerNorm
LayerNorm
$Z^{[i-1]}_{\beta\to\alpha}$
$Z^{[0]}_\alpha$
$Z^{[0]}_\beta$
Block i $(\beta \to \alpha)$
Crossmodal Transformer $(\beta \to \alpha)$
$\times D$ Layers
Layer 0

(a) Crossmodal attention $\mathrm{CM}_{\beta\to\alpha}(X_\alpha, X_\beta)$ between sequences X_α, X_β from distinct modalities.

(b) A crossmodal transformer is a deep stacking of several crossmodal attention blocks.

Figure 3.3: Architectural elements of a crossmodal transformer between two time-series from modality α and β.

3.2.1 Crossmodal Attention

We consider two modalities α and β, with two (potentially non-aligned) sequences from each of them denoted $X_\alpha \in \mathbb{R}^{T_\alpha \times d_\alpha}$ and $X_\beta \in \mathbb{R}^{T_\beta \times d_\beta}$, respectively. For the rest of the chapter, $T_{(\cdot)}$ and $d_{(\cdot)}$ are used to represent sequence length and feature dimension, respectively. Inspired by the decoder transformer in NMT [Vaswani et al., 2017] that translates one language to another, we hypothesize a good way to fuse crossmodal information is providing a latent adaptation across modalities; i.e., β to α. Note that the modalities consider in this Chapter may span very different domains such as facial attributes and spoken words.

We define the Querys as $Q_\alpha = X_\alpha W_{Q_\alpha}$, Keys as $K_\beta = X_\beta W_{K_\beta}$, and Values as $V_\beta = X_\beta W_{V_\beta}$, where $W_{Q_\alpha} \in \mathbb{R}^{d_\alpha \times d_k}$, $W_{K_\beta} \in \mathbb{R}^{d_\beta \times d_k}$ and $W_{V_\beta} \in \mathbb{R}^{d_\beta \times d_v}$ are weights. The latent adaptation from β to α is presented as the crossmodal attention $Y_\alpha := \mathrm{CM}_{\beta\to\alpha}(X_\alpha, X_B) \in \mathbb{R}^{T_\alpha \times d_v}$:

$$\begin{aligned} Y_\alpha &= \mathrm{CM}_{\beta\to\alpha}(X_\alpha, X_\beta) \\ &= \mathrm{softmax}\left(\frac{Q_\alpha K_\beta^\top}{\sqrt{d_k}}\right) V_\beta \\ &= \mathrm{softmax}\left(\frac{X_\alpha W_{Q_\alpha} W_{K_\beta}^\top X_\beta^\top}{\sqrt{d_k}}\right) X_\beta W_{V_\beta}. \end{aligned} \tag{3.1}$$

Note that Y_α has the same length as Q_α (i.e., T_α), but is meanwhile represented in the feature space of V_β. Specifically, the scaled (by $\sqrt{d_k}$) softmax in Equation (3.1) computes a score matrix $\mathrm{softmax}\,(\cdot) \in \mathbb{R}^{T_\alpha \times T_\beta}$, whose (i, j)-th entry measures the attention given by the i-th time step of modality α to the j-th time step of modality β. Hence, the i-th time step of Y_α is a weighted summary of V_β, with the weight determined by i-th row in $\mathrm{softmax}(\cdot)$. We call Equation (3.1) a *single-head* crossmodal attention, which is illustrated in Figure 3.3a.

Following prior works on transformers [Chen et al., 2018b, Dai et al., 2018, Devlin et al., 2018, Vaswani et al., 2017], we add a residual connection to the crossmodal attention computation. Then, another positionwise feed-forward sublayer is injected to complete a *crossmodal attention block* (see Figure 3.3b). Each crossmodal attention block adapts directly from the low-level feature sequence (i.e., $Z_{\beta}^{[0]}$ in Figure 3.3b) and does not rely on self-attention, which makes it different from the NMT encoder-decoder architecture [Shaw et al., 2018, Vaswani et al., 2017] (i.e., taking intermediate-level features). We argue that performing adaptation from low-level feature benefits our model to preserve the low-level information for each modality. We leave the empirical study for adapting from intermediate-level features (i.e., $Z_{\beta}^{[i-1]}$) in Ablation Study in Chapter 3.3.3.

3.2.2 Overall Architecture

Three major modalities are typically involved in multimodal language sequences: language (L), video (V), and audio (A) modalities. We denote with $X_{\{L,V,A\}} \in \mathbb{R}^{T_{\{L,V,A\}} \times d_{\{L,V,A\}}}$ the input feature sequences (and the dimensions thereof) from these 3 modalities. With these notations, we describe in greater details the components of Multimodal Transformer and how crossmodal attention modules are applied.

Temporal Convolutions. To ensure that each element of the input sequences has sufficient awareness of its neighborhood elements, we pass the input sequences through a 1D temporal convolutional layer:

$$\hat{X}_{\{L,V,A\}} = \text{Conv1D}(X_{\{L,V,A\}}, k_{\{L,V,A\}}) \in \mathbb{R}^{T_{\{L,V,A\}} \times d} \tag{3.2}$$

where $k_{\{L,V,A\}}$ are the sizes of the convolutional kernels for modalities $\{L, V, A\}$, and d is a common dimension. The convolved sequences are expected to contain the local structure of the sequence, which is important since the sequences are collected at different sampling rates. Moreover, since the temporal convolutions project the features of different modalities to the same dimension d, the dot-products are admittable in the crossmodal attention module.

Positional Embedding. To enable the sequences to carry temporal information, following prior work [Vaswani et al., 2017], we augment positional embedding (PE) to $\hat{X}_{\{L,V,A\}}$:

$$Z_{\{L,V,A\}}^{[0]} = \hat{X}_{\{L,V,A\}} + \text{PE}(T_{\{L,V,A\}}, d) \tag{3.3}$$

where $\text{PE}(T_{\{L,V,A\}}, d) \in \mathbb{R}^{T_{\{L,V,A\}} \times d}$ computes the (fixed) embeddings for each position index, and $Z_{\{L,V,A\}}^{[0]}$ are the resulting low-level position-aware features for different modalities. We leave more details of the positional embedding to Chapter 3.5.1.

Crossmodal Transformers. Based on the crossmodal attention blocks, we design the crossmodal transformer that enables one modality for receiving information from another modality. In the following, we use the example for passing vision (V) information to language (L), which is denoted by "$V \rightarrow L$". We fix all the dimensions ($d_{\{\alpha,\beta,k,v\}}$) for each crossmodal attention block as d.

Each crossmodal transformer consists of D layers of crossmodal attention blocks (see Figure 3.3b). Formally, a crossmodal transformer computes feed-forwardly for $i = 1, \dots, D$ layers:

$$\begin{aligned} Z_{V\rightarrow L}^{[0]} &= Z_{L}^{[0]} \\ \hat{Z}_{V\rightarrow L}^{[i]} &= \text{CM}_{V\rightarrow L}^{[i],\text{mul}}(\text{LN}(Z_{V\rightarrow L}^{[i-1]}), \text{LN}(Z_{V}^{[0]})) + \text{LN}(Z_{V\rightarrow L}^{[i-1]}) \\ Z_{V\rightarrow L}^{[i]} &= f_{\theta_{V\rightarrow L}^{[i]}}(\text{LN}(\hat{Z}_{V\rightarrow L}^{[i]})) + \text{LN}(\hat{Z}_{V\rightarrow L}^{[i]}) \end{aligned} \tag{3.4}$$

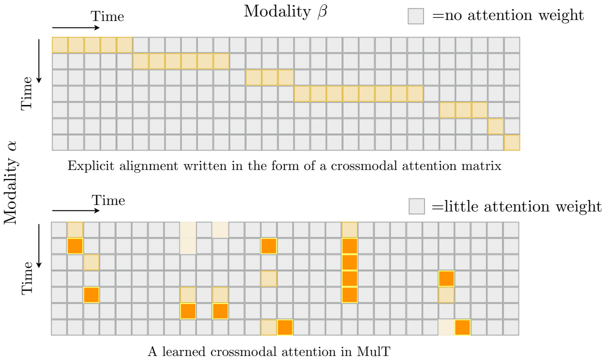

Time
Modality β
=no attention weight
Time
Modality α
Explicit alignment written in the form of a crossmodal attention matrix
Time
=little attention weight
Time
A learned crossmodal attention in MulT

Figure 3.4: An example of visualizing alignment using attention matrix from modality β to α. Multimodal alignment is a special (monotonic) case for crossmodal attention.

where f_θ is a positionwise feed-forward sublayer parametrized by θ, and $\mathrm{CM}^{[i],\mathrm{mul}}_{V\to L}$ means a multi-head (see prior work [Vaswani et al., 2017] for more details) version of $\mathrm{CM}_{V\to L}$ at layer i (note: d should be divisible by the number of heads). LN means layer normalization [Ba et al., 2016].

In this process, each modality keeps updating its sequence via low-level external information from the multi-head crossmodal attention module. At every level of the crossmodal attention block, the low-level signals from source modality are transformed to a different set of Key/Value pairs to interact with the target modality. Empirically, we find that the crossmodal transformer learns to correlate meaningful elements across modalities (see Chapter 3.3 for details). The eventual MulT is based on modeling every pair of crossmodal interactions. Therefore, with 3 modalities (i.e., L, V, A) in consideration, we have 6 crossmodal transformers in total (see Figure 3.2).

Self-Attention Transformers and Prediction. As a final step, we concatenate the outputs from the crossmodal transformers that share the same target modality to yield $Z_{\{L,V,A\}} \in \mathbb{R}^{T_{\{L,V,A\}}\times 2d}$. For example, $Z_L = [Z^{[D]}_{V\to L}; Z^{[D]}_{A\to L}]$. Each of them is then passed through a sequence model to collect temporal information to make predictions. We choose the self-attention transformer [Vaswani et al., 2017]. Eventually, the last elements of the sequences models are extracted to pass through fully-connected layers to make predictions.

3.2.3 Discussion about Attention & Alignment

When modeling unaligned multimodal language sequences, MulT relies on crossmodal attention blocks to merge signals across modalities. While the multimodal sequences were (manually) aligned to the same length in prior works before training [Liang et al., 2018a, Pham et al., 2019, Tsai et al., 2018, Wang et al., 2019, Zadeh et al., 2018c], we note that MulT looks at the non-alignment issue through a completely different lens. Specifically, for MulT, the correlations between elements of multiple modalities are purely based on attention. In other words, MulT does not handle modality non-alignment by (simply) aligning them; instead, the crossmodal attention encourages the model to directly attend to elements in other modalities where strong signals or relevant information is present. As a result, MulT can capture long-range crossmodal contingencies in a way that conventional alignment could not easily reveal. Classical crossmodal alignment, on the other hand, can be expressed as a special (step diagonal) crossmodal attention matrix (i.e., monotonic attention [Yu et al., 2016]). We illustrate their differences in Figure 3.4.

3.3 Experiments

Now, we empirically evaluate the Multimodal Transformer (MulT) on three datasets that are frequently used to benchmark human multimodal affection recognition in prior works [Liang et al., 2018a, Pham et al., 2019, Tsai et al., 2018]. Our goal is to compare MulT with prior competitive approaches on both *word-aligned* (by word, which almost all prior works employ) and *unaligned* (which is more challenging, and which MulT is generically designed for) multimodal language sequences.

3.3.1 Datasets and Evaluation Metrics

Each task consists of a *word-aligned* (processed in the same way as in prior works) and an *unaligned* version. For both versions, the multimodal features are extracted from the textual (GloVe word embeddings [Pennington et al., 2014]), visual (Facet [iMotions, 2017]), and acoustic (COVAREP [Degottex et al., 2014]) data modalities.

For the word-aligned version, following [Pham et al., 2019, Tsai et al., 2018, Zadeh et al., 2018a], we first use P2FA [Yuan and Liberman, 2008] to obtain the aligned timesteps (segmented w.r.t. words) for audio and vision streams, and we then perform averaging on the audio and vision features within these time ranges. All sequences in the word-aligned case have length 50. The process remains the same across all the datasets. On the other hand, for the unaligned version, we keep the original audio and visual features as extracted, without any word-segmented alignment or manual subsampling. As a result, the lengths of each modality vary significantly, where audio and vision sequences may contain up to $> 1,000$ time steps. We elaborate on the three tasks below.

CMU-MOSI & MOSEI. CMU-MOSI [Zadeh et al., 2016] is a human multimodal sentiment analysis dataset consisting of 2,199 short monologue video clips (each lasting the duration of a sentence). Acoustic and visual features of CMU-MOSI are extracted at a sampling rate of 12.5 and 15 Hz, respectively (while textual data are segmented per word and expressed as discrete word embeddings). Meanwhile, CMU-MOSEI [Zadeh et al., 2018c] is a sentiment and emotion analysis dataset made up of 23,454 movie review video clips taken from YouTube (about 10× the size of CMU-MOSI). The unaligned CMU-MOSEI sequences are extracted at a sampling rate of 20 Hz for acoustic and 15 Hz for vision signals.

For both CMU-MOSI and CMU-MOSEI, each sample is labeled by human annotators with a sentiment score from -3 (strongly negative) to 3 (strongly positive). We evaluate the model performances using various metrics, in agreement with those employed in prior works: 7-class accuracy (i.e., Acc_7: sentiment score classification in $\mathbb{Z} \cap [-3, 3]$), binary accuracy (i.e., Acc_2: positive/negative sentiments), F1 score, mean absolute error (MAE) of the score, and the correlation of the model's prediction with human. Both tasks are frequently used to benchmark models' ability to fuse multimodal (sentiment) information [Liang et al., 2018a, Pham et al., 2019, Poria et al., 2017b, Tsai et al., 2018, Wang et al., 2019, Zadeh et al., 2018a].

IEMOCAP. IEMOCAP [Busso et al., 2008b] consists of 10K videos for human emotion analysis. As suggested by Wang et al. [2019], 4 emotions (happy, sad, angry and neutral) were selected for emotion recognition. Unlike CMU-MOSI and CMU-MOSEI, this is a multilabel task (e.g., a person can be sad and angry simultaneously). Its multimodal streams consider fixed sampling rate on audio (12.5 Hz) and vision (15 Hz) signals. We follow [Poria et al., 2017b, Tsai et al., 2018, Wang et al., 2019] to report the binary classification accuracy and the F1 score of the predictions.

Metric	Acc_7^h	Acc_2^h	$\mathrm{F1}^h$	MAE^ℓ	Corr^h
(Word Aligned) CMU-MOSI Sentiment					
EF-LSTM	33.7	75.3	75.2	1.023	0.608
LF-LSTM	35.3	76.8	76.7	1.015	0.625
RMFN [Liang et al., 2018a]	38.3	78.4	78.0	0.922	0.681
MFM [Tsai et al., 2018]	36.2	78.1	78.1	0.951	0.662
RAVEN [Wang et al., 2019]	33.2	78.0	76.6	0.915	**0.691**
MCTN [Pham et al., 2019]	35.6	79.3	79.1	0.909	0.676
MulT (ours)	**40.0**	**83.0**	**82.8**	**0.871**	**0.698**
(Unaligned) CMU-MOSI Sentiment					
CTC [Graves et al., 2006] + EF-LSTM	31.0	73.6	74.5	1.078	0.542
LF-LSTM	33.7	77.6	77.8	0.988	0.624
CTC + MCTN [Pham et al., 2019]	32.7	75.9	76.4	0.991	0.613
CTC + RAVEN [Wang et al., 2019]	31.7	72.7	73.1	1.076	0.544
MulT (ours)	**39.1**	**81.1**	**81.0**	**0.889**	**0.686**

Table 3.1: Results for multimodal sentiment analysis on CMU-MOSI with aligned and non-aligned multimodal sequences. ^{h} means higher is better and $^\ell$ means lower is better. EF stands for early fusion, and LF stands for late fusion.

3.3.2 Baselines

We choose Early Fusion LSTM (EF-LSTM) and Late Fusion LSTM (LF-LSTM) as baseline models, as well as Recurrent Attended Variation Embedding Network (RAVEN) [Wang et al., 2019] and Multimodal Cyclic Translation Network (MCTN) [Pham et al., 2019], that achieved SOTA results on various word-aligned human multimodal language tasks. To compare the models comprehensively, we adapt the *connectionist temporal classification* (CTC) [Graves et al., 2006] method to the prior approaches (e.g., EF-LSTM, MCTN, RAVEN) that cannot be applied directly to the unaligned setting. Specifically, these models train to optimize the CTC alignment objective and the human multimodal objective simultaneously. We leave more detailed treatment of the CTC module to Chapter 3.5.2. For fair comparisons, we control the number of parameters of all models to be approximately the same. The hyperparameters are reported in Chapter 3.5.3. [1]

3.3.3 Quantitative Analysis

Word-Aligned Experiments. We first evaluate MulT on the *word-aligned sequences*— the "home turf" of prior approaches modeling human multimodal language [Pham et al., 2019, Sheikh et al., 2018, Tsai et al., 2018, Wang et al., 2019]. The upper part of the Table 3.1, 3.2, and 3.3 show the results of MulT and baseline approaches on the word-aligned task. With similar model sizes (around 200K parameters), MulT outperforms the other competitive approaches on different metrics on all tasks, with the exception of the "sad" class results on IEMOCAP.

Unaligned Experiments. Next, we evaluate MulT on the same set of datasets in the unaligned setting. Note that MulT can be directly applied to unaligned multimodal stream, while the baseline models (except for LF-LSTM) require the need of additional alignment module (e.g., CTC module).

The results are shown in the bottom part of Table 3.1, 3.2, and 3.3. On the three benchmark datasets, MulT improves upon the prior methods (some with CTC) by 10%-15% on most attributes. Empirically, we find that MulT converges faster to better results at training when compared to other competitive approaches

[1] All experiments are conducted on 1 GTX-1080Ti GPU. The code for our model and experiments can be found in https://github.com/yaohungt/Multimodal-Transformer

Metric	Acc$_7^h$	Acc$_2^h$	F1^h	MAE$^\ell$	Corr^h
(Word Aligned) CMU-MOSEI Sentiment					
EF-LSTM	47.4	78.2	77.9	0.642	0.616
LF-LSTM	48.8	80.6	80.6	0.619	0.659
Graph-MFN [Zadeh et al., 2018c]	45.0	76.9	77.0	0.71	0.54
RAVEN [Wang et al., 2019]	50.0	79.1	79.5	0.614	0.662
MCTN [Pham et al., 2019]	49.6	79.8	80.6	0.609	0.670
MulT (ours)	**51.8**	**82.5**	**82.3**	**0.580**	**0.703**
(Unaligned) CMU-MOSEI Sentiment					
CTC [Graves et al., 2006] + EF-LSTM	46.3	76.1	75.9	0.680	0.585
LF-LSTM	48.8	77.5	78.2	0.624	0.656
CTC + RAVEN [Wang et al., 2019]	45.5	75.4	75.7	0.664	0.599
CTC + MCTN [Pham et al., 2019]	48.2	79.3	79.7	0.631	0.645
MulT (ours)	**50.7**	**81.6**	**81.6**	**0.591**	**0.694**

Table 3.2: Results for multimodal sentiment analysis on (relatively large scale) CMU-MOSEI with aligned and non-aligned multimodal sequences.

Task	Happy		Sad		Angry		Neutral	
Metric	Acc^h	F1^h	Acc^h	F1^h	Acc^h	F1^h	Acc^h	F1^h
(Word Aligned) IEMOCAP Emotions								
EF-LSTM	86.0	84.2	80.2	80.5	85.2	84.5	67.8	67.1
LF-LSTM	85.1	86.3	78.9	81.7	84.7	83.0	67.1	67.6
RMFN [Liang et al., 2018a]	87.5	85.8	83.8	82.9	85.1	84.6	69.5	69.1
MFM [Tsai et al., 2018]	90.2	85.8	**88.4**	**86.1**	**87.5**	86.7	72.1	68.1
RAVEN [Wang et al., 2019]	87.3	85.8	83.4	83.1	**87.3**	86.7	69.7	69.3
MCTN [Pham et al., 2019]	84.9	83.1	80.5	79.6	79.7	80.4	62.3	57.0
MulT (ours)	**90.7**	**88.6**	86.7	**86.0**	**87.4**	**87.0**	**72.4**	**70.7**
(Unaligned) IEMOCAP Emotions								
CTC [Graves et al., 2006] + EF-LSTM	76.2	75.7	70.2	70.5	72.7	67.1	58.1	57.4
LF-LSTM	72.5	71.8	72.9	70.4	68.6	67.9	59.6	56.2
CTC + RAVEN [Wang et al., 2019]	77.0	76.8	67.6	65.6	65.0	64.1	**62.0**	**59.5**
CTC + MCTN [Pham et al., 2019]	80.5	77.5	72.0	71.7	64.9	65.6	49.4	49.3
MulT (ours)	**84.8**	**81.9**	**77.7**	**74.1**	**73.9**	**70.2**	**62.5**	**59.7**

Table 3.3: Results for multimodal emotions analysis on IEMOCAP with aligned and non-aligned multimodal sequences.

(see Figure 3.5). In addition, while we note that in general there is a performance drop on all models when we shift from the word-aligned to unaligned multimodal time-series, the impact MulT takes is much smaller than the other approaches. We hypothesize such performance drop occurs because the asynchronous (and much longer) data streams introduce more difficulty in recognizing important features and computing the appropriate attention.

Ablation Study. To further study the influence of the individual components in MulT, we perform comprehensive ablation analysis using the unaligned version of CMU-MOSEI. The results are shown in Table 3.4.

First, we consider the performance for only using unimodal transformers (i.e., language, audio or vision only). We find that the language transformer outperforms the other two by a large margin. For example, for the Acc$_2^h$ metric, the model improves from 65.6 to 77.4 when comparing audio only to language only unimodal transformer. This fact aligns with the observations in prior work [Pham et al., 2019], where the

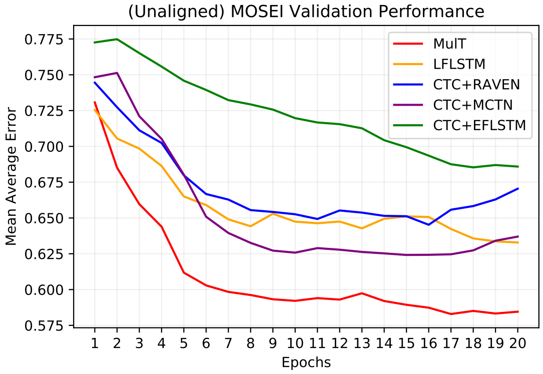


Figure 3.5: Validation set convergence of MulT when compared to other baselines on the unaligned CMU-MOSEI task.

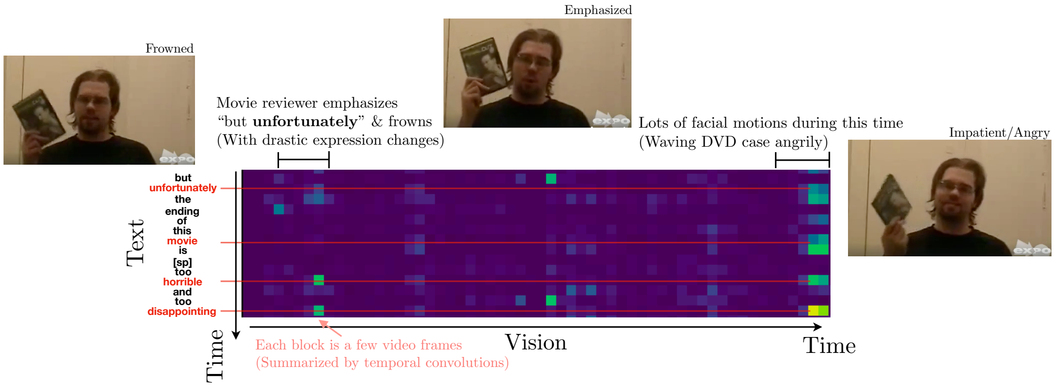

Frowned
Emphasized
Impatient/Angry
Movie reviewer emphasizes "but **unfortunately**" & frowns (With drastic expression changes)
Lots of facial motions during this time (Waving DVD case angrily)
Text
but
unfortunately
the
ending
of
this
movie
is
[sp]
too
horrible
and
too
disappointing
Time
Each block is a few video frames (Summarized by temporal convolutions)
Vision
Time

Figure 3.6: Visualization of sample crossmodal attention weights from layer 3 of $[V \rightarrow L]$ crossmodal transformer on CMU-MOSEI. We found that the crossmodal attention has learned to correlate certain meaningful words (e.g., "movie", "disappointing") with segments of stronger visual signals (typically stronger facial motions or expression change), despite the lack of alignment between original L/V sequences. Note that due to temporal convolution, each textual/visual feature contains the representation of nearby elements.

authors found that a good language network could already achieve good performance at inference time.

Second, we consider 1) a late-fusion transformer that feature-wise concatenates the last elements of three self-attention transformers; and 2) an early-fusion self-attention transformer that takes in a temporal concatenation of three asynchronous sequences $[\hat{X}_L, \hat{X}_V, \hat{X}_A] \in \mathbb{R}^{(T_L+T_V+T_A)\times d_q}$ (see Chapter 3.2.2). Empirically, we find that both EF- and LF-Transformer (which fuse multimodal signals) outperform unimodal transformers.

Finally, we study the importance of individual crossmodal transformers according to the target modalities (i.e., using $[V, A \rightarrow L]$, $[L, A \rightarrow V]$, or $[L, V \rightarrow A]$ network). As shown in Table 3.4, we find crossmodal attention modules consistently improve over the late- and early-fusion transformer models in most metrics on unaligned CMU-MOSEI. In particular, among the three crossmodal transformers, the one where language(L) is the target modality works best. We also additionally study the effect of adapting intermediate-level instead of the low-level features from source modality in crossmodal attention blocks (similar to the NMT encoder-decoder architecture but without self-attention; see Chapter 3.2.1). While

Description	(Unaligned) CMU-MOSEI				
	Sentiment				
	Acc_7^h	Acc_2^h	$\mathrm{F1}^h$	MAE^ℓ	Corr^h
Unimodal Transformers					
Language only	46.5	77.4	78.2	0.653	0.631
Audio only	41.4	65.6	68.8	0.764	0.310
Vision only	43.5	66.4	69.3	0.759	0.343
Late Fusion by using Multiple Unimodal Transformers					
LF-Transformer	47.9	78.6	78.5	0.636	0.658
Temporally Concatenated Early Fusion Transformer					
EF-Transformer	47.8	78.9	78.8	0.648	0.647
Multimodal Transfomers					
Only $[V, A \rightarrow L]$ (ours)	**50.5**	80.1	80.4	0.605	0.670
Only $[L, A \rightarrow V]$ (ours)	48.2	79.7	80.2	0.611	0.651
Only $[L, V \rightarrow A]$ (ours)	47.5	79.2	79.7	0.620	0.648
MulT mixing intermediate-level features (ours)	50.3	80.5	80.6	0.602	0.674
MulT (ours)	**50.7**	**81.6**	**81.6**	**0.591**	**0.691**

Table 3.4: An ablation study on the benefit of MulT's crossmodal transformers using CMU-MOSEI.).

MulT leveraging intermediate-level features still outperform models in other ablative settings, we empirically find adapting from low-level features works best. The ablations suggest that crossmodal attention concretely benefits MulT with better representation learning.

3.3.4 Qualitative Analysis

To understand how crossmodal attention works while modeling unaligned multimodal data, we empirically inspect what kind of signals MulT picks up by visualizing the attention activations. Figure 3.6 shows an example of a section of the crossmodal attention matrix on layer 3 of the $V \rightarrow L$ network of MulT (the original matrix has dimension $T_L \times T_V$; the figure shows the attention corresponding to approximately a 6-sec short window of that matrix). We find that crossmodal attention has learned to attend to meaningful signals across the two modalities. For example, stronger attention is given to the intersection of words that tend to suggest emotions (e.g., "movie", "disappointing") and drastic facial expression changes in the video (start and end of the above vision sequence). This observation advocates one of the aforementioned advantage of MulT over conventional alignment (see Chapter 3.2.3): crossmodal attention enables MulT to directly capture potentially long-range signals, including those off-diagonals on the attention matrix.

3.4 Discussion

In this chapter, we propose Multimodal Transformer (MulT) for analyzing human multimodal language to address the sub-challenge of synchronization and alignment in cross-view learning. At the heart of MulT is the crossmodal attention mechanism, which provides a latent crossmodal adaptation that fuses multimodal information by directly attending to low-level features in other modalities. Whereas prior approaches focused primarily on the aligned multimodal streams, MulT serves as a strong baseline capable of capturing long-range contingencies, regardless of the alignment assumption. Empirically, we show that MulT exhibits the best performance when compared to prior methods.

3.5 Appendix

3.5.1 Positional Embedding

A purely attention-based transformer network is *order-invariant*. In other words, permuting the order of an input sequence does not change transformer's behavior or alter its output. One solution to address this weakness is by embedding the positional information into the hidden units [Vaswani et al., 2017].

Following [Vaswani et al., 2017], we encode the positional information of a sequence of length T via the sin and cos functions with frequencies dictated by the feature index. In particular, we define the positional embedding (PE) of a sequence $X \in \mathbb{R}^{T \times d}$ (where T is length) as a matrix where:

$$\mathrm{PE}[i, 2j] = \sin\left(\frac{i}{10000^{\frac{2j}{d}}}\right)$$

$$\mathrm{PE}[i, 2j+1] = \cos\left(\frac{i}{10000^{\frac{2j}{d}}}\right)$$

for $i = 1, \ldots, T$ and $j = 0, \lfloor \frac{d}{2} \rfloor$. Therefore, each feature dimension (i.e., column) of PE are positional values that exhibit a sinusoidal pattern. Once computed, the positional embedding is added directly to the sequence so that $X + \mathrm{PE}$ encodes the elements' position information at every time step.

3.5.2 Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) [Graves et al., 2006] was first proposed for unsupervised Speech to Text alignment. Particularly, CTC is often combined with the output of recurrent neural network, which enables the model to train end-to-end and simultaneously infer speech-text alignment without supervision. For the ease of explanation, suppose the CTC module now are aiming at aligning an audio signal sequence $[a_1, a_2, a_3, a_4, a_5, a_6]$ with length 6 to a textual sequence “I am really really happy” with length 5. In this example, we refer to audio as the source and texts as target signal, noting that the sequence lengths may be different between the source to target; we also see that the output sequence may have repetitive element (i.e., “really”). The CTC [Graves et al., 2006] module we use comprises two components: alignment predictor and the CTC loss.

First, the alignment predictor is often chosen as a recurrent networks such as LSTM, which performs on the source sequence then outputs the possibility of being the unique words in the target sequence as well as a empty word (i.e., x). In our example, for each individual audio signal, the alignment predictor provides a vector of length 5 regarding the probability being aligned to [x, ‘I’, ‘am’, ‘really’, ‘happy’].

Next, the CTC loss considers the negative log-likelihood loss from only the proper alignment for the alignment predictor outputs. The proper alignment, in our example, can be results such as

i) [x, ‘I’, ‘am’, ‘really’, ‘really’, ‘happy’];
ii) [‘I’, ‘am’, x, ‘really’, ‘really’, ‘happy’];
iii) [‘I’, ‘am’, ‘really’, ‘really’, ‘really’, ‘happy’];
iv) [‘I’, ‘I’, ‘am’, ‘really’, ‘really’, ‘happy’]

In the meantime, some examples of the suboptimal/failure cases would be

i) [x, x, ‘am’, ‘really’, ‘really’, ‘happy’];
ii) [‘I’, ‘am’, ‘I’, ‘really’, ‘really’, ‘happy’];
iii) [‘I’, ‘am’, x, ‘really’, x, ‘happy’]

When the CTC loss is minimized, it implies the source signals are properly aligned to target signals.

	CMU-MOSEI	CMU-MOSI	IEMOCAP
Batch Size	16	128	32
Initial Learning Rate	1e-3	1e-3	2e-3
Optimizer	Adam	Adam	Adam
Transformers Hidden Unit Size d	40	40	40
# of Crossmodal Blocks D	4	4	4
# of Crossmodal Attention Heads	8	10	10
Temporal Convolution Kernel Size ($L/V/A$)	(1 or 3)/3/3	(1 or 3)/3/3	3/3/5
Textual Embedding Dropout	0.3	0.2	0.3
Crossmodal Attention Block Dropout	0.1	0.2	0.25
Output Dropout	0.1	0.1	0.1
Gradient Clip	1.0	0.8	0.8
# of Epochs	20	100	30

Table 3.5: Hyperparameters of Multimodal Transformer (MulT) we use for the various tasks. The “# of Crossmodal Blocks” and “# of Crossmodal Attention Heads” are for each transformer.

To sum up, in the experiments that adopting the CTC module, we train the alignment predictor while minimizing the CTC loss. Then, excluding the probability of blank words, we multiply the probability outputs from the alignment predictor to source signals. The source signal is hence resulting in a pseudo-aligned target singal. In our example, the audio signal is then transforming to a audio signal $[a'_1, a'_2, a'_3, a'_4, a'_5]$ with sequence length 5, which is pseudo-aligned to [’I’, ’am’, ’really’, ’really’, ’happy’].

3.5.3 Hyperparameters

Table 3.5 shows the settings of the various MulTs that we train on human multimodal language tasks. As previously mentioned, the models are contained at roughly the same sizes as in prior works for the purpose of fair comparison. For hyperparameters such as the dropout rate and number of heads in crossmodal attention module, we perform a basic grid search. We decay the learning rate by a factor of 10 when the validation performance plateaus.

3.5.4 Features

The features for multimodal datasets are extracted as follows:

- **Language.** We convert video transcripts into pre-trained Glove word embeddings (glove.840B.300d) [Pennington et al., 2014]. The embedding is a 300 dimensional vector.
- **Vision.** We use Facet [iMotions, 2017] to indicate 35 facial action units, which records facial muscle movement [Ekman, 1992, Ekman et al., 1980] for representing per-frame basic and advanced emotions.
- **Audio.** We use COVAREP [Degottex et al., 2014] for extracting low level acoustic features. The feature includes 12 Mel-frequency cepstral coefficients (MFCCs), pitch tracking and voiced/unvoiced segmenting features, glottal source parameters, peak slope parameters and maxima dispersion quotients. Dimension of the feature is 74.

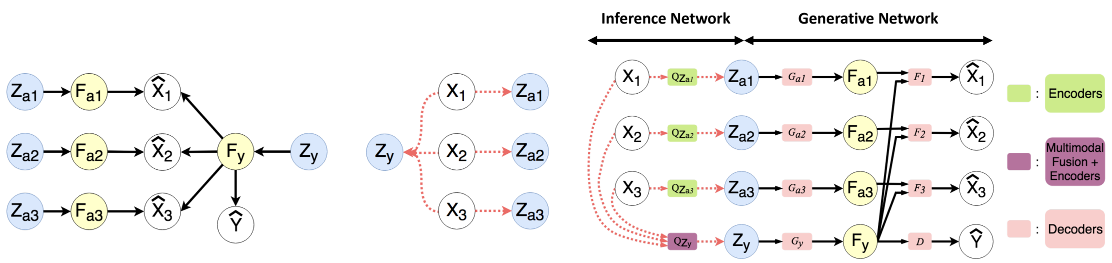
Chapter 4

Heterogeneous Structure - Complementary Factors Disentanglement

In this chapter, we will also use human communication as the cross-view data that contain language, visual and acoustic modalities. We will discuss the sub-challenge of complementary factors disentanglement within the challenge of heterogeneous structure. Although the presence of multiple modalities provides additional valuable information, there are two key difficulties to address when learning from multimodal data: 1) models must learn the complex intra-modal and cross-modal interactions for prediction [Zadeh et al., 2017], and 2) trained models must be robust to unexpected missing or noisy modalities during testing [Ngiam et al., 2011].

We propose to optimize for a joint generative-discriminative objective across multimodal data and labels. The discriminative objective ensures that the representations learned are rich in intra-modal and cross-modal features useful towards predicting the label, while the generative objective allows the model to infer missing modalities at test time and deal with the presence of noisy modalities. To this end, we introduce the Multimodal Factorization Model (MFM in Figure 4.1) that factorizes multimodal representations into *multimodal discriminative* factors and *modality-specific generative* factors. Multimodal discriminative factors are shared across all modalities and contain joint multimodal features required for discriminative tasks. Modality-specific generative factors are unique for each modality and contain the information required for generating each modality. We believe that factorizing multimodal representations into different explanatory factors can help each factor focus on learning from a subset of the joint information across multimodal data and labels. This method is in contrast to jointly learning a single factor that summarizes all generative and discriminative information [Srivastava and Salakhutdinov, 2012]. To sum up, MFM defines a joint distribution over multimodal data, and by the conditional independence assumptions in the assumed graphical model, both generative and discriminative aspects are taken into account. Our model design further provides interpretability of the factorized representations.

Through an extensive set of experiments, we show that MFM learns improved multimodal representations with these characteristics: 1) The multimodal discriminative factors achieve state-of-the-art or competitive performance on six multimodal time series datasets. We also demonstrate that MFM can generalize by integrating it with other existing multimodal discriminative models. 2) MFM allows flexible generation concerning multimodal discriminative factors (labels) and modality-specific generative factors (styles). We further show that we can perform reconstruction of missing modalities from observed modalities without significantly impacting discriminative performance. Finally, we interpret our learned representations using information-based and gradient-based methods, allowing us to understand the contributions of individual factors towards multimodal prediction and generation.



(a) MFM Generative Network

(b) MFM Inference Network

(c) MFM Neural Architecture

Figure 4.1: Illustration of the proposed Multimodal Factorization Model (MFM) with three modalities. MFM factorizes multimodal representations into *multimodal discriminative* factors $\mathbf{F_y}$ and *modality-specific generative* factors $\mathbf{F_a}_{\{1:M\}}$. (a) MFM Generative Network with latent variables $\{\mathbf{Z_y}, \mathbf{Z_a}_{\{1:M\}}\}$, factors $\{\mathbf{F_y}, \mathbf{F_a}_{\{1:M\}}\}$, generated multimodal data $\hat{\mathbf{X}}_{1:3}$ and labels $\hat{\mathbf{Y}}$. (b) MFM Inference Network. (c) MFM Neural Architecture. Best viewed zoomed in and in color.

4.1 Multimodal Factorization Model

Multimodal Factorization Model (MFM) is a latent variable model (Figure 4.1(a)) with conditional independence assumptions over multimodal discriminative factors and modality-specific generative factors. According to these assumptions, we propose a factorization over the joint distribution of multimodal data (Chapter 4.1.1). Since exact posterior inference on this factorized distribution can be intractable, we propose an approximate inference algorithm based on minimizing a joint-distribution Wasserstein distance over multimodal data (Chapter 4.1.2). Finally, we derive the MFM objective by approximating the joint-distribution Wasserstein distance via a generalized mean-field assumption.

Notation: We define $\mathbf{X}_{1:M}$ as the multimodal data from M modalities and $\mathbf{Y}$ as the labels, with joint distribution $P_{\mathbf{X}_{1:M},\mathbf{Y}} = P(\mathbf{X}_{1:M}, \mathbf{Y})$. Let $\hat{\mathbf{X}}_{1:M}$ denote the generated multimodal data and $\hat{\mathbf{Y}}$ denote the generated labels, with joint distribution $P_{\hat{\mathbf{X}}_{1:M},\hat{\mathbf{Y}}} = P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}})$.

4.1.1 Factorized Multimodal Representations

To factorize multimodal representations into multimodal discriminative factors and modality-specific generative factors, MFM assumes a Bayesian network structure as shown in Figure 4.1(a). In this graphical model, factors $\mathbf{F_y}$ and $\mathbf{F_a}_{\{1:M\}}$ are generated from mutually independent latent variables $\mathbf{Z} = [\mathbf{Z_y}, \mathbf{Z_a}_{\{1:M\}}]$ with prior $P_\mathbf{Z}$. In particular, $\mathbf{Z_y}$ generates the multimodal discriminative factor $\mathbf{F_y}$ and $\mathbf{Z_a}_{\{1:M\}}$ generate modality-specific generative factors $\mathbf{F_a}_{\{1:M\}}$. By construction, $\mathbf{F_y}$ contributes to the generation of $\hat{\mathbf{Y}}$ while $\{\mathbf{F_y}, \mathbf{F_a}_i\}$ both contribute to the generation of $\hat{\mathbf{X}}_i$. As a result, the joint distribution $P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}})$ can be factorized as follows:

$$
\begin{aligned}
P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}}) &= \int_{\mathbf{F},\mathbf{Z}} P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}} | \mathbf{F}) P(\mathbf{F}|\mathbf{Z}) P(\mathbf{Z}) d\mathbf{F} d\mathbf{Z} \\
= \int_{\substack{\mathbf{F_y}, \mathbf{F_a}_{\{1:M\}} \\ \mathbf{Z_y}, \mathbf{Z_a}_{\{1:M\}}}} &\Big(P(\hat{\mathbf{Y}}|\mathbf{F_y}) \prod_{i=1}^{M} P(\hat{\mathbf{X}}_i | \mathbf{F_a}_i, \mathbf{F_y}) \Big) \Big(P(\mathbf{F_y}|\mathbf{Z_y}) \prod_{i=1}^{M} P(\mathbf{F_a}_i | \mathbf{Z_a}_i) \Big) \Big(P(\mathbf{Z_y}) \prod_{i=1}^{M} P(\mathbf{Z_a}_i) \Big) d\mathbf{F} d\mathbf{Z},
\end{aligned}
\tag{4.1}
$$

with $d\mathbf{F} = d\mathbf{F_y} \prod_{i=1}^{M} d\mathbf{F_a}_i$ and $d\mathbf{Z} = d\mathbf{Z_y} \prod_{i=1}^{M} d\mathbf{Z_a}_i$.
Exact posterior inference in Equation 4.1 may be analytically intractable due to the integration over $\mathbf{Z}$. We therefore resort to using an approximate inference distribution $Q(\mathbf{Z}|\mathbf{X}_{1:M}, \mathbf{Y})$. As a result, MFM can be viewed as an autoencoding structure that consists of encoder (inference) and decoder (generative) modules (Figure 4.1(c)). The encoder module for $Q(\cdot|\cdot)$ allows us to easily sample $\mathbf{Z}$ from an approximate

posterior. The decoder modules are parametrized according to the factorization of $P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}}|\mathbf{Z})$ as given by Equation 4.1 and Figure 4.1(a).

4.1.2 Minimizing Joint-Distribution Wasserstein Distance over Multimodal Data

Two common choices for approximate inference in autoencoding structures are Variational Autoencoders (VAEs) [Kingma and Welling, 2013] and Wasserstein Autoencoders (WAEs) [Tolstikhin et al., 2017, Zhao et al., 2017]. The former optimizes the evidence lower bound objective (ELBO), and the latter derives an approximation for the primal form of the Wasserstein distance. We consider the latter since it simultaneously results in better latent factor disentanglement [Rubenstein et al., 2018, Zhao et al., 2017] and better sample generation quality than its counterparts [Chen et al., 2016, Higgins et al., 2016, Kingma and Welling, 2013]. However, WAEs are designed for unimodal data and do not consider factorized distributions over latent variables that generate multimodal data. Therefore, we propose a variant for handling factorized joint distributions over multimodal data.

As suggested by Kingma and Welling [2013], we adopt the design of nonlinear mappings (i.e. neural network architectures) in the encoder and decoder (Figure 4.1 (c)). For the encoder $Q(\mathbf{Z}|\mathbf{X}_{1:M}, \mathbf{Y})$, we learn a deterministic mapping $Q_{enc} : \mathbf{X}_{1:M}, \mathbf{Y} \to \mathbf{Z}$ [Rubenstein et al., 2018, Tolstikhin et al., 2017]. For the decoder, we define the generation process from latent variables as $G_y : \mathbf{Z_y} \to \mathbf{F_y}$, $G_{a\{1:M\}} : \mathbf{Z}_{\mathbf{a}\{1:M\}} \to \mathbf{F}_{\mathbf{a}\{1:M\}}$, $D : \mathbf{F_y} \to \hat{\mathbf{Y}}$, and $F_{1:M} : \mathbf{F_y}, \mathbf{F}_{\mathbf{a}\{1:M\}} \to \hat{\mathbf{X}}_{1:M}$, where $G_y, G_{a\{1:M\}}, D$ and $F_{1:M}$ are deterministic functions parametrized by neural networks.

Let $W_c(P_{\mathbf{X}_{1:M},\mathbf{Y}}, P_{\hat{\mathbf{X}}_{1:M},\hat{\mathbf{Y}}})$ denote the joint-distribution Wasserstein distance over multimodal data under cost function c_{X_i} and c_Y. We choose the squared cost $c(a, b) = \|a - b\|_2^2$, allowing us to minimize the 2-Wasserstein distance. The cost function can be defined not only on static data but also on time series data such as text, audio and videos. For example, given time series data $\mathbf{X} = [X^1, X^2, \cdots, X^T]$ and $\hat{\mathbf{X}} = [\hat{X}^1, \hat{X}^2, \cdots, \hat{X}^T]$, we define $c(\mathbf{X}, \hat{\mathbf{X}}) = \sum_{t=1}^{T} \|X^t - \hat{X}^t\|_2^2$.

With conditional independence assumptions in Equation 4.1, we express $W_c(P_{\mathbf{X}_{1:M},\mathbf{Y}}, P_{\hat{\mathbf{X}}_{1:M},\hat{\mathbf{Y}}})$ as:

Proposition 1. For any functions $G_y : \mathbf{Z_y} \to \mathbf{F_y}$, $G_{a\{1:M\}} : \mathbf{Z}_{\mathbf{a}\{1:M\}} \to \mathbf{F}_{\mathbf{a}\{1:M\}}$, $D : \mathbf{F_y} \to \hat{\mathbf{Y}}$, and $F_{1:M} : \mathbf{F}_{\mathbf{a}\{1:M\}}, \mathbf{F_y} \to \hat{\mathbf{X}}_{1:M}$, we have $W_c(P_{\mathbf{X}_{1:M},\mathbf{Y}}, P_{\hat{\mathbf{X}}_{1:M},\hat{\mathbf{Y}}}) =$

$$\inf_{Q_{\mathbf{Z}}=P_{\mathbf{Z}}} \mathbf{E}_{P_{\mathbf{X}_{1:M},\mathbf{Y}}} \mathbf{E}_{Q(\mathbf{Z}|\mathbf{X}_{1:M},\mathbf{Y})} \left[\sum_{i=1}^{M} c_{X_i}\Big(\mathbf{X}_i, F_i\big(G_{ai}(\mathbf{Z}_{\mathbf{a}i}), G_y(\mathbf{Z_y})\big)\Big) + c_Y\Big(\mathbf{Y}, D\big(G_y(\mathbf{Z_y})\big)\Big) \right], \tag{4.2}$$

where $P_{\mathbf{Z}}$ is the prior over $\mathbf{Z} = [\mathbf{Z_y}, \mathbf{Z}_{\mathbf{a}\{1,M\}}]$ and $Q_{\mathbf{Z}}$ is the aggregated posterior of the proposed approximate inference distribution $Q(\mathbf{Z}|\mathbf{X}_{1:M}, \mathbf{Y})$.

Proof. The proof is adapted from Tolstikhin et al. [2017]. The two differences are: (1) we show that $P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}}|\mathbf{Z} = z)$ are Dirac for all $z \in \mathcal{Z}$, and (2) we use the fact that $c((\mathbf{X}_{1:M}, \mathbf{Y}), (\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}})) = \sum_{i=1}^{M} c_{Xi}(\mathbf{X}_i, \hat{\mathbf{X}}_i) + c_Y(\mathbf{Y}, \hat{\mathbf{Y}})$. Please refer to the Chapter 4.5.1 for proof details. ∎

The constraint on $Q_{\mathbf{Z}} = P_{\mathbf{Z}}$ in Proposition 1 is hard to satisfy. To obtain a numerical solution, we first relax the constraint by performing a generalized mean field assumption on Q according to the conditional independence as shown in the inference network of Figure 4.1 (b):

$$Q(\mathbf{Z}|\mathbf{X}_{1:M}, \mathbf{Y}) := Q(\mathbf{Z}|\mathbf{X}_{1:M}) := Q(\mathbf{Z_y}|\mathbf{X}_{1:M}) \prod_{i=1}^{M} Q(\mathbf{Z}_{\mathbf{a}i}|\mathbf{X}_i). \tag{4.3}$$

The intuition here is based on our design that $\mathbf{Z_y}$ generates the multimodal discriminative factor $\mathbf{F_y}$ and $\mathbf{Z}_{\mathbf{a}\{1:M\}}$ generate modality-specific generative factors $\mathbf{F}_{\mathbf{a}\{1:M\}}$. Therefore, the inference for $\mathbf{Z_y}$ should depend on all modalities $\mathbf{X}_{1:M}$ and the inference for $\mathbf{Z}_{\mathbf{a}i}$ should depend only on the specific modality $\mathbf{X}_i$. Following this assumption, we define $\mathcal{Q}$ as a nonparametric set of all encoders that fulfill the factorization in Equation 4.3. A penalty term is added into our objective to find the $Q(\mathbf{Z}|\cdot) \in \mathcal{Q}$ that is the closest to prior $P_{\mathbf{Z}}$, thereby approximately enforcing the constraint $Q_{\mathbf{Z}} = P_{\mathbf{Z}}$:

$$\begin{aligned}&\min_{F,G_{a\{1:M\}},G_y,D} \inf_{Q(\mathbf{Z}|\cdot)\in\mathcal{Q}} \mathbf{E}_{P_{\mathbf{X}_{1:M},\mathbf{Y}}}\mathbf{E}_{Q(\mathbf{Z}_{\mathbf{a}1}|\mathbf{X}_1)}\cdots\mathbf{E}_{Q(\mathbf{Z}_{\mathbf{a}M}|\mathbf{X}_M)}\mathbf{E}_{Q(\mathbf{Z}_{\mathbf{y}}|\mathbf{X}_{1:M})}\\ &\left[\sum_{i=1}^{M} c_{X_i}\Big(\mathbf{X}_i, F\big(G_{ai}(\mathbf{Z}_{\mathbf{a}i}), G_y(\mathbf{Z}_{\mathbf{y}})\big)\Big) + c_Y\Big(\mathbf{Y}, D\big(G_y(\mathbf{Z}_{\mathbf{y}})\big)\Big)\right] + \lambda\mathcal{MMD}(Q_{\mathbf{Z}}, P_{\mathbf{Z}}),\end{aligned} \tag{4.4}$$

where λ is a hyper-parameter and $\mathcal{MMD}$ is the Maximum Mean Discrepancy [Gretton et al., 2012] as a divergence measure between $Q_{\mathbf{Z}}$ and $P_{\mathbf{Z}}$. The prior $P_{\mathbf{Z}}$ is chosen as a centered isotropic Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, so that it implicitly enforces independence between the latent variables $\mathbf{Z} = [\mathbf{Z}_{\mathbf{y}}, \mathbf{Z}_{\mathbf{a}\{1,M\}}]$ [Higgins et al., 2016, Kingma and Welling, 2013, Rubenstein et al., 2018].

Equation 4.4 represents our hybrid generative-discriminative optimization objective over multimodal data: the first loss term $\sum_{i=1}^{M} c_{X_i}(\mathbf{X}_i, F(G_{ai}(\mathbf{Z}_{\mathbf{a}i}), G_y(\mathbf{Z}_{\mathbf{y}})))$ is the generative objective based on reconstruction of multimodal data and the second term $c_Y(\mathbf{Y}, D(G_y(\mathbf{Z}_{\mathbf{y}})))$ is the discriminative objective. In practice we compute the expectations in Equation 4.4 using empirical estimates over the training data. The neural architecture of MFM is illustrated in Figure 4.1(c).

4.1.3 Surrogate Inference for Missing Modalities

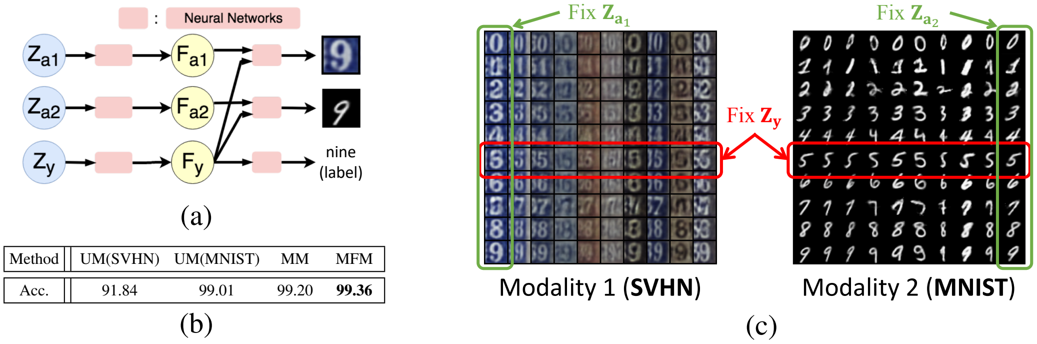
A key difficulty in multimodal learning involves dealing with missing modalities. A good multimodal model should be able to infer the missing modality conditioned on the observed modalities and perform predictions based only on the observed modalities. To achieve this objective, the inference process of MFM can be easily adapted using a surrogate inference network to reconstruct the missing modality given the observed modalities. Formally, let Φ denote the surrogate inference network. The generation of missing modality $\hat{\mathbf{X}}_1$ given the observed modalities $\mathbf{X}_{2:M}$ can be formulated as

$$\begin{aligned}\Phi^* &= \operatorname*{argmin}_{\Phi} \mathbf{E}_{P_{\mathbf{X}_{2:M},\hat{\mathbf{X}}_1}}\Big(-\log P_\Phi(\hat{\mathbf{X}}_1|\mathbf{X}_{2:M})\Big)\\ \text{with } P_\Phi(\hat{\mathbf{X}}_1|\mathbf{X}_{2:M}) &:= \int P(\hat{\mathbf{X}}_1|\mathbf{Z}_{\mathbf{a}1}, \mathbf{Z}_{\mathbf{y}})Q_\Phi(\mathbf{Z}_{\mathbf{a}1}|\mathbf{X}_{2:M})Q_\Phi(\mathbf{Z}_{\mathbf{y}}|\mathbf{X}_{2:M})d\mathbf{Z}_{\mathbf{a}1}d\mathbf{Z}_{\mathbf{y}}.\end{aligned} \tag{4.5}$$

Similar to Chapter 4.1.2, we use deterministic mappings in $Q_\Phi(\cdot|\cdot)$ and $Q_\Phi(\mathbf{Z}_{\mathbf{y}}|\cdot)$ is also used for prediction $P_\Phi(\hat{\mathbf{Y}}|\mathbf{X}_{2:M}) := \int P(\hat{\mathbf{Y}}|\mathbf{Z}_{\mathbf{y}})Q_\Phi(\mathbf{Z}_{\mathbf{y}}|\mathbf{X}_{2:M})d\mathbf{Z}_{\mathbf{y}}$. Equation 4.5 suggests that in the presence of missing modalities, we only need to infer the latent codes rather than the entire modality.

4.1.4 Encoder and Decoder Design

We now discuss the implementation choices for the MFM neural architecture in Figure 4.1(c). The encoder $Q(\mathbf{Z}_{\mathbf{y}}|\mathbf{X}_{1:M})$ can be parametrized by any model that performs multimodal fusion [Morency et al., 2011, Zadeh et al., 2017]. For multimodal image datasets, we adopt Convolutional Neural Networks (CNNs) and Fully-Connected Neural Networks (FCNNs) with late fusion [Nojavanasghari et al., 2016] as our encoder $Q(\mathbf{Z}_{\mathbf{y}}|\mathbf{X}_{1:M})$. The remaining functions in MFM are also parametrized by CNNs and FCNNs. For multimodal time series datasets, we choose the Memory Fusion Network (MFN) [Zadeh et al., 2018a] as our multimodal encoder $Q(\mathbf{Z}_{\mathbf{y}}|\mathbf{X}_{1:M})$. We use Long Short-term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997] for functions $Q(\mathbf{Z}_{\mathbf{a}\{1:M\}}|\mathbf{X}_{1:M})$, decoder LSTM networks [Cho et al., 2014] for functions $F_{1:M}$, and FCNNs for functions G_y, $G_{a\{1:M\}}$ and D. Details are provided in the Chapter 4.5.5 and 4.5.6 and the code is available at `https://github.com/pliang279/factorized/`.



(a)

Method	UM(SVHN)	UM(MNIST)	MM	MFM
Acc.	91.84	99.01	99.20	**99.36**

(b)

(c)

Figure 4.2: (a) MFM generative network for multimodal image dataset SVHN+MNIST, (b) unimodal and multimodal classification accuracies, and (c) conditional generation for SVHN and MNIST digits. MFM shows improved capabilities in digit prediction as well as flexible generation of both images based on labels and styles.

4.2 Experiments

In order to show that MFM learns multimodal representations that are discriminative, generative and interpretable, we design the following experiments. We begin with a multimodal synthetic image dataset that allows us to examine whether MFM displays discriminative and generative capabilities from factorized latent variables. Utilizing image datasets allows us to clearly visualize the generative capabilities of MFM. We then transition to six more challenging real-world multimodal video datasets to 1) rigorously evaluate the discriminative capabilities of MFM in comparison with existing baselines, 2) analyze the importance of each design component through ablation studies, 3) assess the robustness of MFM's modality reconstruction and prediction capabilities to missing modalities, and 4) interpret the learned representations using information-based and gradient-based methods to understand the contributions of individual factors towards multimodal prediction and generation.

4.2.1 Multimodal Synthetic Image Dataset

Here, we study MFM on a synthetic image dataset that considers SVHN [Netzer et al., 2011] and MNIST [Lecun et al., 1998] as the two modalities. SVHN and MNIST are images with different styles but the same labels (digits $0 \sim 9$). We randomly pair $100,000$ SVHN and MNIST images that have the same label, creating a multimodal dataset which we call SVHN+MNIST. $80,000$ pairs are used for training and the rest for testing. To justify that MFM is able to learn improved multimodal representations, we show both classification and generation results on SVHN+MNIST in Figure 4.2.

Prediction: We perform experiments on both unimodal and multimodal classification tasks. UM denotes a unimodal baseline that performs prediction given only one modality as input and MM denotes a multimodal discriminative baseline that performs prediction given both images [Nojavanasghari et al., 2016]. We compare the results for UM(SVHN), UM(MNIST), MM and MFM on SVHN+MNIST in Figure 4.2(b). We achieve better classification performance from unimodal to multimodal which is not surprising since more information is given. More importantly, MFM outperforms MM, which suggests that MFM learns improved factorized representations for discriminative tasks.

Generation: We generate images using the MFM generative network (Figure 4.2(a)). We fix one variable out of $\mathbf{Z} = [\mathbf{Z}_{\mathbf{a}1}, \mathbf{Z}_{\mathbf{a}2}, \text{and } \mathbf{Z}_{\mathbf{y}}]$ and randomly sample the other two variables from prior $P_{\mathbf{Z}}$. From Figure 4.2(c), we observe that MFM shows flexible generation of SVHN and MNIST images based on labels and styles. This suggests that MFM is able to factorize multimodal representations into multimodal discriminative factors (labels) and modality-specific generative factors (styles).

Table 4.1: Results for multimodal speaker traits recognition on POM, multimodal sentiment analysis on CMU-MOSI, ICT-MMMO, YouTube, MOUD, and multimodal emotion recognition on IEMOCAP. SOTA1 and SOTA2 refer to the previous best and second best state-of-the-art respectively, and Δ_{SOTA} shows improvement over SOTA1. Symbols depict the baseline giving the result: # *MFN*, ‡ *MARN*, ∗ *TFN*, † *BC-LSTM*, ◇ *MV-LSTM*, S *EF-LSTM*, ♭ *DF*, ♡ *SVM*, • *RF*. For detailed tables with results for all models, please refer to the Chapter 4.5.2.

Dataset	POM Personality Traits															
Task	Con	Pas	Voi	Dom	Cre	Viv	Exp	Ent	Res	Tru	Rel	Out	Tho	Ner	Per	Hum
Metric	r															
SOTA2	$0.359^{\dagger}$	$0.425^{\dagger}$	$0.166^{\ddagger}$	$0.235^{\ddagger}$	$0.358^{\dagger}$	$0.417^{\dagger}$	$0.450^{\dagger}$	$0.378^{\ddagger}$	$0.295^{\diamond}$	$0.237^{\diamond}$	$0.215^{\ddagger}$	$0.238^{\diamond}$	$0.363^{\dagger}$	$0.258^{\diamond}$	$0.344^{\dagger}$	$0.319^{\dagger}$
SOTA1	$0.395^{\#}$	$0.428^{\#}$	$0.193^{\#}$	$0.313^{\#}$	$0.367^{\#}$	$0.431^{\#}$	$0.452^{\#}$	$0.395^{\#}$	$0.333^{\#}$	$\mathbf{0.296}^{\#}$	$0.255^{\#}$	$0.259^{\#}$	$0.381^{\#}$	$0.318^{\#}$	$\mathbf{0.377}^{\#}$	$0.386^{\#}$
MFM	**0.431**	**0.450**	**0.197**	**0.411**	**0.380**	**0.448**	**0.467**	**0.452**	**0.368**	0.212	**0.309**	**0.333**	**0.404**	**0.333**	0.334	**0.408**
Δ_{SOTA}	↑ 0.036	↑ 0.022	↑ 0.004	↑ 0.097	↑ 0.013	↑ 0.017	↑ 0.015	↑ 0.057	↑ 0.035	–	↑ 0.054	↑ 0.074	↑ 0.023	↑ 0.015	–	↑ 0.022

Dataset	CMU-MOSI					ICT-MMMO		YouTube		MOUD	
Task	Sentiment					Sentiment		Sentiment		Sentiment	
Metric	Acc_7	Acc_2	F1	MAE	r	Acc_2	F1	Acc_3	F1	Acc_2	F1
SOTA2	$34.1^{\#}$	$77.1^{\ddagger}$	$77.0^{\ddagger}$	$0.968^{\ddagger}$	$0.625^{\ddagger}$	72.5^{*}	72.6^{*}	$48.3^{\ddagger}$	$45.1^{\dagger}$	$81.1^{\#}$	$80.4^{\#}$
SOTA1	$34.7^{\ddagger}$	$77.4^{\#}$	$77.3^{\#}$	$0.965^{\#}$	$0.632^{\#}$	$73.8^{\#}$	$73.1^{\#}$	$51.7^{\#}$	$51.6^{\#}$	$81.1^{\ddagger}$	$81.2^{\ddagger}$
MFM	**36.2**	**78.1**	**78.1**	**0.951**	**0.662**	**81.3**	**79.2**	**53.3**	**52.4**	**82.1**	**81.7**
Δ_{SOTA}	↑ 1.5	↑ 0.7	↑ 0.8	↓ 0.014	↑ 0.030	↑ 7.5	↑ 6.1	↑ 1.6	↑ 0.8	↑ 1.0	↑ 0.5

Dataset	IEMOCAP Emotions											
Task	Happy		Sad		Angry		Frustrated		Excited		Neutral	
Metric	Acc_2	F1	Acc_2	F1	Acc_2	F1	Acc_2	F1	Acc_2	F1	Acc_2	F1
SOTA2	$86.7^{\ddagger}$	84.2^{S}	83.4^{*}	$81.7^{\dagger}$	$85.1^{\diamond}$	84.5^{S}	$79.5^{\ddagger}$	$76.6^{\ddagger}$	$89.6^{\ddagger}$	$86.3^{\#}$	68.8^{S}	67.1^{S}
SOTA1	$90.1^{\#}$	$85.3^{\#}$	$85.8^{\#}$	82.8^{*}	$87.0^{\#}$	$86.0^{\#}$	$80.3^{\#}$	$\mathbf{76.8}^{\#}$	$89.8^{\#}$	$\mathbf{87.1}^{\ddagger}$	$71.8^{\#}$	$\mathbf{68.5}^{S}$
MFM	**90.2**	**85.8**	**88.4**	**86.1**	**87.5**	**86.7**	**80.4**	74.5	**90.0**	**87.1**	**72.1**	68.1
Δ_{SOTA}	↑ 0.1	↑ 0.5	↑ 2.6	↑ 3.3	↑ 0.5	↑ 0.7	↑ 0.1	–	↑ 0.2	–	↑ 0.3	–

4.2.2 Multimodal Time Series Datasets

Now, we transition to more challenging multimodal time series datasets. All the datasets consist of monologue videos. Features are extracted from the language (GloVe word embeddings [Pennington et al., 2014]), visual (Facet [iMotions, 2017]), and acoustic (COVAREP [Degottex et al., 2014]) modalities. For a detailed description of feature extraction, please refer to the Chapter 4.5.3.

We consider the following six datasets across three domains: 1) Multimodal Personality Trait Recognition: **POM** [Park et al., 2014] contains 903 movie review videos annotated for the following personality traits: confident (con), passionate (pas), voice pleasant (voi), dominant (dom), credible (cre), vivid (viv), expertise (exp), entertaining (ent), reserved (res), trusting (tru), relaxed (rel), outgoing (out), thorough (tho), nervous (ner), persuasive (per) and humorous (hum). The short form is indicated in parenthesis. 2) Multimodal Sentiment Analysis: **CMU-MOSI** [Zadeh et al., 2016] is a collection of 2199 monologue opinion video clips annotated with sentiment. **ICT-MMMO** [Wöllmer et al., 2013] consists of 340 online social review videos annotated for sentiment. **YouTube** [Morency et al., 2011] contains 269 product review and opinion video segments from YouTube each annotated for sentiment. **MOUD** [Perez-Rosas et al., 2013] consists of 79 product review videos in Spanish. Each video consists of multiple segments labeled as either positive, negative or neutral sentiment. 3) Multimodal Emotion Recognition: **IEMOCAP** [Busso et al., 2008a] consists of 302 videos of recorded dyadic dialogues. The videos are divided into multiple segments each annotated for the presence of 6 discrete emotions (happy, sad, angry, frustrated, excited and neutral), resulting in a total of 7318 segments in the dataset. We report results using the following metrics: Acc_C = multiclass accuracy across C classes, F1 = F1 score, MAE = Mean Absolute Error, r = Pearson's correlation.

Prediction: We first compare the performance of MFM with existing multimodal prediction methods.

Model	Multimodal Disc. Factor	Hybrid Gen.-Disc. Objective	Factorized Gen.-Disc. Factors	Mod.-Spec. Gen. Factors	CMU-MOSI							
					$\hat{\mathbf{X}}$. Reconstruction			$\hat{\mathbf{Y}}$ Prediction				
					MSE (ℓ)	MSE (a)	MSE (v)	Acc_7	Acc_2	F1	MAE	r
$\mathbf{M_A}$	no	no	-	-	-	-	-	33.2	75.2	75.2	1.020	0.616
$\mathbf{M_B}$	yes	no	-	-	-	-	-	34.1	77.4	77.3	0.965	0.632
$\mathbf{M_C}$	no	yes	no	-	0.0413	0.0509	0.0220	34.8	75.9	76.0	0.979	0.640
$\mathbf{M_D}$	yes	yes	no	-	0.0413	0.0486	0.0223	35.0	77.4	77.2	0.960	0.649
$\mathbf{M_E}$	yes	yes	yes	no	0.0397	0.0452	0.0211	35.9	77.3	77.2	0.956	0.661
MFM	yes	yes	yes	yes	**0.0391**	**0.0384**	**0.0183**	**36.2**	**78.1**	**78.1**	**0.951**	**0.662**

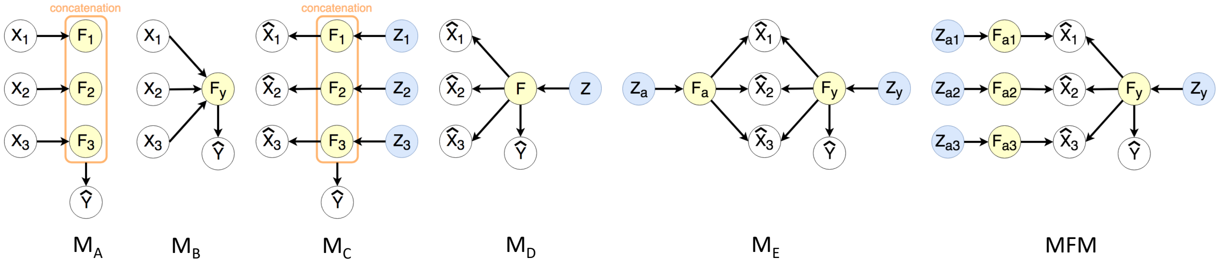


Figure 4.3: Models used in the ablation studies of MFM. Each model removes a design component from our model. Modality reconstruction and sentiment prediction results are reported on CMU-MOSI with best results in bold. Factorizing multimodal representations into multimodal discriminative factors and modality-specific generative factors are crucial for improved performance.

For a detailed description of the baselines, please refer to the Chapter 4.5.2. From Table 4.1, we first observe that the best performing baseline results are achieved by different models across different datasets (most notably MFN, MARN, and TFN). On the other hand, MFM consistently achieves state-of-the-art or competitive results for all six multimodal datasets. We believe that the multimodal discriminative factor $\mathbf{F_y}$ in MFM has successfully learned more meaningful representations by distilling discriminative features. This highlights the benefit of learning factorized multimodal representations towards discriminative tasks. Furthermore, MFM is *model-agnostic* and can be applied to other multimodal encoders $Q(\mathbf{Z_y}|\mathbf{X}_{1:M})$. We perform experiments to show consistent improvements in discriminative performance for several choices of the encoder: EF-LSTM [Morency et al., 2011] and TFN [Zadeh et al., 2017]. For Acc_2 on CMU-MOSI, our factorization framework improves the performance of EF-LSTM from 74.3 to **75.2** and TFN from 74.6 to **75.5**.

Ablation Study: In Figure 4.3, we present the models $\mathbf{M}_{\{\mathbf{A,B,C,D,E}\}}$ used for ablation studies. These models are designed to analyze the effects of using a multimodal discriminative factor, a hybrid generative-discriminative objective, factorized generative-discriminative factors and modality-specific generative factors towards both modality reconstruction and label prediction. The simplest variant is $\mathbf{M_A}$ which represents a purely discriminative model without a joint multimodal discriminative factor (i.e. early fusion [Morency et al., 2011]). $\mathbf{M_B}$ models a joint multimodal discriminative factor which incorporates more general multimodal fusion encoders [Zadeh et al., 2018a]. $\mathbf{M_C}$ extends $\mathbf{M_A}$ by optimizing a hybrid generative-discriminative objective over modality-specific factors. $\mathbf{M_D}$ extends $\mathbf{M_B}$ by optimizing a hybrid generative-discriminative objective over a joint multimodal factor (resembling prior work [Srivastava and Salakhutdinov, 2012]). $\mathbf{M_E}$ factorizes the representation into separate generative and discriminative factors. Finally, MFM is obtained from $\mathbf{M_E}$ by using modality-specific generative factors instead of a joint multimodal generative factor.

From the table in Figure 4.3, we observe the following general trends. For sentiment prediction, using 1) a multimodal discriminative factor outperforms modality-specific discriminative factors ($\mathbf{M_D} > \mathbf{M_C}$, $\mathbf{M_B} > \mathbf{M_A}$), and 2) adding generative capabilities to the model improves performance ($\mathbf{M_C} > \mathbf{M_A}$, $\mathbf{M_E} > \mathbf{M_B}$). For both sentiment prediction and modality reconstruction, 3) factorizing into separate

Table 4.2: The effect of missing modalities on multimodal data reconstruction and sentiment prediction on CMU-MOSI. MFM with surrogate inference is able to better handle missing modalities during test time as compared to the purely generative (Seq2Seq) or purely discriminative baselines.

Task	$\hat{\mathbf{X}}$. Reconstruction			$\hat{\mathbf{Y}}$ Prediction				
Metric	MSE (ℓ)	MSE (a)	MSE (v)	Acc_7	Acc_2	F1	MAE	r
Purely Generative and Discriminative Baselines								
ℓ(anguage) missing	0.0411	-	-	19.4	59.6	59.7	1.386	0.225
a(udio) missing	-	0.0533	-	34.0	73.5	73.4	1.024	0.615
v(isual) missing	-	-	0.0220	33.7	75.4	75.4	0.996	0.634
Multimodal Factorization Model (MFM)								
ℓ(anguage) missing	0.0403	-	-	21.7	62.0	61.7	1.313	0.236
a(udio) missing	-	0.0468	-	35.4	74.3	74.3	1.011	0.603
v(isual) missing	-	-	0.0215	35.0	76.4	76.3	0.990	0.635
all present	**0.0391**	**0.0384**	**0.0182**	**36.2**	**78.1**	**78.1**	**0.951**	**0.662**

generative and discriminative factors improves performance ($\mathbf{M_E} > \mathbf{M_D}$), and 4) using modality-specific generative factors outperforms multimodal generative factors (MFM $> \mathbf{M_E}$). These observations support our design decisions of factorizing multimodal representations into multimodal discriminative factors and modality-specific generative factors.

Missing Modalities: We now evaluate the performance of MFM in the presence of missing modalities using the surrogate inference model as described in Chapter 4.1.3. We compare with two baselines: 1) a purely generative Seq2Seq model [Cho et al., 2014] Φ_G from observed modalities to missing modalities by optimizing $\mathbf{E}_{P_{\mathbf{X}_{1:M}}}(-\log P_{\Phi_D}(\mathbf{X}_1|\mathbf{X}_{2:M}))$, and 2) a purely discriminative model Φ_D from observed modalities to the label by optimizing $\mathbf{E}_{P_{\mathbf{X}_{2:M},\mathbf{Y}}}(-\log P_{\Phi_G}(\mathbf{Y}|\mathbf{X}_{2:M}))$. Both models are modified from MFM by using only the two observed modalities as input and not explicitly accounting for missing modalities. We compare the reconstruction error of each modality (language, visual and acoustic) as well as the performance on sentiment prediction.

Table 4.2 shows that MFM with missing modalities outperforms the generative (Φ_G) or discriminative baselines (Φ_D) in terms of modality reconstruction and sentiment prediction. Additionally, MFM with missing modalities performs close to MFM with all modalities observed. This fact indicates that MFM can learn representations that are relatively robust to missing modalities. In addition, discriminative performance is most affected when the language modality is missing, which is consistent with prior work which indicates that language is most informative in human multimodal language [Zadeh et al., 2017]. On the other hand, sentiment prediction is more robust to missing acoustic and visual features. Finally, we observe that reconstructing the low-level acoustic and visual features is easier as compared to the high-dimensional language features that contain high-level semantic meaning.

Interpretation of Multimodal Representations: We devise two methods to study how individual factors in MFM influence the dynamics of multimodal prediction and generation. These interpretation methods represent both overall trends and fine-grained analysis that could be useful towards deeper understandings of multimodal representation learning. For more details, please refer to the Chapter 4.5.4.

Firstly, an information-based interpretation method is chosen to summarize the contribution of each modality towards the multimodal representations. Since $\mathbf{F}_\mathbf{y}$ is a common cause of $\hat{\mathbf{X}}_{1:M}$, we can compare $\mathrm{MI}(\mathbf{F}_\mathbf{y}, \hat{\mathbf{X}}_1), \cdots, \mathrm{MI}(\mathbf{F}_\mathbf{y}, \hat{\mathbf{X}}_M)$, where $\mathrm{MI}(\cdot,\cdot)$ denotes the mutual information measure between $\mathbf{F}_\mathbf{y}$ and generated modality $\hat{\mathbf{X}}_i$. Higher $\mathrm{MI}(\mathbf{F}_\mathbf{y}, \hat{\mathbf{X}}_i)$ indicates greater contribution from $\mathbf{F}_\mathbf{y}$ to $\hat{\mathbf{X}}_i$. Figure 4.4 reports the ratios $r_i = \mathrm{MI}(\mathbf{F}_\mathbf{y}, \hat{\mathbf{X}}_i)/\mathrm{MI}(\mathbf{F}_{\mathbf{a}i}, \hat{\mathbf{X}}_i)$ which measure a normalized version of the mutual information between $\mathbf{F}_{\mathbf{a}i}$ and $\hat{\mathbf{X}}_i$. We observe that on CMU-MOSI, the language modality is most informative towards

Ratio	r_ℓ	r_v	r_a
CMU-MOSI	0.307	0.030	0.107

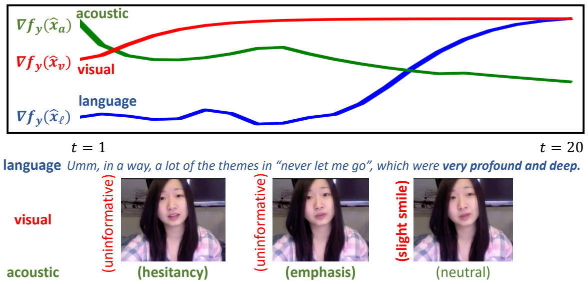


Figure 4.4: Analyzing the multimodal representations learnt in MFM via information-based (entire dataset) and gradient-based interpretation methods (single video) on CMU-MOSI.

sentiment prediction, followed by the acoustic modality. We believe that this result represents a prior over the expression of sentiment in human multimodal language and is closely related to the connections between language and speech [Kuhl, 2000]. Secondly, a gradient-based interpretation method to used analyze the contribution of each modality for every time step in multimodal time series data. We measure the gradient of the generated modality with respect to the target factors (e.g., $\mathbf{F_y}$). Let $\{x_1, x_2, \cdots, x_M\}$ denote multimodal time series data where x_i represents modality i, and $\hat{x}_i = [\hat{x}_i^1, \cdots, \hat{x}_i^t, \cdots, \hat{x}_i^T]$ denote generated modality i across time steps $t \in [1, T]$. The gradient $\nabla_{f_y}(\hat{x}_i)$ measures the extent to which changes in factor $f_y \sim P(\mathbf{F_y}|\mathbf{X}_{1:M} = x_{1:M})$ influences the generation of sequence $\hat{x}_i$. Figure 4.4 plots $\nabla_{f_y}(\hat{x}_i)$ for a video in CMU-MOSI. We observe that multimodal communicative behaviors that are indicative of speaker sentiment such as positive words (e.g. "very profound and deep") and informative acoustic features (e.g. hesitant and emphasized tone of voice) indeed correspond to increases in $\nabla_{f_y}(\hat{x}_i)$.

4.3 Related Work

The two main pillars of research in multimodal representation learning have considered the discriminative and generative objectives individually. Discriminative representation learning [Chaplot et al., 2017, Chen et al., 2017, Frome et al., 2013, Liang et al., 2018b, Socher et al., 2013, Tsai et al., 2017b] models the conditional distribution $P(\mathbf{Y}|\mathbf{X}_{1:M})$. Since these approaches are not concerned with modeling $P(\mathbf{X}_{1:M})$ explicitly, they use parameters more efficiently to model $P(\mathbf{Y}|\mathbf{X}_{1:M})$. For instance, recent works learn visual representations that are maximally dependent with linguistic attributes for improving one-shot image recognition [Tsai and Salakhutdinov, 2017] or introduce tensor product mechanisms to model interactions between the language, visual and acoustic modalities [Liu et al., 2018, Zadeh et al., 2017]. On the other hand, generative representation learning captures the interactions between modalities by modeling the joint distribution $P(\mathbf{X}_1, \cdots, \mathbf{X}_M)$ using either undirected graphical models [Srivastava and Salakhutdinov, 2012], directed graphical models [Suzuki et al., 2016], or neural networks [Sohn et al., 2014]. Some generative approaches compress multimodal data into lower-dimensional feature vectors which can be used for discriminative tasks [Ngiam et al., 2011, Pham et al., 2018]. To unify the advantages of both approaches, MFM factorizes multimodal representations into generative and discriminative components and optimizes for a joint objective.

Factorized representation learning resembles learning disentangled data representations which have been shown to improve the performance on many tasks [Bengio et al., 2013, Higgins et al., 2016, Kulkarni et al., 2015, Lake et al., 2017]. Several methods involve specifying a fixed set of latent attributes that individually control particular variations of data and performing supervised training [Cheung et al., 2014, Karaletsos et al., 2015, Reed et al., 2014, Yang et al., 2015, Zhu et al., 2014], assuming an isotropic Gaussian prior over latent variables to learn disentangled generative representations [Kingma and Welling, 2013, Rubenstein et al., 2018] and learning latent variables in charge of specific variations in the data by maximizing the mutual information between a subset of latent variables and the data [Chen et al., 2016]. However, these methods study factorization of a single modality. MFM factorizes multimodal representations and demonstrates the importance of modality-specific and multimodal factors towards generation and prediction. A concurrent and parallel work that factorizes latent factors in multimodal data was proposed by Hsu and Glass [2018]. They differ from us in the graphical model design, discriminative objective, prior matching criterion, and scale of experiments.

4.4 Discussion

In this chapter, we proposed the Multimodal Factorization Model (MFM) to address the sub-challenge of complemmentary factors disentanglement within the challenge of heterogeneous structure. MFM factorizes the multimodal representations into two sets of independent factors: *multimodal discriminative* factors and *modality-specific generative* factors. The multimodal discriminative factor achieves state-of-the-art or competitive results on six multimodal datasets. The modality-specific generative factors allow us to generate data based on factorized variables, account for missing modalities, and have a deeper understanding of the interactions involved in multimodal learning.

4.5 Appendix

4.5.1 Proof of Proposition 1

To simplify the proof, we first prove it for the unimodal case by considering the Wasserstein distance between $P_{\mathbf{X},\mathbf{Y}}$ and $P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}$.

4.5.1.1 Unimodal Joint-Distribution Wasserstein Distance

Proposition 2. For any functions $G_y : \mathbf{Z_y} \rightarrow \mathbf{F_y}$, $G_a : \mathbf{Z_a} \rightarrow \mathbf{F_a}$, $D : \mathbf{F_y} \rightarrow \hat{\mathbf{Y}}$, and $F : \mathbf{F_a}, \mathbf{F_y} \rightarrow \hat{\mathbf{X}}$, we have

$$W_c(P_{\mathbf{X},\mathbf{Y}}, P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}) = \inf_{Q_{\mathbf{Z}}=P_{\mathbf{Z}}} \mathbf{E}_{P_{\mathbf{X},\mathbf{Y}}} \mathbf{E}_{Q(\mathbf{Z}|\mathbf{X})} \left[c_X\Big(\mathbf{X}, F\big(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y})\big)\Big) + c_Y\Big(\mathbf{Y}, D\big(G_y(\mathbf{Z_y})\big)\Big) \right], \tag{4.6}$$

where W_c is the Wasserstein distance under cost function c_X and c_Y, $P_{\mathbf{Z}}$ is the prior over $\mathbf{Z} = [\mathbf{Z_a}, \mathbf{Z_y}]$ and $Q_{\mathbf{Z}}$ is the aggregated posterior of the proposed inference distribution $Q(\mathbf{Z}|\mathbf{X})$.

Proof:

To begin the proof, we abuse some notations as follows.

By definition, the Wasserstein distance under cost function c between $P_{\mathbf{X},\mathbf{Y}}$ and $P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}$ is

$$W_c(P_{\mathbf{X},\mathbf{Y}}, P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}) := \inf_{\Gamma \in \mathcal{P}\left((\mathbf{X},\mathbf{Y}) \sim P_{\mathbf{X},\mathbf{Y}}, (\hat{\mathbf{X}},\hat{\mathbf{Y}}) \sim P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}\right)} \mathbf{E}_{\left((\mathbf{X},\mathbf{Y}),(\hat{\mathbf{X}},\hat{\mathbf{Y}})\right) \sim \Gamma} \left[c\Big((\mathbf{X},\mathbf{Y}),(\hat{\mathbf{X}},\hat{\mathbf{Y}})\Big) \right], \tag{4.7}$$

where $c\Big((\mathbf{X},\mathbf{Y}),(\hat{\mathbf{X}},\hat{\mathbf{Y}})\Big) : (\mathcal{X},\mathcal{Y}) \times (\mathcal{X},\mathcal{Y}) \rightarrow \mathcal{R}_+$ is any measurable *cost function*. $\mathcal{P}\Big((\mathbf{X},\mathbf{Y}) \sim P_{\mathbf{X},\mathbf{Y}}, (\hat{\mathbf{X}},\hat{\mathbf{Y}}) \sim P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}\Big)$ is the set of all joint distributions of $\Big((\mathbf{X},\mathbf{Y}),(\hat{\mathbf{X}},\hat{\mathbf{Y}})\Big)$ with marginals $P_{\mathbf{X},\mathbf{Y}}$ and $P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}$, respectively. Note that $c\Big((\mathbf{X},\mathbf{Y}),(\hat{\mathbf{X}},\hat{\mathbf{Y}})\Big) = c_X\Big(\mathbf{X},\hat{\mathbf{X}}\Big) + c_Y\Big(\mathbf{Y},\hat{\mathbf{Y}}\Big)$.

Next, we denote the set of all joint distributions of $(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{X}}, \hat{\mathbf{Y}}, \mathbf{Z})$ such that $(\mathbf{X}, \mathbf{Y}) \sim P_{\mathbf{X},\mathbf{Y}}$, $(\hat{\mathbf{X}}, \hat{\mathbf{Y}}, \mathbf{Z}) \sim P_{\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z}}$, and $\Big((\mathbf{X}, \mathbf{Y}) \perp\!\!\!\perp (\hat{\mathbf{X}}, \hat{\mathbf{Y}})|\mathbf{Z}\Big)$ as $\mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z}}$. $\mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}}}$ and $\mathcal{P}_{\mathbf{X},\mathbf{Y},\mathbf{Z}}$ are the sets of the marginals $(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{X}}, \hat{\mathbf{Y}})$ and $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ induced by $\mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z}}$.

We now introduce two Lemmas to help the proof.

Lemma 1. $P(\hat{\mathbf{X}}, \hat{\mathbf{Y}}|\mathbf{Z} = z)$ are Dirac for all $z \in \mathcal{Z}$.

Proof: First, we have $\hat{\mathbf{X}} = F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))$ and $\hat{\mathbf{Y}} = D(G_y(\mathbf{Z_y}))$ with $\mathbf{Z} = \{\mathbf{Z_a}, \mathbf{Z_y}\}$. Since the functions F, G_a, G_y, D are all deterministic, then $P(\hat{\mathbf{X}}, \hat{\mathbf{Y}}|\mathbf{Z})$ are Dirac measures. □

Lemma 2. $\mathcal{P}\Big(P_{\mathbf{X},\mathbf{Y}}, P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}\Big) = \mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}}}$ when $P(\hat{\mathbf{X}}, \hat{\mathbf{Y}}|\mathbf{Z} = z)$ are Dirac for all $z \in \mathcal{Z}$.

Proof: When $\hat{\mathbf{X}}, \hat{\mathbf{Y}}$ are deterministic functions of $\mathbf{Z}$, for any A in the sigma-algebra induced by $\hat{\mathbf{X}}, \hat{\mathbf{Y}}$, we have

$$\mathbf{E}[\mathbb{I}_{[\hat{\mathbf{X}},\hat{\mathbf{Y}} \in A]}|\mathbf{X}, \mathbf{Y}, \mathbf{Z}] = \mathbf{E}[\mathbb{I}_{[\hat{\mathbf{X}},\hat{\mathbf{Y}} \in A]}|\mathbf{Z}].$$

Therefore, this implies that $(\mathbf{X}, \mathbf{Y}) \perp\!\!\!\perp (\hat{\mathbf{X}}, \hat{\mathbf{Y}})|\mathbf{Z}$ which concludes the proof. A similar argument is made in Lemma 1 of [Tolstikhin et al., 2017].

□

Now, we use the fact that $\mathcal{P}\Big(P_{\mathbf{X},\mathbf{Y}}, P_{\hat{\mathbf{X}},\hat{\mathbf{Y}}}\Big) = \mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}}}$ (Lemma 1 + Lemma 2), $c\Big((\mathbf{X}, \mathbf{Y}), (\hat{\mathbf{X}}, \hat{\mathbf{Y}})\Big) = c_X\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) + c_Y\Big(\mathbf{Y}, \hat{\mathbf{Y}}\Big)$, $\hat{\mathbf{X}} = F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))$, and $\hat{\mathbf{Y}} = D(G_y(\mathbf{Z_y}))$, Eq. (4.7) becomes

$$\begin{aligned}
&\inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}}}} \mathbf{E}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}} \sim P}\Big[c_X\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) + c_Y\Big(\mathbf{Y}, \hat{\mathbf{Y}}\Big)\Big] \\
=& \inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z}}} \mathbf{E}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z} \sim P}\Big[c_X\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) + c_Y\Big(\mathbf{Y}, \hat{\mathbf{Y}}\Big)\Big] \\
=& \inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z}}} \mathbf{E}_{P_\mathbf{Z}} \mathbf{E}_{P(\mathbf{X},\mathbf{Y}|\mathbf{Z})} \mathbf{E}_{P(\hat{\mathbf{X}},\hat{\mathbf{Y}}|\mathbf{Z})}\Big[c_X\Big(\mathbf{X}, \hat{\mathbf{X}}\Big) + c_Y\Big(\mathbf{Y}, \hat{\mathbf{Y}}\Big)\Big] \\
=& \inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\hat{\mathbf{X}},\hat{\mathbf{Y}},\mathbf{Z}}} \mathbf{E}_{P_\mathbf{Z}} \mathbf{E}_{P(\mathbf{X},\mathbf{Y}|\mathbf{Z})}\Big[c_X\Big(\mathbf{X}, F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))\Big) + c_Y\Big(\mathbf{Y}, D(G_y(\mathbf{Z_y}))\Big)\Big] \\
=& \inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\mathbf{Z}}} \mathbf{E}_{P_\mathbf{Z}} \mathbf{E}_{P(\mathbf{X},\mathbf{Y}|\mathbf{Z})}\Big[c_X\Big(\mathbf{X}, F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))\Big) + c_Y\Big(\mathbf{Y}, D(G_y(\mathbf{Z_y}))\Big)\Big] \\
=& \inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\mathbf{Z}}} \mathbf{E}_{\mathbf{X},\mathbf{Y},\mathbf{Z} \sim P}\Big[c_X\Big(\mathbf{X}, F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))\Big) + c_Y\Big(\mathbf{Y}, D(G_y(\mathbf{Z_y}))\Big)\Big].
\end{aligned} \tag{4.8}$$

Note that in Eq. (4.8), $\mathcal{P}_{\mathbf{X},\mathbf{Y},\mathbf{Z}} = \mathcal{P}\Big((\mathbf{X}, \mathbf{Y}) \sim P_{\mathbf{X},\mathbf{Y}}, \mathbf{Z} \sim P_\mathbf{Z}\Big)$ and with a proposed $Q(\mathbf{Z}|\mathbf{X})$, we can rewrite Eq. (4.8) as

$$\begin{aligned}
&\inf_{P \in \mathcal{P}_{\mathbf{X},\mathbf{Y},\mathbf{Z}}} \mathbf{E}_{P_{\mathbf{X},\mathbf{Y}}} \mathbf{E}_{P_\mathbf{Z}}\Big[c_X\Big(\mathbf{X}, F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))\Big) + c_Y\Big(\mathbf{Y}, D(G_y(\mathbf{Z_y}))\Big)\Big] \\
=& \inf_{Q_\mathbf{Z} = P_\mathbf{Z}} \mathbf{E}_{P_{\mathbf{X},\mathbf{Y}}} \mathbf{E}_{Q(\mathbf{Z}|\mathbf{X})}\Bigg[c_X\Big(\mathbf{X}, F(G_a(\mathbf{Z_a}), G_y(\mathbf{Z_y}))\Big) + c_Y\Big(\mathbf{Y}, D(G_y(\mathbf{Z_y}))\Big)\Bigg].
\end{aligned} \tag{4.9}$$

■

4.5.1.2 From Unimodal to Multimodal

The proof is similar to Proposition 2, and we present a sketch to it. We can first show $P(\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}}|\mathbf{Z} = z)$ are Dirac for all $z \in \mathcal{Z}$. Then we use the fact that $c\Big((\mathbf{X}_{1:M}, \mathbf{Y}), (\hat{\mathbf{X}}_{1:M}, \hat{\mathbf{Y}})\Big) = \sum_{i=1}^{M} c_{Xi}\Big(\mathbf{X}_i, \hat{\mathbf{X}}_i\Big) + c_Y\Big(\mathbf{Y}, \hat{\mathbf{Y}}\Big)$.

Finally, we follow the tower rule of expectation and the conditional independence property similar to the proof in Proposition 2 and this concludes the proof.

■

4.5.2 Full Baseline Models & Results

For a detailed description of the baselines, we point the reader to MFN [Zadeh et al., 2018a], MARN [Zadeh et al., 2018b], TFN [Zadeh et al., 2017], BC-LSTM [Poria et al., 2017a], MV-LSTM [Rajagopalan et al., 2016], EF-LSTM [Graves et al., 2013, Hochreiter and Schmidhuber, 1997, Schuster and Paliwal, 1997], DF [Nojavanasghari et al., 2016], MV-HCRF [Song et al., 2012, 2013], EF-HCRF [Morency et al., 2007, Quattoni et al., 2007], THMM [Morency et al., 2011], SVM-MD [Zadeh et al., 2016] and RF [Breiman, 2001].

We use the following extra notations for full descriptions of the baseline models described in Chapter 4.2.2, paragraph 3:

Variants of EF-LSTM: **EF-LSTM** (Early Fusion LSTM) uses a single LSTM [Hochreiter and Schmidhuber, 1997] on concatenated multimodal inputs. We also implement the **EF-SLSTM** (stacked) [Graves et al., 2013], **EF-BLSTM** (bidirectional) [Schuster and Paliwal, 1997] and **EF-SBLSTM** (stacked bidirectional) versions.

Variants of EF-HCRF: **EF-HCRF**: (Hidden Conditional Random Field) [Quattoni et al., 2007] uses a HCRF to learn a set of latent variables conditioned on the concatenated input at each time step. **EF-LDHCRF** (Latent Discriminative HCRFs) [Morency et al., 2007] are a class of models that learn hidden states in a CRF using a latent code between observed concatenated input and hidden output. **EF-HSSHCRF**: (Hierarchical Sequence Summarization HCRF) [Song et al., 2013] is a layered model that uses HCRFs with latent variables to learn hidden spatio-temporal dynamics.

Variants of MV-HCRF: **MV-HCRF**: Multi-view HCRF [Song et al., 2012] is an extension of the HCRF for Multi-view data, explicitly capturing view-shared and view specific sub-structures. **MV-LDHCRF**: [Morency et al., 2007] is a variation of the MV-HCRF model that uses LDHCRF instead of HCRF. **MV-HSSHCRF**: [Song et al., 2013] further extends **EF-HSSHCRF** by performing Multi-view hierarchical sequence summary representation.

In the following, we provide the full results for all baselines models described in Chapter 4.2.2, paragraph 3. Table 4.3 contains results for multimodal speaker traits recognition on the POM dataset. Table 4.4 contains results for the multimodal sentiment analysis on the CMU-MOSI, ICT-MMMO, YouTube, and MOUD datasets. Table 4.5 contains results for multimodal emotion recognition on the IEMOCAP dataset. MFM consistently achieves state-of-the-art or competitive results for all six multimodal datasets. We believe that by our MFM design, the multimodal discriminative factor $\mathbf{F_y}$ has successfully learned more meaningful representations by distilling discriminative features. This highlights the benefit of learning factorized multimodal representations towards discriminative tasks.

4.5.3 Multimodal Features

For each of the multimodal time series datasets as mentioned in Chapter 4.2.2, paragraph 3, we extracted the following multimodal features: **Language:** We use pre-trained word embeddings (glove.840B.300d) [Pennington et al., 2014] to convert the video transcripts into a sequence of 300 dimensional word vectors. **Visual:** We use Facet [iMotions, 2017] to extract a set of features including per-frame basic and advanced emotions and facial action units as indicators of facial muscle movement [Ekman, 1992, Ekman et al., 1980]. **Acoustic:** We use COVAREP [Degottex et al., 2014] to extract low level acoustic features including 12 Mel-frequency cepstral coefficients (MFCCs), pitch tracking and voiced/unvoiced segmenting features, glottal source parameters, peak slope parameters and maxima dispersion quotients. To reach the same time alignment between different modalities we choose the granularity of the input to be at the level of words. The words are aligned with audio using P2FA [Yuan and Liberman, 2008] to get their exact utterance times. We use expected feature values across the entire word for visual and acoustic features since they are extracted at a higher frequencies.

We make a note that the features for some of these datasets are constantly being updated. The authors of prior work [Zadeh et al., 2018a] notified us of a discrepancy in the sampling rate for acoustic feature extraction in the ICT-MMMO, YouTube and MOUD datasets which led to inaccurate word-level alignment between the three modalities. They publicly released the updated multimodal features. We performed all experiments on the latest versions of these datasets which can be accessed from `https://github.com/A2Zadeh/CMU-MultimodalSDK`. All baseline models were retrained with extensive hyperparameter search for fair comparison.

Table 4.3: Results for personality trait recognition on the POM dataset. The best results are highlighted in bold and Δ_{SOTA} shows the change in performance over previous state of the art. Improvements are highlighted in green. MFM achieves state-of-the-art or competitive performance on all datasets and metrics.

Dataset	**POM Speaker Personality Traits**															
Task	Con	Pas	Voi	Dom	Cre	Viv	Exp	Ent	Res	Tru	Rel	Out	Tho	Ner	Per	Hum
Metric								r								
Majority	-0.041	-0.029	-0.104	-0.031	-0.122	-0.044	-0.065	-0.105	0.006	-0.077	-0.024	-0.085	-0.130	0.097	-0.127	-0.069
SVM	0.063	0.086	-0.004	0.141	0.113	0.076	0.134	0.141	0.166	0.168	0.104	0.066	0.134	0.068	0.064	0.147
DF	0.240	0.273	0.017	0.139	0.112	0.173	0.118	0.217	0.148	0.143	0.019	0.093	0.041	0.136	0.168	0.259
EF-LSTM	0.200	0.302	0.031	0.079	0.170	0.244	0.265	0.240	0.142	0.062	0.083	0.152	0.260	0.105	0.217	0.227
EF-SLSTM	0.221	0.327	0.042	0.151	0.177	0.239	0.268	0.248	0.204	0.069	0.092	0.215	0.252	0.159	0.218	0.196
EF-BLSTM	0.162	0.289	-0.034	0.135	0.191	0.279	0.274	0.231	0.184	0.154	0.093	0.147	0.245	0.166	0.243	0.272
EF-SBLSTM	0.174	0.310	0.021	0.088	0.170	0.224	0.261	0.241	0.155	0.163	0.097	0.120	0.215	0.121	0.216	0.171
MV-LSTM	0.358	0.416	0.131	0.146	0.280	0.347	0.323	0.326	0.295	0.237	0.119	0.238	0.284	0.258	0.239	0.317
BC-LSTM	0.359	0.425	0.081	0.234	0.358	0.417	0.450	0.361	0.293	0.109	0.075	0.078	0.363	0.184	0.344	0.319
TFN	0.089	0.201	0.030	0.020	0.124	0.204	0.171	0.223	-0.051	-0.064	0.114	0.060	0.048	-0.002	0.106	0.213
MARN	0.340	0.410	0.166	0.235	0.340	0.374	0.406	0.378	0.282	0.147	0.215	0.204	0.348	0.235	0.303	0.287
MFN	0.395	0.428	0.193	0.313	0.367	0.431	0.452	0.395	0.333	0.296	0.255	0.259	0.381	0.318	0.377	0.386
MFM	**0.431**	**0.450**	**0.197**	**0.411**	**0.380**	**0.448**	**0.467**	**0.452**	**0.368**	0.212	**0.309**	**0.333**	**0.404**	**0.333**	0.334	**0.408**
Δ_{SOTA}	↑ 0.036	↑ 0.022	↑ 0.004	↑ 0.097	↑ 0.013	↑ 0.017	↑ 0.015	↑ 0.057	↑ 0.035	–	↑ 0.054	↑ 0.074	↑ 0.023	↑ 0.015	–	↑ 0.022

4.5.4 Information and Gradient-Based Interpretation

Information-Based Interpretation: We choose the normalized Hilbert-Schmidt Independence Criterion [Gretton et al., 2005a, Wu et al., 2018a] as the approximation (see prior work [Sugiyama and Yamada, 2012, Wu et al., 2018a]) of our MI measure:

$$\mathrm{MI}(\mathbf{F}_{\cdot}, \hat{\mathbf{X}}_i) = \mathrm{HSIC}_{norm}(\mathbf{F}_{\cdot}, \hat{\mathbf{X}}_i) = \frac{\mathrm{tr}(\mathbf{K}_{\mathbf{F}_{\cdot}} \mathbf{H} \mathbf{K}_{\hat{\mathbf{X}}_i} \mathbf{H})}{\|\mathbf{H}\mathbf{K}_{\mathbf{F}_{\cdot}} \mathbf{H}\|_F \|\mathbf{H}\mathbf{K}_{\hat{\mathbf{X}}_i} \mathbf{H}\|_F}, \tag{4.10}$$

where $\cdot$ represents y or a_i, n is the number of $\{\mathbf{F}_{\cdot}, \hat{\mathbf{X}}_i\}$ pairs, $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, $\mathbf{K}_{\mathbf{F}_{\cdot}} \in \mathbb{R}^{n \times n}$ is the Gram matrix of $\mathbf{F}_{\cdot}$ with $\mathbf{K}_{\mathbf{F}_{\cdot}ij} = k_1(\mathbf{F}_{\cdot i}, \mathbf{F}_{\cdot j})$, $\mathbf{K}_{\hat{\mathbf{X}}_i} \in \mathbb{R}^{n \times n}$ is the Gram matrix of $\hat{\mathbf{X}}_i$ with $\mathbf{K}_{\hat{\mathbf{X}}_{i}jk} = k_2(\hat{\mathbf{X}}_{ij}, \hat{\mathbf{X}}_{ik})$. $k_1(\cdot, \cdot)$ and $k_2(\cdot, \cdot)$ are predefined kernel functions.

The most common choice for the kernel is the RBF kernel. However, if we consider time series data with various time steps, we need to either perform data augmentation or choose another kernel choice. For example, we can adopt the Global Alignment Kernel [Cuturi et al., 2007] which considers the alignment

Table 4.4: Sentiment prediction results on CMU-MOSI, ICT-MMMO, YouTube and MOUD. The best results are highlighted in bold and Δ_{SOTA} shows the change in performance over previous state of the art (SOTA). Improvements are highlighted in green. MFM achieves state-of-the-art or competitive performance on all datasets and metrics.

Dataset	**CMU-MOSI**					**ICT-MMMO**		**YouTube**		**MOUD**	
Task	Sentiment					Sentiment		Sentiment		Sentiment	
Metric	Acc_7	Acc_2	F1	MAE	*r*	Acc_2	F1	Acc_3	F1	Acc_2	F1
Majority	17.5	50.2	50.1	1.864	0.057	40.0	22.9	42.4	25.2	60.4	45.5
RF	21.3	56.4	56.3	-	-	70.0	69.8	33.3	32.3	64.2	63.3
SVM-MD	26.5	71.6	72.3	1.100	0.559	68.8	68.7	42.4	37.9	59.4	45.5
THMM	17.8	53.8	53.0	-		50.7	45.4	42.4	27.9	61.3	57.0
SAL-CNN	-	73.0	-	-	-	-	-	-	-	-	-
C-MKL	30.2	72.3	72.0	-	-	-	-	-	-	-	-
EF-HCRF	24.6	65.3	65.4	-	-	50.0	50.3	44.1	43.8	54.7	54.7
EF-LDHCRF	24.6	64.0	64.0	-	-	73.8	73.1	45.8	45.0	52.8	49.3
MV-HCRF	22.6	44.8	27.7	-	-	36.3	19.3	27.1	19.7	60.4	45.5
MV-LDHCRF	24.6	64.0	64.0	-	-	68.8	67.1	44.1	44.0	53.8	46.9
CMV-HCRF	22.3	44.8	27.7	-	-	36.3	19.3	30.5	14.3	60.4	45.5
CMV-LDHCRF	24.6	63.6	63.6	-	-	51.3	51.4	42.4	42.0	53.8	47.8
EF-HSSHCRF	24.6	63.3	63.4	-	-	50.0	51.3	37.3	35.6	52.8	49.3
MV-HSSHCRF	24.6	65.6	65.7	-	-	62.5	63.1	44.1	44.0	47.2	46.4
DF	26.8	72.3	72.1	1.143	0.518	65.0	58.7	45.8	32.0	67.0	67.1
EF-LSTM	32.4	74.3	74.3	1.023	0.622	66.3	65.0	44.1	43.6	67.0	64.3
EF-SLSTM	29.3	72.7	72.8	1.081	0.600	72.5	70.9	40.7	41.2	56.6	51.4
EF-BLSTM	28.9	72.0	72.0	1.080	0.577	63.8	49.6	42.4	38.1	58.5	58.9
EF-SBLSTM	26.8	73.3	73.2	1.037	0.619	62.5	49.0	37.3	33.2	63.2	63.3
MV-LSTM	33.2	73.9	74.0	1.019	0.601	72.5	72.3	45.8	43.3	57.6	48.2
BC-LSTM	28.7	73.9	73.9	1.079	0.581	70.0	70.1	45.0	45.1	72.6	72.9
TFN	28.7	74.6	74.5	1.040	0.587	72.5	72.6	45.0	41.0	63.2	61.7
MARN	34.7	77.1	77.0	0.968	0.625	71.3	70.2	48.3	44.9	81.1	81.2
MFN	34.1	77.4	77.3	0.965	0.632	73.8	73.1	51.7	51.6	81.1	80.4
MFM	**36.2**	**78.1**	**78.1**	**0.951**	**0.662**	**81.3**	**79.2**	**53.3**	**52.4**	**82.1**	**81.7**
Δ_{SOTA}	↑ 1.5	↑ 0.7	↑ 0.8	↓ 0.014	↑ 0.030	↑ 7.5	↑ 6.1	↑ 1.6	↑ 0.8	↑ 1.0	↑ 0.5

Table 4.5: Emotion recognition results on IEMOCAP test set. The best results are highlighted in bold and Δ_{SOTA} shows the change in performance over previous SOTA. Improvements are highlighted in green. MFM achieves state-of-the-art or competitive performance on all datasets and metrics.

Dataset	**IEMOCAP Emotions**											
Task	Happy		Sad		Angry		Frustrated		Excited		Neutral	
Metric	Acc_2	F1	Acc_2	F1	Acc_2	F1	Acc_2	F1	Acc_2	F1	Acc_2	F1
Majority	85.6	79.0	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
SVM	86.1	81.5	81.1	78.8	82.5	82.4	77.3	71.1	86.4	86.0	65.2	64.9
RF	85.5	80.7	80.1	76.5	81.9	82.0	78.6	75.3	88.9	85.1	63.2	57.3
THMM	85.6	79.2	79.5	79.8	79.3	73.0	71.6	69.6	86.0	84.6	58.6	46.4
EF-HCRF	85.7	79.2	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
EF-LDHCRF	85.8	79.5	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
MV-HCRF	15.0	4.9	79.4	70.3	24.2	9.4	79.5	70.4	89.6	84.7	59.1	44.0
MV-LDHCRF	85.7	79.2	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
CMV-HCRF	14.4	3.6	79.4	70.3	24.2	9.4	79.5	70.4	89.6	84.7	59.1	44.0
CMV-LDHCRF	85.8	79.5	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
EF-HSSHCRF	85.8	79.5	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
MV-HSSHCRF	85.8	79.5	79.4	70.3	75.8	65.4	79.5	70.4	89.6	84.7	59.1	44.0
DF	86.0	81.0	81.8	81.2	75.8	65.4	78.4	76.8	89.6	84.7	59.1	44.0
EF-LSTM	85.2	83.3	82.1	81.1	84.5	84.3	79.5	70.4	89.6	84.7	68.2	67.1
EF-SLSTM	85.6	79.0	80.7	80.2	82.8	82.2	77.5	69.7	89.3	86.2	68.8	**68.5**
EF-BLSTM	85.0	83.7	81.8	81.6	84.2	83.3	79.5	70.4	89.6	84.7	67.1	66.6
EF-SBLSTM	86.0	84.2	80.2	80.5	85.2	84.5	79.5	70.4	89.6	84.7	67.8	67.1
MV-LSTM	85.9	81.3	80.4	74.0	85.1	84.3	79.5	73.8	88.9	85.8	67.0	66.7
BC-LSTM	84.9	81.7	83.2	81.7	83.5	84.2	80.0	76.1	86.9	85.4	67.5	64.1
TFN	84.8	83.6	83.4	82.8	83.4	84.2	74.1	74.3	75.6	78.0	67.5	65.4
MARN	86.7	83.6	82.0	81.2	84.6	84.2	79.5	76.6	89.6	**87.1**	66.8	65.9
MFN	90.1	85.3	85.8	79.2	87.0	86.0	80.3	**76.9**	89.8	86.3	71.8	61.7
MFM	**90.2**	**85.8**	**88.4**	**86.1**	**87.5**	**86.7**	**80.4**	74.5	**90.0**	**87.1**	**72.1**	68.1
Δ_{SOTA}	↑ 0.1	↑ 0.5	↑ 2.6	↑ 3.3	↑ 0.5	↑ 0.7	↑ 0.1	–	↑ 0.2	–	↑ 0.3	–

between two varying-length time series when computing the kernel score between them. To simplify our analysis, we choose to augment data before we calculate the kernel score with the RBF kernel. More specifically, we perform averaging over time series data:

$$\mathbf{X}_{aug} = \frac{1}{n}\sum_{t=1}^{T} X^t \text{ with } \mathbf{X} = [X^1, X^2, \cdots, X^T]. \tag{4.11}$$

The bandwidth of the RBF kernel is set as 1.0 throughout the experiments.

Table 4.6: Information-Based interpretation results showing ratios $r_i = \frac{\text{MI}(\mathbf{F_y}, \hat{\mathbf{X}}_i)}{\text{MI}(\mathbf{F_{a}}_i, \hat{\mathbf{X}}_i)}$, $i \in \{(\ell)anguage, (v)isual, (a)coustic\}$ for the POM dataset for personality traits prediction.

Ratio	r_ℓ (language)	r_v (visual)	r_a (acoustic)
POM	1.090	0.996	0.898

Here, we provide an additional interpretation result for the POM dataset in Table 4.6. We observe that the language modality is also the most informative while the visual and acoustic modalities are almost equally informative. This result is in agreement with behavioral studies which have observed that non-verbal behaviors are particularly informative of personality traits [Guimond and Massrieh, 2012, Levine et al., 2009, Mohammadi et al., 2010]. For example, the same sentence "this movie was great" can convey significantly different messages on speaker confidence depending on whether it was said in a loud and exciting voice, with eye contact, or powerful gesticulation.

Gradient-Based Interpretation: MFM reconstructs x_i as follows:

$$\hat{x}_i = F_i(f_{ai}, f_y), f_{ai} = G_{ai}(z_{ai}), f_y = G_y(z_y), z_{ai} \sim Q(\mathbf{Z_{ai}}|\mathbf{X}_i = x_i), z_y \sim Q(\mathbf{Z_y}|\mathbf{X}_{1:M} = x_{1:M}). \tag{4.12}$$

Equation (4.12) also explains how we obtain $f_y \sim P(\mathbf{F_y}|\mathbf{X}_{1:M} = x_{1:M})$. The gradient flow through time is defined as:

$$\nabla_{f_y}(\hat{x}_i) := [\|\nabla_{f_y}\hat{x}_i^1\|_F^2, \|\nabla_{f_y}\hat{x}_i^2\|_F^2, \cdots, \|\nabla_{f_y}\hat{x}_i^T\|_F^2]. \tag{4.13}$$

4.5.5 Encoder and Decoder Design for Multimodal Synthetic Image Dataset

For experiments on the multimodal synthetic image dataset, we use convolutional+fully-connected layers for the encoder and deconvolutional+fully-connected layers for the decoder [Zeiler et al., 2010]. Different convolutional layers are each applied on the input SVHN and MNIST images to learn modality-specific generative factors. Next, we concatenate the features from two more convolutional layers on SVHN and MNIST to learn the multimodal-discriminative factor. The multimodal discriminative factor is passed through fully-connected layers to predict the label. For generation, we concatenate the multimodal discriminative factors and the modality-specific generative factor together and use a deconvolutional layer to generate digits.

4.5.6 Encoder and Decoder Design for Multimodal Time Series Datasets

Figure 4.5 illustrates how MFM operates on multimodal time series data. The encoder $Q(\mathbf{Z_y}|\mathbf{X}_{1:M})$ can be parametrized by any model that performs multimodal fusion [Nojavanasghari et al., 2016, Zadeh et al., 2018a]. We choose the Memory Fusion Network (MFN) [Zadeh et al., 2018a] as our encoder $Q(\mathbf{Z_y}|\mathbf{X}_{1:M})$. We use encoder LSTM networks and decoder LSTM networks [Cho et al., 2014] to parametrize functions $Q(\mathbf{Z_{a}}_{1:M}|\mathbf{X}_{1:M})$ and $F_{1:M}$ respectively, and FCNNs to parametrize functions G_y, $G_{a\{1:M\}}$ and D.

4.5.7 Surrogate Inference Graphical Model

We illustrate the surrogate inference for addressing the missing modalities issue in Figure 4.6. The surrogate inference model infers the latent codes given the present modalities. These inferred latent codes can then be used for reconstructing the missing modalities or label prediction in the presence of missing modalities.

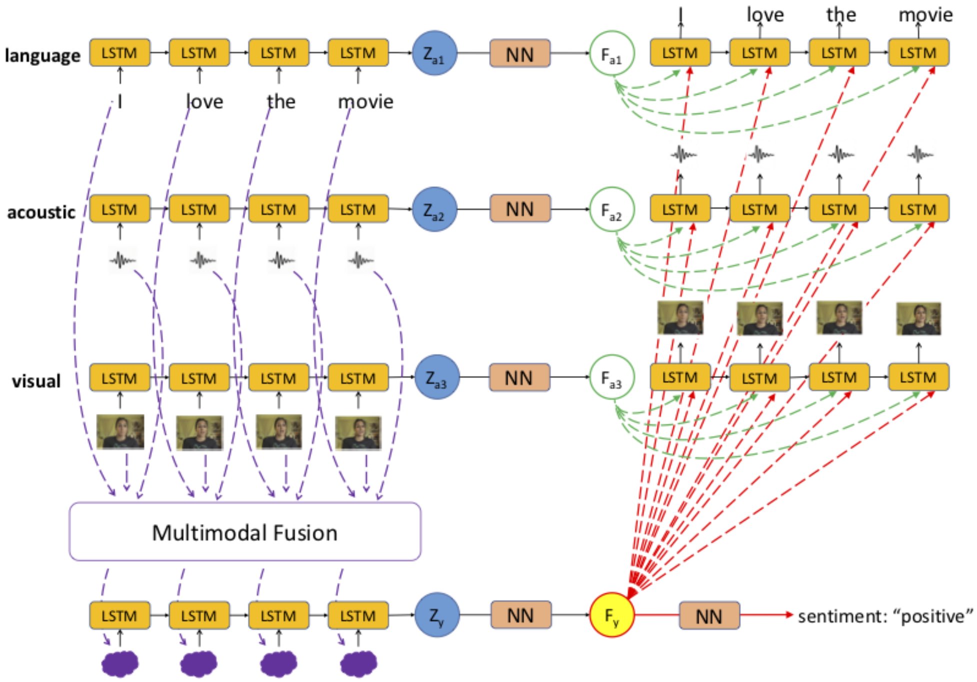


Figure 4.5: Recurrent neural architecture for MFM. The encoder $Q(\mathbf{Z_y}|\mathbf{X}_{1:M})$ can be parametrized by any model that performs multimodal fusion [Nojavanasghari et al., 2016, Zadeh et al., 2018a]. We use encoder LSTM networks and decoder LSTM networks [Cho et al., 2014] to parametrize functions $Q(\mathbf{Z}_{\mathbf{a}1:M}|\mathbf{X}_{1:M})$ and $F_{1:M}$ respectively, and FCNNs to parametrize functions G_y, $G_{a\{1:M\}}$ and D.

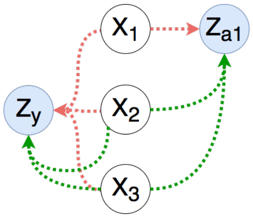


Figure 4.6: The surrogate inference graphical model to deal with missing modalities in MFM. Red lines denote original inference in MFM and green lines denote surrogate inference to infer latent codes given present modalities.

Chapter 5

Relationship Quantification - Mutual Information Estimation

In this chapter, we will discuss the sub-challenge of mutual information estimation within the challenge of relationship quantification. Mutual Information (MI) measures the average statistical dependency between two random variables, and it has found abundant applications in practice, such as feature selection [Chen et al., 2018a, Peng et al., 2005], interpretable factor discovery [Chen et al., 2016, Tsai et al., 2018], genetic association studies [Zhang et al., 2012], to name a few. Recent work [Belghazi et al., 2018, Poole et al., 2019] proposed to use neural networks with gradient descent to estimate MI, which empirically scales better in high-dimension settings as compared to classic approaches (e.g., Kraskov (KSG) [Kraskov et al., 2004] estimator), which are known to suffer from the curse of dimensionality. Inspired by this line of work, we take a step further to present neural methods for point-wise dependency (PD) estimation. At a colloquial level, PD serves to understand the instance-level dependency between a pair of events taken by two random variables, which gives us a fine-grained understanding of the outcome. Formally, it can be realized as the ratio between likelihood of their co-occurrence to the likelihood of the product events: $p(x,y)/p(x)p(y)$ with x and y being the corresponding outcomes.

At first glance, it may seem straightforward to estimate PD by adopting prior density ratio estimation approaches [Sugiyama et al., 2012a,b] to directly calculate the ratio between $p(x,y)$ and $p(x)p(y)$. Nonetheless, for the sake of tractability, previous methods are mainly kernel-based approaches that might be inadequate to scale to high-dimensional and complex-structured data. In this work, we introduce approaches for PD estimation that leverage the recent advances in rich and flexible neural networks. We show that we can naturally obtain PD when we are optimizing MI neural variational bounds [Belghazi et al., 2018, Poole et al., 2019].

However, estimating these MI bounds often results in inevitably large variance [Song and Ermon, 2019]. To address this concern, we develop two data-driven approaches: *Probabilistic Classifier* and *Density-Ratio Fitting*. *Probabilistic Classifier* turns PD estimation into a supervised binary classification task, where we train a classifier to distinguish the observed joint distribution from the product of marginal distribution. This approach adopts cross-entropy loss using neural networks, which is favorable for optimization and exhibits a stable training trajectory with less variance. *Density-Ratio Fitting* seeks to minimize the least-square difference between the true and the estimated PD. Its objective involves no logarithm and exponentiation; hence, it is practically preferable due to its numerical stability.

We empirically analyze the advantages of PD neural estimation on three applications. First, we cast the challenging MI estimation problem to be a PD estimation problem. The re-formulation bypasses calculating MI lower bounds in prior work [Belghazi et al., 2018, Poole et al., 2019], which suffers from large variance [Song and Ermon, 2019] in practice. Our empirical results demonstrate the low variance and bias of the proposed approach when comparing to prior MI neural estimators. Second, our PD estimation objectives also inspire new losses for contrastive self-supervised representation learning. Surprisingly, *Density-Ratio Fitting* inspired loss results in a consistent improvement over prior work in both shallow [Tschannen et al., 2019] and deep [Bachman et al., 2019] neural architectures. Third, we study the use of PD estimation for data containing information across modalities. More specifically, we analyze the cross-modal retrieval task on human speech and text corpora. We make our experiments publicly available at `https://github.com/yaohungt/Pointwise_Dependency_Neural_Estimation`.

5.1 Related Work

Point-wise Dependency Estimation Prior literature studies point-wise dependency (PD) with three groups of estimation methods: *counting-based* [Bouma, 2009, Church and Hanks, 1990, Levy and Goldberg, 2014], *kernel-based* [Yokoi et al., 2018], and *likelihood-based* [Li et al., 2015]. *Counting-based* methods approximate the joint density by counting the occurrence of the pair (i.e., (x, y)) and the marginal density by counting the presence of the individual outcome (i.e., x or y). Counting based approaches can only work on discrete data and may be unrealistic when the data is sparse. *Kernel-based* method, particularly pointwise HSIC [Yokoi et al., 2018], can be seen as a smoothed variant of the counting-based methods, which adopts the kernel to measure the similarity between sparse data. Although this method manifests nice robustness to sparse data, its computational cost is high with high-dimensional data. *Likelihood-based* approaches instead approximate conditional likelihood (i.e., $p(y|x)$) and marginal likelihood (i.e., $p(y)$) using function approximators such as neural networks. Although this approach can be adapted to continuous data, it involves marginal likelihood estimation, which is challenging [Goodfellow et al., 2014, Kingma and Welling, 2013] and may perform poorly in practice. On the other hand, our presented approaches involve no marginal likelihood estimation, can work on both discrete and continuous data, and leverage neural networks with gradient descent in high-dimensional settings.

Density Ratio Estimation To calculate the ratio between densities $(p(x)/q(x))$, prior density ratio estimation approaches [Sugiyama et al., 2012a,b] propose to estimate the ratio directly and avoid estimating the density ($p(x)$ and $q(x)$). For example, Sugiyama et al. [2012b] fit the true density ratio model under the Bregman divergence [Bregman, 1967] and further develop a robust density estimation method under the power divergence [Basu et al., 1998]. While it is straightforward to apply these approaches to PD estimation, these approaches are studied in the context of kernel-based methods, which can make it difficult to apply in practice when data is high-dimensional and complex-structured. Our approaches contrarily take advantage of high-capacity neural networks.

Neural Methods for Mutual Information Estimation Recent approaches [Belghazi et al., 2018, Poole et al., 2019] present neural methods that estimate mutual information (MI) via its variational bounds. They consider MI 1) lower bounds such as Donsker-Varadhan bound [Donsker and Varadhan, 1983] and Nguyen-Wainwright-Jordan bound [Nguyen et al., 2010]; and 2) upper bound such as Barber-Agakov bound [Barber and Agakov, 2003]. These bounds exhibit inevitable large variance [Song and Ermon, 2019] and have severe training instability in practice [Hjelm et al., 2018, Tschannen et al., 2019]. In our discussion, we show that we can obtain PD when optimizing these bounds. Additionally, we present alternative PD estimation methods that do not involve calculating MI variational bounds and are favorable in practice.

5.2 Point-wise Dependency Neural Estimation

This chapter aims to identify the association for a pair of outcomes $(x, y) \in \mathcal{X} \times \mathcal{Y}$ by studying their point-wise dependency. We use an uppercase letter to denote a random variable (i.e., X), a lowercase letter to indicate an outcome x drawn from a particular distribution (i.e., $x \sim P_X$), and a calligraphy letter $\mathcal{X}$ to represent a sample space (i.e., $x \in \mathcal{X}$). The joint distribution of X, Y is represented by $P_{X,Y}$, and the product of their marginals is represented by $P_X P_Y$. Throughout the chapter, we use the conventional notation $I(X;Y)$ to denote the mutual information between random variables X and Y.

Formally, we define the following point-wise dependency (PD) to quantitatively measure the discrepancy between *the probability of their co-occurrence* and *the probability of independent occurrences*.

Definition 1 (Point-wise Dependency)**.** Given a pair of outcomes $(x, y) \sim P_{X,Y}$, their point-wise dependency is defined as $r(x, y) := p(x, y)/p(x)p(y)$.

PD is non-negative. Intuitively, when $r(x, y) > 1$, it means (x, y) co-occur more often than their independent occurances. Similarly, when $r(x, y) \leq 1$, it means they co-occur less frequently. Our goal is to estimate $r(x, y)$ by approximating it using neural network $\hat{r}_\theta(x, y)$ with parameter $\theta \in \Theta$.

5.2.1 Mutual Information and Point-wise Dependency

A related quantitative measurement of point-wise dependency is Point-wise mutual information (PMI) [Bouma, 2009], which is the logarithm of PD (PMI $:= f(x, y) = \log r(x, y)$). In what follows, we discuss parametrized estimation of PMI using neural networks $\hat{f}_\theta(x, y)$ with parameter θ. By definition, mutual information $I(X;Y)$ is the expected value of PMI: $I(X;Y) = \mathbb{E}_P[\log r(X, Y)] = \mathbb{E}_P[f(X, Y)]$. Hence by using $\hat{f}_\theta$ as a plug-in, we can obtain an approximation of the mutual information with $\mathbb{E}_P[\hat{f}_\theta(X, Y)]$. Reversely, we will show that PMI can be obtained when optimizing MI (neural) variational bounds and present two methods to do so, one as unconstrained optimization and the other as constrained optimization problem.

(Unconstrained Optimization) Variational Bounds of Mutual Information Recent work [Belghazi et al., 2018, Poole et al., 2019] proposes to estimate MI using neural networks by exploiting either the variational MI lower bounds [Belghazi et al., 2018] or the variational MI form Poole et al. [2019]. In particular, Belghazi et al. [2018] proposed the I_{DV} estimator, standing for Donsker-Varadhan (DV) lower bound [Donsker and Varadhan, 1983] of MI. On the other hand, Poole et al. [2019] proposed the I_{JS} estimator, corresponding to using f-GAN objective [Nowozin et al., 2016] as a lower bound of Jensen-Shannon (JS) divergence between $P_{X,Y}$ and $P_X P_Y$. I_{JS} is found to be more stable then I_{DV} and other variational lower bounds, and thus it is widely used in prior work [Hjelm et al., 2018, Poole et al., 2019, Song and Ermon, 2019], defined as follows:

$$I_{\mathrm{JS}} := \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}\Big[-\mathrm{softplus}\Big(-\hat{f}_\theta(x, y)\Big)\Big] - \mathbb{E}_{P_X P_Y}\Big[\mathrm{softplus}\Big(\hat{f}_\theta(x, y)\Big)\Big], \tag{5.1}$$

where we use softplus to denote softplus $(x) = \log\,(1 + \exp\,(x))$. It could be readily verified that the optimal $\hat{f}^*_\theta(x, y) = \log\left(p(x, y)/p(x)p(y)\right)$ [Poole et al., 2019]. We refer this objective as *Variational Bounds of Mutual Information* approach for PMI estimation.

(Constrained Optimization) Density Matching This method considers to match the true joint density $p(x, y)$ and the estimated joint density $\hat{p}_\theta(x, y) := e^{\hat{f}_\theta(x,y)} p(x)p(y)$ by minimizing the following KL divergence:

$$\inf_{\theta\in\Theta} D_{\mathrm{KL}}(P_{X,Y} \parallel \hat{P}_{\theta X,Y}) := \inf_{\theta\in\Theta} I(X;Y) - \mathbb{E}_{P_{X,Y}}\left[\hat{f}_\theta(x,y)\right] \Leftrightarrow \sup_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}\left[\hat{f}_\theta(x,y)\right].$$

Since KL divergence has a minimum value of 0, it is easy to see that $\forall\theta \in \Theta$, $\mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)]$ is a lower bound of MI. Note that this objective is a constrained optimization problem, since we need to ensure the estimated joint density is a valid density function: $\hat{p}_\theta(x,y) \geq 0$ and $\iint \hat{p}_\theta(x,y)\,\mathrm{d}x\mathrm{d}y = 1$. Equivalently, the constraints could be formed as $e^{\hat{f}_\theta(x,y)} \geq 0$ (trivially true) and $\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}] = 1$. Putting everything together, we can reformulate the following constrained optimization problem:

$$\max_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)], \quad \text{subject to } \mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}] = 1,$$

which is also called KL importance estimation procedure [Sugiyama et al., 2008] with a unique solution $\hat{f}^*_\theta(x,y) = \log\left(p(x,y)/p(x)p(y)\right)$. The Lagrangian of the above constrained problem is

$$\max_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \lambda\cdot\left(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}] - 1\right), \tag{5.2}$$

where $\lambda \in \mathbb{R}$ is the dual variable. Furthermore, penalty method could also be used to transform the original constrained optimization problem to an unconstrained one:

$$\max_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \eta\cdot\left(\log \mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]\right)^2, \tag{5.3}$$

where $\eta > 0$ is the penalty coefficient. We refer Eq. (5.2) as *Density Matching I* and Eq. (5.3) as *Density Matching II* for PMI estimation.

5.2.2 Proposed Methods for Point-wise Dependency (PD) Estimation

So far, we introduce how to obtain PMI by optimizing various MI variational bounds. Now, instead of estimating PMI, we present two methods to estimate PD ($p(x,y)/p(x)p(y)$), i.e., the *Probabilistic Classifier* method and the *Density-Ratio Fitting* method. We argue that the presented PD estimation methods admit better training stability than the PMI estimation methods discussed previously. On the one hand, the Probabilistic Classifier method casts PD estimation as a binary classification task, where the binary cross-entropy loss can be used and optimized in existing optimization packages [Abadi et al., 2016, Paszke et al., 2019]. On the other hand, the Density-Ratio Fitting method contains no logarithm or exponentiation, which are often the roots of the instability in MI (or PMI) estimation [Poole et al., 2019, Song and Ermon, 2019]. In what follows, we present both methods in a sequel.

Probabilistic Classifier Method This approach casts the PD estimation as the problem of estimating the 'class'-posterior probability. First, we use a Bernoulli random variable C to classify the samples drawn from the joint density ($C = 1$ for $(x,y) \sim P_{X,Y}$) and the samples drawn from product of the marginal densities ($C = 0$ for $(x,y) \sim P_X P_Y$). Equivalently, the likelihood function $p(x,y \mid C = 1) := p(x,y)$ and $p(x,y \mid C = 0) := p(x)p(y)$. By Bayes' Theorem, we re-express PD by the ratio of two class-posterior probability:

$$r(x,y) = \frac{p(x,y)}{p(x)p(y)} = \frac{p(x,y \mid C=1)}{p(x,y \mid C=0)} = \frac{p(C=0)}{p(C=1)}\frac{p(C=1 \mid x,y)}{p(C=0 \mid x,y)}.$$

In the above equation, the ratio $\frac{p(C=0)}{p(C=1)}$ can be approximated by the ratio of the sample size:

$$\frac{\hat{p}(C=0)}{\hat{p}(C=1)} = \frac{(n_{P_X P_Y})/(n_{P_X P_Y} + n_{P_{X,Y}})}{(n_{P_{X,Y}})/(n_{P_X P_Y} + n_{P_{X,Y}})} = \frac{n_{P_X P_Y}}{n_{P_{X,Y}}},$$

and we use a probability classifier $\hat{p}_\theta(C \mid x, y)$ parameterized by a neural network θ to approximate the class-posterior classifier $p(C \mid x, y)$. By adopting the binary cross-entropy loss, the objective has the following form:

$$\max_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\log \hat{p}_\theta(C=1 \mid x, y)] + \mathbb{E}_{P_X P_Y}[\log (1 - \hat{p}_\theta(C=1 \mid x, y))]. \tag{5.4}$$

Then, bringing all the equations together, we obtain the *Probabilistic Classifier* PD estimator:

$$\hat{r}_\theta(x, y) = \frac{n_{P_X P_Y}}{n_{P_{X,Y}}} \frac{\hat{p}_\theta(C=1 \mid x, y)}{\hat{p}_\theta(C=0 \mid x, y)}, \quad \text{with } (x, y) \sim P_{X,Y} \text{ or } (x, y) \sim P_X P_Y. \tag{5.5}$$

Density-Ratio Fitting Method This approach considers to minimize the expected least-square difference between the true PD $r(x, y)$ and the estimated PD $\hat{r}_\theta(x, y)$:

$$\inf_{\theta \in \Theta} \mathbb{E}_{P_X P_Y}[\left(r(x, y) - \hat{r}_\theta(x, y)\right)^2] \Leftrightarrow \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\hat{r}_\theta(x, y)] - \frac{1}{2}\mathbb{E}_{P_X P_Y}[\hat{r}_\theta^2(x, y)]. \tag{5.6}$$

The objective is also called least-square density-ratio fitting method [Kanamori et al., 2009] and has a unique solution $\hat{r}_\theta^*(x, y) = p(x, y)/p(x)p(y)$. We refer Eq. (5.6) as *Density-Ratio Fitting* PD estimation.

5.3 Application I: Mutual Information Estimation

By definition, as the average effect of point-wise dependency (PD), Mutual Information (MI) measures the statistical independence between random variables:

$$\begin{aligned} I(X; Y) = D_{\mathrm{KL}}(P_{X,Y} \parallel P_X P_Y) &= \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \, \mathrm{d}x\mathrm{d}y = \mathbb{E}_{P_{X,Y}}[\log r(x, y)] \\ &\approx \mathbb{E}_{P_{X,Y}}[\log \hat{r}_\theta(x, y)] \approx \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x, y)], \end{aligned} \tag{5.7}$$

where we estimate MI by directly plugging-in PD (i.e., $\hat{r}_\theta$ in Eq. (5.5), (5.6)) or PMI (i.e., $\hat{f}_\theta$ in Eq. (5.1), (5.2), and (5.3)). In summary, we cast the MI estimation problem to a PD or PMI estimation problem.

Baseline Models Instead of approximating MI by plugging-in the estimated PD or PMI, prior work focuses on establishing tractable and scalable bounds for MI [Belghazi et al., 2018, Oord et al., 2018, Poole et al., 2019, Song and Ermon, 2019], in which the bounds can be computed via gradient descent over neural networks. Strong baselines include CPC [Oord et al., 2018], NWJ [Belghazi et al., 2018], JS [Poole et al., 2019], DV (MINE) [Belghazi et al., 2018], and SMILE [Song and Ermon, 2019]. To understand the differences, we separate MI neural estimation methods into two procedures: *learning* and *inference*. The learning step learns the parameters when estimating 1) point-wise dependency/ logarithm of point-wise dependency; or 2) MI lower bound. The inference step considers the parameters from the learning step and infers value for 1) MI itself; or 2) a lower bound of MI. We summarize different approaches in Table 5.1. For completeness, one may see Chapter 5.7.1 and 5.7.2 for more details about these bounds.

Table 5.1: MI neural estimation methods. The estimation procedure is dissected into learning and inference phases, which may use different objectives. Baselines consider to estimate MI via lower bounds, while ours consider to estimate MI via plugging in PD ($\hat{r}_\theta$) or PMI ($\hat{f}_\theta$) estimators.

Baselines	Learning	Inference
CPC [Oord et al., 2018]	I_{CPC} [Oord et al., 2018]	I_{CPC} [Oord et al., 2018]
NWJ [Belghazi et al., 2018]	I_{NWJ} [Belghazi et al., 2018, Nguyen et al., 2010]	I_{NWJ} [Belghazi et al., 2018, Nguyen et al., 2010]
JS [Poole et al., 2019]	I_{JS} [Nowozin et al., 2016] (Eq. (5.1))	I_{NWJ} [Belghazi et al., 2018, Nguyen et al., 2010]
DV (MINE) [Belghazi et al., 2018]	I_{DV} [Belghazi et al., 2018]	I_{DV} [Belghazi et al., 2018, Donsker and Varadhan, 1983]
SMILE Song and Ermon [2019]	I_{JS} [Nowozin et al., 2016] (Eq. (5.1))	I_{DV} [Belghazi et al., 2018, Donsker and Varadhan, 1983]

Ours	Learning	Inference
Variational MI Bounds	I_{JS} [Nowozin et al., 2016] (Eq. (5.1))	Eq. (5.7) with $\hat{f}_\theta$
Probabilistic Classifier	Eq. (5.4)	Eq. (5.7) with $\hat{r}_\theta$ in Eq. (5.5)
Density Matching I	Eq. (5.2)	Eq. (5.7) with $\hat{f}_\theta$
Density Matching II	Eq. (5.3)	Eq. (5.7) with $\hat{f}_\theta$
Density-Ratio Fitting	Eq. (5.6)	Eq. (5.7) with $\hat{r}_\theta$

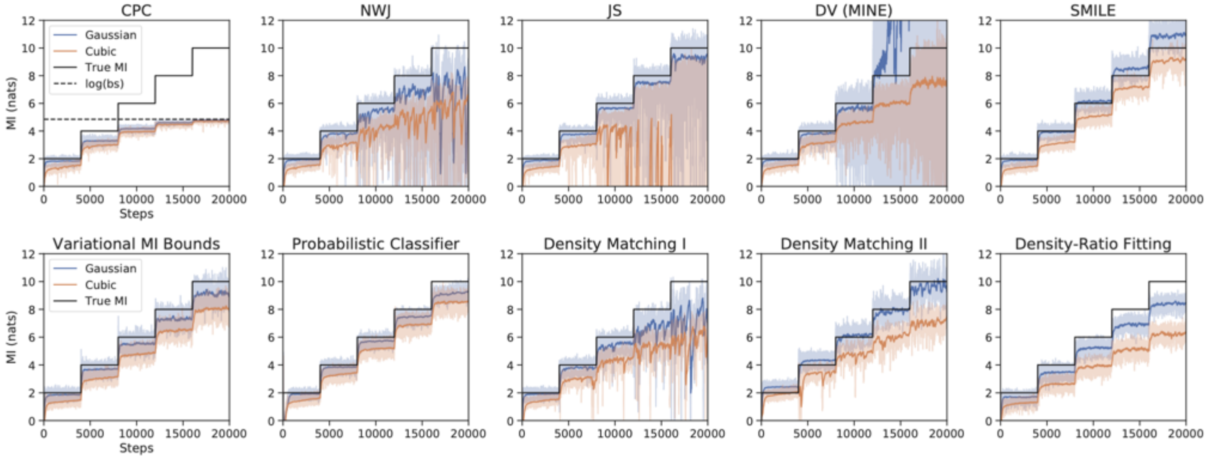


Figure 5.1: **Gaussian** and **Cubic** task for correlated Guassians with tractable ground truth MI. The upper row are the baselines and the lower row are our methods. Network, learning rate, optimizer, and batch size are fixed for all MI neural estimators. The only differences are the learning and inference objectives shown in Table 5.1.

Benchmarking on Correlated Gaussians To evaluate the performance between different MI neural estimators, we consider the standard tasks on correlated Gaussians [Belghazi et al., 2018, Poole et al., 2019, Song and Ermon, 2019]. In particular, we draw (x, y) from two 20-dimensional Gaussians with correlation ρ, which is referred as **Gaussian** task. Then, we apply a cubic transformation on y so that $y \mapsto y^3$, which is referred to as **Cubic** task. These two tasks have tractable ground truth MI $= -10 \log(1-\rho^2)$. We train all models for $20,000$ iterations, starting from MI $= 2$ and increasing it by 2 per $4,000$ iterations. We fix the network, learning rate, optimizer, and batch size across all the estimators for a fair comparison. The only differences are the objectives considered in the learning and inference in MI estimation (shown in Table 5.1).

Results & Discussions We present the results in Figure 5.1 and leave more training details in Chapter 5.7.2. In the following, we discuss bias-variance trade-offs for different approaches. We first discuss general observations. Most of the estimators have both larger bias and variance with larger ground truth MI. The only exception is CPC [Oord et al., 2018], where its value is upper bounded by $\log(\text{batch\_size})$ [Poole et al., 2019]. The bias is also larger in **Gaussian** task than in **Cubic** task except for DV [Belghazi et al., 2018]. Next, we discuss the differences among estimators in detail. CPC [Oord et al., 2018] has the smallest variance, yet it is highly biased. Although having larger variance than CPC, SMILE [Song and Ermon, 2019]/ Variational MI Bounds/ Probabilistic Classifier/ Density Matching I & II/ Density-Ratio Fitting approaches have a much lower bias. Among them, Probabilistic Classifier and Density-Ratio Fitting approaches have the smallest variance. NWJ [Belghazi et al., 2018]/ JS [Poole et al., 2019]/ DV [Belghazi et al., 2018], whereas, have both large variance and bias. Note that JS [Poole et al., 2019] has larger variance is because using I_{NWJ} objective during inference. To sum up, we see that the plug-in MI estimators enjoy smaller variance and bias when comparing to most of the lower bound methods.

Theoretical Analysis In Eq. (5.7), we present a high-level intuition that a good estimation of the PD function $\hat{r}_\theta(x,y)$ could be used to estimate the mutual information. In what follows, we present a formal justification for this argument. To begin with, let $P^{(n)}_{X,Y}$ denote the empirical distribution of the ground-truth joint distribution $P_{X,Y}$ estimated from n samples drawn uniformly at random from $P_{X,Y}$. Then our estimator of the mutual information is given by $\widehat{I}^{(n)}_\theta(X;Y) := \mathbb{E}_{P^{(n)}_{X,Y}}[\log \hat{r}_\theta(x,y)]$.

At a high level, our arguments contain two parts. In the first part, we show that w.h.p. (with high probability) $\widehat{I}^{(n)}_\theta(X;Y)$ is close to $\mathbb{E}_{P_{X,Y}}[\log \hat{r}_\theta(x,y)]$. In the second part, we apply the universal approximation lemma of neural networks [Hornik et al., 1989] to show that there exists $\hat{r}_\theta(\cdot,\cdot)$ that is close to $r(\cdot,\cdot)$. Formally, let $\mathcal{F} := \{\hat{r}_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ be the set of neural networks where the parameter θ is a d-dimensional vector. Throughout the analysis, we assume the following assumptions:

Assumption 1 (Boundedness of the density ratio)**.** There exist universal constants $C_l \le C_u$ such that $\forall \hat{r}_\theta \in \mathcal{F}$ and $\forall x, y$, $C_l \le \log \hat{r}_\theta(x,y) \le C_u$.

Assumption 2 (log-smoothness of the density ratio)**.** There exists $\rho > 0$ such that for $\forall x, y$ and $\forall \theta_1, \theta_2 \in \Theta$, $|\log \hat{r}_{\theta_1}(x,y) - \log \hat{r}_{\theta_2}(x,y)| \le \rho \cdot \|\theta_1 - \theta_2\|$.

Assumption 1 basically asks the output of a neural net to be bounded and Assumption 2 says that for any given input pair, the output of the network should only change slightly if we just slightly perturb the network weights. Both assumptions are mostly verified in practical networks. Based on these two assumptions, the following lemma is adapted from Bartlett [1998] that bounds the rate of uniform convergence of a function class in terms of its covering number. The original lemma is based on the L_∞ norm of the function class; whereas the following one, we use the L_2 norm on Θ.

Lemma 3. (estimation). Let $\varepsilon > 0$ and $\mathcal{N}(\Theta, \varepsilon)$ be the covering number of Θ with radius ε under L_2 norm. Let $P_{X,Y}$ be any distribution where $S = \{x_i, y_i\}_{i=1}^n$ are sampled from and define $M := C_u - C_l$, then

$$\Pr_S\left(\sup_{\hat{r}_\theta \in \mathcal{F}} \left|\widehat{I}^{(n)}_\theta(X;Y) - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_\theta(x,y)]\right| \ge \varepsilon\right) \le 2\mathcal{N}(\Theta, \varepsilon/4\rho)\exp\left(-\frac{n\varepsilon^2}{2M^2}\right). \tag{5.8}$$

Next lemma is derived from [Hornik et al., 1989], which shows that neural networks are universal approximators:

Lemma 4 (Hornik et al. [1989], approximation)**.** Let $\varepsilon > 0$. There exists $d \in \mathbb{N}$ and a family of neural networks $\mathcal{F} := \{\hat{r}_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ where Θ is compact, such that $\inf_{\hat{r}_\theta \in \mathcal{F}} \left|\mathbb{E}_{P_{X,Y}}[\log \hat{r}_\theta(x,y)] - I(X;Y)\right| \le \varepsilon$.

Combining both lemmas, we are ready to state the following main result:

Theorem 1. Let $0 < \delta < 1$. There exists $d \in \mathbb{N}$ and a family of neural networks $\mathcal{F} := \{\hat{r}_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ where Θ is compact, so that $\exists \theta^* \in \Theta$, with probability at least $1 - \delta$ over the draw of

$S = \{x_i, y_i\}_{i=1}^n \sim P_{X,Y}^{\otimes n}$,

$$\left|\widehat{I}_{\theta^*}^{(n)}(X;Y) - I(X;Y)\right| \leq O\left(\sqrt{\frac{d+\log(1/\delta)}{n}}\right). \tag{5.9}$$

It is worth pointing out that the above theorem is a theorem of existence, but *not* a constructive theorem, meaning that it does not give an estimator explicitly. To sum up, it shows that there exists a neural network θ^* such that, w.h.p., $\widehat{I}_{\theta^*}^{(n)}(X;Y)$ can approximate $I(X;Y)$ with n samples at a rate of $O(1/\sqrt{n})$.

5.4 Application II: Self-supervised Representation Learning

Self-supervised representation learning aims at extracting task-relevant information without access to label or downstream signals. Among different self-supervised representation learning techniques, *contrastive learning* may be the most popular one with empirical [Agrawal et al., 2015, Arandjelovic and Zisserman, 2017, Bachman et al., 2019, Chen et al., 2020a, He et al., 2019, Hénaff et al., 2019, Hjelm et al., 2018, Jayaraman and Grauman, 2015, Kong et al., 2019, Oord et al., 2018, Ozair et al., 2019, Tian et al., 2019] and theoretical [Arora et al., 2019, Tsai et al., 2020c] support. The core of contrastive learning is having the representations sampled from similar pairs be differentiated from random pairs. In other words, we hope that the representations learned from the similar pairs have higher point-wise dependency than the random pairs. Let v_1/v_2 denote two different views for the same data, v_2' represent a view from a different data, and F/G be two mapping functions from data to representations. In short, contrastive learning objective learns F/G such that $r(F(v_1), G(v_2))$ is much larger than $r(F(v_1), G(v_2'))$.

Connection between Contrastive Learning and PD Our goal is to show that our learning objectives resemble contrastive learning. We first take the *Probabilistic Classifier* approach as an example and incorporate the learning of F/G, which we name it as *Probabilistic Classifier Coding* (PCC):

$$\sup_{F,G}\sup_{\theta\in\Theta} \mathbb{E}_{P_{\mathcal{V}_1,\mathcal{V}_2}}[\log \hat{p}_\theta(c=1|(F(v_1),G(v_2)))] + \mathbb{E}_{P_{\mathcal{V}_1}P_{\mathcal{V}_2}}[\log\Big(1-\hat{p}_\theta(c=1|(F(v_1),G(v_2')))\Big)], \tag{5.10}$$

which aims at learning F/G to better classify (i.e., differentiate) between similar or random data pairs. Next, we consider the *Density-Ratio Fitting* approach, which we refer to the objective as *Density-Ratio Fitting Coding* (D-RFC):

$$\sup_{F,G}\sup_{\theta\in\Theta} \mathbb{E}_{P_{\mathcal{V}_1,\mathcal{V}_2}}[\hat{r}_\theta(F(v_1),G(v_2))] - \frac{1}{2}\mathbb{E}_{P_{\mathcal{V}_1}P_{\mathcal{V}_2}}[\hat{r}_\theta^2(F(v_1),G(v_2'))], \tag{5.11}$$

which aims at learning F/G to maximize $\hat{r}_\theta(F(v_1),G(v_2))$ and minimize $\hat{r}_\theta(F(v_1),G(v_2'))$. We leave the discussion for the adaptations of Variational MI Bounds, Density Matching I ,and Density Matching II in Chapter 5.7.3.

Baseline Model The most adopted contrastive representation learning objective is Contrastive Predictive Coding (CPC) [Oord et al., 2018]:

$$\sup_{F,G}\sup_{\theta\in\Theta} \mathbb{E}_{(v_1^1,v_2^1)\sim P_{\mathcal{V}_1,\mathcal{V}_2},\cdots(v_1^n,v_2^n)\sim P_{\mathcal{V}_1,\mathcal{V}_2}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(F(v_1^i),G(v_2^i))}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(F(v_1^i),G(v_2^j))}}],$$

where $\{v_1^i, v_2^i\}_{i=1}^n$ are independently and identically sampled from $P_{\mathcal{V}_1,\mathcal{V}_2}$. $\hat{c}_\theta(\cdot)$ is a function that takes the representations learned from the data pairs and returns a scalar.

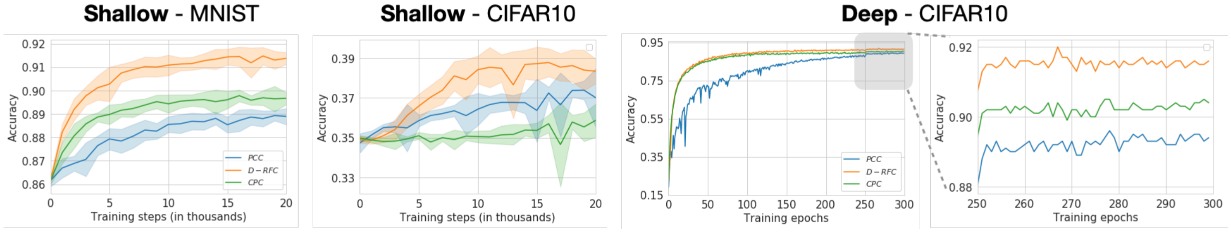


Figure 5.2: **Shallow** [Tschannen et al., 2019] and **Deep** [Bachman et al., 2019] task for self-supervised visual representation learning using *downstream linear evaluation protocol*. We compare the presented Probabilistic Classifier Coding (PCC) and Density-Ratio Fitting Coding (D-RFC) with baseline Contrastive Predictive Coding (CPC). Network, learning rate, optimizer, and batch size are fixed for all the methods. The only differences are the learning objectives.

Experimental Setup We compare our proposed approaches with CPC [Oord et al., 2018] on two tasks [Bachman et al., 2019, Tschannen et al., 2019]. Due to the fact that the performance of the self-supervisedly learned representations strongly depends on the choice of feature extractor architectures and the parametrization of the employed MI estimators [Tschannen et al., 2019]. For a fair comparison, we fix the network, learning rate, optimizer, and batch size when comparing between different objectives. In the first set of experiments, we choose a relatively shallow network as suggested by Tschannen et al. [2019], performing self-supervised learning experiments on MNIST [LeCun et al., 1998] and CIFAR10 [Krizhevsky et al., 2009]. We report the average and standard deviations from 10 random trials. This task is referred to as **shallow** experiment. In the second set of experiments, we choose a relatively deep network as suggested by Bachman et al. [2019], performing experiments on CIFAR10. This task is referred to as **deep** experiment. Both the **shallow** and **deep** tasks perform representation learning without access to the label information, and then the performance is evaluated by *downstream linear evaluation protocol* [Bachman et al., 2019, Hénaff et al., 2019, Hjelm et al., 2018, Kolesnikov et al., 2019, Oord et al., 2018, Tian et al., 2019, Tschannen et al., 2019]. Specifically, a linear classifier is trained from the self-supervisedly learned (fixed) representation to the labels on the training set. We present the results with convergence in Figure 5.2. One may see Chapter 5.7.3 for more details.

Results & Discussions Prior approaches [Ozair et al., 2019, Poole et al., 2019, Song and Ermon, 2019, Tschannen et al., 2019] contend that a valid MI lower bound or an objective with better MI estimation may not result in better representations. We have a similar observation that D-RFC performs the best (when comparing to CPC and PCC) while it is neither a lower bound of MI nor the best objective of MI estimation. Next, we see an inconsistent trend when comparing PCC to CPC. In the Shallow task on CIFAR10, PCC performs better than CPC, while it performs worse on the other experiments. To sum up, we show our PD estimation objectives can be used for self-supervised representation learning, which is either at par or better than prior approaches.

5.5 Application III: Cross-modal Learning

Here, we discuss the usage of point-wise dependency (PD) estimation for data containing information across modalities - audio and text.

Table 5.2: Cross-modal Retrieval task with unsupervised word features across acoustic and textual modalities. *Probabilistic Classifier* approach is used to estimate PD between the audio and textual features of a given word. The estimator is trained on the training split. We report the $1:5$ matching results from audio to textual features on the test split, where we obtain 96.24% top-1 retrieval accuracy.

Correct Audio-Textual Retrieval Examples (Top-1 Accuracy: 96.24%)					
Audio Feature	Textual Features (Ranked by logarithm of point-wise dependency)				
depths	**depths (15.22)**	mildewed (-58.62)	lugged (-92.24)	alison (-108.02)	raffleshurst (-161.74)
receptacle	**receptacle (1.32)**	bloated (-15.41)	recreate (-39.77)	sting (-90.51)	pity (-104.44)
frontiers	**frontiers (3.36)**	institution (-31.01)	laterally (-54.17)	pretends (-105.11)	vibrating (-124.88)

Incorrect Audio-Textual Retrieval Examples					
Audio Feature	Textual Features (Ranked by logarithm of point-wise dependency)				
cos	tortoise (-2.33)	**cos (-10.72)**	tickling (-12.53)	undressed (-18.11)	cromwell's (-44.31)
elbowing	itinerary (-6.51)	**elbowing (-8.22)**	swims (-12.98)	rigid (-24.14)	integrity (-39.76)
alma's	roughness (-3.11)	**alma's (-3.67)**	montreal (-11.81)	tuneful (-12.22)	levant (-18.26)

Experimental Setup - Cross-modal Retrieval We instantiate the discussion using unsupervised word features[1] which are learned from text corpora (i.e., Word2Vec [Mikolov et al., 2013] method) and human speech (i.e., Speech2Vec [Chung and Glass, 2018] method). In particular, in this dataset, a word feature has two distinct features: audio and textual feature. We denote $\mathcal{X}$ as the audio sample space and $\mathcal{Y}$ as the textual sample space. Since our goal is not comparing between different approaches but presenting the usage of PD estimation for cross-modal learning, we select only one approach *Probabilistic Classifier* as our objective for estimating PD. Note that we report the logarithm of PD, which is PMI in the results. One may refer to Chapter 5.7.4 for more details on training and datasets.

By definition, given an audio feature x and a textual feature y, their point-wise dependency $r(x, y)$ measures their statistical dependency. For example, if x_1 and y_1 are the features for the same word, and y_2 is the feature for another word, then $r(x_1, y_1) > r(x_1, y_2)$ (in most cases). As a consequence, we can train PD estimators using the training split, and computing PD values for cross-modal retrieval on the test split.

Results & Discussions In Table 5.2, we report the results on $1:5$ matching[2] from audio to textual features. First, we obtain 96.24% top-1 retrieval accuracy using PD estimation (with *Probabilistic Classifier* approach). Another approach such as *Density-Ratio Fitting* obtains 92.26% top-1 retrieval accuracy. Then, we study the success and failure retrieval cases. The success examples show the highest statistical dependency (i.e., the highest PMI) between the audio and textual features of the same word. The failure examples, on the contrary, (all of them) have the second-highest PMI between the audio and textual features of the same word. Last, we observe that only the correctly retrieved cross-modal features have positive PMI values, which suggest two features are statistically dependent. As a summary, PD acts as a statistical dependency measurement, and we show its estimation can be generalized from training to test split for cross-modal retrieval.

[1] The word features can be downloaded from `https://github.com/iamyuanchung/speech2vec-pretrained-vectors`.

[2] One trial contains an audio feature, its corresponding textual feature, and 4 randomly sampled textual features.

5.6 Discussion

In this chapter, we discuss the sub-challenge of mutual information estimation within the challenge of relationship quantification. We study both mutual information, which is an aggregate statistic of the dependency between two random variables, and instance-level dependency. To overcome the curse of dimensionality in classical kernel-based approaches, we leverage the power of rich and flexible neural networks to model high-dimensional data. In particular, we first show that point-wise dependency is a natural product from optimizing mutual information variational bounds. Then, we further develop two point-wise dependency estimation approaches: Probabilistic Classifier and Density-Ratio Fitting that are free of optimizing mutual information variational bounds. A diversified set of experiments manifest the advantages of using our approaches.

5.7 Appendix

5.7.1 Optimization Objectives for Point-wise Dependency Neural Estimation

In what follows, we shall show detailed derivations for the point-wise dependency estimation methods. Four approaches are discussed: *Variational Bounds of Mutual Information*, *Density Matching*, *Probabilistic Classifier*, and *Density-Ratio Fitting*. For convenience, we define $\Omega = \mathcal{X} \times \mathcal{Y}$. We have $P_{X,Y}$ and $P_X P_Y$ (can also be written as $P_X \otimes P_Y$) be the probability measures over $\sigma-$algebras over Ω with their probability densities being the Radon-Nikodym derivatives (i.e., $p(x,y) = dP_{X,Y}/d\mu$ and $p(x)p(y) = dP_X P_Y/d\mu$ with μ being the Lebesgue measure).

5.7.1.1 Method I: Variational Bounds of Mutual Information

Recent advances [Belghazi et al., 2018, Poole et al., 2019] propose to estimate mutual information (MI) using neural network either from variational MI lower bounds (e.g., I_{NWJ} [Belghazi et al., 2018] and I_{DV} [Belghazi et al., 2018]) or a variational form of MI (e.g., I_{JS} [Poole et al., 2019]). These estimators have the logarithm of point-wise dependency (PMI) as the intermediate product, which we will show in the following. We denote $\mathcal{M}$ be any class of functions $m : \Omega \to \mathbb{R}$.

Proposition 3 (I_{NWJ} and its neural estimation, restating Nguyen-Wainwright-Jordan bound [Belghazi et al., 2018, Nguyen et al., 2010])**.**

$$I_{\text{NWJ}} := \sup_{m \in \mathcal{M}} \mathbb{E}_{P_{X,Y}}[m(x,y)] - e^{-1}\mathbb{E}_{P_X P_Y}[e^{m(x,y)}] = \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - e^{-1}\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]$$

has the optimal function $m^*(x,y) = 1 + \log \frac{p(x,y)}{p(x)p(y)}$. And when Θ is large enough, the optimal $\hat{f}^*_\theta(x,y) = 1 + \log \frac{p(x,y)}{p(x)p(y)}$.

Proof. The second-order functional derivative of the objective is $-e^{-1} \cdot e^{m(x,y)} \cdot dP_X P_Y$, which is always negative. The negative second-order functional derivative implies the objective has a supreme value. Then, take the first-order functional derivative $\frac{\partial I_{\text{NWJ}}}{\partial m}$ and set it to zero:

$$dP_{X,Y} - e^{-1} \cdot e^{m(x,y)} \cdot dP_X P_Y = 0.$$

We then get optimal $m^*(x,y) = 1 + \log \frac{dP_{X,Y}}{dP_X P_Y} = 1 + \log \frac{p(x,y)}{p(x)p(y)}$. When Θ is large enough, by universal approximation theorem of neural networks [Hornik et al., 1989], the approximation in Proposition 3 is tight, which means $\hat{f}^*_\theta(x,y) = m^*(x,y) = 1 + \log \frac{p(x,y)}{p(x)p(y)}$. ■

Proposition 4 (I_{DV} and its neural estimation, restating Donsker-Varadhan bound [Belghazi et al., 2018, Donsker and Varadhan, 1983]).

$$I_{\mathrm{DV}} := \sup_{m \in \mathcal{M}} \mathbb{E}_{P_{X,Y}}[m(x,y)] - \log\Big(\mathbb{E}_{P_X P_Y}[e^{m(x,y)}]\Big)] = \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \log\Big(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]\Big)]$$

has optimal functions $m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)} + \text{Const.}$. And when Θ is large enough, the optimal $\hat{f}^*_\theta(x,y) = \log \frac{p(x,y)}{p(x)p(y)} + \text{Const.}$.

Proof. Let $\mathbf{1}_.$ be an indicator function, and the second-order functional derivative of the objective is

$$-\frac{e^{m(x,y)} \cdot \mathbb{E}_{(x',y') \sim P_X P_Y}\Big[e^{m(x',y')} \cdot \mathbf{1}_{(x',y') \neq (x,y)}\Big]}{\Big(\mathbb{E}_{P_X P_Y}[e^{m(x,y)}]\Big)^2} \cdot dP_X P_Y,$$

which is always negative. The negative second-order functional derivative implies the objective has a supreme value. Then, take the first-order functional derivative $\frac{\partial I_{\mathrm{DV}}}{\partial m}$ and set it to zero:

$$dP_{X,Y} - \frac{e^{m(x,y)}}{\mathbb{E}_{P_X P_Y}[e^{m(x,y)}]} \cdot dP_X P_Y = 0.$$

We then have $m^*(x,y)$ take the forms $m^*(x,y) = \log \frac{dP_{X,Y}}{dP_X P_Y} + \text{Const.} = \log \frac{p(x,y)}{p(x)p(y)} + \text{Const.}$. When Θ is large enough, by universal approximation theorem of neural networks [Hornik et al., 1989], the approximation in Proposition 4 is tight, which means $\hat{f}^*_\theta(x,y) = m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)} + \text{Const.}$. ■

Proposition 5 (I_{JS} and its neural estimation, restating Jensen-Shannon bound with f-GAN objective [Poole et al., 2019]).

$$\begin{aligned} I_{\mathrm{JS}} :&= \sup_{m \in \mathcal{M}} \mathbb{E}_{P_{X,Y}}\Big[-\text{softplus}\Big(-m(x,y)\Big)\Big] - \mathbb{E}_{P_X P_Y}\Big[\text{softplus}\Big(m(x,y)\Big)\Big] \\ &= \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}\Big[-\text{softplus}\Big(-\hat{f}_\theta(x,y)\Big)\Big] - \mathbb{E}_{P_X P_Y}\Big[\text{softplus}\Big(\hat{f}_\theta(x,y)\Big)\Big] \end{aligned}$$

with softplus function being $\text{softplus}\,(x) = \log\Big(1 + \exp\,(x)\Big)$ and the optimal solution $m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$. And when Θ is large enough, the optimal $\hat{f}^*_\theta(x,y) = m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$.

Proof. The second-order functional derivative of the objective is

$$-\frac{1}{\Big(1 + e^{m(x,y)}\Big)^2} \cdot e^{m(x,y)} \cdot dP_{X,Y} - \frac{1}{\Big(1 + e^{-m(x,y)}\Big)^2} \cdot e^{-m(x,y)} \cdot dP_X P_Y,$$

which is always negative. The negative second-order functional derivative implies the objective has a supreme value. Then, take the first-order functional derivative $\frac{\partial I_{\mathrm{JS}}}{\partial m}$ and set it to zero:

$$\frac{1}{1 + e^{-m(x,y)}} \cdot e^{-m(x,y)} \cdot dP_{X,Y} - \frac{1}{1 + e^{m(x,y)}} \cdot e^{m(x,y)} \cdot dP_X P_Y = 0.$$

We then get $m^*(x,y) = \log \frac{dP_{X,Y}}{dP_X P_Y} = \log \frac{p(x,y)}{p(x)p(y)}$. When Θ is large enough, by universal approximation theorem of neural networks [Hornik et al., 1989], the approximation in Proposition 5 is tight, which means $\hat{f}^*_\theta(x,y) = m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$. ■

We see that either I_{NWJ} (Proposition 3) or I_{JS} (Proposition 5) gives us the optimal PMI estimation, while I_{DV} (Proposition 4) is less preferable since its optimal solution includes an arbitrary constant. In practice, we prefer I_{JS} over $I_{\text{NWJ}}/I_{\text{DV}}$ due to its better training stability [Poole et al., 2019].

5.7.1.2 Method II: Density Matching

This method considers to match the true joint density $p(x,y)$ and the estimated joint density via KL-divergence. We let the estimated joint probability be $P_{m,X,Y}$ with its joint density being $e^{m(x,y)}p(x)p(y)$, where $e^{m(x,y)}$ acts to ensure the estimated joint density is a valid probability density function. Hence, we let $m \in \mathcal{M}''$ with $\mathcal{M}''$ being 1) any class of functions $m : \Omega \to \mathbb{R}$; and 2) $\int e^{m(x,y)}\, dP_X P_Y = \mathbb{E}_{P_X P_Y}[e^{m(x,y)}] = 1$.

Proposition 6 (KL Loss in Density Matching and its neural estimation)**.**

$$\begin{aligned} L_{\text{KL}_{\text{DM}}} &:= \sup_{m \in \mathcal{M}''} \mathbb{E}_{P_{X,Y}}[m(x,y)] \\ &= \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] \text{ s.t. } \mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}] = 1 \end{aligned}$$

with the optimal $m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$. And when Θ is large enough, the optimal $\hat{f}^*_\theta(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$.

Proof. First, we compute the KL-divergence:

$$\begin{aligned} L_{\text{KL}_{\text{DM}}} &= \inf_{m \in \mathcal{M}''} D_{\text{KL}}(P_{X,Y} \parallel \hat{P}_{X,Y}) = \inf_{m \in \mathcal{M}''} H(P_{X,Y}) - \mathbb{E}_{P_{X,Y}}\Big[\log e^{m(x,y)} p(x)p(y)\Big] \\ &= \inf_{m \in \mathcal{M}''} H(P_{X,Y}) - \mathbb{E}_{P_{X,Y}}\Big[\log p(x)p(y)\Big] - \mathbb{E}_{P_{X,Y}}\Big[m(x,y)\Big] \\ &= \inf_{m \in \mathcal{M}''} I(X;Y) - \mathbb{E}_{P_{X,Y}}\Big[m(x,y)\Big] = \text{Const.} + \sup_{m \in \mathcal{M}''} \mathbb{E}_{P_{X,Y}}\Big[m(x,y)\Big] \\ &\Leftrightarrow \sup_{m \in \mathcal{M}} \mathbb{E}_{P_{X,Y}}[m(x,y)] \text{ s.t. } \mathbb{E}_{P_X P_Y}[e^{m(x,y)}] = 1. \end{aligned}$$

Consider the following Lagrangian:

$$h(m, \lambda_1, \lambda_2) := \mathbb{E}_{P_{X,Y}}[m] - \lambda(\mathbb{E}_{P_X P_Y}[e^m] - 1),$$

where $\lambda \in \mathbb{R}$. Taking the functional derivative and setting it to be zero, we see

$$dP_{X,Y} - \lambda \cdot e^m \cdot dP_X dP_Y = 0.$$

To satisfy the constraint, we obtain

$$\mathbb{E}_{P_X P_Y}[e^m] = 1 \iff E_{P_X P_Y}[\frac{1}{\lambda}\frac{dP_{X,Y}}{dP_X P_Y}] = \frac{1}{\lambda} E_{P_X P_Y}[\frac{dP_{X,Y}}{dP_X P_Y}] = \frac{1}{\lambda} = 1 \iff \lambda = 1.$$

Plugging-in $\lambda = 1$, the optimal $m^*(x,y) = \log \frac{dP_{XY}}{dP_X P_Y} = \log \frac{p(x,y)}{p(x)p(y)}$. When Θ is large enough, by universal approximation theorem of neural networks [Hornik et al., 1989], the approximation in Proposition 6 is tight, which means $\hat{f}^*_\theta(x,y) = m^*(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$. ■

The objective function in Proposition 6 is a constrained optimization problem, and we present two relaxed optimization objectives. The first one is Lagrange relaxation:

$$\sup_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \lambda\Big(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}] - 1\Big)$$

with the optimal Lagrange coefficient $\lambda = 1$ (see proof for Proposition 6).

The second one is log barrier method:

$$\sup_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \eta\Big(\log \mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]\Big)^2,$$

where $\eta > 0$ is a hyper-parameter controlling the regularization term.

5.7.1.3 Method III: Probabilistic Classifier

This approach casts the PD estimation as the problem of estimating the 'class'-posterior probability. We use a Bernoulli random variable C to classify the samples drawn from the joint density ($C = 1$ for $(x,y) \sim P_{X,Y}$) and the samples drawn from product of the marginal densities ($C = 0$ for $(x,y) \sim P_X P_Y$). In order to present our derivation, we define $H(\cdot)$ as the entropy and $H(\cdot,\cdot)$ as the cross entropy. Slightly abusing notation, we define $\Omega' = \mathcal{X} \times \mathcal{Y} \times \{0,1\}$ and $\mathcal{M}'$ is 1) any class of functions $m : \Omega' \to (0,1)$; and 2) $m(x,y,0) + m(x,y,1) = 1$ for any x and y. Note that since $m(x,y,c)$ is always positive and $m(x,y,0) + m(x,y,1) = 1$ for any x, y, $m(x,y,c)$ is a proper probability mass function with respect to C given any x, y. Consider the binary cross entropy loss:

Proposition 7 (Binary Cross Entropy Loss in Probabilistic Classifier Method and its neural estimation)**.**

$$\begin{aligned} L_{\mathrm{BCE_{PC}}} :=& \sup_{m\in\mathcal{M}'} \mathbb{E}_{P_{X,Y}}[\log m(x,y,C=1)] + \mathbb{E}_{P_X P_Y}[\log\Big(1 - m(x,y,C=1)\Big)] \\ =& \sup_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\log \hat{p}_\theta(C=1|(x,y))] + \mathbb{E}_{P_X P_Y}[\log\Big(1 - \hat{p}_\theta(C=1|(x,y))\Big)] \end{aligned}$$

with the optimal $m^*(x,y,c) = p(c|(x,y))$. And when Θ is large enough, the optimal $\hat{p}^*_\theta(c|(x,y)) = p(c|(x,y))$.

Proof. We see

$$\begin{aligned}
L_{\mathrm{BCE}_{\mathrm{PC}}} &= \inf_{m\in\mathcal{M}'} \mathbb{E}_{P_{XY}}\Big[H\Big(P(C|(x,y)), m(x,y,C)\Big)\Big] + \mathbb{E}_{P_XP_Y}\Big[H\Big(P(C|(x,y)), m(x,y,C))\Big)\Big] \\
&= \inf_{m\in\mathcal{M}'} \mathbb{E}_{P_{XY}}\Big[H\Big(P(C|(x,y))\Big) + D_{\mathrm{KL}}(P(C|(x,y)) \parallel m(x,y,C))\Big] \\
&\qquad + \mathbb{E}_{P_XP_Y}\Big[H\Big(P(C|(x,y))\Big) + D_{\mathrm{KL}}(P(C|(x,y)) \parallel m(x,y,C))\Big] \\
&= \text{Const.} + \inf_{m\in\mathcal{M}'} \mathbb{E}_{P_{XY}}\Big[D_{\mathrm{KL}}(P(C|(x,y)) \parallel m(x,y,C))\Big] \\
&\qquad\qquad + \mathbb{E}_{P_XP_Y}\Big[D_{\mathrm{KL}}(P(C|(x,y)) \parallel m(x,y,C))\Big] \\
&= \text{Const.} + \inf_{m\in\mathcal{M}'} \mathbb{E}_{P_{XY}}\Big[\mathbb{E}_{P(C|(x,y))}[-\log m(x,y,c)]\Big] \\
&\qquad\qquad + \mathbb{E}_{P_XP_Y}\Big[\mathbb{E}_{P(C|(x,y))}[-\log m(x,y,c)]\Big] \\
&= \text{Const.} + \inf_{m\in\mathcal{M}'} \mathbb{E}_{P_{XY}}[-\log m(x,y,C=1)] + \mathbb{E}_{P_XP_Y}[-\log m(x,y,C=0)] \\
&\Leftrightarrow \sup_{m\in\mathcal{M}'} \mathbb{E}_{P_{X,Y}}[\log m(x,y,C=1)] + \mathbb{E}_{P_XP_Y}[\log\Big(1 - m(x,y,C=1)\Big)].
\end{aligned}$$

The optimal m^* happens when $D_{\mathrm{KL}}(P(C|(x,y)) \parallel m^*(x,y,C)) = 0$ for any (x,y), which implies $m^*(x,y,c) = p(c|(x,y))$. When Θ is large enough, by universal approximation theorem of neural networks [Hornik et al., 1989], the approximation in Proposition 7 is tight, which means $\hat{p}^*_\theta(c|(x,y)) = m^*(x,y,c) = p(c|(x,y))$. ■

The obtained estimated class-posterior classifier can be used for approximating point-wise dependency (PD):

$$\hat{r}_\theta(x,y) = \frac{n_{P_XP_Y}}{n_{P_{X,Y}}}\frac{\hat{p}_\theta(C=1|(x,y))}{\hat{p}_\theta(C=0|(x,y))} \text{ with } (x,y) \sim P_{X,Y} \text{ or } (x,y) \sim P_XP_Y.$$

5.7.1.4 Method IV: Density-Ratio Fitting

Let $\mathcal{M}$ be any class of functions $m : \Omega \to \mathbb{R}$. This approach considers to minimize the expected (in $\mathbb{E}_{P_XP_Y}[\cdot]$) least-square difference between the true PD $r(x,y)$ and the estimated PD $m(x,y)$:

Proposition 8 (Least-Square Loss in Density-Ratio Fitting and its neural estimation)**.**

$$L_{\mathrm{LS}_{\mathrm{D-RF}}} := \sup_{m\in\mathcal{M}} \mathbb{E}_{P_{X,Y}}[m(x,y)] - \frac{1}{2}\mathbb{E}_{P_XP_Y}[m^2(x,y)] = \sup_{\theta\in\Theta} \mathbb{E}_{P_{X,Y}}[\hat{r}_\theta(x,y)] - \frac{1}{2}\mathbb{E}_{P_XP_Y}[\hat{r}^2_\theta(x,y)]$$

with the optimal $m^*(x,y) = \frac{p(x,y)}{p(x)p(y)}$. And when Θ is larger enough, the optimal $\hat{r}^*_\theta(x,y) = \frac{p(x,y)}{p(x)p(y)}$.

Proof.

$$\begin{aligned}
L_{\mathrm{LS_{D-RF}}} &= \inf_{m \in \mathcal{M}} \mathbb{E}_{P_X P_Y}\big[\big(r(x,y) - m(x,y)\big)^2\big] \\
&= \inf_{m \in \mathcal{M}} \mathbb{E}_{P_X P_Y}[r^2(x,y)] - 2\mathbb{E}_{P_X P_Y}[r(x,y)m(x,y)] + \mathbb{E}_{P_X P_Y}[m^2(x,y)] \\
&= \mathrm{Const.} + \inf_{m \in \mathcal{M}} \; -2\mathbb{E}_{P_X P_Y}[r(x,y)m(x,y)] + \mathbb{E}_{P_X P_Y}[m^2(x,y)] \\
&= \mathrm{Const.} + \inf_{m \in \mathcal{M}} \; -2\mathbb{E}_{P_{XY}}[m(x,y)] + \mathbb{E}_{P_X P_Y}[m^2(x,y)] \\
&\Leftrightarrow \sup_{m \in \mathcal{M}} \mathbb{E}_{P_{XY}}[m(x,y)] - \frac{1}{2}\mathbb{E}_{P_X P_Y}[m^2(x,y)].
\end{aligned}$$

Take the first-order functional derivative and set it to zero:

$$dP_{XY} - m(x,y) \cdot dP_X P_Y = 0.$$

We then get $m^*(x,y) = \frac{dP_{X,Y}}{dP_X P_Y} = \frac{p(x,y)}{p(x)p(y)}$. When Θ is large enough, by universal approximation theorem of neural networks [Hornik et al., 1989], the approximation in Proposition 8 is tight, which means $\hat{r}^*_\theta(x,y) = m^*(x,y) = \frac{p(x,y)}{p(x)p(y)}$. ■

5.7.2 More on Mutual Information Neural Estimation

In what follows, we present more analysis on estimating mutual information (MI) using neural networks. Before going into more details, we would like to 1) show I_{NWJ} and I_{DV} are MI lower bounds; and 2) present I_{CPC} [Oord et al., 2018] objective.

Lemma 5 (I_{NWJ} as a MI lower bound)**.**

$$\forall \theta \in \Theta, \quad I(X;Y) \geq \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - e^{-1}\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}].$$

Therefore,

$$I(X;Y) \geq I_{\mathrm{NWJ}} := \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - e^{-1}\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}].$$

Proof. In Proposition 3, we show the supreme value of $\mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - e^{-1}\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]$ happens when $\hat{f}^*_\theta(x,y) = 1 + \log \frac{p(x,y)}{p(x)p(y)}$. Plugging-in $\hat{f}^*_\theta(x,y)$, we get

$$\begin{aligned}
\mathbb{E}_{P_{X,Y}}[\hat{f}^*_\theta(x,y)] - e^{-1}\mathbb{E}_{P_X P_Y}[e^{\hat{f}^*_\theta(x,y)}] &= \mathbb{E}_{P_{X,Y}}[1 + \log \frac{p(x,y)}{p(x)p(y)}] - e^{-1}\mathbb{E}_{P_X P_Y}[e^1 \cdot \frac{p(x,y)}{p(x)p(y)}] \\
=& 1 + \mathbb{E}_{P_{X,Y}}[\log \frac{p(x,y)}{p(x)p(y)}] - e^{-1} \cdot e^1 \cdot \mathbb{E}_{P_X P_Y}[\frac{p(x,y)}{p(x)p(y)}] = 1 + I(X;Y) - 1 = I(X;Y).
\end{aligned}$$ ■

Lemma 6 (I_{DV} as a MI lower bound)**.**

$$\forall \theta \in \Theta, \quad I(X;Y) \geq \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \log\Big(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]\Big).$$

Therefore,

$$I(X;Y) \geq I_{\mathrm{DV}} := \sup_{\theta \in \Theta} \mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - -\log\Big(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]\Big).$$

Proof. In Proposition 4, we show the supreme value of $\mathbb{E}_{P_{X,Y}}[\hat{f}_\theta(x,y)] - \log\left(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta(x,y)}]\right)$ happens when $\hat{f}_\theta^*(x,y) = \text{Const.} + \log\frac{p(x,y)}{p(x)p(y)}$. Plugging-in $\hat{f}_\theta^*(x,y)$, we get

$$\begin{aligned}
&\mathbb{E}_{P_{X,Y}}[\hat{f}_\theta^*(x,y)] - \log\left(\mathbb{E}_{P_X P_Y}[e^{\hat{f}_\theta^*(x,y)}]\right)\\
=&\mathbb{E}_{P_{X,Y}}[\text{Const.} + \log\frac{p(x,y)}{p(x)p(y)}] - \log\left(\mathbb{E}_{P_X P_Y}[e^{\text{Const.}+\log\frac{p(x,y)}{p(x)p(y)}}]\right)\\
=&\text{Const.} + \mathbb{E}_{P_{X,Y}}[\log\frac{p(x,y)}{p(x)p(y)}] - \text{Const.}\cdot\mathbb{E}_{P_X P_Y}[\frac{p(x,y)}{p(x)p(y)}] = I(X;Y).
\end{aligned}$$

■

Proposition 9 (I_{CPC}, restating Contrastive Predictive Coding [Oord et al., 2018])**.** With $\hat{c}_\theta(x,y)$ representing a real-valued measureable function on $\mathcal{X}\times\mathcal{Y}$ which is parametrized by a neural network θ,

$$L_{\text{CPC}} := \sup_{\theta\in\Theta}\mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i,y_j)}}]$$

with an upper bound value $\log n$.

Proof.

$$\begin{aligned}
L_{\text{CPC}} &= \sup_{\theta\in\Theta}\mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i,y_j)}}]\\
&= \sup_{\theta\in\Theta}\mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(x_i,y_i)}}{\sum_{j=1}^n e^{\hat{c}_\theta(x_i,y_j)}}] + \log n\\
&\leq \sup_{\theta\in\Theta}\mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(x_i,y_i)}}{e^{\hat{c}_\theta(x_i,y_i)}}] + \log n\\
&= \sup_{\theta\in\Theta}\mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log 1] + \log n\\
&= \log n.
\end{aligned}$$

■

Lemma 7 (I_{CPC} as a MI lower bound)**.**

$$\forall\theta\in\Theta,\quad I(X;Y) \geq \mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i,y_j)}}].$$

Therefore,

$$I(X;Y) \geq I_{\text{CPC}} := \sup_{\theta\in\Theta}\mathbb{E}_{(x_1,y_1)\sim P_{X,Y},\cdots(x_n,y_n)\sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log\frac{e^{\hat{c}_\theta(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i,y_j)}}].$$

Proof. First, we use independent and identical random variables $X_1, X_2, \cdots, X_n$ and $Y_1, Y_2, \cdots, Y_n$ to represent the copies of X and Y, where $(x_i, y_i) \sim P_{X_i, Y_i}$. Replacing the random variables in Lemma 5, we obtain

$$\forall \theta \in \Theta, \quad I(X_i; Y_{1:n}) \geq \mathbb{E}_{P_{X_i, Y_{1:n}}}[\hat{f}_\theta(x_i, y_{1:k})] - e^{-1}\mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[e^{\hat{f}_\theta(x_i, y_{1:k})}].$$

Next, we define $\hat{f}_\theta(x_i, y_{1:k}) = 1 + \log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}$ and get

$$\forall \theta \in \Theta, \quad I(X_i; Y_{1:n}) \geq 1 + \mathbb{E}_{P_{X_i, Y_{1:n}}}[\log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] - \mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[\frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}].$$

Since $Y_1, Y_2, \cdots, Y_n$ are independent and identical samples, $\mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[\frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] = \mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[\frac{e^{\hat{c}_\theta(x_i, y_{i'})}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}]$ $\forall i' \in \{1, 2, \cdots, n\}$. Therefore, $\mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[\frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] = \frac{1}{n}\sum_{i'=1}^n \mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[\frac{e^{\hat{c}_\theta(x_i, y_{i'})}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] = \mathbb{E}_{P_{X_i} P_{Y_{1:n}}}[\frac{\frac{1}{n}\sum_{i'=1}^n e^{\hat{c}_\theta(x_i, y_{i'})}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] = 1$. Plugging-in this result, we have

$$\forall \theta \in \Theta, \quad I(X_i; Y_{1:n}) \geq 1 + \mathbb{E}_{P_{X_i, Y_{1:n}}}[\log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] - 1 = \mathbb{E}_{P_{X_i, Y_{1:n}}}[\log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}].$$

Note that $Y_{i'}$ is independent to X_i when $i' \neq i$, and therefore $I(X_i; Y_{1:n}) = I(X_i; Y_i) = I(X; Y)$.

Bringing everything together, the original objective can be reformulated as

$$\begin{aligned}
&\mathbb{E}_{(x_1, y_1) \sim P_{X,Y}, \cdots (x_n, y_n) \sim P_{X,Y}}[\frac{1}{n}\sum_{i=1}^n \log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] \\
=&\mathbb{E}_{P_{X_{1:n}, Y_{1:n}}}[\frac{1}{n}\sum_{i=1}^n \log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{P_{X_i, Y_{1:n}}}[\log \frac{e^{\hat{c}_\theta(x_i, y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{\hat{c}_\theta(x_i, y_j)}}] \\
\leq&\frac{1}{n}\sum_{i=1}^n I(X_i; Y_{1:n}) = \frac{1}{n}\sum_{i=1}^n I(X; Y) = I(X; Y).
\end{aligned}$$

■

5.7.2.1 Learning/ Inference in MI Neural Estimation and Baselines

The MI neural estimation methods can be dissected into two procedures: *learning* and *inference*. The learning step learns the parameters when estimating 1) point-wise dependency (PD)/ logarithm of point-wise dependency (PMI); or 2) MI lower bound. The inference step considers the parameters from the learning step and infers value for 1) MI itself; or 2) a lower bound of MI. We summarize different approaches in Table 5.1, and we discuss the baselines here. We present the comparisons between baselines and our methods in Table 5.1/ Figure 5.1.

CPC Oord et al. [2018] presented **C**ontrastive **P**redictive **C**oding (**CPC**) as an unsupervised learning objective, which adopts I_{CPC} (see Proposition 9) in both learning and inference stages. From Proposition 9 and Lemma 7, we conclude

$$I_{\mathrm{CPC}} \leq \min\Big(\log n, I(X; Y)\Big).$$

Hence, the difference between I_{CPC} and $I(X; Y)$ is large when n is small. This fact implies a large bias when using I_{CPC} to estimate MI. Nevertheless, empirical evidences [Poole et al., 2019, Song and Ermon, 2019] showed that I_{CPC} has low variance, which is also verified in our experiments.

NWJ Belghazi et al. [2018] presented to use neural networks to estimate **N**guyen-**W**ainwright-**J**ordan bound [Belghazi et al., 2018, Nguyen et al., 2010] (**NWJ**) bound of MI, which adopts I_{NWJ} (see Proposition 3) in both learning and inference stages. In Proposition 3 and Lemma 5, we show that when Θ is large enough, the supreme value of I_{NWJ} is $I(X;Y)$. Hence, we can expect a smaller bias when comparing I_{NWJ} to I_{CPC}. Song and Ermon [2019] acknowledged the variance of an empirical I_{NWJ} estimation is $\Omega(e^{I(X;Y)})$, suggesting a large variance when the true MI is large. We verify these facts in our experiments.

DV (MINE) Belghazi et al. [2018] presented to use neural networks to estimate **D**onsker-**V**aradhan bound [Belghazi et al., 2018, Nguyen et al., 2010] (**DV**) bound of MI, which adopts I_{DV} (see Proposition 4) in both learning and inference stages. The author also refers this MI estimation procedure as **M**utual **I**nformation **N**eural **E**stimation (**MINE**). In Proposition 4 and Lemma 6, we show that when Θ is large enough, the supreme value of I_{DV} is $I(X;Y)$. Hence, we can expect a smaller bias when comparing I_{DV} to I_{CPC}. Song and Ermon [2019] acknowledged the limiting variance of an empirical I_{DV} estimation is $\Omega(e^{I(X;Y)})$, which implies the variance is large when the true MI is large. We verify these facts in our experiments.

JS Unlike **CPC**, **NWJ**, and **DV**, Poole et al. [2019] presented to adopt different objectives in learning and inference stages for MI estimation. Precisely, the author uses Jensen-Shannon F-GAN [Nowozin et al., 2016] objective (see Proposition 5) to estimate PMI and then plugs in the PMI into I_{NWJ} (see Proposition 3) for the inference. The author refers this MI estimation method as **JS** since it considers **J**ensen-**S**hannon divergence during learning. Unfortunately, this estimation method still considers I_{NWJ} as its inference objective, and therefore the variance is still $\Omega(e^{I(X;Y)})$. Empirical results are shown in our experiments.

SMILE To overcome the large variance issue in **NWJ**, **DV**, and **JS**, Song and Ermon [2019] presented to use I_{JS} (see Proposition 5) for estimating PMI and then plug in the PMI to a modified I_{DV} (see Proposition 4). Specifically, the author clipped the value of $e^{\hat{f}_\theta(x,y)}$ in the second term of I_{DV} to control the variance during the inference stage. Although the modification introduces some bias for MI estimation, it is empirically admitting a small variance, which we also find in our experiments.

5.7.2.2 Architecture Design in Experiments

We follow the same training and evaluation protocal for Correlated Gaussians experiments in prior work [Poole et al., 2019, Song and Ermon, 2019]. We adopt the "concatenate critic" design [Oord et al., 2018, Poole et al., 2019, Song and Ermon, 2019] for our neural network parametrized function. The neural network parametrized functions are $\hat{c}_\theta$ in **CPC**, $\hat{f}_\theta$ in **NWJ/JS/DV/SMILE**/Variational MI Bounds/Density Matching I/Density Matchinig II, $\hat{r}_\theta$ in Density-Ratio Fitting, and $\hat{p}_\theta$ in Probabilistic Classifier. Take $\hat{c}_\theta$ as an example, the concatenate critic design admits $\hat{c}_\theta(x,y) = g_\theta([x,y])$ with g_θ being multiple-layer perceptrons. We consider g_θ to be 1-hidden-layer neural network with 512 neurons for each layer and ReLU function as the activation. The optimization considers batch size 128 and Adam optimizer [Kingma and Ba, 2015] with learning rate 0.001. For a fair comparison, we fix everything except for the learning and inference objectives. Note that Probabilistic Classifier method applies sigmoid function to the outputs to ensure probabilistic outputs. We set $\eta = 1.0$ in Density Matching II.

5.7.2.3 Theoretical Analysis

We restate the Assumptions in the main text:

Assumption 3 (Boundedness of the density ratio; restating Assumption 1)**.** There exist universal constants $C_l \leq C_u$ such that $\forall \hat{r}_\theta \in \mathcal{F}$ and $\forall x, y$, $C_l \leq \log \hat{r}_\theta(x, y) \leq C_u$.

Assumption 4 (log-smoothness of the density ratio; restating Assumption 2)**.** There exists $\rho > 0$ such that for $\forall x, y$ and $\forall \theta_1, \theta_2 \in \Theta$, $|\log \hat{r}_{\theta_1}(x, y) - \log \hat{r}_{\theta_2}(x, y)| \leq \rho \cdot \|\theta_1 - \theta_2\|$.

In what follows, we first prove the following lemma. The main idea is from Bartlett [1998], while here we focus on the covering number of the parameter space Θ using L_2 norm.

Lemma 8 (estimation; restating Lemma 3)**.** Let $\varepsilon > 0$ and $\mathcal{N}(\Theta, \varepsilon)$ be the covering number of Θ with radius ε under L_2 norm. Let $P_{X,Y}$ be any distribution where $S = \{x_i, y_i\}_{i=1}^n$ are sampled from and define $M := C_u - C_l$, then

$$\Pr_S\left(\sup_{\hat{r}_\theta \in \mathcal{F}} \left|\widehat{I}_\theta^{(n)}(X;Y) - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_\theta(x, y)]\right| \geq \varepsilon\right) \leq 2\mathcal{N}(\Theta, \varepsilon/4\rho) \exp\left(-\frac{n\varepsilon^2}{2M^2}\right). \tag{5.12}$$

Proof. Define $l_S(\theta) := \widehat{I}_\theta^{(n)}(X;Y) - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_\theta(x, y)]$. For $\theta_1, \theta_2 \in \Theta$, we first bound the difference $|l_S(\theta_1) - l_S(\theta_2)|$ in terms of the distance between θ_1 and θ_2. To do so, for any joint distribution P over $X \times Y$, we first bound the following difference:

$$\begin{aligned}
|\mathbb{E}_P[\log \hat{r}_{\theta_1}(x, y)] - \mathbb{E}_P[\log \hat{r}_{\theta_2}(x, y)]| &\leq \mathbb{E}_P[|\log \hat{r}_{\theta_1}(x, y) - \log \hat{r}_{\theta_2}(x, y)|] \\
&\leq \mathbb{E}_P[\rho \cdot \|\theta_1 - \theta_2\|_2] \\
&= \rho \cdot \|\theta_1 - \theta_2\|_2,
\end{aligned}$$

where the first inequality is due to the triangle inequality and the second one is from Assumption 4. Next we bound $|l_S(\theta_1) - l_S(\theta_2)|$ by applying the above inequality twice:

$$\begin{aligned}
|l_S(\theta_1) - l_S(\theta_2)| &= \left|\left(\widehat{I}_{\theta_1}^{(n)}(X;Y) - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_{\theta_1}(x, y)]\right) - \left(\widehat{I}_{\theta_2}^{(n)}(X;Y) - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_{\theta_2}(x, y)]\right)\right| \\
&\leq \left|\widehat{I}_{\theta_1}^{(n)}(X;Y) - \widehat{I}_{\theta_2}^{(n)}(X;Y)\right| + |\mathbb{E}_{P_{X,Y}}[\log \hat{r}_{\theta_1}(x, y)] - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_{\theta_2}(x, y)]| \\
&\leq \rho \cdot \|\theta_1 - \theta_2\| + \rho \cdot \|\theta_1 - \theta_2\|_2 \\
&= 2\rho \cdot \|\theta_1 - \theta_2\|.
\end{aligned}$$

Now we consider the covering of Θ. Since Θ is compact, it admits a finite covering. To simplify the notation, let $T := \mathcal{N}(\Theta, \varepsilon/4\rho)$ and let $\cup_{k=1}^T \Theta_k$ be a finite cover of Θ. Furthermore, assume $\theta_i \in \Theta_i$ be the center of the L_2 ball Θ_i with radius $\varepsilon/4\rho$. As a result, the following bound holds:

$$\begin{aligned}
\Pr_S(\sup_{\hat{r}_\theta \in \mathcal{F}} |l_S(\theta)| \geq \varepsilon) &= \Pr_S(\sup_{\theta \in \Theta} |l_S(\theta)| \geq \varepsilon) \\
&\leq \Pr_S(\cup_{k \in [T]} \sup_{\theta \in \Theta_k} |l_S(\theta)| \geq \varepsilon) \\
&\leq \sum_{k \in [T]} \Pr_S(\sup_{\theta \in \Theta_k} |l_S(\theta)| \geq \varepsilon).
\end{aligned}$$

The last inequality above is due to the union bound. Next, $\forall k \in [T]$, realize that the following inequality holds:

$$\Pr_S(\sup_{\theta \in \Theta_k} |l_S(\theta)| \geq \varepsilon) \leq \Pr_S(|l_S(\theta_k)| \geq \varepsilon/2).$$

To see this, note that the L_2 ball of Θ_k has radius $\varepsilon/4\rho$, hence $\sup_{\theta\in\Theta_k}|l_S(\theta)-l_S(\theta_k)|\leq 2\rho\cdot\varepsilon/4\rho=\varepsilon/2$, which yields:

$$\Pr_S(\sup_{\theta\in\Theta_k}|l_S(\theta)|\geq\varepsilon)\leq\Pr_S(\sup_{\theta\in\Theta_k}|l_S(\theta)-l_S(\theta_k)|+|l_S(\theta_k)|\geq\varepsilon)$$
$$\leq\Pr_S(|l_S(\theta_k)|\geq\varepsilon/2).$$

To proceed, it suffices if we could provide an upper bound for $\Pr_S(|l_S(\theta_k)|\geq\varepsilon/2)$. Now since $\log\hat{r}_{\theta_k}(x,y)$ is bounded for any pair of input x,y by Assumption 3, it follows from the Hoeffding's inequality that

$$\Pr_S(|l_S(\theta_k)|\geq\varepsilon/2)=\Pr_S\left(\left|\widehat{I}^{(n)}_{\theta_k}(X;Y)-\mathbb{E}_{P_{X,Y}}[\log\hat{r}_{\theta_k}(x,y)]\right|\geq\varepsilon/2\right)$$
$$\leq 2\exp\left(-\frac{n\varepsilon^2}{2M^2}\right).$$

Now, combine all the pieces together, we have:

$$\begin{aligned}\Pr_S(\sup_{\hat{r}_\theta\in\mathcal{F}}\left|\widehat{I}^{(n)}_\theta(X;Y)-\mathbb{E}_{P_{X,Y}}[\log\hat{r}_\theta(x,y)]\right|\geq\varepsilon)&=\Pr_S(\sup_{\theta\in\Theta}|l_S(\theta)|\geq\varepsilon)\\&\leq\sum_{k\in[T]}\Pr_S(\sup_{\theta\in\Theta_k}|l_S(\theta)|\geq\varepsilon)\\&\leq\mathcal{N}(\Theta,\varepsilon/4\rho)\Pr_S(\sup_{\theta\in\Theta_k}|l_S(\theta)|\geq\varepsilon)\\&\leq\mathcal{N}(\Theta,\varepsilon/4\rho)\Pr_S(|l_S(\theta_k)|\geq\varepsilon/2)\\&\leq 2\mathcal{N}(\Theta,\varepsilon/4\rho)\exp\left(-\frac{n\varepsilon^2}{2M^2}\right).\end{aligned}$$ ■

We restate the Lemma 4:

Lemma 9 (Hornik et al. [1989], approximation; restating Lemma 4)**.** Let $\varepsilon>0$. There exists $d\in\mathbb{N}$ and a family of neural networks $\mathcal{F}:=\{\hat{r}_\theta:\theta\in\Theta\subseteq\mathbb{R}^d\}$ where Θ is compact, such that $\inf_{\hat{r}_\theta\in\mathcal{F}}\left|\mathbb{E}_{P_{X,Y}}[\log\hat{r}_\theta(x,y)]-I(X;Y)\right|\leq\varepsilon$.

Now, we are ready the present our theorem:

Theorem 2. Let $0<\delta<1$. There exists $d\in\mathbb{N}$ and a family of neural networks $\mathcal{F}:=\{\hat{r}_\theta:\theta\in\Theta\subseteq\mathbb{R}^d\}$ where Θ is compact, so that $\exists\theta^*\in\Theta$, with probability at least $1-\delta$ over the draw of $S=\{x_i,y_i\}_{i=1}^n\sim P_{X,Y}^{\otimes n}$,

$$\left|\widehat{I}^{(n)}_{\theta^*}(X;Y)-I(X;Y)\right|\leq O\left(\sqrt{\frac{d+\log(1/\delta)}{n}}\right). \tag{5.13}$$

Proof. This theorem simply follows a combination of Lemma 8 and Lemma 9. First, by Lemma 9, for $\varepsilon>0$, there exists $d\in\mathbb{N}$ and a family of neural networks $\mathcal{F}:=\{\hat{r}_\theta:\theta\in\Theta\subseteq\mathbb{R}^d\}$ where Θ is compact, such that there $\exists\theta^*\in\Theta$,

$$\left|\mathbb{E}_{P_{X,Y}}[\log\hat{r}_{\theta^*}(x,y)]-I(X;Y)\right|\leq\frac{\varepsilon}{2}.$$

Next, we perform analysis on the estimation error $\left|\widehat{I}^{(n)}_{\theta^*}(X;Y)-\mathbb{E}_{P_{X,Y}}[\log\hat{r}_{\theta^*}(x,y)]\right|\leq\frac{\varepsilon}{2}$. Applying Lemma 8 with the fact [Anthony and Bartlett, 2009] that for $\Theta\subseteq\mathbb{R}^d$, $\log\mathcal{N}(\Theta,\varepsilon/4\rho)=O(d\log(\rho/\varepsilon))$, we can solve for ε in terms of the given δ. It suffices for us to find $\varepsilon \rightarrow \frac{\varepsilon}{2}$ such that:

$$2\mathcal{N}(\Theta, \varepsilon/8\rho) \exp\left(-\frac{n\varepsilon^2}{8M^2}\right) \leq \delta,$$

which is equivalent to finding ε such that the following inequality holds:

$$c \cdot d \log \frac{\varepsilon}{8\rho} + \frac{n\varepsilon^2}{8M^2} \geq \log \frac{2}{\delta},$$

where c is a universal constant that is independent of d. Now, using the inequality $\log(x) \leq x - 1$, it suffices for us to find ε such that

$$c \cdot d \left(\frac{\varepsilon}{8\rho} - 1\right) + \frac{n\varepsilon^2}{8M^2} \geq c \cdot d \log \frac{\varepsilon}{8\rho} + \frac{n\varepsilon^2}{8M^2} \geq \log \frac{2}{\delta},$$

which is in turn equivalent to solving:

$$\varepsilon^2 + c'\varepsilon \geq \left(\log \frac{2}{\delta} + cd\right) \cdot \frac{8M^2}{n},$$

where $c' = c'(c, d, \rho, n, M)$. Nevertheless, in order for the above inequality to hold, it suffices if we choose

$$\varepsilon = O\left(\sqrt{\frac{d + \log(1/\delta)}{n}}\right).$$

The final step is to combine the above two inequalities together:

$$\begin{aligned} \left|\widehat{I}^{(n)}_{\theta^*}(X;Y) - I(X;Y)\right| &\leq \left|\widehat{I}^{(n)}_{\theta^*}(X;Y) - \mathbb{E}_{P_{X,Y}}[\log \hat{r}_{\theta^*}(x,y)]\right| + \left|\mathbb{E}_{P_{X,Y}}[\log \hat{r}_{\theta^*}(x,y)] - I(X;Y)\right| \\ &\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = O\left(\sqrt{\frac{d + \log(1/\delta)}{n}}\right). \end{aligned}$$ ■

5.7.3 More on Self-supervised Representation Learning

We have shown how we adapt the proposed point-wise dependency estimation approaches (Probabilistic Classifier and Density-Ratio Fitting) to contrastive learning objectives (Probabilistic Classifier Coding and Density-Ratio Fitting Coding) for self-supervised representation learning. Following the adaptation, it is straightforward to define new contrastive learning objectives that are inspired by other presented approaches such as Variational MI Bounds, Density Matching I ,and Density Matching II. Nevertheless, instead of presenting new objectives, we would like to discuss 1) the connection between Probabilistic Classifier and Variational MI Bounds; 2) the connection between Density Matchinig I/II and I_{NWJ} (see Proposition 3); and 3) the potential limitations of the new objectives. Next, we will discuss the baseline method Contrastive Predictive Coding (CPC). Last, we present the experimental details.

5.7.3.1 Connection between Probabilistic Classifier and Variational MI Bounds

Proposition 7 states that the Probabilistic Classifier approach admits a classification task to differentiate the pairs sampled from a joint distribution or the product of marginal distribution. This classification task minimizes the binary cross entropy loss, which is highly optimized and stabilized in popular optimization packages such as PyTorch [Paszke et al., 2019] and TensorFlow [Abadi et al., 2016] (e.g., log-sum-exp trick for numerical stability). Note that, if we let $\hat{p}_\theta = \text{sigmoid}\left(l_\theta\right)$ with l_θ being the logits model, then reformulating Probabilistic Classifier to optimizing l_θ leads to the same objective as I_{JS} (see Proposition 5), which is the learning objective of *Variational MI Bounds* method. Although being the same objective as the Probabilistic Classifier approach, I_{JS} may encounter a relatively higher training instability (unless a particular take-care on its numerical instability). As pointed out by Tschannen et al. [2019], contrastive learning approaches with higher variance may result in a lower down-stream task performance, which accords with our empirical observation.

5.7.3.2 Connection between Density Matching I/II and I_{NWJ}

Density Matching I/II approaches are derived from the KL loss between the true joint density and estimated joint density ($L_{\text{KL}_{\text{DM}}}$ in Proposition 6). Specifically, Density Matching I is a Lagrange relaxation of $L_{\text{KL}_{\text{DM}}}$. If we change $\hat{f}_\theta + 1 = \hat{f}'_\theta$ in Density Matching I approach, then reformulating our objective to optimizing $\hat{f}'_\theta$ leads to the same objective as I_{NWJ} (see Proposition 3). Song and Ermon [2019] acknowledged the variance of an empirical I_{NWJ} estimation is $\Omega(e^{I(X;Y)})$, and hence the variance is large unless $I(X;Y)$ is small. Having the same conclusion in Chapter 5.7.3.1, our empirical observation finds Density Matching I/II lead to worsened representation as comparing to other contrastive learning objectives.

5.7.3.3 Contrastive Predictive Coding (CPC) for Contrastive Representation Learning

Contrastive Predictive Coding (CPC) [Oord et al., 2018] adapts I_{CPC} (see Proposition 9) to a contrastive representation learning objective:

$$\sup_{F,G} \sup_{\theta \in \Theta} \mathbb{E}_{(v_1^1, v_2^1) \sim P_{\mathcal{V}_1,\mathcal{V}_2}, \cdots (v_1^n, v_2^n) \sim P_{\mathcal{V}_1,\mathcal{V}_2}} [\frac{1}{n} \sum_{i=1}^{n} \log \frac{e^{\hat{c}_\theta(F(v_1^i), G(v_2^i))}}{\frac{1}{n} \sum_{j=1}^{n} e^{\hat{c}_\theta(F(v_1^i), G(v_2^j))}}],$$

where $\{v_1^i, v_2^i\}_{i=1}^n$ are independently and identically sampled from $P_{\mathcal{V}_1,\mathcal{V}_2}$. $\hat{c}_\theta(\cdot)$ is a function that takes the representations learned from the data pairs and returns a scalar.

5.7.3.4 Experiments Details

Datasets We adopt MNIST [LeCun et al., 1998] and CIFAR10 [Krizhevsky et al., 2009] as the datasets in our experiments. MNIST contains $60,000$ training and $10,000$ test examples. Each example is a grey-scale digit image ($0 \sim 9$) with size 28×28. CIFAR10 contains $50,000$ training and $10,000$ test examples. Each example is a 32×32 colour image from 10 mutual exclusive classes: {airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck}.

Pre-training and Fine-tuning Our self-supervised learning experiments contain two stages: *pre-training* and *fine-tuning*. In pre-training stage, we learn representation from the training samples using contrastive learning objectives (e.g., Probabilistic Classifier Coding (PCC), Density-Ratio Fitting Coding (D-RFC), and Contrastive Predictive Coding (CPC) [Oord et al., 2018]). View 1 ($\mathcal{V}_1$) and 2 ($\mathcal{V}_2$) are generated by augmenting the input with different transformations. For example, given an input, v_1 can be the 15-degree-rotated one and v_2 can be the horizontally flipped one. For **shallow** experiment, we consider the same data augmentations adopted by Tschannen et al. [2019]; for **deep** experiment, we consider the same data augmentations adopted by Bachman et al. [2019]. In fine-tuning stage, the network in the pre-training stage is fixed; we train only the classifier for minimizing classification loss from the representations. We follow linear evaluation protocol [Bachman et al., 2019, Hénaff et al., 2019, Hjelm et al., 2018, Kolesnikov et al., 2019, Oord et al., 2018, Tian et al., 2019, Tschannen et al., 2019] such that the classifier is a linear layer. After the pre-training and fine-tuning stages, we evaluate the performance of the model on the test samples.

Architectures To clearly understand how contrastive learning objectives affect the down-stream performance, we fix the network, learnnig rate, optimizer, and batch size across different objectives. To be more precise, we stick to the official implementations by Tschannen et al. [2019] (for **shallow** experiment) and Bachman et al. [2019] (for **deep** experiment). The only change is the contrastive learning objective, which is the loss in the pre-training stage for self-supervised learning experiments.

Reproducibility One can refer to `https://github.com/google-research/google-research/tree/master/mutual_information_representation_learning` and `https://github.com/Philip-Bachman/amdim-public` for the authors' official implementations, or checking the details in our released code.

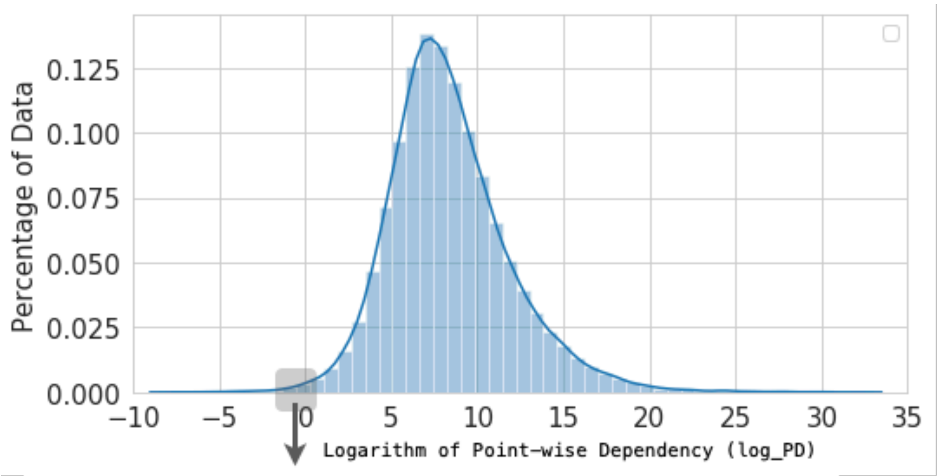
Consistent Trend on SimCLR [Chen et al., 2020a] We also evaluate CPC, PCC, and D-RFC in SimCLR [Chen et al., 2020a], which is a SOTA model and method on self-supervised representation learning. Note that the default contrastive learning objective considered in SimCLR [Chen et al., 2020a] is CPC, which obtains 91.04% test accuracy on CIFAR-10 (average for 5 runs). Details can be found in `https://github.com/google-research/simclr`. Similar to our **shallow** and **deep** experiments, we only change the contrastive learning objectives in SimCLR, and observing 91.51% and 88.69% average test accuracy for D-RFC and PCC, respectively. The trend is consistent with our **deep** experiment, where D-RFC works slightly better than CPC and PCC works slightly worse than CPC.

5.7.4 More on Cross-Modal Learning

Another Case Study: Cross-modal Adversarial Samples Debugging One important topic in interpretable machine learning [Molnar, 2019] is dataset debugging, which detects adversarial samples in a given dataset. For instance, in this dataset, an adversarial word feature would have low statistical dependency between its audio and textual representations. In Fig. 5.3, we report the PMI distribution and highlight the training words with PMI < 0 (i.e., the adversarial samples). We note that a negative PMI means the audio and textual features are either statistically independent or even co-occur less frequently than the independent assumption.

First, we find the distribution of PMI resembles a Gaussian distribution. The mean of the PMI values is MI, and our empirical estimation for it is 8.37. Our goal is to identify the training samples with PMI that deviates far from MI, and especially for the samples with negative PMI. There are 147 words have negative PMI values, approximately 0.45% of the training words. Next, we select some of these words and categorize them into two groups. The first group contains the words end in "ly" and another group contains the words end in "s". That is to say, the words end in "ly" and "s" are adversarial training sample in our analysis. To sum up, we demonstrate how our PD estimation approach can be used to detect adversarial training examples in a cross-modal dataset.

Dataset We construct a dataset that contains features from Word2Vec [Mikolov et al., 2013] and Speech2Vec [Chung and Glass, 2018]. Word2Vec is an unsupervised word embedding learning technique that takes a large text corpus of text as input and produces a fixed-length vector space. Specifically, each word in the corpus is assigned a real-valued and fixed-dimensional feature embedding. Similar to



Group1: seasonably, judicially, buckley, distantly, imaginatively, irresolutely, irresolutely, everlastingly, mistakenly, auspiciously

Group2: aristides, stallings, student's, instants, fosters, crates, mens, surrenders, successions, cottons, contrives, adieus, pensioners, benefactions, innkeepers, reverences

Figure 5.3: **Dataset Debugging** task with unsupervised word features across acoustic and textual modalities. *Probabilistic Classifier* approach is used to estimate PD between the audio and textual feature of a given word. The estimator is trained on the training split. We plot the logarithm of PD (i.e., PMI) distribution for the training words. We select the words with negative PMI values and categorize them into two groups: one contains the words end in "ly" and another containts the words end in "s".

Word2Vec, Speech2Vec takes a large corpus of human speech as input and produces a fixed-length vector space. Specifically, it transforms a variable-length speech segment (a word in the speech corpus) as a real-valued and fixed-dimensional feature embedding. There are $37,622$ words shared across Word2Vec and Speech2Vec, where we consider $32,622$ words of them (randomly selected) to be the training split and $5,000$ of them to be the test split. That is to say, each word contains a textual feature (from Word2Vec) and an audio feature (from Speech2Vec), with both feature being $100-$dimensional. The dataset can be downloaded from `https://github.com/iamyuanchung/speech2vec-pretrained-vectors` and we include the training/test split in our released code.

Training and Architectures We adopt the "separate critic" design [Oord et al., 2018, Poole et al., 2019, Song and Ermon, 2019] for our neural network parametrized function. Suppose $\hat{l}_\theta$ is the logits model in Probabilistic Classifier approach, and the separate critic design admits $\hat{l}_\theta(x,y) = g_{x_\theta}(x)^\top g_{y_\theta}(y)$ with g_{x_θ} and g_{y_θ} being different multiple layer perceptrons. We consider g_{x_θ} and g_{y_θ} to be 1-hidden-layer neural network with 512 neurons for intermediate layers, 128 neurons for the output layer, and ReLU function as the activation. The optimization considers batch size 512 and Adam optimizer [Kingma and Ba, 2015] with learning rate 0.001. A sigmoid function is applied to $\hat{l}_\theta$ ($\hat{p}_\theta = \text{sigmoid}(\hat{l}_\theta)$) to ensure $\hat{p}_\theta$ is a probabilistic output. We consider 100 training epochs.

Reproducibility Please refer to our released code, where we also include the dataset and its training/ test split.

5.7.5 Practical Deployment for Expectation(s)

In practice, the expectations in Propositions 3, 4, 5, 6, 7, 8, and 9 are estimated using empirical samples from $P_{X,Y}$ and $P_X P_Y$. With mild assumptions on the compactness of Θ and the boundness of our measurement, the estimation error would be small by uniform law of large numbers [Van der Vaart, 2000].

Chapter 6

Learning with Limited Supervision - Cross-view Learning with only Pairing Information

In this chapter, we study the sub-challenge of cross-view learning with only pairing information within the challenge of learning with limited supervision. We instantiate the discussion using the self-supervised representation learning (SSL) [Devlin et al., 2018, Oord et al., 2018, Tian et al., 2019, Zhang et al., 2016], where many proposed approaches for self-supervised learning follow naturally a multi-view perspective, with the input (e.g., original images) and the self-supervised signals (e.g., augmented images) being seen as two redundant views of the data. Then, SSL learns representations using a proxy objective (i.e., SSL objective) between inputs and self-defined signals. Empirical evidence suggests that the learned representations can generalize well to a wide range of downstream tasks, even when the SSL objective has not utilize any downstream supervision during training. For example, SimCLR [Chen et al., 2020a] defines a contrastive loss (i.e., an SSL objective) between images with different augmentations (i.e., one as the input and the other as the self-supervised signal). Then, one can take SimCLR as features extractor and adopt the features to various computer vision applications, spanning image classification, object detection, instance segmentation, and pose estimation [He et al., 2019]. Despite success in practice, only a few work [Arora et al., 2019, Lee et al., 2020, Tosh et al., 2020] provide theoretical insights into the learning efficacy of SSL. Our work shares a similar goal to explain the success of SSL, from the perspectives of Information Theory [Cover and Thomas, 2012] and multi-view representation.

To understand (a subset[1] of) SSL, we start by the following *multi-view assumption*. First, we regard the input and the self-supervised signals as two corresponding views of the data. Using our running example, in SimCLR [Chen et al., 2020a], the augmented images (i.e., the input and the self-supervised signal) are an image with different views. Second, we adopt a common assumption in multi-view learning: either view alone is (approximately) sufficient for the downstream tasks (see Assumption 1 in prior work [Sridharan and Kakade, 2008]). The assumption suggests that the image augmentations (e.g., changing the style of an image) should not affect the labels of images, or analogously, the self-supervised signal contains most (if not all) of the information that the input has about the downstream tasks. With this assumption, our first contribution is to formally show that the self-supervised learned representations can 1) extract all the task-relevant information (from the input) with a potential loss; and 2) discard all the task-irrelevant information (from the input) with a fixed gap. Then, using classification task as an example, we are able

[1] We discuss the limitations of the multi-view assumption in Chapter 6.1.1.

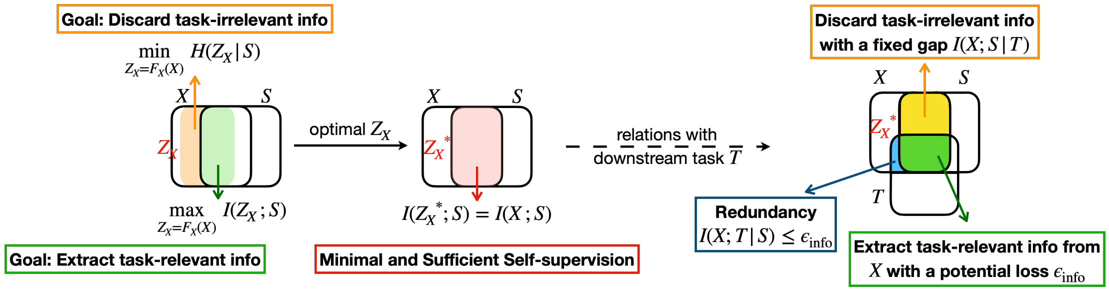

Goal: Discard task-irrelevant info
$\min_{Z_X=F_X(X)} H(Z_X|S)$
X
S
Z_X
$\max_{Z_X=F_X(X)} I(Z_X;S)$
Goal: Extract task-relevant info
optimal Z_X
X
S
$Z_X{}^*$
$I(Z_X{}^*;S) = I(X;S)$
Minimal and Sufficient Self-supervision
relations with downstream task T
Discard task-irrelevant info with a fixed gap $I(X;S|T)$
X
S
$Z_X{}^*$
T
Redundancy $I(X;T|S) \leq \epsilon_{\text{info}}$
Extract task-relevant info from X with a potential loss ϵ_{info}

Figure 6.1: High-level takeaways for our main results using information diagrams. (a) We present to learn minimal and sufficient self-supervision: minimize $H(Z_X|S)$ for discarding task-irrelevant information and maximize $I(Z_X;S)$ for extracting task-relevant information. (b) The resulting learned representation $Z_X{}^*$ contains all task relevant information from the input with a potential loss ϵ_{info} and discards task-irrelevant information with a fixed gap $I(X;S|T)$. (c) Our core assumption: the self-supervised signal is approximately redundant to the input for the task-relevant information.

the quantify the smallest generalization error (Bayes error rate) given the discussed task-relevant and task-irrelevant information.

As the second contribution, our analysis 1) connects prior arts for SSL on contrastive [Bachman et al., 2019, Chen et al., 2020a, Oord et al., 2018, Tian et al., 2019] and predictive learning [Devlin et al., 2018, Tulyakov et al., 2018, Vondrick et al., 2016, Zhang et al., 2016] approaches; and 2) paves the way to a larger space of composing SSL objectives to extract task-relevant and discard task-irrelevant information simultaneously. For instance, the combination between the contrastive and predictive learning approaches achieves better performance than contrastive- or predictive-alone objective and enjoys less over-fitting problem. We also present a new objective to discard task-irrelevant information. The objective can be easily incorporated with prior self-supervised learning objectives.

We conduct controlled experiments on visual (the first set) and visual-textual (the second set) self-supervised representation learning. The first set of experiments are performed when the multi-view assumption is likely to hold. The goal is to compare different compositions of SSL objectives on extracting task-relevant and discarding task-irrelevant information. The second set of experiments are performed when the input and the self-supervised signal lie in very different modalities. Under this cross-modality setting, the task-relevant information may not mostly lie in the shared information between the input and the self-supervised signal. The goal is to examine SSL objectives' generalization, where the multi-view assumption is likely to fail.

6.1 A Multi-view Information-Theoretical Framework

Notations. For the input, we denote its random variable as X, sample space as $\mathcal{X}$, and outcome as x. We learn a representation (Z_X/ $\mathcal{Z}$/ z_x) from the input through a deterministic mapping F_X: $Z_X = F_X(X)$. For the self-supervised signal, we denote its random variable/ sample space/ outcome as S/ $\mathcal{S}$/ s. Two sample spaces can be different between the input and the self-supervised signal: $\mathcal{X} \neq \mathcal{S}$. The information required for downstream tasks is referred to as "task-relevant information": T/ $\mathcal{T}$/ t. Note that SSL has no access to the task-relevant information. Lastly, we use $I(A;B)$ to represent mutual information, $I(A;B|C)$ to represent conditional mutual information, $H(A)$ to represent the entropy, and $H(A|B)$ to represent conditional entropy for random variables A/B/C. We provide high-level takeaways for our main results in Figure 6.1. We defer all proofs to Chapter 6.5.2, 6.5.3 and 6.5.4.

6.1.1 Multi-view Assumption

We regard the input (X) and the self-supervised signals (S) as two views of the data. Here, we provide a table showing different X/S in various SSL frameworks:

Framework	BERT [Devlin et al., 2018]	Look & Listen [Arandjelovic and Zisserman, 2017]	SimCLR [Chen et al., 2020a]	Colorization [Zhang et al., 2016]
Inputs (X)	Non-masked Words	Image	Image	Image Lightness
Self-supervised Signals (S)	Masked Words	Audio Stream	Same Image with Augmentation	Image Color

We note that not all SSL frameworks realize the inputs and the self-supervised signals as corresponding views. For instance, Jigsaw puzzle [Noroozi and Favaro, 2016] considers (shuffled) image patches as the input and the positions of the patches as the self-supervised signals. Another example is Learning by Predicting Rotations [Gidaris et al., 2018], which considers an image (rotating with a specific angle) as the input and the rotation angle of the image as the self-supervised signal. We point out that the frameworks that regard X/S as two corresponding views [Chen et al., 2020a, He et al., 2019] have a much better empirical downstream performance than the frameworks that do not [Gidaris et al., 2018, Noroozi and Favaro, 2016]. We hence focuse on the multi-view setting between X/S.

Next, we adopt the common assumption (i.e., *multi-view assumption* [Sridharan and Kakade, 2008, Xu et al., 2013]) in the multi-view learning between the input and the self-supervised signal:

Assumption 5 (Multi-view, restating Assumption 1 in prior work [Sridharan and Kakade, 2008])**.** The self-supervised signal is approximately redundant to the input for the task-relevant information. In other words, there exist an $\epsilon_{\text{info}} > 0$ such that $I(X;T|S) \leq \epsilon_{\text{info}}$.

Assumption 5 states that, when ϵ_{info} is small, the task-relevant information lies mostly in the shared information between the input and the self-supervised signals. We argue this assumption is mild with the following example. For self-supervised visual contrastive learning [Chen et al., 2020a, Hjelm et al., 2018], the input and the self-supervised signal are the same image with different augmentations. Using image augmentations can be seen as changing the style of an image while not affecting the content. And we argue that the information required for downstream tasks should only be retained in the content but not the style. Next, we point out the failure cases of the assumption (or have large ϵ_{info}): the input and the self-supervised signal contain very different task-relevant information. For instance, a drastic image augmentation (e.g., adding large noise) may change the content of the image (e.g., the noise completely occludes the objects). Another example is BERT [Devlin et al., 2018], with too much masking, downstream information may exist differently in the masked (i.e., the self-supervised signals) and the non-masked (i.e., the input) words. Analogously, too much masking makes the non-masked words have insufficient context to predict the masked words.

6.1.2 Learning Minimal and Sufficient Representations for Self-supervision

We start by discussing the supervised representation learning. The Information Bottleneck (IB) method [Achille and Soatto, 2018, Tishby et al., 2000] generalizes minimal sufficient statistics to the representations that are minimal (i.e., less complexity) and sufficient (i.e., better fidelity). To learn such representations for downstream supervision, we consider the following objectives:

Definition 2 (Minimal and Sufficient Representations for Downstream Supervision)**.** Let Z_X^{sup} be the sufficient supervised representation and $Z_X^{\text{sup}_{\text{min}}}$ be the minimal and sufficient representation:

$$Z_X^{\text{sup}} = \underset{Z_X}{\arg\max}\, I(Z_X;T) \text{ and } Z_X^{\text{sup}_{\text{min}}} = \underset{Z_X}{\arg\min}\, H(Z_X|T) \text{ s.t. } I(Z_X;T) \text{ is maximized.}$$

To reduce the complexity of the representation Z_X, the prior methods [Achille and Soatto, 2018, Tishby et al., 2000] presented to minimize $I(Z_X;X)$ while ours presents to minimize $H(Z_X|T)$. We provide a justification: minimizing $H(Z_X|T)$ reduces the randomness from T to Z_X, and the randomness is regarded as a form of incompressibility [Calude, 2013]. Hence, minimizing $H(Z_X|T)$ leads to a more compressed representation (discarding redundant information)[2]. Note that we do not constrain the downstream task T as classification, regression, or clustering.

Then, we present SSL objectives to learn sufficient (and minimal) representations for self-supervision:

Definition 3 (Minimal and Sufficient Representations for Self-supervision)**.** Let Z_X^{ssl} be the sufficient self-supervised representation and $Z_X^{\text{ssl}_{\text{min}}}$ be the minimal and sufficient representation:

$$Z_X^{\text{ssl}} = \underset{Z_X}{\arg\max}\, I(Z_X;S) \text{ and } Z_X^{\text{ssl}_{\text{min}}} = \underset{Z_X}{\arg\min}\, H(Z_X|S) \text{ s.t. } I(Z_X;S) \text{ is maximized.}$$

Definition 3 defines our self-supervised representation learning strategy. Now, we are ready to associate the supervised and self-supervised learned representations:

Theorem 3 (Task-relevant information with a potential loss ϵ_{info})**.** The supervised learned representations (i.e., Z_X^{sup} and $Z_X^{\text{sup}_{\text{min}}}$) contain all the task-relevant information in the input (i.e., $I(X;T)$). The self-supervised learned representations (i.e., Z_X^{ssl} and $Z_X^{\text{ssl}_{\text{min}}}$) contain all the task-relevant information in the input with a potential loss ϵ_{info}. Formally,

$$I(X;T) = I(Z_X^{\text{sup}};T) = I(Z_X^{\text{sup}_{\text{min}}};T) \geq I(Z_X^{\text{ssl}};T) \geq I(Z_X^{\text{ssl}_{\text{min}}};T) \geq I(X;T) - \epsilon_{\text{info}}.$$

When ϵ_{info} is small, Theorem 3 indicates that the self-supervised learned representations can extract almost as much task-relevant information as the supervised one. While when ϵ_{info} is non-trivial, the learned representations may not always lead to good downstream performance. This result has also been observed in prior work [Tschannen et al., 2019] and InfoMin [Tian et al., 2020], which claim the representations with maximal mutual information may not have the best performance.

Theorem 4 (Task-irrelevant information with a fixed compression gap $I(X;S|T)$)**.** The sufficient self-supervised representation (i.e., $I(Z_X^{\text{ssl}};T)$) contains more task-irrelevant information in the input than the sufficient and minimal self-supervised representation (i.e., $I(Z_X^{\text{ssl}_{\text{min}}};T)$). The latter contains an amount of the information, $I(X;S|T)$, that cannot be discarded from the input. Formally,

$$I(Z_X^{\text{ssl}};X|T) = I(X;S|T) + I(Z_X^{\text{ssl}};X|S,T) \geq I(Z_X^{\text{ssl}_{\text{min}}};X|T) = I(X;S|T) \geq I(Z_X^{\text{sup}_{\text{min}}};X|T) = 0.$$

Theorem 4 indicates that a compression gap (i.e., $I(X;S|T)$) exists when we discard the task-irrelevant information from the input. To be specific, $I(X;S|T)$ is the amount of the shared information between the input and the self-supervised signal excluding the task-relevant information. Hence, $I(X;S|T)$ would be large if the downstream tasks requires only a portion of the shared information.

6.1.3 Connections with Contrastive and Predictive Learning Objectives

Theorem 3 and 4 state that our self-supervised learning strategies (i.e., min $H(Z_X|S)$ and max $I(Z_X;S)$ defined in Definition 3) can extract task-relevant and discard task-irrelevant information. A question emerges:

[2] We do not claim $H(Z_X|T)$ minimization is better than $I(Z_X;X)$ minimization for reducing the complexity in the representations Z_X. In Chapter 6.5.1, we will show that $H(Z_X|T)$ minimization and $I(Z_X;X)$ minimization are interchangeable under our framework's setting.

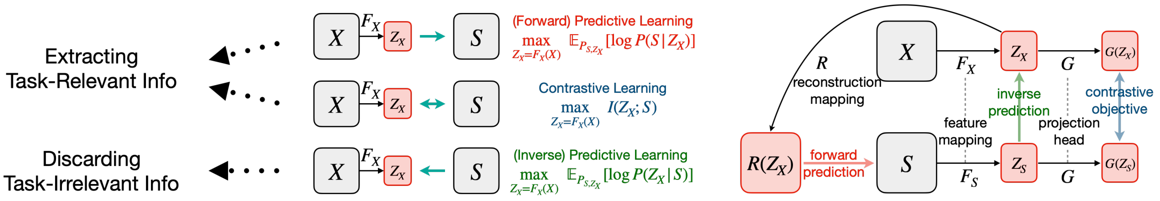


Figure 6.2: Remarks on contrastive and predictive learning objectives for self-supervised learning. Between the representation Z_X and the self-supervised signal S, *contrastive objective* performs mutual information maximization and *predictive objectives* perform log conditional likelihood maximization. We show that the SSL objectives aim at extracting task-relevant and discarding task-irrelevant information. Last, we summarize the computational blocks for practical deployments for these objectives.

"What are the practical aspects of the presented self-supervised learning strategies?"

To answer this question, we present 1) the connections with prior SSL objectives, especially for contrastive [Bachman et al., 2019, Chen et al., 2020a, He et al., 2019, Hjelm et al., 2018, Oord et al., 2018, Tian et al., 2019] and predictive [Devlin et al., 2018, Pathak et al., 2016, Peters et al., 2018, Tulyakov et al., 2018, Vondrick et al., 2016, Zhang et al., 2016] learning objectives, showing that these objectives are extracting task-relevant information; and 2) a new *inverse predictive learning* objective to discard task-irrelevant information. We illustrate important remarks in Figure 6.2.

Contrastive Learning (is extracting task-relevant information). Contrastive learning objective [Oord et al., 2018] maximizes the dependency/contrastiveness between the learned representation Z_X and the self-supervised signal S, which suggests maximizing the the mutual information $I(Z_X;S)$. Theorem 3 suggests that maximizing $I(Z_X;S)$ results in Z_X containing (approximately) all the information required for the downstream tasks from the input X. To deploy the contrastive learning objective, we suggest contrastive predictive coding (CPC) [Oord et al., 2018][3], which is a mutual information lower bound with low variance [Poole et al., 2019, Song and Ermon, 2019]:

$$\mathrm{L}_{CL} := \max_{\substack{Z_S=F_S(S),\\ Z_X=F_X(X),G}} \mathbb{E}_{(z_{s1},z_{x1}),\cdots,(z_{sn},z_{xn})\sim P^n(Z_S,Z_X)}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{\langle G(z_{xi}),G(z_{si})\rangle}}{\frac{1}{n}\sum_{j=1}^{n}e^{\langle G(z_{xi}),G(z_{sj})\rangle}}\right], \tag{6.1}$$

where $F_S:\mathcal{S}\to\mathcal{Z}$ is a deterministic mapping and G is a project head that projects a representation in $\mathcal{Z}$ into a lower-dimensional vector. If the input and self-supervised signals share the same sample space, i.e., $\mathcal{X}=\mathcal{S}$, we can impose $F_X=F_S$ (e.g., self-supervised visual representation learning [Chen et al., 2020a]). The projection head, G, can be an identity, a linear, or a non-linear mapping. Last, we note that modeling eq. (6.1) often requires a large batch size (e.g., large n in eq. (6.1)) to ensure a good downstream performance [Chen et al., 2020a, He et al., 2019].

Forward Predictive Learning (is extracting task-relevant information). Forward predictive learning encourages the learned representation Z_X to reconstruct the self-supervised signal S, which suggests maximizing the log conditional likelihood $\mathbb{E}_{P_{S,Z_X}}[\log P(S|Z_X)]$. By the chain rule, $I(Z_X;S)=H(S)-H(S|Z_X)$, where $H(S)$ is irrelevant to Z_X. Hence, maximizing $I(Z_X;S)$ is equivalent to maximizing

[3] Other contrastive learning objectives can be other mutual information lower bounds such as DV-bound or NWJ-bound [Belghazi et al., 2018] or its JS-divergence [Hjelm et al., 2018, Poole et al., 2019] variants. Among different objectives, Tschannen et al. [2019] have suggested that the objectives with large variance (e.g., DV-/NWJ-bound [Belghazi et al., 2018]) may lead to worsen performance compared to the low variance counterparts (e.g., CPC [Oord et al., 2018] and JS-div. [Poole et al., 2019]).

$-H(S|Z_X) = \mathbb{E}_{P_{S,Z_X}}[\log P(S|Z_X)]$, which is the predictive learning objective. Together with Theorem 3, if z_x can perfectly reconstruct s for any $(s, z_x) \sim P_{S,Z_X}$, then Z_X contains (approximately) all the information required for the downstream tasks from the input X. A common approach to avoid intractability in computing $\mathbb{E}_{P_{S,Z_X}}[\log P(S|Z_X)]$ is assuming a variational distribution $Q_\phi(S|Z_X)$ with ϕ representing the parameters in $Q_\phi(\cdot|\cdot)$. Specifically, we present to maximize $\mathbb{E}_{P_{S,Z_X}}[\log Q_\phi(S|Z_X)]$, which is a lower bound of $\mathbb{E}_{P_{S,Z_X}}[\log P(S|Z_X)]$[4]. $Q_\phi(\cdot|\cdot)$ can be any distribution such as Gaussian or Laplacian and ϕ can be a linear model, a kernel method, or a neural network. Note that the choice of the reconstruction type of loss depends on the distribution type of $Q_\phi(\cdot|\cdot)$, and is not fixed. For instance, if we let $Q_\phi(S|Z_X)$ be Gaussian $\mathcal{N}\Big(S|R(Z_X), \sigma\mathbf{I}\Big)$ with $\sigma\mathbf{I}$ as a diagonal matrix[5], the objective becomes:

$$\mathrm{L}_{FP} := \max_{Z_X = F_X(X), R} \mathbb{E}_{s, z_x \sim P_{S,Z_X}} \Big[- \|s - R(z_x)\|_2^2 \Big], \tag{6.2}$$

where $R : \mathcal{Z} \to \mathcal{S}$ is a deterministic mapping to reconstruct S from Z and we ignore the constants derived from the Gaussian distribution. Last, in most real-world applications, the self-supervised signal S has a much higher dimension (e.g., a $224 \times 224 \times 3$ image) than the representation Z_X (e.g., a 64-dim. vector). Hence, modeling a conditional generative model $Q_\phi(S|Z_X)$ will be challenging.

Inverse Predictive Learning (is discarding task-irrelevant information). Inverse predictive learning encourages the self-supervised signal S to reconstruct the learned representation Z_X, which suggests maximizing the log conditional likelihood $\mathbb{E}_{P_{S,Z_X}}[\log P(Z_X|S)]$. Given Theorem 4 together with $-H(Z_X|S) = \mathbb{E}_{P_{S,Z_X}}[\log P(Z_X|S)]$, we know if s can perfectly reconstruct z_x for any $(s, z_x) \sim P_{S,Z_X}$ under the constraint that $I(Z_X; S)$ is maximized, then Z_X discards the task-irrelevant information, excluding $I(X; S|T)$. Similar to the forward predictive learning, we use $\mathbb{E}_{P_{S,Z_X}}[\log Q_\phi(Z_X|S)]$ as a lower bound of $\mathbb{E}_{P_{S,Z_X}}[\log P(Z_X|S)]$. In our deployment, we take the advantage of the design in eq. (6.1) and let $Q_\phi(Z_X|S)$ be Gaussian $\mathcal{N}\Big(Z_X|F_S(S), \sigma\mathbf{I}\Big)$:

$$\mathrm{L}_{IP} := \max_{Z_S = F_S(S), Z_X = F_X(X)} \mathbb{E}_{z_s, z_x \sim P_{Z_S,Z_X}} \Big[- \|z_x - z_s\|_2^2 \Big]. \tag{6.3}$$

Note that optimizing eq. (6.3) alone results in a degenerated solution, e.g., learning Z_X and Z_S to be the same constant.

Composing SSL Objectives (to extract task-relevant and discard task-irrelevant information simultaneously). So far, we discussed how prior self-supervised learning approaches extract task-relevant information via the contrastive or the forward predictive learning objectives. Our analysis also inspires a new loss, the inverse predictive learning objective, to discard task-irrelevant information. Now, We present a composite loss to combine them together:

$$\mathrm{L}_{SSL} = \lambda_{CL}\mathrm{L}_{CL} + \lambda_{FP}\mathrm{L}_{FP} + \lambda_{IP}\mathrm{L}_{IP}, \tag{6.4}$$

[4] $\mathbb{E}_{P_{S,Z_X}}[\log P(S|Z_X)] = \max_{Q_\phi} \mathbb{E}_{P_{S,Z_X}}[\log Q_\phi(S|Z_X)] + D_{\mathrm{KL}}\Big(P(S|Z_X) \,\|\, Q_\phi(S|Z_X)\Big) \geq \max_{Q_\phi} \mathbb{E}_{P_{S,Z_X}}[\log Q_\phi(S|Z_X)]$.

[5] The assumption of identity covariance in the Gaussian is only a particular parameterization of the distribution $Q(\cdot|\cdot)$. Other examples are MocoGAN [Tulyakov et al., 2018], which assumes Q is Laplacian (i.e., ℓ_1 reconstruction loss) and ϕ is a deconvolutional network [Long et al., 2015]. Transformer-XL [Dai et al., 2019] assumes Q is a categorical distribution (i.e., cross entropy loss) and ϕ is a Transformer network [Vaswani et al., 2017]. Although Gaussian with diagonal covariance is not the best assumption, it is perhaps the simplest one.

where λ_{CL}, λ_{FP}, and λ_{IP} are hyper-parameters. This composite loss enables us to extract task-relevant and discard task-irrelevant information simultaneously.

6.1.4 Theoretical Analysis - Bayes Error Rate for Downstream Classification

So far, we see the practical aspects of our designed SSL strategies. Now, we provide an theoretical analysis on the representations' generalization error when T is a categorical variable. We use Bayes error rate as an example, which stands for the irreducible error (smallest generalization error [Feder and Merhav, 1994]) when learning an arbitrary classifier from the representation to infer the labels. In specific, let P_e be the Bayes error rate of arbitrary learned representations Z_X and $\hat{T}$ as the estimation for T from our classifier,

$$P_e := \mathbb{E}_{z_x \sim P_{Z_X}}[1 - \max_{t \in T} P(\hat{T} = t | z_x)].$$

To begin with, we present a general form of sample complexity with mutual information ($I(Z_X; S)$) estimation using empirical samples from distribution $P_{Z_X,S}$. Let $P^{(n)}_{Z_X,S}$ denote the (uniformly sampled) empirical distribution of $P_{Z_X,S}$ and $\hat{I}^{(n)}_\theta(Z_X; S) := \mathbb{E}_{P^{(n)}_{Z_X,S}}[\hat{f}_\theta(z_x, s)]$ with $\hat{f}_\theta$ being the estimated log density ratio (i.e., $\log p(s|z_x)/p(s)$).

Proposition 10 (Mutual Information Neural Estimation, restating Theorem 1 by Tsai et al. [2020d]). Let $0 < \delta < 1$. There exists $d \in \mathbb{N}$ and a family of neural networks $\mathcal{F} := \{\hat{f}_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ where Θ is compact, so that $\exists \theta^* \in \Theta$, with probability at least $1 - \delta$ over the draw of $\{z_{xi}, s_i\}_{i=1}^n \sim P^{\otimes n}_{Z_X,S}$, $\left| \widehat{I}^{(n)}_{\theta^*}(Z_X; S) - I(Z_X; S) \right| \leq O\left(\sqrt{\frac{d + \log(1/\delta)}{n}}\right)$.

This proposition shows that there exists a neural network θ^*, with high probability, $\widehat{I}^{(n)}_{\theta^*}(Z_X; S)$ can approximate $I(Z_X; S)$ with n samples at rate $O(1/\sqrt{n})$. Under this network θ^* and the same parameters d and δ, we are ready to present our main results on the Bayes error rate. Formally, let $|T|$ be T's cardinalitiy and $\text{Th}(x) = \min\{\max\{x, 0\}, 1 - 1/|T|\}$ as a thresholding function:

Theorem 5 (Bayes Error Rates for Arbitrary Learned Representations). For an arbitrary learned representations Z_X, $P_e = \text{Th}(\bar{P}_e)$ with

$$\bar{P}_e \leq 1 - \exp\left(- \Big(H(T) + I(X; S|T) + I(Z; X|S, T) - \hat{I}^{(n)}_{\theta^*}(Z_X; S) + O\big(\sqrt{\frac{d + \log(1/\delta)}{n}}\big) \Big) \right).$$

Given arbitrary learned representations (Z_X), Theorem 5 suggests the corresponding Bayes error rate (P_e) is small when: 1) the estimated mutual information $\left(\widehat{I}^{(n)}_{\theta^*}(Z_X; S)\right)$ is large; 2) a larger number of samples n are used for estimating the mutual information; and 3) the task-irrelevant information (the compression gap $I(X; S|T)$ and the superfluous information $I(Z; X|S, T)$, defined in Theorem 4) is small. The first and the second results supports the claim that maximizing $I(Z_X; S)$ may learn the representations that are beneficial to downstream tasks. The third result implies the learned representations may perform better on the downstream task when the compression gap is small. Additionally, $Z^{\text{ssl}_{\text{min}}}$ is preferable than Z^{ssl} since $I(Z^{\text{ssl}_{\text{min}}}; X|S, T) = 0$ and $I(Z^{\text{ssl}}; X|S, T) \geq 0$.

Theorem 6 (Bayes Error Rates for Self-supervised Learned Representations). Let $P_e^{\text{sup}}/P_e^{\text{ssl}}/P_e^{\text{ssl}_{\text{min}}}$ be the Bayes error rate of the supervised or the self-supervised learned representations $Z_X^{\text{sup}}/Z_X^{\text{ssl}}/Z_X^{\text{ssl}_{\text{min}}}$. Then, $P_e^{\text{ssl}} = \text{Th}(\bar{P}_e^{\text{ssl}})$ and $P_e^{\text{ssl}_{\text{min}}} = \text{Th}(\bar{P}_e^{\text{ssl}_{\text{min}}})$ with

$$-\frac{\log(1 - P_e^{\text{sup}}) + \log 2}{\log(|T|)} \leq \{\bar{P}_e^{\text{ssl}}, \bar{P}_e^{\text{ssl}_{\text{min}}}\} \leq 1 - \exp\Big(- (\log 2 + P_e^{\text{sup}} \cdot \log|T| + \epsilon_{\text{info}})\Big).$$

Given our self-supervised learned representations (Z_X^{ssl} and $Z_X^{\text{ssl}_{\text{min}}}$), Theorem 6 suggests a smaller upper bound of P_e^{ssl} (or $P_e^{\text{ssl}_{\text{min}}}$) when the redundancy between the input and the self-supervised signal (ϵ_{info}, defined in Assumption 5) is small. This result implies the self-supervised learned representations may perform better on the downstream task when the multi-view redundancy is small.

6.2 Controlled Experiments

We aim at providing empirical supports for Theorems 3 and 4 and comparing different SSL objectives. In particular, we present information inequalities in Theorems 3 and 4 regarding the amount of the task-relevant and the task-irrelevant information that will be extracted and discarded when learning self-supervised representations. Nonetheless, quantifying the information is notoriously hard and often leads to inaccurate quantifications in practice [McAllester and Stratos, 2020, Song and Ermon, 2019]. Not to mention the information we aim to quantify is the conditional information, which is believed to be even more challenging than quantifying the unconditional one [Póczos and Schneider, 2012]. To address this concern, we instead study the generalization error of the self-supervised learned representations, theoretically (Bayes error rate discussed in Chapter 6.1.4) and empirically (test performance discussed here).

Another important aspect of the experimental design is examining eq. (6.4), which can be viewed as a Lagrangian relaxation to learn representations that contain minimal and sufficient self-supervision (see Definition 3): a weighted combination between $I(Z_X;S)$ and $-H(Z_X|S)$. In particular, the contrastive loss L_{CL} and the forward-predictive loss L_{FP} represent different realizations of modeling $I(Z_X;S)$ and the inverse-predictive loss L_{FP} represents a realization of modeling $-H(Z_X|S)$.

We design two sets of experiments: The first one is when the input and self-supervised signals lie in the same modality (visual) and are likely to satisfy the multi-view redundancy assumption (Assumption 5). The second one is when the input and self-supervised signals lie in very different modalities (visual and textual), thus challenging the SSL objective's generalization ability.

Experiment I - Visual Representation Learning. We use Omniglot dataset [Lake et al., 2015] [6] in this experiment. The training set contains images from 964 characters, and the test set contains 659 characters. There are no characters overlap between the training and test set. Each character contains twenty examples drawn from twenty different people. We regard image as input (X) and generate self-supervised signal (S) by first sampling an image from the same character as the input image and then applying translation/ rotation to it. Furthermore, we represent task-relevant information (T) by the labels of the image. Under this self-supervised signal construction, the exclusive information in X or S are drawing styles (i.e., by different people) and image augmentations, and only their shared information contribute to T. To formally show the later, if T representing the label for X/S, then $P(T|X)$ and $P(T|S)$ are Dirac. Hence, $T \perp\!\!\!\perp S|X$ and $T \perp\!\!\!\perp X|S$, suggesting Assumption 5 holds.

We train the feature mapping $F_X(\cdot)$ with SSL objectives (see eq. (6.4)), set $F_S(\cdot) = F_X(\cdot)$, let $R(\cdot)$ be symmetrical to $F_X(\cdot)$, and $G(\cdot)$ be an identity mapping. On the test set, we fix the mapping and randomly select 5 examples per character as the labeled examples. Then, we classify the rest of the examples using

[6] More complex datasets such as CIFAR10 [Krizhevsky et al., 2009] or ImageNet [Deng et al., 2009], to achieve similar performance, require a much larger training scale from contrastive to forward predictive objective. For example, on ImageNet, MoCo [He et al., 2019] uses 8 GPUs for its contrastive objective and ImageGPT [Chen et al.] uses 2048 TPUs for its forward predictive objective. We choose the Omniglot to ensure fair comparisons among different self-supervised learning objectives under reasonable computation constraint.

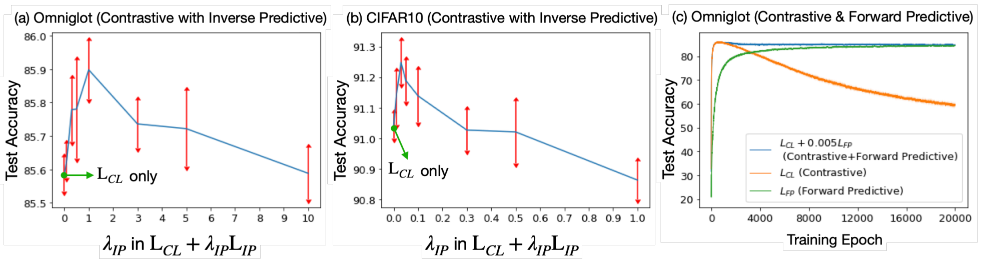


Figure 6.3: Comparisons for different compositions of SSL objectives on Omniglot and CIFAR10.

the 1-nearest neighbor classifier based on feature (i.e., $Z_X = F_X(X)$) cosine similarity. The random performance on this task stands at $\frac{1}{659} \approx 0.15\%$. One may refer to Chapter 6.5.6 for more details.

▷ *Results & Discussions.* In Figure 6.3, we evaluate the generalization ability on the test set for different SSL objectives. First, we examine how the introduced inverse predictive learning objective L_{IP} can help improve the performance along with the contrastive learning objective L_{CL}. We present the results in Figure 6.3 (a) and also provide experiments with SimCLR [Chen et al., 2020a] on CIFAR10 [Krizhevsky et al., 2009] in Figure 6.3 (b), where $\lambda_{IP} = 0$ refers to the exact same setup as in SimCLR (which considers only L_{CL}). We find that adding L_{IP} in the objective can boost model performance, although being sensitive to the hyper-parameter λ_{IP}. According to Theorem 4, the improved performance suggests a more compressed representation results in better performance for the downstream tasks. Second, we add the discussions with the forward predictive learning objective L_{FP}. We present the results in Figure 6.3 (c). Comparing to L_{FP}, L_{CL} 1) reaches better test accuracy; 2) requires shorter training epochs to reach the best performance; and 3) suffers from overfitting with long-epoch training. Combining both of them ($\mathrm{L}_{CL} + 0.005\mathrm{L}_{FP}$) brings their advantages together.

Experiment II - Visual-Textual Representation Learning. We provide experiments using MS COCO dataset [Lin et al., 2014] that contains 328k multi-labeled images with 2.5 million labeled instances from 91 objects. Each image has 5 annotated captions describing the relationships between objects in the scenes. We regard image as input (X) and its textual descriptions as self-supervised signal (S). Since vision and text are two very different modalities, the multi-view redundancy may not be satisfied, which means ϵ_{info} may be large in Assumption 5.

We adopt L_{CL} $(+\lambda_{IP}\mathrm{L}_{\mathrm{IP}})$ as our SSL objective. We use ResNet18 [He et al., 2016] image encoder for $F_X(\cdot)$ (trained from scratch or fine-tuned on ImageNet [Deng et al., 2009] pre-trained weights), BERT-uncased [Devlin et al., 2018] text encoder for $F_S(\cdot)$ (trained from scratch or BookCorpus [Zhu et al., 2015]/Wikipedia pre-trained weights), and a linear layer for $G(\cdot)$. After performing self-supervised visual-textual representation learning, we consider the downstream multi-label classification over 91 categories. We evaluate learned visual representation (Z_X) using *downstream linear evaluation protocol* [Bachman et al., 2019, Hénaff et al., 2019, Hjelm et al., 2018, Oord et al., 2018, Tian et al., 2019, Tschannen et al., 2019]. Specifically, a linear classifier is trained from the self-supervised learned (fixed) representation to the labels on the training set. Commonly used metrics for multi-label classification are reported on MS COCO validation set: Micro ROC-AUC and Subset Accuracy. One may refer to Chapter 6.5.7 for more details on these metrics.

▷ *Results & Discussions.* First, Figure 6.4 (a) suggests that the SSL strategy can still work when the input and self-supervised signals lie in different modalities. For example, pre-trained ResNet with BERT (either raw or the pre-trained one) outperforms pre-trained ResNet alone. We also see that the self-supervised learned representations benefit more if the ResNet is pre-trained but not the BERT. This

(a) MS COCO (Using L_{CL} as SSL objective)

Setting	Micro ROC-AUC	Subset Acc.
Cross-modality Self-supervised Learning		
Raw BERT + Raw ResNet	0.5963 ± 0.0034	0.0166 ± 0.0017
Pre-trained BERT + Raw ResNet	0.5915 ± 0.0035	0.0163 ± 0.0011
Raw BERT + Pre-trained ResNet	0.7049 ± 0.0040	0.2081 ± 0.0063
Pre-trained BERT + Pre-trained ResNet	0.7065 ± 0.0026	0.2123 ± 0.0040
Non Self-supervised Learning		
Only Pre-trained ResNet	0.6761 ± 0.0045	0.1719 ± 0.0015

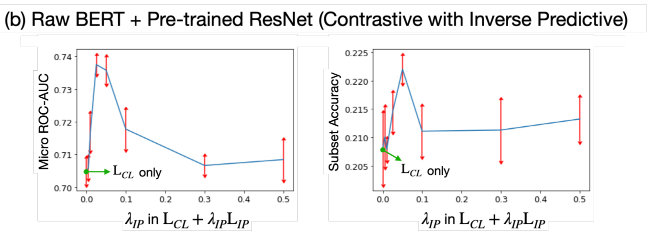

(b) Raw BERT + Pre-trained ResNet (Contrastive with Inverse Predictive)

Figure 6.4: Comparisons for different settings on self-supervised visual-textual representation training. We report metrics on MS COCO validation set with mean and standard deviation from 5 random trials.

result is in accord with the fact that object recognition requires more understanding in vision, and hence the pre-trained ResNet is preferrable than the pre-trained BERT. Next, Figure 6.4 (b) suggests that the self-supervised learned representations can be further improved by combining L_{CL} and L_{IP}, suggesting L_{IP} may be a useful objective to discard task-irrelevant information.

Remarks on λ_{IP} and L_{IP}. As observed in the experimental results, λ_{IP} is a sensitive hyper-parameter to the performance. We provide an optimization perspective to address this concern. Note that one of the our goals is to examine the setting when learning the minimal and sufficient representations for self-supervision (see Definition 3): minimize $H(Z_X|S)$ under the constraint that $I(Z_X;S)$ is maximized. However, this constrained optimization is not feasible when considering gradients methods in neural networks. Hence, our approach can be seen as its Lagrangian Relaxation by a weighted combination between L_{CL} (or L_{FP}, representing $I(Z_X;S)$) and L_{IP} (representing $H(Z_X|S)$) with the λ_{IP} being the Lagrangian coefficient.

The optimal λ_{IP} can be obtained by solving the Lagrangian dual, which depends on the parametrization of L_{CL} (or L_{FP}) and L_{IP}. Different parameterizations lead to different loss and gradient landscapes, and hence the optimal λ_{IP} differs across experiments. This conclusion is verified by the results presented in Figure 6.3 (a) and (b) and Figure 6.4 (b). Lastly, we point out that even not solving the Lagrangian dual, an empirical observation across experiments is that λ_{IP} which leads to the best performance is when the scale of L_{IP} is one-tenth to the scale of L_{CL} (or L_{FP}).

6.3 Related Work

Prior work by Arora et al. [2019] and the recent concurrent work [Lee et al., 2020, Tosh et al., 2020] are landmarks for theoretically understanding the success of SSL. In particular, Arora et al. [2019], Lee et al. [2020] showed a decreased sample complexity for downstream supervised tasks when adopting contrastive learning objectives [Arora et al., 2019] or predicting the known information in the data [Lee et al., 2020]. Tosh et al. [2020] showed that the linear functions of the learned representations are nearly optimal on downstream prediction tasks. By viewing the input and the self-supervised signal as two corresponding views of the data, we discuss the differences among these works and ours. On the one hand, the work by Arora et al. [2019], Lee et al. [2020] assume strong independence between the views conditioning on the downstream tasks, i.e., $I(X;S|T) \approx 0$. On the other hand, the work by Tosh et al. [2020] and ours assume strong independence between the downstream task and one view conditioning on the other view, i.e., $I(T;X|S) \approx 0$. Prior work [Balcan et al., 2005, Du et al., 2010] have compared these two assumptions and pointed out the former one ($I(X;S|T) \approx 0$) is too strong and not likely to hold in practice. We note that all these related work and ours have shown that the self-supervised learning methods are learning to extract task-relevant information. Our work additionally presents to discard task-irrelevant information and quantifies the amount of information that cannot be discarded.

Our method also resembles the InfoMax principle [Hjelm et al., 2018, Linsker, 1988] and the Multi-view Information Bottleneck method [Federici et al., 2020]. The InfoMax principle aims at preserving the information of itself, while ours aims at extracting the information in the self-supervised signal. On the other hand, to reduce the redundant information across views, the Multi-view Information Bottleneck method proposed to minimize the conditional mutual information $I(Z_X; X|S)$, while ours propose to minimize the conditional entropy $H(Z_X|S)$. The conditional entropy minimization problem can be easily optimized via our proposed inverse predictive learning objective.

Another related work is InfoMin [Tian et al., 2020], where both InfoMin and our method suggest to learn the representations that contain "not" too much information. In particular, InfoMin presents to augment the data (i.e., by constructing learnable data augmentations) such that the shared information between augmented variants is as minimal as possible, followed by the mutual information maximization between the learned features from the augmented variants. Our method instead considers standard augmentations (e.g., rotations and translations), followed by learning representations that contain no more than the shared information between the augmented variants of the data.

On the empirical side, we explain why contrastive [Bachman et al., 2019, Chen et al., 2020a, Oord et al., 2018] and predictive learning [Chen et al., Pathak et al., 2016, Vondrick et al., 2016, Zhang et al., 2016] approaches can unsupervised extract task-relevant information. Different from these work, we present an objective to discard task-irrelevant information and show its combination with existing contrastive or predictive objectives benefits the performance.

6.4 Discussion

In this chapter, we study both theoretical and empirical perspectives on self-supervised learning to address the sub-challenge of cross-view learning with only pairing information. We show that the self-supervised learned representations could extract task-relevant information (with a potential loss) and discard task-irrelevant information (with a fixed gap), along with their practical deployments such as contrastive and predictive learning objectives.

6.5 Appendix

6.5.1 Remarks on Learning Minimal and Sufficient Representations

So far, we discussed the objectives to learn minimal and sufficient representations (Definition 2). Here, we discuss the similarities and differences between the prior methods [Achille and Soatto, 2018, Tishby et al., 2000] and ours. First, to obtain sufficient representations (for the downstream task T), all the methods presented to maximize $I(Z_X; T)$. Then, to maintain minimal amount of information in the representations, the prior methods [Achille and Soatto, 2018, Tishby et al., 2000] presented to minimize $I(Z_X; X)$ and the ours presents to minimize $H(Z_X|T)$. Our goal is to relate $I(Z_X; X)$ minimization and $H(Z_X|T)$ minimization in our framework.

To begin with, under the constraint $I(Z_X; T)$ is maximized, we see that minimizing $I(Z_X; X)$ is equivalent to minimizing $I(Z_X; X|T)$. The reason is that $I(Z_X; X) = I(Z_X; X|T) + I(Z_X; X; T)$, where $I(Z_X; X; T) = I(Z_X; T)$ due to the determinism from X to Z_X (our framework learns a deterministic function from X to Z_X) and $I(Z_X; T)$ is maximized in our constraint. Then, $I(Z_X; X|T) = H(Z_X|T) - H(Z_X|X, T)$, where $H(Z_X|T)$ contains no randomness (no information) as Z_X being deterministic from X. Hence, $I(Z_X; X|T)$ minimization and $H(Z_X|T)$ minimization are interchangeable.

The same claim can be made from the downstream task T to the self-supervised signal S. In specific, when X to Z_X is deterministic, $I(Z_X; X|S)$ minimization and $H(Z_X|S)$ minimization are interchangeable. As discussed in the related work, for reducing the amount of the redundant information, Federici et al. [2020] presented to use $I(Z_X; X|S)$ minimization and ours presented to use $H(Z_X|T)$ minimization. We also note that directly minimizing the conditional mutual information (i.e., $I(Z_X; X|S)$) requires a min-max optimization [Mukherjee et al., 2020], which may cause instability in practice. To overcome the issue, Federici et al. [2020] assumes a Gaussian encoder for $X \to Z_X$ and presents an upper bound of the original objective.

6.5.2 Proofs for Theorem 3 and 4

We start by presenting a useful lemma from the fact that $F_X(\cdot)$ is a deterministic function:

Lemma 10 (Determinism)**.** If $P(Z_X|X)$ is Dirac, then the following conditional independence holds: $T \perp\!\!\!\perp Z_X|X$ and $S \perp\!\!\!\perp Z_X|X$, inducing a Markov chain $S \leftrightarrow T \leftrightarrow X \to Z_X$.

Proof. When Z_X is a deterministic function of X, for any A in the sigma-algebra induced by Z_X we have $\mathbb{E}[\mathbf{1}_{[Z_X \in A]}|X, \{T, S\}] = \mathbb{E}[\mathbf{1}_{[Z_X \in A]}|X, S] = \mathbb{E}[\mathbf{1}_{[Z_X \in A]}|X]$, which implies $T \perp\!\!\!\perp Z_X|X$ and $S \perp\!\!\!\perp Z_X|X$. ■

Theorem 7 (Task-relevant information with a potential loss ϵ_{info}, restating Theorem 3)**.** The supervised learned representations (i.e., $I(Z_X^{\text{sup}}; T)$ and $I(Z_X^{\text{sup}_{\text{min}}}; T)$) contain all the task-relevant information in the input (i.e., $I(X; T)$). The self-supervised learned representations (i.e., $I(Z_X^{\text{ssl}}; T)$ and $I(Z_X^{\text{ssl}_{\text{min}}}; T)$) contain all the task-relevant information in the input with a potential loss ϵ_{info}. Formally,

$$I(X; T) = I(Z_X^{\text{sup}}; T) = I(Z_X^{\text{sup}_{\text{min}}}; T) \geq I(Z_X^{\text{ssl}}; T) \geq I(Z_X^{\text{ssl}_{\text{min}}}; T) \geq I(X; T) - \epsilon_{\text{info}}.$$

Proof. The proofs contain two parts. The first one is showing the results for the supervised learned representations and the second one is for the self-supervised learned representations.

Supervised Learned Representations: Adopting Data Processing Inequality (DPI by Cover and Thomas [2012]) in the Markov chain $S \leftrightarrow T \leftrightarrow X \to Z_X$ (Lemma 10), $I(Z_X; T)$ is maximized at $I(X; T)$. Since both supervised learned representations (Z_X^{sup} and $Z_X^{\text{sup}_{\text{min}}}$) maximize $I(Z_X; T)$, we conclude $I(Z_X^{\text{sup}}; T) = I(Z_X^{\text{sup}_{\text{min}}}; T) = I(X; T)$.

Self-supervised Learned Representations: First, we have

$$I(Z_X; S) = I(Z_X; T) - I(Z_X; T|S) + I(Z_X; S|T) = I(Z_X; T; S) + I(Z_X; S|T)$$

and

$$I(X; S) = I(X; T) - I(X; T|S) + I(X; S|T) = I(X; T; S) + I(X; S|T).$$

By DPI in the Markov chain $S \leftrightarrow T \leftrightarrow X \to Z_X$ (Lemma 10), we know

- $I(Z_X; S)$ is maximized at $I(X; S)$
- $I(Z_X; S; T)$ is maximized at $I(X; S; T)$
- $I(Z_X; S|T)$ is maximized at $I(X; S|T)$

Since both self-supervised learned representations (Z_X^{ssl} and $Z_X^{\text{ssl}_{\text{min}}}$) maximize $I(Z_X; S)$, we have $I(Z_X^{\text{ssl}}; S) = I(Z_X^{\text{ssl}_{\text{min}}}; S) = I(X; S)$. Hence, $I(Z_X^{\text{ssl}}; S; T) = I(Z_X^{\text{ssl}_{\text{min}}}; S; T) = I(X; S; T)$ and $I(Z_X^{\text{ssl}}; S|T) = I(Z_X^{\text{ssl}_{\text{min}}}; S|T) = I(X; S|T)$. Using the result $I(Z_X^{\text{ssl}}; S; T) = I(Z_X^{\text{ssl}_{\text{min}}}; S; T) = I(X; S; T)$, we get

$$I(Z_X^{\text{ssl}}; T) = I(X; T) - I(X; T|S) + I(Z_X^{\text{ssl}}; T|S)$$

and

$$I(Z_X^{\text{ssl}_{\text{min}}}; T) = I(X; T) - I(X; T|S) + I(Z_X^{\text{ssl}_{\text{min}}}; T|S).$$

Now, we are ready to present the inequalities:

1. $I(X; T) \geq I(Z_X^{\text{ssl}}; T)$ due to $I(X; T|S) \geq I(Z_X^{\text{ssl}}; T|S)$ by DPI.
2. $I(Z_X^{\text{ssl}}; T) \geq I(Z_X^{\text{ssl}_{\text{min}}}; T)$ due to $I(Z_X^{\text{ssl}}; T|S) \geq I(Z_X^{\text{ssl}_{\text{min}}}; T|S) = 0$. Since $H(Z_X|S)$ is minimized at $Z_X^{\text{ssl}_{\text{min}}}$, $I(Z_X^{\text{ssl}_{\text{min}}}; T|S) = 0$.
3. $I(Z_X^{\text{ssl}_{\text{min}}}; T) \geq I(X; T) - \epsilon_{\text{info}}$ due to

$$I(X; T) - I(X; T|S) + I(Z_X^{\text{ssl}_{\text{min}}}; T|S) \geq I(X; T) - I(X; T|S) \geq I(X; T) - \epsilon_{\text{info}},$$

where $I(X; T|S) \leq \epsilon_{\text{info}}$ by the redundancy assumption.

■

Theorem 8 (Task-irrelevant information with a fixed compression gap $I(X; S|T)$, restating Theorem 4)**.** The sufficient self-supervised representation (i.e., $I(Z_X^{\text{ssl}}; T)$) contains more task-irrelevant information in the input than then the sufficient and minimal self-supervised representation (i.e., $I(Z_X^{\text{ssl}_{\text{min}}}; T)$). The later contains an amount of the information, $I(X; S|T)$, that cannot be discarded from the input. Formally,

$$I(Z_X^{\text{ssl}}; X|T) = I(X; S|T) + I(Z_X^{\text{ssl}}; X|S, T) \geq I(Z_X^{\text{ssl}_{\text{min}}}; X|T) = I(X; S|T) \geq I(Z_X^{\text{sup}_{\text{min}}}; X|T) = 0.$$

Proof. First, we see that

$$I(Z_X; X|T) = I(Z_X; X; S|T) + I(Z_X; X|S, T) = I(Z_X; S|T) + I(Z_X; X|S, T),$$

where $I(Z_X; X; S|T) = I(Z_X; S|T)$ by DPI in the Markov chain $S \leftrightarrow T \leftrightarrow X \rightarrow Z_X$.

We conclude the proof by combining the following:

- From the proof in Theorem 7, we showed $I(Z_X^{\text{ssl}}; S|T) = I(Z_X^{\text{ssl}_{\text{min}}}; S|T) = I(X; S|T)$.
- Since $H(Z_X|S)$ is minimized at $Z_X^{\text{ssl}_{\text{min}}}$, $I(Z_X^{\text{ssl}_{\text{min}}}; X|S, T) = 0$.
- Since $H(Z_X|T)$ is minimized at $Z_X^{\text{sup}_{\text{min}}}$, $I(Z_X^{\text{sup}_{\text{min}}}; X|T) = 0$.

■

6.5.3 Proof for Proposition 10

Proposition 11 (Mutual Information Neural Estimation, restating Proposition 10)**.** Let $0 < \delta < 1$. There exists $d \in \mathbb{N}$ and a family of neural networks $\mathcal{F} := \{\hat{f}_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$ where Θ is compact, so that $\exists \theta^* \in \Theta$, with probability at least $1-\delta$ over the draw of $\{z_{x_i}, s_i\}_{i=1}^n \sim P_{Z_X,S}^{\otimes n}$, $\left|\widehat{I}_{\theta^*}^{(n)}(Z_X;S) - I(Z_X;S)\right| \leq O\left(\sqrt{\frac{d+\log(1/\delta)}{n}}\right)$.

Sketch of Proof. The proof is a standard instance of uniform convergence bound. First, we assume the boundness and the Lipschitzness of $\hat{f}_\theta$. Then, we use the universal approximation lemma of neural networks [Hornik et al., 1989]. Last, combing all these two along with the uniform convergence in terms of the covering number [Bartlett, 1998], we complete the proof. ■

We note that the complete proof can be found in the prior work [Tsai et al., 2020d]. An alternative but similar proof can be found in another prior work [Belghazi et al., 2018], which gives us $\left|\widehat{I}_{\theta^*}^{(n)}(Z_X;S) - I(Z_X;S)\right| \leq O\left(\sqrt{\frac{d\log d+\log(1/\delta)}{n}}\right)$. The subtle difference between them is that, given a neural network function space $\Theta \subseteq \mathbb{R}^d$ and its covering number $\mathcal{N}(\Theta,\eta)$, Tsai et al. [2020d] has $\mathcal{N}(\Theta,\eta) = O\Big((\eta)^{-d}\Big)$ by Bartlett [1998] and Belghazi et al. [2018] has $\mathcal{N}(\Theta,\eta) = O\Big((\eta/\sqrt{d})^{-d}\Big)$ by Shalev-Shwartz and Ben-David [2014]. Both are valid and the one used by Tsai et al. [2020d] is tighter.

6.5.4 Proofs for Theorem 5 and 6

To begin with, we see that

$$\begin{aligned} I(Z_X;T) &= I(Z_X;X) - I(Z_X;X|T) + I(Z_X;T|X) = I(Z_X;X) - I(Z_X;X|T) \\ &= I(Z_X;S) - I(Z_X;S|X) + I(Z_X;X|S) - I(Z_X;X|T) \\ &= I(Z_X;S) + I(Z_X;X|S) - I(Z_X;X|T) \\ &\geq I(Z_X;S) - I(Z_X;X|T), \end{aligned}$$

where $I(Z_X;T|X) = I(Z_X;S|X) = 0$ due to the determinism from X to Z_X. Then, in the proof of Theorem 8, we have shown $I(Z_X;X|T) = I(Z_X;S|T) + I(Z_X;X|S,T)$. Hence,

$$\begin{aligned} I(Z_X;T) &\geq I(Z_X;S) - I(Z_X;S|T) - I(Z_X;X|S,T) \\ &\geq I(Z_X;S) - I(X;S|T) - I(Z_X;X|S,T), \end{aligned}$$

where $I(Z_X;S|T) \leq I(X;S|T)$ by DPI.

Theorem 9 (Bayes Error Rates for Arbitrary Learned Representations, restating Theorem 5)**.** For an arbitrary learned representations Z_X, $P_e = \mathrm{Th}(\bar{P}_e)$ with

$$\bar{P}_e \leq 1 - \exp^{-\left(H(T)+I(X;S|T)+I(Z;X|S,T)-\hat{I}_{\theta^*}^{(n)}(Z_X;S)+O\left(\sqrt{\frac{d+\log(1/\delta)}{n}}\right)\right)}.$$

Proof. We use the inequality between P_e and $H(T|Z_X)$ indicated by Feder and Merhav [1994]:

$$-\log(1-P_e) \leq H(T|Z_X).$$

Combining with $I(Z_X;T) = H(T) - H(T|Z_X)$ and $I(Z_X;T) \geq I(Z_X;S) - I(X;S|T) - I(Z_X;X|S,T)$, we have

$$\log(1-P_e) \geq -H(T) + I(Z_X;S) - I(X;S|T) - I(Z_X;X|S,T).$$

Hence,

$$P_e \leq 1 - \exp^{-\left(H(T)+I(X;S|T)+I(Z;X|S,T)-I(Z_X;S)\right)}.$$

Next, by definition of the Bayes error rate, we know $0 \leq P_e \leq 1 - \frac{1}{|T|}$.

We conclude the proof by combining Proposition 11, $\left|\widehat{I}^{(n)}_{\theta^*}(Z_X;S) - I(Z_X;S)\right| \leq O\left(\sqrt{\frac{d+\log(1/\delta)}{n}}\right)$. ■

Theorem 10 (Bayes Error Rates for Self-supervised Learned Representations, restating Theorem 6)**.** Let $P_e^{\text{sup}}/P_e^{\text{ssl}}/P_e^{\text{ssl}_{\text{min}}}$ be the Bayes error rate of the supervised or the self-supervised learned representations $Z_X^{\text{sup}}/Z_X^{\text{ssl}}/Z_X^{\text{ssl}_{\text{min}}}$. Then, $P_e^{\text{ssl}} = \text{Th}(\bar{P}_e^{\text{ssl}})$ and $P_e^{\text{ssl}_{\text{min}}} = \text{Th}(\bar{P}_e^{\text{ssl}_{\text{min}}})$ with

$$-\frac{\log\left(1-P_e^{\text{sup}}\right)+\log 2}{\log\left(|T|\right)} \leq \{\bar{P}_e^{\text{ssl}}, \bar{P}_e^{\text{ssl}_{\text{min}}}\} \leq 1 - \exp^{-(\log 2 + P_e^{\text{sup}} \cdot \log|T| + \epsilon_{\text{info}})}.$$

Proof. We use the two inequalities between P_e and $H(T|Z_X)$ by Feder and Merhav [1994] and Cover and Thomas [2012]:

$$-\log(1-P_e) \leq H(T|Z_X)$$

and

$$H(T|Z_X) \leq \log 2 + P_e \log|T|.$$

Combining the results from Theorem 7:

$$I(Z_X^{\text{sup}};T) \geq I(Z_X^{\text{ssl}};T) \geq I(Z_X^{\text{ssl}_{\text{min}}};T) \geq I(Z_X^{\text{sup}};T) - \epsilon_{\text{info}},$$

we have

- the upper bound of the self-supervised learned representations' Bayes error rate:

$$\begin{aligned}\{-\log(1-P_e^{\text{ssl}}), -\log(1-P_e^{\text{ssl}_{\text{min}}})\} &\leq \{H(T|Z_X^{\text{ssl}}), H(T|Z_X^{\text{ssl}_{\text{min}}})\} \\ &\leq H(T|Z_X^{\text{sup}}) + \epsilon_{\text{info}} \\ &\leq \log 2 + P_e^{\text{sup}} \log|T| + \epsilon_{\text{info}},\end{aligned}$$

  which suggests $\{P_e^{\text{ssl}}, P_e^{\text{ssl}_{\text{min}}}\} \leq 1 - \exp^{-(\log 2 + P_e^{\text{sup}} \cdot \log|T| + \epsilon_{\text{info}})}$.
- the lower bound of the self-supervised learned representations' Bayes error rate:

$$\begin{aligned}-\log(1-P_e^{\text{sup}}) &\leq H(T|Z_X^{\text{sup}}) \\ &\leq \{H(T|Z_X^{\text{ssl}}), H(T|Z_X^{\text{ssl}_{\text{min}}})\} \\ &\leq \{\log 2 + P_e^{\text{ssl}} \log|T|, \leq \{\log 2 + P_e^{\text{ssl}_{\text{min}}} \log|T|\},\end{aligned}$$

  which suggests $-\frac{\log\left(1-P_e^{\text{sup}}\right)+\log 2}{\log\left(|T|\right)} \leq \{P_e^{\text{ssl}}, P_e^{\text{ssl}_{\text{min}}}\}$.

We conclude the proof by having P_e lie in the feasible range: $0 \leq P_e \leq 1 - \frac{1}{|T|}$. ■

6.5.5 Tighter Bounds for the Bayes Error Rates

We note that the bound used in Theorems 9 and 10: $-\log(1-P_e) \leq H(T|Z_X) \leq \log 2 + P_e \log|T|$ is not tight. A tighter bound is $H^-(P_e) \leq H(T|Z_X) \leq H^+(P_e)$ with

$$H^-(P_e) := H\Big(k(1-P_e)\Big) + k(1-P_e)\log k \text{ when } \frac{k-1}{k} \leq P_e \leq \frac{k}{k+1}\,,\; 1 \leq k \leq |T|-1,$$

$$\text{and} \qquad H^+(P_e) := H(P_e) + P_e \log\left(|T|-1\right),$$

where $H(x) = -x\log(x) - (1-x)\log(1-x)$.

It is clear that $-\log(1-P_e) \leq H^-(P_e)$ and $H^+(P_e) \leq \log 2 + P_e \log(|T|)$.

Hence, Theorem 9 and 10 can be improved as follows:

Theorem 11 (Tighter Bayes Error Rates for Arbitrary Learned Representations)**.** For an arbitrary learned representations Z_X, $P_e = \text{Th}(\bar{P}_e)$ with $\bar{P}_e \leq P_{e_{\text{upper}}}$. $P_{e_{\text{upper}}}$ is derived from the program

$$\arg\max_{P_e} H^-(P_e) \leq H(T) - \hat{I}_\theta^{(n)}(Z_X^{\text{ssl}}; S) + I(X;S|T) + I(Z_X; X|S,T) + O\Big(\sqrt{\frac{d+\log(1/\delta)}{n}}\Big).$$

Theorem 12 (Tighter Bayes Error Rates for Self-supervised Learned Representations)**.** Let $P_e^{\text{sup}}/P_e^{\text{ssl}}/P_e^{\text{ssl}_{\text{min}}}$ be the Bayes error rate of the supervised or the self-supervised learned representations $Z_X^{\text{sup}}/Z_X^{\text{ssl}}/Z_X^{\text{ssl}_{\text{min}}}$. Then, $P_e^{\text{ssl}} = \text{Th}(\bar{P}_e^{\text{ssl}})$ and $P_e^{\text{ssl}_{\text{min}}} = \text{Th}(\bar{P}_e^{\text{ssl}_{\text{min}}})$ with

$$P_{e_{\text{lower}}}^{\text{ssl}} \leq \{\bar{P}_e^{\text{ssl}}, \bar{P}_e^{\text{ssl}_{\text{min}}}\} \leq P_{e_{\text{upper}}}^{\text{ssl}}.$$

$P_{e_{\text{lower}}}^{\text{ssl}}$ is derived from the following program

$$\arg\min_{P_e^{\text{ssl}}} H^-(P_e^{\text{sup}}) \leq H^+(P_e^{\text{ssl}})$$

and $P_{e_{\text{upper}}}^{\text{ssl}}$ is derived from the following program

$$\arg\max_{P_e^{\text{ssl}}} H^-(P_e^{\text{ssl}}) \leq H^+(P_e^{\text{sup}}) + \epsilon_{\text{info}}.$$

6.5.6 More on Visual Representation Learning Experiments

We designed controlled experiments on self-supervised visual representation learning to empirically support our theorem and examine different compositions of SSL objectives. Here, we will discuss 1) the architecture design; 2) different deployments of contrastive/ forward predictive learning; and 3) different self-supervised signal construction strategy.

6.5.6.1 Architecture Design

The input image has size 105 × 105. For image augmentations, we adopt 1) rotation with degrees from -10° to $+10^\circ$; 2) translation from -15 pixels to $+15$ pixels; 3) scaling both width and height from 0.85 to 1.0; 4) scaling width from 0.85 to 1.25 while fixing the height; and 5) resizing the image to 28 × 28. Then, a deep network takes a 28 × 28 image and outputs a $1024-$dim. feature vector. The deep network has the structure:

Conv − BN − ReLU − Conv − BN − ReLU − MaxPool − Conv − BN − ReLU − MaxPool − Conv −BN − ReLU − Ma Conv has 3x3 kernel size with 128 output channels, MaxPool has 2x2 kernel size, and Linear is a 1152 to 1024 weight matrix. $R(\cdot)$ is symmetric to $F_X(\cdot)$, which has Linear − BN − ReLU − UnFlatten − DeConv − BN − ReLU − −BN − ReLU − DeConv. $R(\cdot)$ has the exact same number of parameters as $F_X(\cdot)$. Note that we use the same network designs in $I(\cdot,\cdot)$ and $H(\cdot|\cdot)$ estimations.

6.5.6.2 Different Deployments for Contrastive and Predictive Learning Objectives

For practical deployments, we suggested Contrastive Predictive Coding (CPC) [Oord et al., 2018] for L_{CL} and assume Gaussian distribution for the variational distributions in L_{FP}/ L_{IP}. The practical deployments can be abundant by using different mutual information approximations for L_{CL} and having different distribution assumptions for L_{FP}/ L_{IP}. In the following, we discuss a few examples.

Contrastive Learning. Other than CPC [Oord et al., 2018], another popular contrastive learning objective is JS [Bachman et al., 2019], which is the lower bound of Jensen-Shannon divergence between $P(Z_S, Z_X)$ and $P(Z_S)P(Z_X)$ (a variational bound of mutual information). Its objective can be written as

$$\max_{Z_S=F_S(S),Z_X=F_X(X),G} \mathbb{E}_{P(Z_S,Z_X)}\Big[-\mathrm{softplus}\Big(-\langle G(z_x),G(z_s)\rangle\Big)\Big] - \mathbb{E}_{P(Z_S)P(Z_X)}\Big[\mathrm{softplus}\Big(\langle G(z_x),G(z_s)\rangle\Big)\Big],$$

where we use softplus to denote $\mathrm{softplus}\,(x) = \log\,(1+\exp\,(x))$.

Predictive Learning. Gaussian distribution may be the simplest distribution form that we can imagine, which leads to Mean Square Error (MSE) reconstruction loss. Here, we use forward predictive learning as an example, and we discuss the case when $\mathcal{S}$ lies in discrete $\{0,1\}$ sample space. Specifically, we let $Q_\phi(S|Z_X)$ be factorized multivariate Bernoulli:

$$\max_{Z_X=F_X(X),R} \mathbb{E}_{P_{S,Z_X}}\left[\sum_{i=1}^{p} s_i \cdot \log\,[R(z_x)]_i + (1-s_i)\cdot\log\,[1-R(z_x)]_i\right]. \tag{6.5}$$

This objective leads to Binary Cross Entropy (BCE) reconstruction loss.

If we assume each reconstruction loss corresponds to a particular distribution form, then by ignoring which variatioinal distribution we choose, we are free to choose arbitrary reconstruction loss. For instance, by switching s and z in eq. (6.5), the objective can be regarded as Reverse Binary Cross Entropy Loss (RevBCE) reconstruction loss. In our experiments, we find RevBCE works the best among {MSE, BCE, and RevBCE}. Therefore, we choose RevBCE as the example reconstruction loss as L_{FP}.

More Experiments. We provide an additional set of experiments by having {CPC, JS} for L_{CL} and {MSE, BCE, RevBCE} reconstruction loss for L_{FP} in Figure 6.5. From the results, we find different formulation of objectives bring very different test generalization performance. We argue that, given a particular task, it is challenging but important to find the best deployments for contrastive and predictive learning objectives.

6.5.6.3 Different Self-supervised Signal Construction Strategy

We designed a self-supervised signal construction strategy that the input (X) and the self-supervised signal (S) differ in {drawing styles, image augmentations}. This self-supervised signal construction strategy is different from the one that is commonly adopted in most self-supervised visual representation learning work [Bachman et al., 2019, Chen et al., 2020a, Tian et al., 2019]. Specifically, prior work consider the difference between input and the self-supervised signal only in image augmentations. We provide additional experiments in Fig. 6.6 to compare these two different self-supervised signal construction strategies.

(a) Omniglot (Composing SSL Objectives with L_{FP} as MSE)

Objective	Trained for	Test Accuracy
L_{CL}	500 epochs	$85.59 \pm 0.05\%$
$\mathrm{L}_{CL} + \mathrm{L}_{IP}$	500 epochs	$85.90 \pm 0.09\%$
L_{FP}	20000 epochs	$84.83 \pm 0.07\%$
$\mathrm{L}_{FP} + 10\mathrm{L}_{IP}$	20000 epochs	$84.96 \pm 0.04\%$
$\mathrm{L}_{CL} + 10\mathrm{L}_{FP}$	9000 epochs	$86.13 \pm 0.21\%$
$\mathrm{L}_{CL} + 10\mathrm{L}_{FP} + \mathrm{L}_{IP}$	9000 epochs	$86.17 \pm 0.13\%$

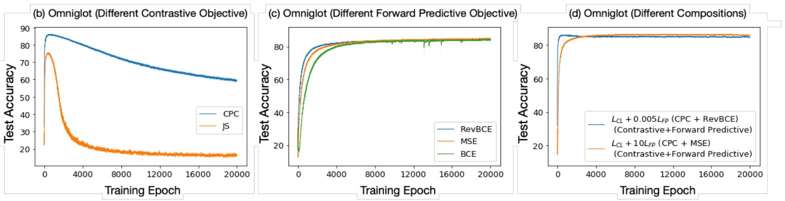


Figure 6.5: Comparisons for different objectives/compositions of SSL objectives on self-supervised visual representation training. We report mean and its standard error from 5 random trials.

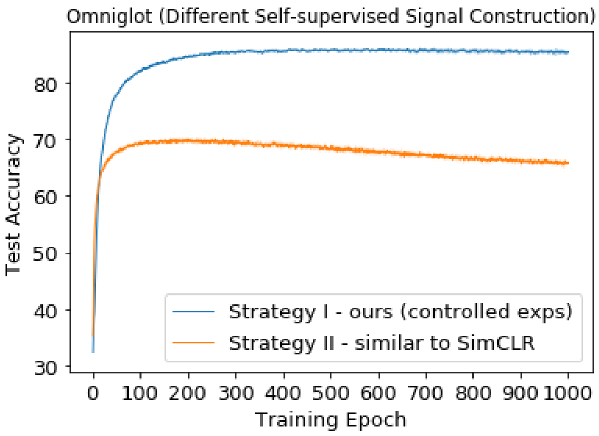


Figure 6.6: Comparisons for different self-supervised signal construction strategies. The differences between the input and the self-supervised signals are {drawing styles, image augmentations} for our construction strategy and only {image augmentations} for SimCLR [Chen et al., 2020a]'s strategy. We choose L_{CL} as our objective, reporting mean and its standard error from 5 random trials.

We see that, comparing to the common self-supervised signal construction strategy [Bachman et al., 2019, Chen et al., 2020a, Tian et al., 2019], the strategy introduced in our controlled experiments has much better generalization ability to test set. It is worth noting that, although our construction strategy has access to the label information (i.e., we sample the self-supervised signal image from the same character with the input image), our SSL objectives do not train with the labels. Nonetheless, since we implicitly utilize the label information in our self-supervised construction strategy, it will be unfair to directly compare our strategy and prior one. An interesting future research direction is examining different self-supervised signal construction strategy and even combine full/part of label information into self-supervised learning.

6.5.7 Metrics in Visual-Textual Representation Learning

- Subset Accuracy (A) [Sorower], also know as the Exact Match Ratio (MR), ignores all partially correct (consider them incorrect) outputs and extend accuracy from the single label case to the multi-label setting.

$$MR = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{[Y_i = H_i]}$$

- Micro AUC ROC score [Fawcett, 2006] computes the AUC (Area under the curve) of a receiver operating characteristic (ROC) curve.

Chapter 7

Learning with Limited Supervision - Cross-view Learning with Auxiliary Information

Self-supervised learning (SSL) considers the learning objectives that use data's self-information but not labels, where the labels are often expensive to collect. As a result, SSL empowers us to leverage a large amount of unlabeled data to learn good representations, and its applications span computer vision [Chen et al., 2020a, He et al., 2020], natural language processing [Devlin et al., 2018, Peters et al., 2018] and speech processing [Baevski et al., 2020, Schneider et al., 2019]. In addition to labels, we may sometimes access additional sources as auxiliary information for data, such as additional attributes information or data hierarchy information. The auxiliary information often naturally comes with the data, and hence it is cheaper to collect than labels. For example, Instagram images contain a mass amount of hashtags as additional attributes information. Nonetheless, the auxiliary information is often noisy. Hence, it raises a research challenge of effectively leveraging useful information from the auxiliary information in the SSL process.

We argue that a form of the valuable information provided by the auxiliary information is its implied clustering information of data. For example, we can expect an Instagram image to be semantically more similar to the image with the same hashtags than the image with different hashtags. Hence, our first step for leveraging the auxiliary information in SSL is to construct auxiliary-information-determined clusters. Specifically, we build data clusters such that the data from the same cluster have similar auxiliary information, such as having the same data attributes or belonging to the same data hierarchy. Then, our second step is to minimize the intra-cluster difference for the self-supervised learned representations. Particularly, we present the clustering InfoNCE (Cl-InfoNCE) objective to learn similar representations for augmented variants of data within the same cluster and dissimilar representations for data from different clusters. To conclude, the presented two-step approach leverages the structural information from the auxiliary information, then integrating the structural information into the SSL process. See Figure 7.1 for an overview of the chapter.

We highlight several properties of our approach. First, we characterize the goodness of the Cl-InfoNCE-learned representations via the statistical relationships between the constructed clusters and the downstream labels. A resulting implication is that we can expect better downstream performance for our auxiliary-information-infused self-supervised representations when having i) higher mutual information between the labels and the auxiliary-information-determined clusters and ii) lower conditional entropy of the clusters given the labels. Second, Cl-InfoNCE generalizes recent contrastive learning objectives by changing

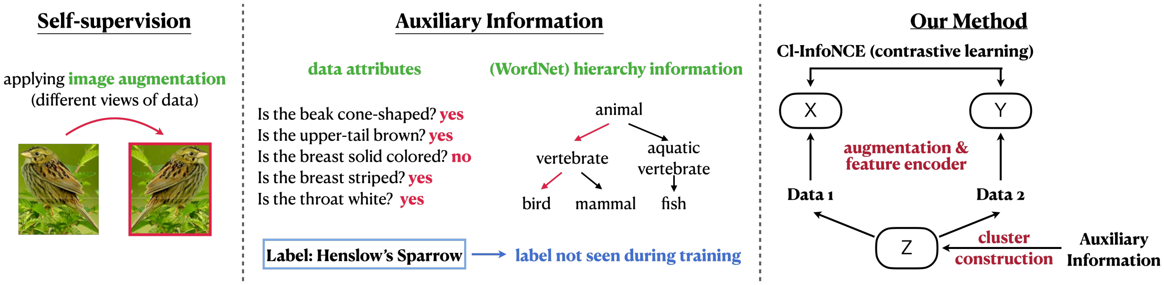

Self-supervision
applying image augmentation
(different views of data)
Auxiliary Information
data attributes
(WordNet) hierarchy information
Is the beak cone-shaped? yes
Is the upper-tail brown? yes
Is the breast solid colored? no
Is the breast striped? yes
Is the throat white? yes
animal
vertebrate
aquatic vertebrate
bird
mammal
fish
Label: Henslow's Sparrow
label not seen during training
Our Method
Cl-InfoNCE (contrastive learning)
X
Y
augmentation & feature encoder
Data 1
Data 2
Z
cluster construction
Auxiliary Information

Figure 7.1: **Left: Self-supervision.** Self-supervised learning (SSL) uses self-supervision (the supervision from the data itself) for learning representations. An example of self-supervision is the augmented variant of the original data. **Middle: Auxiliary Information.** This chapter aims to integrate the auxiliary information into SSL. We consider two types of auxiliary information: data attributes and (WordNet) hierarchy information. In our example, the data attributes are binary indicators, and the hierarchy information is the hierarchy information for the label. **Right: Our Method.** We first construct data clusters according to auxiliary information. We argue the formed clusters can provide valuable structural information of data for learning better representations. Second, we present the clustering InfoNCE (Cl-InfoNCE) objective to leverage the constructed clusters.

the way to construct the clusters. In particular, when each cluster contains only one data, Cl-InfoNCE specializes in conventional self-supervised contrastive objective (e.g., the InfoNCE objective [Oord et al., 2018]). When the clusters are labels, Cl-InfoNCE specializes in supervised contrastive objective (e.g., the objective considered by Khosla et al. [2020]). The generalization implies that our approach (auxiliary-information-determined clusters + Cl-InfoNCE) works between conventional self-supervised and supervised representation learning. Third, Cl-InfoNCE is a computationally efficient method as it can scale up even with many clusters. The reason is that Cl-InfoNCE is a contrastive-based approach, which is non-parametric. Particularly, the number of the parameters in Cl-InfoNCE is independent of the number of clusters.

We conduct experiments on learning visual representations using UT-zappos50K [Yu and Grauman, 2014], CUB-200-2011 [Wah et al., 2011], Wider Attribute [Li et al., 2016] and ImageNet-100 [Russakovsky et al., 2015] datasets. For the first set of experiments, we focus on the analysis of Cl-InfoNCE to study how well it works with unsupervised constructed clusters (K-means clusters). We find it achieves better performance comparing to the clustering-based self-supervised learning approaches, such as the Prototypical Contrastive Learning (PCL) [Li et al., 2020a] method. The result suggests that the K-means method + Cl-InfoNCE can be a strong baseline for the conventional self-supervised learning setting. For the second set of experiments, we like to see how much improvement can the auxiliary information bring to us. We consider the discrete attributes and the WordNet hierarchy information [Miller, 1995] as the auxiliary information. We show that the auxiliary-information-infused self-supervised representations, compared to conventional self-supervised representation, have a much better performance on downstream tasks. We also find that Cl-InfoNCE has a better performance than the baseline - predicting the clustering assignments with cross-entropy loss.

7.1 Related Work

Self-supervised Learning. Self-supervised learning (SSL) defines a pretext task as a pre-training step and uses the pre-trained features for a wide range of downstream tasks, such as object detection and segmentation in Computer Vision [Chen et al., 2020a, He et al., 2020], question answering, and language understanding in Natual Language Processing [Devlin et al., 2018, Peters et al., 2018] and automatic speech recognition in Speech Processing [Baevski et al., 2020, Schneider et al., 2019]. In this chapter, we focus on discussing two types of pretext tasks: clustering approaches [Caron et al., 2018, 2020] and contrastive approaches [Chen et al., 2020a, He et al., 2020].

On the one hand, the clustering approaches jointly learn the networks' parameters and the cluster assignments of the resulting features. The cluster assignments are obtained through unsupervised clustering methods such as k-means [Caron et al., 2018], the optimal transportation algorithms such as Sinkhorn algorithm [Caron et al., 2020], etc. It is worth noting that the clustering approaches enforce consistency between cluster assignments for different augmentations of the same data. On the other hand, the contrastive approaches learn similar representations for augmented variants of a data and dissimilar representations for different data. The objectives considered for contrastive approaches are the InfoNCE objective [Chen et al., 2020a, He et al., 2020, Oord et al., 2018], Wasserstein Predictive Coding [Ozair et al., 2019], Relative Predictive Coding [Tsai et al., 2021c], etc. Both the clustering and the contrastive approaches aim to learn representations that are invariant to data augmentations.

There is another line of work combining clustering and contrastive approaches, such as HUBERT [Hsu et al., 2020], Prototypical Contrastive Learning [Li et al., 2020a] and Wav2Vec [Baevski et al., 2020, Schneider et al., 2019]. They first construct (unsupervised) clusters from the data. Then, they perform a contrastive approach to learn similar representations for the data within the same cluster. Our approach relates to these work with two differences: 1) we construct the clusters from the auxiliary information; and 2) we present Cl-InfoNCE as a new contrastive approach and characterize the goodness for the resulting representations.

Learning to Predict Auxiliary Information. Our study also relates to work on learning to predict weak labels [Mahajan et al., 2018, Radford et al., 2021, Sun et al., 2017, Wen et al., 2018]. The weak labels can be hashtags for Instagram images [Mahajan et al., 2018], metadata such as identity and nationality for a person [Wen et al., 2018] or corresponding textual descriptions for an image [Radford et al., 2021]. Compared to labels, the weak labels are noisy but require much less manual annotation work. This line of work shows that the network learned by weakly supervised pre-training tasks can generalize well to various downstream tasks, including object detection and segmentation, cross-modality matching, and video action recognition. The main difference between this line of work and ours is that our approach does not consider a prediction objective but a contrastive learning objective (i.e., the Cl-InfoNCE objective).

7.2 Method

We present a two-step approach to leverage the structural information from the auxiliary information and then integrate this structural information into the self-supervised learning process. The first step (Section 7.2.1) clusters data according to auxiliary information. And we consider discrete attributes and data hierarchy as the auxiliary information. The second step (Section 7.2.2) presents the clustering InfoNCE (Cl-InfoNCE) objective, a contrastive-learning-based approach, to leverage the constructed clusters. Last, in Section 7.2.3, we discuss the implications and provide the investigations for our approach. For notations, we use the upper case (e.g., X) letter to denote the random variable and the lower case (e.g., x) to denote the outcome from the random variable.

7.2.1 Cluster Construction for Discrete Attributes and Data Hierarchy Information

This sub-Section discusses how we construct data clusters according to auxiliary information. And in this chapter, we consider the data attributes and data hierarchy information as the auxiliary information. Note that the cluster constructions may differ with different types of auxiliary information. Below, we present our specific ways to determine data clusters according to our selected types of auxiliary information. We focus on providing overviews of our method, and more details can be found in our released code. We provide the illustration in Figure 7.2.

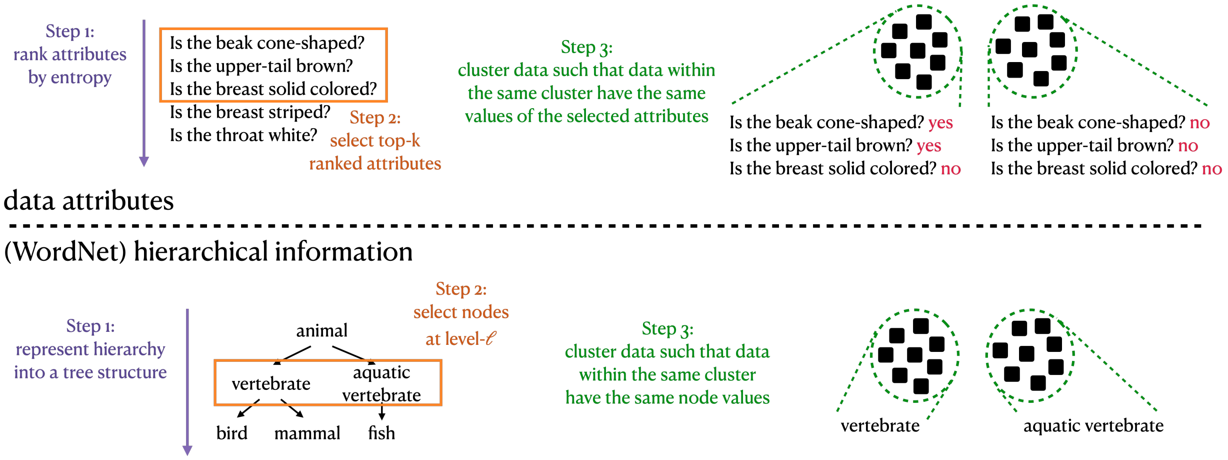

Step 1:
rank attributes
by entropy
Is the beak cone-shaped?
Is the upper-tail brown?
Is the breast solid colored?
Is the breast striped?
Is the throat white?
Step 2:
select top-k
ranked attributes
Step 3:
cluster data such that data within
the same cluster have the same
values of the selected attributes
Is the beak cone-shaped? yes
Is the upper-tail brown? yes
Is the breast solid colored? no
Is the beak cone-shaped? no
Is the upper-tail brown? no
Is the breast solid colored? no
data attributes
(WordNet) hierarchical information
Step 1:
represent hierarchy
into a tree structure
animal
vertebrate
aquatic
vertebrate
bird
mammal
fish
Step 2:
select nodes
at level-ℓ
Step 3:
cluster data such that data
within the same cluster
have the same node values
vertebrate
aquatic vertebrate

Figure 7.2: Cluster construction according to auxiliary information. We consider data attributes and (WordNet) hierarchical information as auxiliary information.

Clustering according to Discrete Attributes. We consider the discrete attributes as the first type of auxiliary information. An example of such auxiliary information is binary indicators of attributes, such as "short/long hair", "with/without sunglasses" or "short/long sleeves", for human photos. We construct the clusters such that data within each cluster will have the same values for a set of attributes. In our running example, if picking the set of attributes being hair and sunglasses, the human photos having both the "long hair" and "with sunglasses" will form a cluster. Then, how we determine the set of attributes? First, we rank each attribute according to its entropy in the dataset. Note that if an attribute has high entropy, it means this attribute is distributed diversely. Then, we select the attributes with top-k highest entropy, where k is a hyper-parameter.

Clustering according to Hierarchy Information. As the second type of auxiliary information, we consider hierarchy information - more specifically, the WordNet hierarchy [Miller, 1995]. The WordNet hierarchy describes the hierarchy information for data labels. For instance, assuming "human" and "mouse" as the labels, WordNet hierarchy suggests 1) "mammal" is the parent of "human" and "mouse"; and 2) "vertebrate" is the parent of "mammal". In this running example, "mammal" and "vertebrate" can be seen as the coarse labels of data, and we construct the clusters such that data within each cluster will have the same coarse label. Then, how we choose the coarse labels? We first represent the WordNet hierarchy into a tree structure (each children node has only one parent node). Then, we choose the coarse labels to be the nodes in the level l in the WordNet tree hierarchy (the root node is level 1). l is a hyper-parameter.

7.2.2 Clustering InfoNCE Objective

So far, we see how we determine the data clusters from discrete data attributes or data hierarchy information (as the auxiliary information). Now, we shall show how we integrate this clustering information into the self-supervised learning process. We note that most of the self-supervised learning approaches present to learn representations invariant to data augmentations [Caron et al., 2020, Chen et al., 2020a]. And on this basis, we present to learn representations that will also be similar for data with the same cluster assignment. To this end, we introduce the clustering InfoNCE (Cl-InfoNCE) objective, which is inspired by the InfoNCE objective [Oord et al., 2018] (which is widely used in conventional self-supervised representation learning). For a better presentation flow, we leave the discussion of InfoNCE later (in Section 7.2.3) but do not present it as a technical background first. We use the alphabets X and Y to denote the representations from augmented data:

$X = \text{Feature\_Encoder}\Big(\text{Augmentation\_1}\big(\text{Data\_1}\big)\Big)$ and $Y = \text{Feature\_Encoder}\Big(\text{Augmentation\_2}\big(\text{Data\_2}\big)\Big)$,

and the alphabet Z to denote the constructed clusters. Then, we formulate Cl-InfoNCE as

Proposition 12 (Clustering-based InfoNCE (Cl-InfoNCE))**.**

$$\mathrm{Cl}-\mathrm{InfoNCE} := \sup_f \mathbb{E}_{(x_i,y_i)\sim \mathbb{E}_{z\sim P_Z}\left[P_{X|z}P_{Y|z}\right]^{\otimes n}} \left[\frac{1}{n}\sum_{i=1}^{n} \log \frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n} e^{f(x_i,y_j)}}\right], \tag{7.1}$$

where $f(x,y)$ is any function that returns a scalar from the input (x,y). As suggested by prior work [Chen et al., 2020a, He et al., 2020], we choose $f(x,y) = \mathrm{cosine}\big(g(x), g(y)\big)/\tau$ to be the cosine similarity between non-linear projected $g(x)$ and $g(y)$. $g(\cdot)$ is a neural network (also known as the projection head [Chen et al., 2020a, He et al., 2020]) and τ is the temperature hyper-parameter. $\{(x_i, y_i)\}_{i=1}^n$ are n independent copies of $(x,y) \sim \mathbb{E}_{z\sim P_Z}\left[P_{X|z}P_{Y|z}\right]$, where it first samples a cluster $z \sim P_Z$ and then samples (x,y) pair with $x \sim P_{X|z}$ and $y \sim P_{Y|z}$. Furthermore, we call (x_i, y_i) as the positively-paired data (x_i and y_i have the same cluster assignment) and (x_i, y_j) $(i \neq j)$ as the negatively-paired data (x_i and y_j have independent cluster assignment). Note that, in practice, the expectation in eq. (7.1) is replaced by the empirical mean of a batch of samples.

Our objective is learning the representations X and Y (by updating the parameters in the feature encoder) to maximize Cl-InfoNCE. At a colloquial level, the maximization pulls towards the representations of the augmented data within the same cluster and push away the representations of the augmented data from different clusters. Theoretically, we present the following:

Theorem 13 (informal, Cl-InfoNCE maximization learns to include the clustering information)**.**

$$\begin{gathered}\mathrm{Cl}-\mathrm{InfoNCE} \leq D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right) \leq H(Z) \\ \text{and the equality holds only when } H(Z|X) = H(Z|Y) = 0,\end{gathered} \tag{7.2}$$

where $H(Z)$ is the entropy of Z and $H(Z|X)$ (or $H(Z|Y)$) are the conditional entropy of Z given X (or Y). Please find detailed derivations and proofs in Appendix.

The theorem suggests that Cl-InfoNCE has an upper bound $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right)$, which measures the distribution divergence between the product of clustering-conditional marginal distributions (i.e., $\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]$) and the product of marginal distributions (i.e., $P_X P_Y$). We give an intuition for $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right)$: if $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right)$ is high, then we can easily tell whether (x,y) have the same cluster assignment or not. The theorem also suggests that maximizing Cl-InfoNCE results in the representations X and Y including the clustering information Z ($\because H(Z|X) = H(Z|Y) = 0$).

7.2.3 Implications and Investigations

Goodness of the Learned Representations. In Theorem 13, we show that maximizing Cl-InfoNCE learns the representations (X and Y) to include the clustering (Z) information. Therefore, to characterize how good is the learned representations by maximizing Cl-InfoNCE, we can instead study the relations between Z and the downstream labels (denoting by T). In particular, we can use information-theoretical metrics such as the mutual information $I(Z;T)$ and the conditional entropy $H(Z|T)$ to characterize the goodness of the learned representations. $I(Z;T)$ measures how relevant the clusters and the labels, and $H(Z|T)$ measures how much redundant information in the clusters that are irrelevant to the labels. For instance, we can expect good downstream performance for our auxiliary-information-infused representations when having high mutual information and low conditional entropy between the auxiliary-information-determined clusters and the labels.

Generalization of Recent Self-supervised and Supervised Contrastive Approaches. Cl-InfoNCE (eq. (7.1)) serves as an objective that generalizes to different levels of supervision according to how we construct the clusters (Z). When $Z =$ instance id (i.e., each cluster only contains an instance), $\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]$ specializes to P_{XY} and Cl-InfoNCE specializes to the InfoNCE objective [Oord et al., 2018], which aims to learn similar representations for augmented variants of the same data and dissimilar representations for different data. InfoNCE is the most popular used self-supervised contrastive learning objective [Chen et al., 2020a, He et al., 2020, Tsai et al., 2021e]. When $Z =$ downstream labels, Cl-InfoNCE specializes to the objective described in *Supervised Contrastive Learning* [Khosla et al., 2020], which aims to learn similar representations for data that are from the same downstream labels and vice versa. In our chapter, the clusters Z are determined by the auxiliary information, and we aim to learn similar representations for data sharing the same auxiliary information and vice versa. This process can be understood as weakly supervised contrastive learning. To conclude, Cl-InfoNCE is a clustering-based contrastive learning objective. By differing its cluster construction, Cl-InfoNCE interpolates among unsupervised, weakly supervised, and supervised representation learning.

Advantages over Learning to Predict the Clusters Assignments. An alternative way to leverage the data clustering information is learning to predict the cluster assignment (Z) from the representations (X and Y). An example is learning to predict the hashtags for Instagram images [Mahajan et al., 2018], where the author shows that this prediction process serves as a good pre-training step. Nonetheless, comparing to our presented Cl-InfoNCE objective, learning to predict the cluster assignment requires building an additional classifier between the representations and the cluster. It will be non-ideal and inefficient to optimize this classifier when having a large number of clusters. The reason is that the number of the classifier's parameters is proportional to the number of clusters. An example is that, when $Z =$ instance id, the number of the clusters will be the total number of data, which can be billions. Learning to predict the clustering assignment may work poorly under this case, while InfoNCE (Cl-InfoNCE when $Z =$ instance id) can reach a good performance [Chen et al., 2020a]. Last, the most used objective for learning to predict the clusters is the cross-entropy loss. And evidences [Khosla et al., 2020] show that, compared to the cross-entropy loss, the contrastive objective (e.g., our presented Cl-InfoNCE) is more robust to natural corruptions of data and stable to hyper-parameters and optimizers settings.

7.3 Experiments

In the beginning, we discuss the datasets used in the chapter in Section 7.3.1. We consider either discrete attributes or data hierarchy information as auxiliary information for data. Then, in Section 7.3.2, we explain the methodology that will be used in the experiments. In Section 7.3.3, we present the first set of the experiments, which focuses on studying the presented Cl-InfoNCE objective (see Section 7.2.2) under conventional self-supervised setting. To this end, we consider unsupervised constructed clusters (e.g., k-means) along with Cl-InfoNCE. And we compare Cl-InfoNCE with other clustering-based self-supervised approaches. In Section 7.3.4 and 7.3.5, we further present experiments under the scenario when auxiliary information is available. We compare our method with the baseline approach - learning to predict the clustering assignment with cross-entropy loss. We also compare with conventional self-supervised representations and supervised representations.

7.3.1 Datasets

We consider the following datasets. **UT-zappos50K** [Yu and Grauman, 2014]: It contains $50,025$ shoes images along with 7 discrete attributes as auxiliary information. Each attribute follows a binomial distribution, and we convert each attribute into a set of Bernoulli attributes, resulting in a total of 126 binary attributes. There are 21 shoe categories. **Wider Attribute** [Li et al., 2016]: It contains $13,789$ images, and there are several bounding boxes in an image. The attributes are annotated per bounding box. We perform OR operation on attributes from different bounding boxes in an image, resulting in 14 binary attributes per image as the auxiliary information. There are 30 scene categories. **CUB-200-2011** [Wah et al., 2011]: It contains $11,788$ bird images with 312 binary attributes as the auxiliary information. There are 200 bird species. **ImageNet-100** [Russakovsky et al., 2015]: It is a subset of the ImageNet-1k object recognition dataset [Russakovsky et al., 2015], where we select 100 categories out of $1,000$, resulting in around 0.12 million images. We consider WordNet hierarchy information as the auxiliary information.

7.3.2 Methodology

Following Chen et al. [2020a], we conduct experiments on pre-training visual representations and then evaluating the learned representations using the linear evaluation protocol. In precise, after the pre-training stage, we fix the pre-trained feature encoder and then categorize test images by linear classification results. We select ResNet-50 [He et al., 2016] as our feature encoder across all settings. Note that our goal is learning representations (i.e, X and Y) for maximizing the Cl-InfoNCE objective (equation (7.1)). Within Cl-InfoNCE, the positively-paired representations $(x, y^+) \sim \mathbb{E}_{z\sim P_Z}\big[P_{X|z}P_{Y|z}\big]$ are the learned representations from augmented images from the same cluster $z \sim P_Z$ and the negatively-paired representations $(x, y^-) \sim P_X P_Y$ are the representations from arbitrary two images. We leave the network designs, the optimizer choices, and more details for the datasets in Appendix.

Before delving into the experiments, we like to recall that, in Section 7.2.3, we discussed using the mutual information $I(Z;T)$ and the conditional entropy $H(Z|T)$ between the clusters (Z) and the labels (T) to characterize the goodness of Cl-InfoNCE's learned representations. To prove this concept, on UT-Zappos50K, we synthetically construct clusters for various $I(Z;T)$ and $H(Z|T)$ followed by applying Cl-InfoNCE. We present the results in the right figure. Our empirical results are in accordance with the statements that the clusters with higher $I(Z;T)$ and lower $H(Z|T)$ will lead to higher downstream performance. In later experiments, we will also discuss these two information-theoretical metrics.

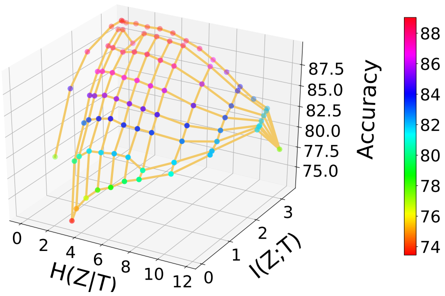


Figure 7.3: $I(Z;T)$ represents how relevant the clusters and the labels; higher is better. $H(Z|T)$ represents the redundant information in the clusters for the labels; lower is better.

7.3.3 Experiment I: K-means Clusters + Cl-InfoNCE

We study how Cl-InfoNCE can learn good self-supervised representations even without auxiliary information. To this end, we construct unsupervised clusters (e.g., k-means clusters on top of the learned representations) for Cl-InfoNCE. Similar to the EM algorithm, we iteratively perform the k-means clustering to determine the clusters for the representations, and then we adopt Cl-InfoNCE to leverage the k-means clusters to update the representations. We select the Prototypical Contrastive Learning (PCL) [Li et al., 2020a] as the baseline of the clustering-based self-supervised approach. In particular, PCL performs data log-likelihood maximization by assuming data are generated from isotropic Gaussians. It considers

Method	UT-Zappos50K Top-1 (Accuracy)	Wider Attribute Top-1 (Accuracy)	CUB-200-2011 Top-1 (Accuracy)	ImageNet-100 Top-1 (Accuracy)
Non-clustering-based Self-supervised Approaches				
SimCLR [Chen et al., 2020a]	77.8±1.5	40.2±0.9	14.1±0.7	58.2±1.7
MoCo [He et al., 2020]	83.4±0.5	41.0±0.7	13.8±0.5	59.4±1.6
Clustering-based Self-supervised Approaches (# of clusters = 1K/ 1K/ 1K/ 2.5K)				
PCL [Li et al., 2020a]	82.4±0.5	41.0±0.4	14.4±0.5	68.9±0.7
K-means + Cl-InfoNCE (ours)	**84.5±0.4**	**43.6±0.4**	**17.6±0.2**	**77.9±0.7**

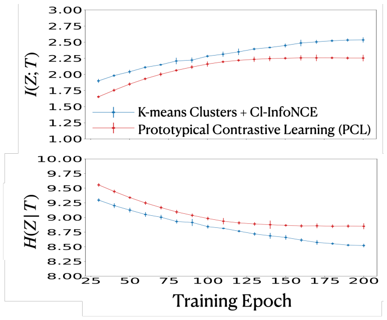


Figure 7.4: Experimental results under conventional self-supervised setting (pre-training using no label supervision and no auxiliary information). **Left:** We compare our method (K-means clusters + Cl-InfoNCE) with self-supervised approaches that leverage and do not consider unsupervised clustering. The downstream performance is reported using the linear evaluation protocol [Chen et al., 2020a]. **Right:** For our method and Prototypical Contrastive Learning (PCL), we plot the mutual information ($I(Z;T)$) and the conditional entropy ($H(Z|T)$) versus training epochs. Z are the unsupervised clusters, and T are the downstream labels.

the MLE objective, where the author makes a connection with contrastive approaches [Chen et al., 2020a, He et al., 2020]. The clusters in PCL are determined via MAP estimation. For the sake of the completeness of the experiments, we also include the non-clustering-based self-supervised approaches, including SimCLR [Chen et al., 2020a] and MoCo [He et al., 2020]. Note that this set of experiments considers the conventional self-supervised setting, in which we can leverage the information neither from labels nor from auxiliary information.

Results. We first look at the left table in Figure 7.4. We observe that, except for ImageNet-100, there is no obvious performance difference between the non-clustering-based (i.e., SimCLR and MoCo) and the clustering-based baseline (i.e., PCL). Since ImageNet-100 is a more complex dataset comparing to the other three datasets, we argue that, when performing self-supervised learning, discovering latent structures in data (via unsupervised clustering) may best benefit larger-sized datasets. Additionally, among all the approaches, our method reaches the best performance. The result suggests our method can be as competitive as other conventional self-supervised approaches.

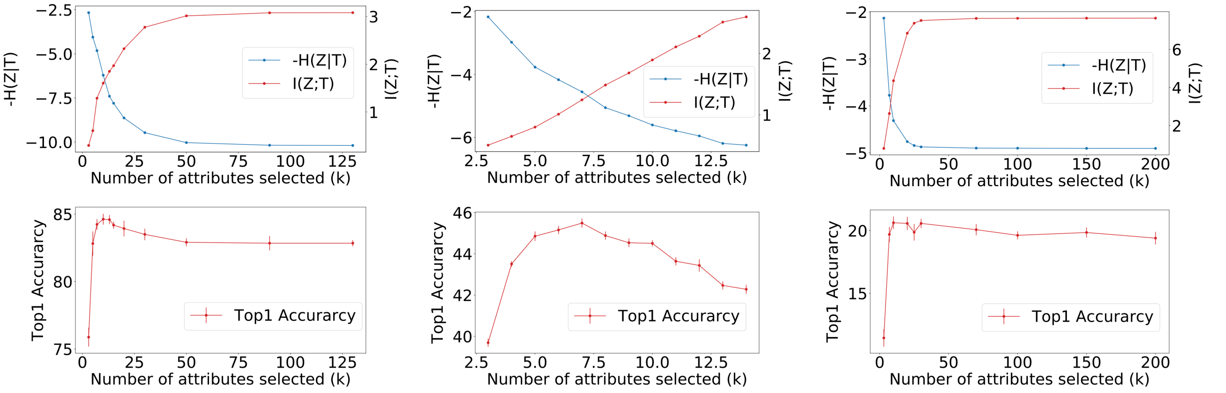
Next, we look at the right plot in Figure 7.4. We study the mutual information $I(Z;T)$ and the conditional entropy $H(Z|T)$ between the unsupervised constructed clusters Z and the downstream labels T. We select our method and PCL, providing the plot of the two information-theoretical metrics versus the training epoch. We find that, as the number of training epochs increases, both methods can construct unsupervised clusters that are more relevant (higher $I(Z;T)$) and contain less redundant information (lower $H(Z|T)$) about the downstream label. This result suggests that the clustering-based self-supervised approaches are discovering the latent structures that are more useful for the downstream tasks. It is worth noting that our method consistently has higher $I(Z;T)$ and lower $H(Z|T)$ comparing to PCL.

7.3.4 Experiment II: Data-Attributes-Determined Clusters + Cl-InfoNCE

We like to understand how well Cl-InfoNCE can be combined with the auxiliary information. For this purpose, we select the data discrete attributes as the auxiliary information, construct the clusters (Z) using the discrete attributes (see Section 7.2.1 and Figure 7.2), and then adopt attributes-determined clusters for Cl-InfoNCE. Recall our construction of data-attributes-determined clusters: we select the attributes with top-k highest entropy and then construct the clusters such that the data within a cluster will have the same values over the selected attributes. k is the hyper-parameter. Note that our method considers a weakly supervised setting since the data attributes can be seen as the data's weak supervision. For the completeness

Method (Contrastive Learning† / Predictive Learning‡)	UT-Zappos50K		Wider Attribute		CUB-200-2011	
	Top-1 Acc.	Top-5 Acc.	Top-1 Acc.	Top-5 Acc.	Top-1 Acc.	Top-5 Acc.
Supervised Representation Learning (Z = downstream labels T)						
‡Cross-Entropy Loss	89.2±0.5	99.6±0.4	44.7±1.5	71.2± 0.5	60.5±1.2	81.7±0.7
†(Labels + Cl-InfoNCE) SupCon [Khosla et al., 2020]	89.0±0.4	99.4± 0.3	49.9±0.8	76.2±0.2	59.9±0.7	78.8± 0.3
Weakly Supervised Representation Learning (Z = attributes-determined clusters)						
‡Cross-Entropy Loss	82.7±0.7	99.04±0.3	39.4±0.6	68.6±0.2	17.5±1.0	46.0±0.8
†Attributes-Determined Clusters + Cl-InfoNCE (ours)	84.6±0.4	99.1±0.2	45.5±0.2	75.4±0.2	20.6± 0.5	47.0±0.5
Self-supervised Representation Learning (Z = instance id)						
†MoCo [He et al., 2020]	83.4±0.2	99.1±0.3	41.03±0.7	74.0±0.4	13.8±0.7	36.5±0.5
†(Instance-ID + Cl-InfoNCE) SimCLR [Chen et al., 2020a]	77.8±1.0	97.9±0.8	40.2±0.9	73.0±0.3	14.1± 0.7	35.2±0.6

Table 7.1: Experimental results under supervised (pre-training using label supervision), weakly supervised (pre-training using data attributes), and conventional self-supervised (pre-training using neither label supervision nor data attributes) setting. Each setting refers to a particular cluster (Z) construction. The methods presented in this table are either contrastive or predictive learning approaches. We report the best results for weakly supervised methods by tuning the hyper-parameter k for attributes-determined clusters.



(a) UT-zappos50K (b) Wider Attribute (c) CUB-200-2011

Figure 7.5: Experimental results for attributes-determined clusters + Cl-InfoNCE by tuning the hyper-parameter k when constructing the clusters. Note that we select attributes with top-k highest entropy, and we construct the clusters such that the data within a cluster would have the same values for the selected attributes. Z are the constructed clusters, and T are the downstream labels. We find the intersection between the mutual information ($I(Z;T)$) and the negative conditional entropy ($-H(Z|T)$) gives us the best downstream performance.

of the experiments, we include the comparisons with the supervised (Z = downstream labels T) and the conventional self-supervised (Z = instance ID) setting for our method. We show in Section 7.2.3, the supervised setting is equivalent to the Supervised Contrastive Learning objective [Khosla et al., 2020] and the conventional self-supervised setting is equivalent to SimCLR [Chen et al., 2020a]. We also include another baseline that leverages the data clustering information - learning to predict the clusters assignments using cross-entropy loss.

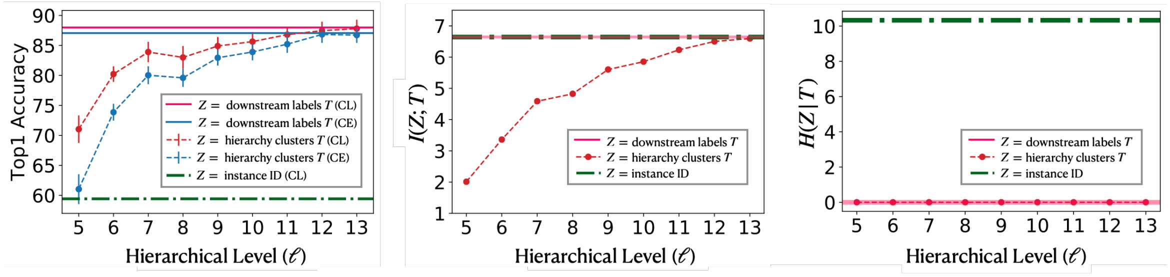
Results. Table 7.1 presents our results. First, we compare different cluster constructions along with Cl-InfoNCE and use the top-1 accuracy on Wider Attribute for discussions. We find the performance grows from low to high when having the clusters as instance ID (40.2), attributes-determined clusters (45.5) to labels (49.9). This result suggests that CL-InfoNCE can better bridge the gap with the supervised learned representations by using auxiliary information. Second, we find that using auxiliary information does not always guarantee better performance than not using it. For instance, predicting the attributes-determined clusters using the cross-entropy loss (39.4) performs worse than the SimCLR method (40.2), which does not utilize the auxiliary information. Hence, how to effectively leverage the auxiliary information is crucial. Third, we observe the predictive method always performs worse than the contrastive method under the weakly supervised setting. For example, on UT-Zappos50K, although predicting the labels using the cross-entropy loss (89.2) performs at par with SupCon (89.0), predicting attributes-determined clusters using the cross-entropy loss (82.7) performs worse than attributes-determined clusters + Cl-InfoNCE (84.6). This result implies that the contrastive method (e.g., Cl-InfoNCE) can generally be applied across various supervision levels.

To better understand the effect of the hyper-parameter k for constructing the attributes-determined clusters, we study the information-theoretical metrics between Z and T and report in Figure 7.5. First, as k increases, the mutual information $I(Z;T)$ increases but the conditional entropy $H(Z|T)$ also increases. Hence, although considering more attributes leads to the clusters that are more correlated to the downstream labels, the clusters may also contain more downstream-irrelevant information. This is in accord with our second observation that, as k increases, the downstream performance first increases then decreases. Therefore, we only need a partial set of the most informative attributes (those with high entropy) to determine the clusters. Our last observation is that the best performing clusters happen at the intersection between $I(Z;T)$ and negative $H(Z|T)$. This observation helps us study the trade-off between $I(Z;T)$ and $H(Z|T)$ and suggests that the clusters, when used for Cl-InfoNCE, having the highest $I(Z;T) - H(Z|T)$ could achieve the best performance.

7.3.5 Experiment III: Data-Hierarchy-Determined Clusters + Cl-InfoNCE

The experimental setup and the comparing baselines are similar to Section 7.3.4, but now we consider the WordNet [Miller, 1995] hierarchy as the auxiliary information. As discussed in Section 7.2.1 and Figure 7.2, we construct the clusters Z such that the data within a cluster have the same parent node in the level l in the data's WordNet tree hierarchy. l is the hyper-parameter.

Results. Figure 7.6 presents our results. First, we look at the leftmost plot, and we have several similar observations when having the data attributes as the auxiliary information. One of them is that the contrastive method consistently outperforms the predictive method. Another of them is that the weakly supervised representations better close the gap with the supervised representations. Second, as discussed in Section 7.2.1, the WordNet data hierarchy clusters can be regarded as the coarse labels of the data. Hence, when increasing the hierarchy level l, we can observe the performance improvement (see the leftmost plot) and the increasing mutual information $I(Z;T)$ (see the middle plot) between the clusters Z and the labels T. Note that $H(Z|T)$ remains zero (see the rightmost plot) since the coarse labels (the intermediate nodes) can be determined by the downstream labels (the leaf nodes) under the tree hierarchy structure. Third, we discuss the conventional self-supervised setting with the special case when $Z =$ instanced ID. Z as the instance ID has the highest $I(Z;T)$ (see the middle plot) but also the highest $H(Z|T)$ (see the rightmost plot). And we observe that the conventional self-supervised representations perform the worse (see the leftmost plot). We conclude that, when using cluster-based representation learning approaches, we shall not rely purely on the mutual information between the data clusters and the downstream labels to determine the goodness of the learned representations. We shall also take the redundant information in the clusters into account.



CL represents Cl-InfoNCE and **CE** represents learning to predict the cluster assignments using Cross-Entropy loss)

Figure 7.6: Experimental results on ImageNet-100 for Cl-InfoNCE under supervised (clusters Z = downstream labels T), weakly supervised (Z = hierarchy clusters) and conventional self-supervised (Z = instance ID) setting. We also consider the baseline - learning to predict the clustering assignment using the cross-entropy loss. Note that we construct the clusters such that the data within a cluster have the same parent node in the level ℓ in the data's WordNet tree hierarchy. Under this construction, the root node is of the level 1, and the downstream labels are of the level 14. $I(Z;T)$ is the mutual information, and $H(Z|T)$ is the conditional entropy.

7.4 Discussion

In this chapter, we present to integrate auxiliary information of data into the self-supervised learning process. We first construct data clusters according to auxiliary information. Then, we introduce the clustering InfoNCE (Cl-InfoNCE) objective to leverage the built clusters. Our method brings the performance closer to the supervised learned representations compared to the conventional self-supervised learning approaches. Moreover, even without auxiliary information, Cl-InfoNCE can work with unsupervised K-means clusters as a strong method under the conventional self-supervised learning setting. We believe this work sheds light on the advantage of exploiting 1) noisy but cheap-to-collect sources of information in the wild and 2) data structure information for learning better representations.

Limitations. Our approach requires determining data clusters from auxiliary information. In this chapter, we present different data cluster construction methods for discrete attributes and data hierarchy information. Nonetheless, some types of auxiliary information may be highly unstructured. And determining the clusters according to such auxiliary information may require additional effort. For instance, if having continuous attributes as auxiliary information, binning or quantization cannot be avoided when constructing the clusters.

Negative Social Impacts. Certain auxiliary information may contain private information. For example, in medical applications, physical conditions as auxiliary information may reveal a person's identity. Therefore, we should be careful in choosing auxiliary information for privacy concerns.

7.5 Appendix

7.5.1 Theoretical Analysis

In this section, we provide theoretical analysis on the presented Cl-InfoNCE objective. We recall the proposition of Cl-InfoNCE and our presented theorem:

Proposition 13 (Clustering-based InfoNCE (Cl-InfoNCE), restating Proposition 12 in the main text)**.**

$$\mathrm{Cl-InfoNCE} := \sup_f \mathbb{E}_{(x_i,y_i)\sim \mathbb{E}_{z\sim P_Z}\left[P_{X|z}P_{Y|z}\right]^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right],$$

Theorem 14 (informal, Cl-InfoNCE maximization learns to include the clustering information, restating Theorem 13 in the main text)**.**

$$\mathrm{Cl-InfoNCE} \leq D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right) \leq H(Z)$$
$$\text{and the equality holds only when } H(Z|X) = H(Z|Y) = 0.$$

Our goal is to prove Theorem 14. For a better presentation flow, we split the proof into three parts:

- Proving $\mathrm{Cl-InfoNCE} \leq D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right)$ in Section 7.5.1.1
- Proving $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right) \leq H(Z)$ in Section 7.5.1.2
- Proving $\mathrm{Cl-InfoNCE}$ maximizes at $H(Z)$ when $H(Z|X) = H(Z|Y) = 0$ in Section 7.5.1.3

7.5.1.1 Part I - Proving $\mathrm{Cl-InfoNCE} \leq D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right)$

The proof requires the following lemma.

Lemma 11 (Theorem 1 by Song and Ermon [2020])**.** Let $\mathcal{X}$ and $\mathcal{Y}$ be the sample spaces for X and Y, f be any function: $(\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$, and $\mathcal{P}$ and $\mathcal{Q}$ be the probability measures on $\mathcal{X} \times \mathcal{Y}$. Then,

$$\sup_f \mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\log \frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right] \leq D_{\mathrm{KL}}\left(\mathcal{P} \parallel \mathcal{Q}\right).$$

Now, we are ready to prove the following lemma:

Lemma 12 (Proof Part I)**.** $\mathrm{Cl-InfoNCE} := \sup_f \mathbb{E}_{(x_i,y_i)\sim\mathbb{E}_{z\sim P_Z}\left[P_{X|z}P_{Y|z}\right]^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^n \log \frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x_i,y_j)}}\right] \leq D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right)$.

Proof. By defining $\mathcal{P} = \mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]$ and $\mathcal{Q} = P_X P_Y$, we have

$$\mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\log \frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right] = \mathbb{E}_{(x_i,y_i)\sim\mathbb{E}_{z\sim P_Z}\left[P_{X|z}P_{Y|z}\right]^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^n \log \frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x_i,y_j)}}\right].$$

Plug in this result into Lemma 11 and we conclude the proof. ■

7.5.1.2 Part II - Proving $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right) \leq H(Z)$

The proof requires the following lemma:

Lemma 13. $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \parallel P_X P_Y\right) \leq \min\left\{\mathrm{MI}(Z;X), \mathrm{MI}(Z;Y)\right\}$.

Proof.

$$
\begin{aligned}
&\mathrm{MI}(Z;X) - D_{\mathrm{KL}}\Big(\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big] \,\|\, P_X P_Y\Big)\\
&= \int_z p(z) \int_x p(x|z) \,\log\, \frac{p(x|z)}{p(x)} \mathrm{d}x\mathrm{d}z - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{\int_{z'} p(z')p(x|z')p(y|z')\mathrm{d}z'}{p(x)p(y)} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&= \int_z p(z) \int_x p(x|z) \,\log\, \frac{p(x|z)}{p(x)} \mathrm{d}x\mathrm{d}z - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{\int_{z'} p(z'|y)p(x|z')\mathrm{d}z'}{p(x)} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{p(x|z)}{\int_{z'} p(z'|y)p(x|z')\mathrm{d}z'} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&= - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{\int_{z'} p(z'|y)p(x|z')\mathrm{d}z'}{p(x|z)} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&\geq - \int_z p(z) \int_x p(x|z) \int_y p(y|z) \left(\frac{\int_{z'} p(z'|y)p(x|z')\mathrm{d}z'}{p(x|z)} - 1\right) \mathrm{d}x\mathrm{d}y\mathrm{d}z \,\Big(\because \log t \leq t-1\Big)\\
&= 0.
\end{aligned}
$$

Hence, $\mathrm{MI}(Z;X) \geq D_{\mathrm{KL}}\Big(\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big] \,\|\, P_X P_Y\Big)$. Likewise, $\mathrm{MI}(Z;Y) \geq D_{\mathrm{KL}}\Big(\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big] \,\|\, P_X P_Y\Big)$. We complete the proof by combining the two results. ■

Now, we are ready to prove the following lemma:

Lemma 14 (Proof Part II)**.** $D_{\mathrm{KL}}\Big(\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big] \,\|\, P_X P_Y\Big) \leq H(Z)$.

Proof. Combining Lemma 13 and the fact that $\min\Big\{\mathrm{MI}(Z;X), \mathrm{MI}(Z;Y)\Big\} \leq H(Z)$, we complete the proof. Note that we consider Z as the clustering assignment, which is discrete but not continuous. And the inequality holds for the discrete Z, but may not hold for the continuous Z. ■

7.5.1.3 Part III - Proving $\mathrm{Cl-InfoNCE}$ maximizes at $H(Z)$ when $H(Z|X) = H(Z|Y) = 0$

We directly provide the following lemma:

Lemma 15 (Proof Part III)**.** $\mathrm{Cl-InfoNCE}$ max. at $H(Z)$ when $H(Z|X) = H(Z|Y) = 0$.

Proof. When $H(Z|Y) = 0$, $p(Z|Y = y)$ is Dirac. The objective

$$
\begin{aligned}
&D_{\mathrm{KL}}\Big(\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big] \,\|\, P_X P_Y\Big)\\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{\int_{z'} p(z')p(x|z')p(y|z')\mathrm{d}z'}{p(x)p(y)} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{\int_{z'} p(z'|y)p(x|z')\mathrm{d}z'}{p(x)} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{\int_{z'} p(z')p(x|z')p(y|z')\mathrm{d}z'}{p(x)p(y)} \mathrm{d}x\mathrm{d}y\mathrm{d}z\\
&= \int_z p(z) \int_x p(x|z) \int_y p(y|z) \,\log\, \frac{p(x|z)}{p(x)} \mathrm{d}x\mathrm{d}y\mathrm{d}z = \mathrm{MI}\Big(Z;X\Big).
\end{aligned}
$$

The second-last equality comes with the fact that: when $p(Z|Y=y)$ is Dirac, $p(z'|y)=1\ \forall z'=z$ and $p(z'|y)=0\ \forall z'\neq z$. Combining with the fact that $\mathrm{MI}\Big(Z;X\Big)=H(Z)$ when $H(Z|X)=0$, we know $D_{\mathrm{KL}}\left(\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right] \,\|\, P_XP_Y\right)=H(Z)$ when $H(Z|X)=H(Z|Y)=0$.

Furthermore, by Lemma 12 and Lemma 14, we complete the proof. ■

7.5.1.4 Bringing Everything Together

We bring Lemmas 12, 14, and 15 together and complete the proof of Theorem 14.

7.5.2 Algorithms

In this section, we provide algorithms for our experiments. We consider two sets of the experiments. The first one is K-means clusters + Cl-InfoNCE (see Section 7.3.3 in the main text), where the clusters involved in Cl-InfoNCE are iteratively obtained via K-means clustering on top of data representations. The second one is auxiliary-information-determined clusters + Cl-InfoNCE (see Section 7.3.4 and 7.3.5 in the main text), where the clusters involved in Cl-InfoNCE are pre-determined accordingly to data attributes (see Section 7.3.4) or data hierarchy information (see Section 7.3.5).

K-means clusters + Cl-InfoNCE We present here the algorithm for K-means clusters + Cl-InfoNCE. At each iteration in our algorithm, we perform K-means Clustering algorithm on top of data representations for obtaining cluster assignments. The cluster assignment will then be used in our Cl-InfoNCE objective.

Algorithm 1: K-means Clusters + Cl-InfoNCE

```
Result: Pretrained Encoder $f_\theta(\cdot)$
$f_\theta(\cdot)$ ← Base Encoder Network;
Aug $(\cdot)$ ← Obtaining Two Variants of Augmented Data via Augmentation Functions;
Embedding ← Gathering data representations by passing data through $f_\theta(\cdot)$;
Clusters ←K-means-clustering(Embedding);
for epoch in 1,2,...,N do
    for batch in 1,2,...,M do
        data1, data2 ← Aug(data_batch);
        feature1, feature2 ← $f_\theta$(data1), $f_\theta$(data2);
        $L_{\text{Cl-infoNCE}}$ ← Cl-InfoNCE(feature1, feature2, Clusters);
        $f_\theta \leftarrow f_\theta - lr * \frac{\partial}{\partial\theta} L_{\text{Cl-infoNCE}}$;
    end
    Embedding ← gather embeddings for all data through $f_\theta(\cdot)$;
    Clusters ←K-means-clustering(Embedding);
end
```

Auxiliary information determined clusters + Cl-InfoNCE We present the algorithm to combine auxiliary-information-determined clusters with Cl-InfoNCE. We select data attributes or data hierarchy information as the auxiliary information, and we present their clustering determining steps in Section 7.2.1

in the main text.

Algorithm 2: Pre-Determined Clusters + Cl-InfoNCE

```
Result: Pretrained Encoder $f_\theta(\cdot)$
$f_\theta(\cdot)$ ← Base Encoder Network;
Aug $(\cdot)$ ← Obtaining Two Variants of Augmented Data via Augmentation Functions;
Clusters ←Pre-determining Data Clusters from Auxiliary Information;
for epoch in 1,2,...,N do
    for batch in 1,2,...,M do
        data1, data2 ← Aug(data_batch);
        feature1, feature2 ← $f_\theta$(data1), $f_\theta$(data2);
        $L_{\text{Cl-infoNCE}}$ ← Cl-InfoNCE(feature1, feature2, Clusters);
        $f_\theta \leftarrow f_\theta - lr * \frac{\partial}{\partial\theta} L_{\text{Cl-infoNCE}}$;
    end
end
```

7.5.3 Experimental details

The following content describes our experiments settings in details. For reference, our code is available at https://anonymous.4open.science/r/Cl-InfoNCE-02AB/README.md.

7.5.3.1 UT-Zappos50K

The following section describes the experiments we performed on UT-Zappos50K dataset in Section 7.3 in the main text.

Accessiblity The dataset is attributed to [Yu and Grauman, 2014] and available at the link: http://vision.cs.utexas.edu/projects/finegrained/utzap50k. The dataset is for non-commercial use only.

Data Processing The dataset contains images of shoe from Zappos.com. We rescale the images to 32×32. The official dataset has 4 large categories following 21 sub-categories. We utilize the 21 subcategories for all our classification tasks. The dataset comes with 7 attributes as auxiliary information. We binarize the 7 discrete attributes into 126 binary attributes. We rank the binarized attributes based on their entropy and use the top-k binary attributes to form clusters. Note that different k result in different data clusters (see Figure 7.5 (a) in the main text).

Training and Test Split: We randomly split train-validation images by $7:3$ ratio, resulting in $35,017$ train data and $15,008$ validation dataset.

Network Design We use ResNet-50 architecture to serve as a backbone for encoder. To compensate the 32x32 image size, we change the first 7x7 2D convolution to 3x3 2D convolution and remove the first max pooling layer in the normal ResNet-50 (See code for detail). This allows finer grain of information processing. After using the modified ResNet-50 as encoder, we include a 2048-2048-128 Multi-Layer Perceptron (MLP) as the projection head $\Big($i.e., $g(\cdot)$ in $f(\cdot,\cdot)$ equation 7.1 in the main text$\Big)$ for Cl-InfoNCE. During evaluation, we discard the projection head and train a linear layer on top of the encoder's output. For both K-means clusters + Cl-InfoNCE and auxiliary-information-determined clusters + Cl-InfoNCE, we adopt the same network architecture, including the same encoder, the same MLP projection head and the same linear evaluation protocol. In the K-means + Cl-InfoNCE settings, the number of the K-means clusters is $1,000$. Kmeans clustering is performed every epoch during training. We find performing Kmeans for every epoch benefits the performance. For fair comparsion, we use the same network architecture and cluster number for PCL.

Optimization We choose SGD with momentum of 0.95 for optimizer with a weight decay of 0.0001 to prevent network over-fitting. To allow stable training, we employ a linear warm-up and cosine decay scheduler for learning rate. For experiments shown in Figure 7.5 (a) in the main text, the learning rate is set to be 0.17 and the temperature is chosen to be 0.07 in Cl-InfoNCE. And for experiments shown in Figure 7.4 in the main text, learning rate is set to be 0.1 and the temperature is chosen to be 0.1 in Cl-InfoNCE.

Computational Resource We conduct experiments on machines with 4 NVIDIA Tesla P100. It takes about 16 hours to run 1000 epochs of training with batch size 128 for both auxiliary information aided and unsupervised Cl-InfoNCE.

7.5.3.2 Wider Attributes

The following section describes the experiments we performed on Wider Attributes dataset in Section 7.3 in the main text.

Accessiblity The dataset is credited to [Li et al., 2016] and can be downloaded from the link: `http://mmlab.ie.cuhk.edu.hk/projects/WIDERAttribute.html`. The dataset is for public and non-commercial usage.

Data Processing The dataset contains $13,789$ images with multiple semantic bounding boxes attached to each image. Each bounding is annotated with 14 binary attributes, and different bounding boxes in an image may have different attributes. Here, we perform the OR operation among the attributes in the bounding boxes in an image. Hence, each image is linked to 14 binary attributes. We rank the 14 attributes by their entropy and use the top-k of them when performing experiments in Figure 7.5 (b) in the main text. We consider a classification task consisting of 30 scene categories.

Training and Test Split: The dataset comes with its training, validation, and test split. Due to a small number of data, we combine the original training and validation set as our training set and use the original test set as our validation set. The resulting training set contains $6,871$ images and the validation set contains $6,918$ images.

Computational Resource To speed up computation, on Wider Attribute dataset we use a batch size of 40, resulting in 16-hour computation in a single NVIDIA Tesla P100 GPU for $1,000$ epochs training.

Network Design and Optimization We use ResNet-50 architecture as an encoder for Wider Attributed dataset. We choose 2048-2048-128 MLP as the projection head $\Big($i.e., $g(\cdot)$ in $f(\cdot,\cdot)$ equation (7.1) in the main text$\Big)$ for Cl-InfoNCE. The MLP projection head is discarded during the linear evaluation protocol. Particularly, during the linear evaluation protocol, the encoder is frozen and a linear layer on top of the encoder is fine-tuned with downstream labels. For Kmeans + Cl-InfoNCE and Auxiliary information + Cl-InfoNCE, we consider the same architectures for the encoder, the MLP head and the linear evaluation classifier. For K-means + Cl-InfoNCE, we consider $1,000$ K-means clusters. For fair comparsion, the same network architecture and cluster number is used for experiments with PCL.

For Optimization, we use SGD with momentum of 0.95. Additionally, 0.0001 weight decay is adopted in the network to prevent over-fitting. We use a learning rate of 0.1 and temperature of 0.1 in Cl-InfoNCE for all experiments. A linear warm-up following a cosine decay is used for the learning rate scheduling, providing a more stable learning process.

7.5.3.3 CUB-200-2011

The following section describes the experiments we performed on CUB-200-2011 dataset in Section 7.3 in the main text.

Accessiblity CUB-200-2011 is created by Wah et al. [2011] and is a fine-grained dataset for bird species. It can be downloaded from the link: `http://www.vision.caltech.edu/visipedia/CUB-200-2011.html`. The usage is restricted to non-commercial research and educational purposes.

Data Processing The original dataset contains 200 birds categories over $11,788$ images with 312 binary attributes attached to each image. We utilize those attributes and rank them based on their entropy. In Figure 7.5 (c), we use the top-k of those attributes to constrcut clusters with which we perform in Cl-InfoNCE. The image is rescaled to 224×224.

Train Test Split: We follow the original train-validation split, resulting in $5,994$ train images and $5,794$ validation images.

Computational Resource It takes about 8 hours to train for 1000 epochs with 128 batch size on 4 NVIDIA Tesla P100 GPUs.

Network Design and Optimization We choose ResNet-50 for CUB-200-2011 as the encoder. After extracting features from the encoder, a 2048-2048-128 MLP projection head $\Big($i.e., $g(\cdot)$ in $f(\cdot,\cdot)$ equation (7.1) in the main text$\Big)$ is used for Cl-InfoNCE. During the linear evaluation protocal, the MLP projection head is removed and the features extracted from the pre-trained encoder is fed into a linear classifier layer. The linear classifier layer is fine-tuned with the downstream labels. The network architectures remain the same for both K-means clusters + Cl-InfoNCE and auxiliary-information-determined clusters + Cl-InfoNCE settings. In the K-means clusters + Cl-InfoNCE settings, we consider $1,000$ K-means clusters. For fair comparsion, the same network architecture and cluster number is used for experiments with PCL.

SGD with momentum of 0.95 is used during the optimization. We select a linear warm-up following a cosine decay learning rate scheduler. The peak learning rate is chosen to be 0.1 and the temperature is set to be 0.1 for both K-means + Cl-InfoNCE and Auxiliary information + Cl-InfoNCE settings.

7.5.3.4 ImageNet-100

The following section describes the experiments we performed on ImageNet-100 dataset in Section 7.3 in the main text.

Accessibility This dataset is a subset of ImageNet-1K dataset, which comes from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012-2017 [Russakovsky et al., 2015]. ILSVRC is for non-commercial research and educational purposes and we refer to the ImageNet official site for more information: `https://www.image-net.org/download.php`.

Data Processing In the Section 7.3.5 in the main text, we select 100 classes from ImageNet-1K to conduct experiments (the selected categories can be found in `https://anonymous.4open.science/r/Cl-InfoNCE-02AB/data_processing/imagenet100/selected_100_classes.txt`). We also conduct a slight pre-processing (via pruning a small number of edges in the WordNet graph) on the WordNet hierarchy structure to ensure it admits a tree structure. Specifically, each of the selected categories and their ancestors only have one path to the root. We refer the pruning procedure in `https://anonymous.4open.science/r/Cl-InfoNCE-02AB/data_processing/imagenet100/hierarchy_processing/imagenet_hierarchy.py` (line 222 to 251).

We cluster data according to their common ancestor in the pruned tree structure and determine the level l of each cluster by the step needed to traverse from root to that node in the pruned tree. Therefore, the larger the l, the closer the common ancestor is to the real class labels, hence more accurate clusters will be formed. Particularly, the real class labels is at level 14.

Training and Test Split: Please refer to the following file for the training and validation split.

- training: `https://anonymous.4open.science/r/Cl-InfoNCE-02AB/data_processing/imagenet100/hier/meta_data_train.csv`
- validation: `https://anonymous.4open.science/r/Cl-InfoNCE-02AB/data_processing/imagenet100/hier/meta_data_val.csv`

The training split contains $128,783$ images and the test split contains $5,000$ images. The images are rescaled to size 224×224.

Computational Resource It takes 48-hour training for 200 epochs with batch size 128 using 4 NVIDIA Tesla P100 machines. All the experiments on ImageNet-100 is trained with the same batch size and number of epochs.

Network Design and Optimization Hyper-parameters We use conventional ResNet-50 as the backbone for the encoder. 2048-2048-128 MLP layer and $l2$ normalization layer is used after the encoder during training and discarded in the linear evaluation protocal. We maintain the same architecture for Kmeans + Cl-InfoNCE and auxiliary information aided Cl-InfoNCE. For Kmeans + Cl-InfoNCE, we choose 2500 as the cluster number. For fair comparsion, the same network architecture and cluster number is used for experiments with PCL. The Optimizer is SGD with 0.95 momentum. For K-means + Cl-InfoNCE used in Figure 7.4 in the main text, we use the learning rate of 0.03 and the temperature of 0.2. We use the learning rate of 0.1 and temperature of 0.1 for auxiliary information + Cl-InfoNCE in Figure 7.6 in the main text. A linear warm-up and cosine decay is used for the learning rate scheduling. To stablize the training and reduce overfitting, we adopt 0.0001 weight decay for the encoder network.

7.5.4 Comparisons with Swapping Clustering Assignments between Views

In this section, we provide additional comparisons between Kmeans + Cl-InfoNCE and Swapping Clustering Assignments between Views (SwAV) [Caron et al., 2020]. The experiment is performed on ImageNet-100 dataset. SwAV is a recent art for clustering-based self-supervised approach. In particular, SwAV adopts Sinkhorn algorithm [Cuturi, 2013] to determine the data clustering assignments for a batch of data samples, and SwAV also ensures augmented views of samples will have the same clustering assignments. We present the results in Table 7.2, where we see SwAV has similar performance with the Prototypical Contrastive Learning method [Li et al., 2020a] and has worse performance than our method (i.e., K-means +Cl-InfoNCE).

Method	Top-1 Accuracy (%)
Non-clustering-based Self-supervised Approaches	
SimCLR [Chen et al., 2020a]	58.2±1.7
MoCo [He et al., 2020]	59.4±1.6
Clustering-based Self-supervised Approaches (# of clusters = 2.5K)	
SwAV [Caron et al., 2020]	68.5±1.0
PCL [Li et al., 2020a]	68.9±0.7
K-means + Cl-InfoNCE (ours)	**77.9±0.7**

Table 7.2: Additional Comparsion with SwAV [Caron et al., 2020] showing its similar performance as PCL on ImageNet-100 dataset.

7.5.5 Preliminary results on ImageNet-1K with Cl-InfoNCE

We have performed experiments on ImageNet-100 dataset, which is a subset of the ImageNet-1K dataset [Russakovsky et al., 2015]. We use the batch size of $1,024$ for all the methods and consider 100 training epochs. We present the comparisons among Supervised Contrastive Learning [Khosla et al., 2020], our method (i.e., WordNet-hierarchy-information-determined clusters + Cl-InfoNCE), and SimCLR [Chen et al., 2020a]. We select the level-12 nodes in the WordNet tree hierarchy structures as our hierarchy-determined clusters for Cl-InfoNCE. We report the results in Table 7.3. We find that our method (i.e., hierarchy-determined clusters + Cl-InfoNCE) performs in between the supervised representations and conventional self-supervised representations.

Method	Top-1 Accuracy (%)
Supervised Representation Learning (Z = downstream labels T)	
SupCon [Khosla et al., 2020]	76.1±1.7
Weakly Supervised Representation Learning (Z = level 12 WordNet hierarchy labels)	
Hierarchy-Clusters + Cl-InfoNCE (ours)	67.9±1.5
Self-supervised Representation Learning (Z = instance ID)	
SimCLR [Chen et al., 2020a]	62.9±1.2

Table 7.3: Preliminary results for WordNet-hierarchy-determined clusters + Cl-InfoNCE on ImageNet-1K.

7.5.6 Synthetically Constructed Clusters in Section 7.3.2 in the Main Text

In Section 7.3.2 in the main text, on the UT-Zappos50K dataset, we synthesize clusters Z for various $I(Z;T)$ and $H(Z|T)$ with T being the downstream labels. There are 86 configurations of Z in total. Note that the configuration process has no access to data's auxiliary information and among the

86 configurations we consider the special cases for the supervised $\big(Z = T\big)$ and the unsupervised setting $\big(Z = \text{instance ID}\big)$. In specific, when $Z = T$, $I(Z;T)$ reaches its maximum at $H(T)$ and $H(Z|T)$ reaches its minimum at 0; when $Z = \text{instance ID}$, both $I(Z;T)$ $\big(\text{to be } H(T)\big)$ and $H(Z|T)$ $\big(\text{to be } H(\text{instance ID})\big)$ reaches their maximum. The code for generating these 86 configurations can be found in lines 177-299 in `https://anonymous.4open.science/r/Cl-InfoNCE-02AB/data_processing/UT-zappos50K/synthetic/generate.py`.

Chapter 8

Learning with Limited Supervision - Cross-view Learning with Undesirable Information

The prevalence of data has created many opportunities for machine learning systems to leverage information from it, especially for self-supervised learning (SSL). This unsupervised learning paradigm requires neither labels nor prior task knowledge to learn good representations. Nonetheless, the data may contain undesirable information that we should exclude. For example, protected attributes such as gender or ethnicity are present in many datasets [Mehrabi et al., 2019]. Without careful intervention, a conventional SSL process will inevitably also learn from these attributes. As a result, the learned representations may lead to unfair decisions on the downstream tasks, which should not have taken these protected attributes into account [Madras et al., 2018]. Another case of undesirable information is the presence of meta-information in datasets. For example, in a speech recognition setting, speaker ID is often presented as meta-information, but speech representations, in general, should be independent of the concrete speaker identity. There are two reasons: the speaker's information should not be leaked, and good representations of the speech signals should generalize well to new speakers. Hence, we should consider removing task-irrelevant meta-information in SSL. In the domain generalization setting, we may want to learn representations from multiple domains and expect the learned representations to generalize well across domains. This setup is related to Domain Adaptation [Pan et al., 2010], where the trained classifiers [Wang and Hebert, 2016] or the learned features [Tzeng et al., 2017] are expected to be domain-invariant. From a practical perspective, we may want to remove the domain-specific information for better generalization when considering SSL data in new domains.

One way to remove undesirable information in SSL is by adding a second objective to minimize the mutual information between the self-supervised representations and the underlying undesirable variable [Song et al., 2019], or minimize the prediction ability from the representations to the variable [Zemel et al., 2013]. However, these are not ideal due to the challenge of optimizing multiple objectives together and balancing the trade-off between the downstream performance and the undesirable variable's effect [Zhao and Gordon, 2019]. In this work, to remove the undesirable information in SSL, we present a conditional SSL method that uses only a single objective. In particular, our SSL method removes the effect of variations of the undesirable variable by conditioning on its value. Intuitively, since the variations are fixed, the effect of the variable will not be accounted for in SSL.

For the conditional SSL, we present the conditional contrastive learning objectives given that the contrastive objectives [Chen et al., 2020a, Tsai et al., 2021e, Tschannen et al., 2019] are most widely used in conventional SSL. One representative contrastive objective is InfoNCE [Oord et al., 2018], which aims to learn similar representations for correlated data pairs and dissimilar representations for unrelated data pairs. Inspired by InfoNCE, we develop the Conditional InfoNCE (C-InfoNCE) objective, which learns similar representations for *conditionally*-correlated data pairs and dissimilar representations for *conditionally*-unrelated data pairs. For example, if we choose the speaker ID and condition on speaker ID being # 1, the *conditionally*-correlated data pairs are representations of the *same* sequence from speaker # 1, and *conditionally*-unrelated data pairs are representations of *different* sequences from speaker # 1. In this example, both conditionally-correlated and -unrelated data come from the speaker # 1, which in other words, the part of the information for speaker identification is fixed.

From an information-theoretical perspective, we show that learning representations using C-InfoNCE relates to conditional mutual information maximization within representations. In particular, conditional mutual information measures the shared information between representations when removing the effect of the conditioned variable, which in our case is the undesirable variable. Since recent theoretical studies show that mutual information maximization within representations can result in representations that perform well on the downstream tasks [Arora et al., 2019, Tsai et al., 2021e], C-InfoNCE learned representations (using conditional mutual information maximization) might also enjoy competitive downstream performance with the additional benefit of removing undesirable information. However, C-InfoNCE requires the conditioned variable as input, resulting in extra computational cost compared to InfoNCE. To address this issue, we introduce a second objective, the Weak-Conditional InfoNCE (WeaC-InfoNCE), as a simplified form of C-InfoNCE, which does not require the conditioned variable as input like C-InfoNCE. We show that WeaC-InfoNCE is a lower bound of C-InfoNCE. Hence, learning representations using WeaC-InfoNCE also relates to the conditional mutual information maximization within representations.

To verify the effectiveness of our method, we conduct several experiments. First, we consider the speaker ID and sequence ID as the conditioned variables for self-supervised speech representation learning. The speaker and the sequence ID are the meta-information for human speech. We find removing this meta-information in the representations leads to better downstream performance on phoneme classification. Second, we consider age and gender as the conditioned variable for fair representation learning. We observe our methods can effectively remove a greater level of sensitive information compared to baseline methods. Finally, we consider the domain specification when performing self-supervised representation learning on data from multiple domains. We find removing the domain-specific information in learned representation leads to better downstream performance. To conclude, we find learning representations using either C-InfoNCE or WeaC-InfoNCE achieve competitive downstream task performance while successfully removing a significant level of the effect of the conditioned variable compared to conventional self-supervised representation baselines.

8.1 Method

In this section, we introduce conditional contrastive learning methods to remove undesirable information in self-supervised learning. In Subsection 8.1.1, we discuss the technical background - unconditional contrastive learning and the corresponding InfoNCE objective [Oord et al., 2018] under the conventional self-supervised learning setup. Next, Subsection 8.1.2 presents the Conditional InfoNCE (C-InfoNCE) objective, and in Subsection 8.1.3 we discuss the higher computational cost of C-InfoNCE (compared to InfoNCE) and then introduce the Weak-Conditional InfoNCE (WeaC-InfoNCE) as a more computationally efficient variant of C-InfoNCE.

Notation. We use uppercase letters (e.g., X) to denote random variables and lowercase letters (e.g., x) to denote outcomes from the random variables. In this work, we denote X and Y as the learned representations of data. For instance, in visual self-supervised learning [Chen et al., 2020a,b]:

$$X = \text{Feature\_Encoder}\Big(\text{Augmentation\_1}\big(\text{Data\_1}\big)\Big) \text{ and } Y = \text{Feature\_Encoder}\Big(\text{Augmentation\_2}\big(\text{Data\_2}\big)\Big).$$

where Data_1 and Data_2 could be augmented views from the same image or different images [Tschannen et al., 2019]. Next, we denote Z as the conditioned variable between X and Y. In other words, Z is a variable that contains undesirable information we hope to remove from our representations. We use P_X as the distribution of X and $D_{\text{KL}}\left(\cdot \,\|\, \cdot\right)$ as the Kullback–Leibler divergence between distributions.

8.1.1 Unconditional Contrastive Learning

Unconditional contrastive learning [Bachman et al., 2019, Chen et al., 2020a, He et al., 2020] aims at learning similar representations for correlated data and dissimilar representations for unrelated data. Examples of the correlated data could be different crops [Bachman et al., 2019] or distortions [Chen et al., 2020a, He et al., 2020] of the same image or the cross-modality pair (image-caption pair) [Tsai et al., 2021e] of a sample. Examples of the unrelated data include different images [Chen et al., 2020a, He et al., 2020] or an image and a caption from another image [Tsai et al., 2021e]. Below we briefly review a probabilistic interpretation of unconditional contrastive learning.

We refer to the representation (x, y) from correlated data as $(x, y) \sim P_{X,Y}$, where $P_{X,Y}$ is the joint distribution on $X \times Y$. Similarly, we use $(x, y) \sim P_X P_Y$ to mean that the representations (x, y) are from uncorrelated data, where $P_X P_Y$ is the product of marginal distributions. Poole et al. [2019], Tschannen et al. [2019] and Tsai et al. [2021c] have shown that the unconditional contrastive learning is essentially maximizing the divergence between $P_{X,Y}$ and $P_X P_Y$. For instance, a common contrastive approach, the InfoNCE [Oord et al., 2018] method, is maximizing $D_{\text{KL}}\left(P_{X,Y} \,\|\, P_X P_Y\right)$ as follows:

Definition 4 (InfoNCE [Oord et al., 2018] for unconditional contrastive learning)**.**

$$\text{InfoNCE} := \sup_f \mathbb{E}_{(x_i,y_i)\sim P_{X,Y}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right] \leq D_{\text{KL}}\left(P_{X,Y} \,\|\, P_X P_Y\right) = \text{MI}\left(X;Y\right), \tag{8.1}$$

where $\{(x_i, y_i)\}_{i=1}^n$ represents n independent copies of $(x, y) \sim P_{X,Y}$, $f(x, y)$ is any critic function that considers the input (x, y) and output a scalar, and $\text{MI}\left(X;Y\right)$ is the mutual information (MI) between X and Y. A common choice of $f(x, y)$ is to consider the cosine similarity $f(x, y) = \cos\Big(g(x), g(y)\Big)/\tau$ with τ being the temperature hyper-parameter and $g(\cdot)$ being a shallow network (usually a two-layer fully connected neural network [Chen et al., 2020a, He et al., 2020]). At a high-level, InfoNCE is maximizing similarities for $(x_i, y_i) \sim P_{X,Y}$ and minimizing similarities for $(x_i, y_j) \sim P_X P_Y$ $(i \neq j)$. As shown in Equation (8.1), InfoNCE is a lower bound of $\text{MI}\left(X;Y\right)$, and Arora et al. [2019], Tosh et al. [2021], Tsai et al. [2021e] have shown that maximizing lower bounds of $\text{MI}\left(X;Y\right)$ often leads to better representations for downstream tasks.

8.1.2 Conditional Contrastive Learning

We now discuss one idea to remove undesirable information from a variable from the self-supervised representations: conditioning on it [Cover, 1999]. Intuitively, conditioning on a variable means fixing the variations of this variable, and hence its effect can be removed.

Our conditional contrastive learning hence aims at learning similar representations for *conditionally*-correlated data and dissimilar representations for *conditionally*-unrelated data. For instance, let us consider an example of conditional speech self-supervised learning, where we choose the speaker ID to be the conditioned variable and let the outcome of this variable be speaker # 1. Then, the *conditionally*-correlated data are two representations learned from the *same* sequence of speaker # 1, and the *conditionally*-unrelated data are two representations learned from *different* sequences of speaker # 1. In both cases, *all* sequences are from speaker # 1 because we condition on it. We update the representations by calculating the contrastive objective from the constructed data pairs. Then we condition on another speaker ID outcome, say speaker # 2, and construct new data pairs and update the representations using the new data pairs. In this example, the conditional contrastive learning method is learning representations by taking data pairs from the same speaker in each update step and different speakers across the update steps. We hope that such a paradigm can exclude the information about speakers' identity in the representations (assuming the information regarding the speakers is undesirable information). Similar to the last section, below, we also provide a probabilistic interpretation of conditional contrastive learning.

We refer to the representation (x, y) from the conditionally-correlated data as $(x, y) \sim P_{X,Y|z}$, where $P_{X,Y|z}$ is the joint distribution on $X \times Y$ conditioned on the undesirable variable $Z = z$. And we refer to the representation (x, y) from the conditionally-unrelated data pair as $(x, y) \sim P_{X|z}P_{Y|z}$, where $P_{X|z}P_{Y|z}$ is the product of conditional marginal distributions. Recall that the unconditional contrastive learning aims to maximize the probability divergence between P_{XY} and $P_X P_Y$, resulting in a connection with mutual information $\mathrm{MI}(X;Y)$. Similarly, the conditional contrastive learning aims to maximize the divergence between $P_{XY|z}$ and $P_{X|z}P_{Y|z}$ for all $z \sim P_Z$, leading to a connection with conditional mutual information $\mathrm{MI}(X;Y|Z)$:

Definition 5 (Conditional Mutual Information)**.**

$$\begin{aligned} \mathrm{MI}\,(X;Y|Z) &:= \mathbb{E}_{z\sim Z}[D_{\mathrm{KL}}(P_{X,Y|Z=z} \| P_{X|Z=z}P_{Y|Z=z})] = \int_{\mathcal{Z}} D_{\mathrm{KL}}(P_{X,Y|Z} \| P_{X|Z}P_{Y|Z})\,\mathrm{d}P_Z \\ &= \int_{\mathcal{Z}} p_Z(z) \int_{\mathcal{Y}} \int_{\mathcal{X}} p_{X,Y|Z}(x,y|z) \log \frac{p_{X,Y|Z}(x,y|z)}{p_{X|Z}(x|z)p_{Y|Z}(y|z)} \mathrm{d}x\mathrm{d}y\mathrm{d}z. \end{aligned} \tag{8.2}$$

The conditional mutual information measures the expected mutual information of X and Y given Z. In other words, it measures the averaged shared information between X and Y conditioning on Z, and by conditioning on Z, we fix its variations and exclude its effect.

Inspired by InfoNCE for unconditional contrastive learning, we present the Conditional InfoNCE (C-InfoNCE) objective for conditional contrastive learning:

Proposition 14 (Conditional InfoNCE (C-InfoNCE) for conditional contrastive learning)**.**

$$\mathrm{C-InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right]\right] \leq \mathrm{MI}\,(X;Y|Z), \tag{8.3}$$

where $\{(x_i, y_i)\}_{i=1}^n$ represents n independent copies of $(x, y) \sim P_{X,Y|z}$ and $f(x, y, z)$ takes in the input (x, y, z) and outputs a scalar. We leave the derivations and proofs in Appendix. We design $f(x, y, z)$ as the cosine similarity $f(x, y, z) = \cos\Big(g(x, z), g(y, z)\Big)/\tau$ with τ being the temperature hyper-parameter and $g(\cdot, \cdot)$ being a shallow network. As shown in Equation (8.3), C-InfoNCE is a lower bound of $\mathrm{MI}\,(X;Y|Z)$, and hence learning representation using C-InfoNCE results in conditional mutual information maximization within representations.

8.1.3 Weak-Conditional Contrastive Learning

Compared to InfoNCE (Equation (8.1)), a disadvantage of C-InfoNCE (Equation (8.3)) is the need to consider three variables (i.e., X, Y, and Z) instead of two (i.e., X and Y) in $f(\cdot)$. In particular, introducing a third variable Z increases computational cost since the function $f(\cdot)$ becomes more complex. To alleviate this problem, we introduce the following Weak-Conditional InfoNCE (WeaC-InfoNCE) objective to avoid having Z as an extra input variable:

Proposition 15 (Weak-Conditional InfoNCE (WeaC-InfoNCE) for conditional contrastive learning)**.**

$$\begin{aligned}\mathrm{WeaC-InfoNCE} := \sup_{f} \mathbb{E}_{z \sim P_Z}\Bigg[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}{}^{\otimes n}}\Big[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\Big]\Bigg] \\ \leq D_{\mathrm{KL}}\Big(P_{X,Y} \,\|\, \mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]\Big) = \mathrm{Weak-MI}\,(X;Y|Z) \leq \mathrm{MI}\,(X;Y|Z),\end{aligned} \tag{8.4}$$

where $f(x,y)$ takes in the input (x,y) and outputs a scalar, instead of $f(x,y,z)$ which takes in (x,y,z) in C-InfoNCE. Same as InfoNCE, we consider $f(x,y)$ in WeaC-InfoNCE as the cosine similarity $f(x,y) = \cos\Big(g(x),g(y)\Big)/\tau$ with τ being the temperature hyper-parameter and $g(\cdot)$ being a shallow network. $\mathrm{Weak-MI}\,(X;Y|Z)$ represents the weak-conditional mutual information, which is the KL-divergence between $P_{X,Y}$ and $\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]$. The notion of the weak-conditional mutual information comes from weak-conditional independence [Daudin, 1980, Fukumizu et al., 2004, 2007], which is defined as the case when $P_{X,Y} = \mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]$.

We highlight the differences between the weak-conditional mutual information (i.e., $\mathrm{Weak-MI}\,(X;Y|Z)$) and the standard conditional mutual information (i.e., $\mathrm{MI}\,(X;Y|Z)$). First, $\mathrm{Weak-MI}\,(X;Y|Z)$ measures $D_{\mathrm{KL}}\big(P_{X,Y} \,\|\, \mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]\big)$ and $\mathrm{MI}\,(X;Y|Z)$ measures $\mathbb{E}_Z\big[D_{\mathrm{KL}}\big(P_{X,Y|Z} \,\|\, P_{X|Z}P_{Y|Z}\big)\big]$, where $D_{\mathrm{KL}}\big(P_{X,Y} \,\|\, \mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]\big) \leq \mathbb{E}_Z\big[D_{\mathrm{KL}}\big(P_{X,Y|Z} \,\|\, P_{X|Z}P_{Y|Z}\big)\big]$. Hence, $\mathrm{Weak-MI}\,(X;Y|Z)$ is a lower bound on $\mathrm{MI}\,(X;Y|Z)$ and can be seen as a more "conservative" measurement of $\mathrm{MI}\,(X;Y|Z)$, capturing only part of information in $\mathrm{MI}\,(X;Y|Z)$. Second, $\mathrm{MI}\,(X;Y|Z) = 0$ is a sufficient condition for $\mathrm{Weak-MI}\,(X;Y|Z) = 0$, which suggests that conditional independence implies weak-conditional independence. See detailed derivations and proofs in Appendix.

To conclude, WeaC-InfoNCE (Equation (8.4)) objective is a surrogate for C-InfoNCE objective (Equation (8.3)), with the additional benefit of considering only two variables instead of three in $f(\cdot)$. Since WeaC-InfoNCE is also a lower bound on $\mathrm{MI}\,(X;Y|Z)$ (looser than C-InfoNCE), learning representations using WeaC-InfoNCE results in conditional mutual information maximization within representations, just like C-InfoNCE.

8.2 Experiments

We evaluate the proposed conditional contrastive learning on several tasks. In Section 8.2.1, we study self-supervised speech representation learning and consider the meta-information such as the speaker ID and sequence ID as the conditioned variable. We study whether removing the meta-information in the learned representations impacts downstream performance. In Section 8.2.2, we examine fair representation learning and consider the sensitive attributes, age and gender, as the conditioned variable. We hope to reduce the amount of sensitive information in our learned representations. In Section 8.2.3, we investigate multi-domain self-supervised learning and consider the domain specification (domain ID) as the conditioned variable. We aim to reduce domain-specific information in the representations so that they can transfer well across different domains.

In practice, the expectations in InfoNCE (Equation (8.1)), C-InfoNCE (Equation (8.3)), and WC-InfoNCE (Equation (8.4)) are replaced by the empirical mean of a batch of samples. For a fair comparison, when comparing InfoNCE, C-InfoNCE, and WC-InfoNCE, we will only alter the training objectives and leave the network design and the optimization procedure identical. The small difference is the design of the critic function $f(\cdot)$, and we ensure $g(\cdot)$ in $f(\cdot)$ has similar size across different objectives. We leave

Objective	Phoneme Classification	Speaker Classification
Unconditional Self-supervised Learning		
InfoNCE [Oord et al., 2018]	$63.6_{\pm 0.12}$	$93.4_{\pm 0.18}$
Conditional Self-supervised Learning (Z = Speaker ID)		
C-InfoNCE (ours)	$64.3_{\pm 0.13}$	$71.2_{\pm 0.14}$
WeaC-InfoNCE (ours)	$63.8_{\pm 0.12}$	$72.0_{\pm 0.13}$
Conditional Self-supervised Learning (Z = Sequence ID)		
C-InfoNCE (ours)	$64.5_{\pm 0.11}$	$71.6_{\pm 0.12}$
WeaC-InfoNCE (ours)	$64.2_{\pm 0.13}$	$72.3_{\pm 0.13}$

Table 8.1: Accuracy (%) for LibriSpeech-100h phoneme and speaker classification results using self-supervised representations (conditional versus unconditional contrastive learning methods). We consider the meta information including the speaker and sequence ID as the conditioned variable Z.

the details for the networks, the optimizers, the hyper-parameters, and more details for the datasets in Appendix.

8.2.1 Speech Representation Learning: Removing Effect from Meta-Information

For the first set of experiments, we consider learning self-supervised speech representations on Librispeech-100h dataset [Panayotov et al., 2015], which contains 100 hours of English speech from 251 speakers and $28,538$ sequences. Following prior work [Oord et al., 2018, Rivière et al., 2020], we first pretrain the model using self-supervised objectives without downstream label access. Then, we fix the pretrained model, add an additional linear classifier on top of the pretrained representations, and fine-tune the linear classifier with labels. Note that the above steps are performed on the training set. For evaluation, we fix both the pretrained model and the linear classifier, and we report the top-1 accuracy metric on the evaluation set.

We consider the unconditional (conventional setup, InfoNCE [Oord et al., 2018] in Equation (8.1)) and the conditional (ours, C-InfoNCE in Equation (8.3) and WeaC-InfoNCE in Equation (8.4)) contrastive learning methods as the self-supervised objectives. And we select the meta-information, including the speaker and sequence ID, as the conditioned variable Z in C-InfoNCE and WeaC-InfoNCE. Note that C-InfoNCE and WeaC-InfoNCE aim to remove the effect of the conditioned variable in the resulting representations. Hence, our goal is to see whether removing the meta-information will affect the representation's performance on the downstream tasks, including Phoneme Classification and Speaker Classification.

We follow Oord et al. [2018], Rivière et al. [2020] for the experimental setup, where they consider InfoNCE as the objective. In particular, the correlatedly-paired representations $(x, y^+) \sim P_{X,Y}$ in InfoNCE are a pair of past and future states of a sequence. And unrelatedly-paired representations $(x, y^-) \sim P_X P_Y$ are two random states from different sequences. InfoNCE aims to learn similar correlatedly-paired representations and dissimilar unrelatedly-paired representations. Hence, this process is regarded as forward modeling, i.e., predicting the future states of a sequence from its past. Next, we discuss C-InfoNCE and WeaC-InfoNCE's deployments. Different from InfoNCE, C-InfoNCE and WeaC-InfoNCE consider the conditionally-correlatedly-paired representations (e.g., $(x, y^+) \sim P_{X,Y|Z=z}$) and the conditionally-unrelatedly-paired representations (e.g., $(x, y^-) \sim P_{X|Z=z} P_{Y|Z=z}$) given the conditioned outcome $z \sim P_Z$. Take Z as the speaker ID as an example and assume its outcome z is speaker 1, then (x, y^+) are a pair of past and future states of a speech sequence for speaker 1, and (x, y^-) are two random states from different sequences for speaker 1. If Z is the sequence ID, then both (x, y^+) and (x, y^-) are from the same sequence. Comparing to InfoNCE, C-InfoNCE and WeaC-InfoNCE also perform the forward modeling but have different unrelatedly-pairs construction.

Table 8.1 shows results. First, we observe a performance improvement on phoneme classification by comparing C-InfoNCE and WeaC-InfoNCE with InfoNCE. Since C-InfoNCE and WeaC-InfoNCE aim to remove the meta-information (speaker and sequence ID) in the self-supervised representations, the improvement suggests that the meta-information should be irrelevant to the phoneme classification. Second, on speaker classification, we see an over 20% performance deterioration from 93.4% of InfoNCE to C-InfoNCE and WeaC-InfoNCE. This suggests that by conditioning on the speaker or sequence ID, the C-InfoNCE and WeaC-InfoNCE successfully remove a decent amount of speaker information. Note that the sequence ID implicitly contains speaker information because each sequence must be from only one single speaker. Last, we compare C-InfoNCE and WeaC-InfoNCE and find that WeaC-InfoNCE performs in between InfoNCE and C-InfoNCE. The result suggests that C-InfoNCE is better at removing the effect from the conditioned variable than WeaC-InfoNCE. Yet, as discussed in Section 8.1.3, WeaC-InfoNCE is more computationally efficient than C-InfoNCE.

8.2.2 Fair Representation Learning: Removing Effect from Sensitive Attributes

For our second set of experiments, we consider removing sensitive information in self-supervised fair representation learning. We follow the setting in prior work [Song et al., 2019], which learns the representations to maximally preserve information from data while removing the information from the sensitive attributes. In particular, let the input data be X, the learned representations be Y, and the sensitive attributes be Z, Song et al. [2019] presents the pretraining stage as maximizing the conditional mutual information $\text{MI}(X;Y|Z)$ under the constraint $\text{MI}\,(Y,Z) < \epsilon$ (we set $\epsilon = 0.1$ in our experiments). Then, we fix the pretrained network and fine-tune the representations using logistic regression classifiers on top of the learned representations. The above steps are performed on the training set. For evaluation, the reported metrics are the demographic parity distance Δ_{DP} [Madras et al., 2018] and the representation quality based on the area under the ROC curve (AUC) on downstream tasks. Specifically, the demographic parity distance Δ_{DP} [Madras et al., 2018] is defined as the absolute expected difference in classifier outcomes between two sensitive groups (e.g., the male group and the female group from gender), and a lower Δ_{DP} suggests a fairer representation.

We consider two datasets: UCI German credit [Dua and Graff, 2017] and Adult [Dua and Graff, 2017] datasets. The German credit dataset contains $1,000$ samples with 20 attributes, a binary age indicator variable as the sensitive attribute, and predicting credit approval as the downstream task. The Adult dataset includes $48,842$ samples with 14 attributes, gender as the sensitive attribute, and predicting whether an adult makes $50K$ per year as the downstream task. We consider the unconditional contrastive learning (i.e., $\max\ \text{MI}\,(X;Y)$ s.t. $\text{MI}\,(Y;Z) < 0.1$) and the conditional contrastive learning (i.e., $\max\ \text{MI}\,(X;Y|Z)$ s.t. $\text{MI}\,(Y;Z) < 0.1$) approaches. All the comparing methods are adapted from the L-MIFR framework described in prior work [Song et al., 2019], and here we only highlight the differences: the self-supervised learning part, particularly $\text{MI}\,(X;Y)$ and $\text{MI}\,(X;Y|Z)$. More details about L-MIFR's optimization process can be found in Appendix.

The first baseline is having InfoNCE (Equation (8.1)) as the lower bound of $\text{MI}\,(X;Y)$. In particular, a correlated-pair $(x, y^+) \sim P_{X,Y}$ in InfoNCE is a pair of input and its learned representation. And an unrelated pair $(x, y^-) \sim P_X P_Y$ is a pair of input and a representation learned from another input. InfoNCE is learning representations to preserve the information from the input, while it does not remove the effect from the sensitive attributes. Next, our methods (C-InfoNCE in Equation (8.3) and WeaC-InfoNCE in Equation (8.4)) are the lower bounds of $\text{MI}(X;Y|Z)$. Different from InfoNCE, C-InfoNCE and WeaC-InfoNCE consider

Objective	UCI German credit		UCI Adult	
	Δ_{DP} ($\downarrow$)	ROC AUC ($\uparrow$)	Δ_{DP} ($\downarrow$)	ROC AUC ($\uparrow$)
Unconditional Self-supervised Learning $\Rightarrow$ *max* $\mathrm{MI}\,(X;Y)$ *s.t.* $\mathrm{MI}\,(Y;Z) < 0.1$				
InfoNCE [Oord et al., 2018]	0.04	0.56	0.23	0.62
Conditional Self-supervised Learning ($Z = $ *Age or Gender*) $\Rightarrow$ *max* $\mathrm{MI}\,(X;Y\|Z)$ *s.t.* $\mathrm{MI}\,(Y;Z) < 0.1$				
L-MIFR [Song et al., 2019]	0.02	0.61	0.07	0.68
MIFR [Song et al., 2019]	0.02	0.60	0.08	0.66
C-InfoNCE (ours)	0.02	**0.62**	**0.06**	**0.69**
WeaC-InfoNCE (ours)	0.02	**0.62**	**0.06**	0.68

Table 8.2: Results for demographic parity distance (Δ_{DP}, lower means fairer representations) and the area under the ROC curve (ROC AUC, higher means better downstream performance) for fair representation learning on UCI German credit and Adult datasets. The conditioned variable (Z, the sensitive attributes) in the German credit dataset is age and in the Adult dataset is gender. X are the input and Y are the learned representations.

the conditionally-correlated pair (e.g., $(x, y^+) \sim P_{X,Y|Z=z}$) and the conditionally-unrelated pair (e.g., $(x, y^-) \sim P_{X|Z=z}P_{Y|Z=z}$) given the conditioned outcome $z \sim P_Z$. For instance, let Z be gender, and we condition on the gender is female. Then the conditionally-correlated pair would be the data from a female and the learned representation from the same female. The conditionally-unrelated pair would be the data from a female and the learned representation from the data of another female. Lastly, we have the MIFR and the L-MIFR methods [Song et al., 2019] to be the baselines. These two methods variationally lower-bound the conditional mutual information: $\mathrm{MI}\,(X;Y|Z) \geq \mathbb{E}_{q_\phi(X,Y,Z)}\,[\log p_\theta(X|Y,Z)]$, with q_ϕ and p_θ being two separate networks modeling different distributions. MIFR and L-MIFR consider the same objective but different optimization processes. To conclude, L-MIFR, MIFR, C-InfoNCE, and WeaC-InfoNCE belong to conditional learning objectives, which consider removing the sensitive information from the data representations.

Table 8.2 presents our results. First, we find the unconditional (InfoNCE) self-supervised method has lower fairness and lower downstream task performance than the conditional (L-MIFR, MIFR, C-InfoNCE, WeaC-InfoNCE) self-supervised methods. For instance, on UCI Adult dataset, InfoNCE has lower Δ_{DP} (0.23 v.s. 0.06) and lower ROC AUC (0.62 v.s. 0.69) than C-InfoNCE. Note that both unconditional and conditional methods aim to preserve more information from the data in the representation, but the conditional methods additionally consider removing the effect from the sensitive attributes. The result suggests that the conditional methods can achieve better fairness and would not sacrifice downstream performances. Second, we find that our methods (C-InfoNCE and WeaC-InfoNCE) achieve at par or even better fairness and downstream performances compared to the strong baselines, L-MIFR and MIFR. The result suggests that our methods can be competitive for self-supervised fair representation learning.

8.2.3 Multi-domain Representation Learning: Removing Effect from Domain Specification

For the third set of experiments, we consider learning self-supervised representations from multiple domains. While more domains provide more information, we argue that domain-invariant information particularly can effectively be shared across domains. Hence, we aim to improve the generalization by removing domain-specific information in the representations. To this end, we consider applying the conditional

Objective	CIFAR-10	Tiny ImageNet	SUN 397
Uni-Domain Unconditional Self-supervised Learning			
InfoNCE [Oord et al., 2018]	$92.67_{\pm 0.12}$	$53.42_{\pm 0.43}$	$70.89_{\pm 0.35}$
Multi-Domain Unconditional Self-supervised Learning			
InfoNCE [Oord et al., 2018]	$91.13_{\pm 0.11}$	$50.38_{\pm 0.32}$	$68.23_{\pm 0.45}$
Multi-Domain Conditional Self-supervised Learning (Z = Domain Specification)			
C-InfoNCE (ours)	$93.54_{\pm 0.21}$	$\mathbf{57.46}_{\pm 0.23}$	$\mathbf{74.62}_{\pm 0.26}$
WeaC-InfoNCE (ours)	$\mathbf{94.23}_{\pm 0.31}$	$57.01_{\pm 0.19}$	$74.23_{\pm 0.31}$

Table 8.3: Accuracy (%) for object detection and scene understanding using self-supervised representation learning with the presence of data from multiple domains. In the experiments, we regard a dataset as a domain with the selected datasets having similar scales but different purposes (object detection and scene understanding). The unconditional contrastive learning represents the setting in SimCLR [Chen et al., 2020a], which utilizes the InfoNCE objective. The notion of uni-domain refers the setting that we pre-train using a single dataset and the notion of multi-domain considers the pre-training using the mixture of the three selected datasets. The conditional contrastive learning considers the domain specification as the conditioned variable. We adopt the linear evaluation protocal [Chen et al., 2020a, He et al., 2020].

contrastive learning objectives by conditioning on the domain specification (i.e., the domain id). For our experiments, we select the following three visual datasets: 1) CIFAR-10 [Krizhevsky et al., 2009], an object detection dataset with $60,000$ 32×32 images in 10 classes; 2) Tiny ImageNet [Le and Yang, 2015], a scaled-down version of ImageNet dataset for object detection, with $100,000$ 64×64 images in 200 classes; and 3) SUN 397 dataset [Xiao et al., 2010], a scene understanding dataset with $108,753$ images of 397 categories. We regard each dataset as a domain since the datasets have different image sources and tasks (object detection v.s. scene understanding). Following prior work [Chen et al., 2020a, He et al., 2020], we first pretrain the model using self-supervised objectives without access to downstream labels. Then, we freeze the pretrained model, add an additional linear classifier, and fine-tune the linear classifier with labels. Both steps are performed on the training set. For evaluation, we report the top-1 accuracy metric on the evaluation set. We select the ResNet-50 [He et al., 2016] as our pretrained feature model.

We now discuss the implementations of C-InfoNCE in Equation (8.3) and WeaC-InfoNCE in Equation (8.4), as well as various baseline methods. The results are presented in Table 8.3. First, the uni-domain unconditional contrastive learning is SimCLR [Chen et al., 2020a] - a standard visual contrastive representation learning approach which considers InfoNCE [Oord et al., 2018] (Equation (8.1)) as its objective. Note that it performs pretraining on a single dataset and then evaluates on the same dataset. In particular, the correlated-paired representations $(x, y^+) \sim P_{X,Y}$ in InfoNCE are the learned representations from augmented variants of an image and the unrelated-paired representations $(x, y^-) \sim P_X P_Y$ are the representations of two random images in the dataset.

The multiple-domain unconditional contrastive learning represents the same setting as the previous one except that we perform pretraining on the mixture of the three selected datasets. Hence, the unrelated-paired representations $(x, y^-) \sim P_X P_Y$ can be the representations from different datasets (e.g., x from CIFAR-10 and y^- from Tiny ImageNet). Finally, the multiple-domain conditional contrastive learning considers C-InfoNCE and WeaC-InfoNCE as the pretraining objectives on the mixed datasets, with the domain specification being the conditioned variable Z. Thus, in C-InfoNCE and WeaC-InfoNCE, the conditionally-correlatedly-paired representations (e.g., $(x, y^+) \sim P_{X,Y|Z=z}$) and the conditionally-unrelatedly-paired representations (e.g., $(x, y^-) \sim P_{X|Z=z}P_{Y|Z=z}$) are always from the same dataset. We consider the same batch size across all settings for a fair comparison.

Let us first consider CIFAR-10's performance in Table 8.3. First, we see the performance drop from uni-domain unconditional contrastive learning (92.67) to multi-domain unconditional contrastive learning (91.13), where both methods use InfoNCE as the objective. Note that multi-domain unconditional one considers all data from the three datasets for the pretraining. In contrast, the uni-domain unconditional one only considers the data from a single dataset. The performance drop suggests that merely increasing the data in self-supervised learning may not result in better performance, especially when the data come from multiple domains. Second, we see the performance improvement from the uni-modal and multi-modal unconditional methods (92.67 and 91.13) to the multi-domain conditional methods (93.54 and 94.23), where the conditional methods consider removing domain-specific information in the learned representations. The performance improvement suggests that the domain-specific information may be irrelevant to the downstream task. Third, we observe no obvious performance differences between C-InfoNCE and WeaC-InfoNCE. For instance, C-InfoNCE achieves worse performance than WeaC-InfoNCE on CIFAR-10 (93.54 v.s. 94.23), while it performs better on Tiny ImageNet (57.46 v.s. 57.01). Since WeaC-InfoNCE is computationally more efficient than C-InfoNCE, it may be a better choice for multi-domain self-supervised learning.

8.3 Related Work

Self-Supervised Learning Self-supervised learning takes pre-defined tasks from unlabeled samples for pretraining representations [Jaiswal et al., 2021] and uses the learned representations for downstream tasks, such as visual object detection [Chen et al., 2020b, He et al., 2020], language understanding and Question Answering [Devlin et al., 2018, Lan et al., 2019], and automatic speech recognition [Oord et al., 2018, Rivière et al., 2020]. One major way of self-supervised learning is contrastive learning [Chen et al., 2020a, He et al., 2020, Oord et al., 2018], which tries to construct pairs of related data (termed positive pairs) and pairs of unrelated samples (termed negative pairs) and learn a model that scores the positive and negative pairs differently [Kipf et al., 2019]. Prior works [Chen et al., 2020a, Oord et al., 2018, Tsai et al., 2021e] construct the correlated pairs and unrelated pairs from the unconditional joint and the product of marginal distributions. Meanwhile, the proposed C-InfoNCE and WeaC-InfoNCE consider the conditionally-correlated pairs and conditionally-unrelated pairs from the conditional joint and the product of conditional marginal distributions, respectively. Both distributions are conditioned on the same outcome of an undesirable variable.

Fair Representation Learning Fair representation learning [McNamara et al., 2019] considers different measures to quantify algorithm's fairness, including statistical parity [Dwork et al., 2012], equality of opportunity [Hardt et al., 2016], equalized odds [Hardt et al., 2016], and individual fairness [Dwork et al., 2012]. While most of the study focuses on a supervised setup [Dwork et al., 2012, Hardt et al., 2016, McNamara et al., 2019], this chapter studies the self-supervised setup where downstream tasks are unknown, but the learned representations still need to preserve fairness criterion [Calmon et al., 2017, Madras et al., 2018]. Song et al. [2019] presents a suitable framework that maximizes the expressiveness of the representation while satisfying controllable levels of fairness using conditional mutual information. The difference between Song et al. [2019] and the proposed method is how we construct and maximize the lower bound of conditional mutual information.

Multi-Domain Learning Domain adaptation [Patel et al., 2015, Wang and Deng, 2018] aims to solve a distribution shift [Zhang et al., 2013] or domain change [Gopalan et al., 2011] between two domains that could degrade performance. Domain adaptation methods can achieve domain-invariant representations by minimizing the probability divergences [Long et al., 2017, Shrivastava et al., 2017, Zhuang et al., 2015] between source and target data domains. Our approach, on the other hand, reduces the effect of domain specifications to achieve domain-invariant representation by conditioning on the domain specifications (domain ID).

8.4 Discussion

In this chapter, we developed conditional contrastive learning methods to remove the effect of an undesirable variable in self-supervised learning by conditioning on it. The proposed C-InfoNCE and WeaC-InfoNCE objectives lead to representations that perform better in downstream tasks and exclude a greater level of information of undesirable variables compared to baseline models in speech representation learning, fairness representation, and multi-domain visual representation learning. Nevertheless, we do recognize the limitations of the approach. First, it is not always easy to know which exact variables to condition when we want to remove certain undesirable information. Moreover, conditioning on the incorrect information may lead to suboptimal representations for downstream tasks. In terms of the potential broader impact of this work, the methods in the chapter can bring a positive social impact by removing privacy-related information from representations. For example, in medical applications, our approach can be used to help protect patient information from being leaked out in learned representations. However, if the conditioned variable in a machine learning system also contains vital information useful for the downstream task, removing its effect may result in a performance drop and thus affect users of the system.

8.5 Appendix

8.5.1 Theoretical Analysis

This section provides the theoretical analysis of Proposition 14 and Proposition 15 in the main text. The full set of assumptions of all theoretical results and complete proofs of all theoretical results are presented below.

8.5.1.1 Lemmas before Proof

We first present the following lemmas, which will be later used in the proof:

Lemma 16 (Nguyen et al. [2010] with two variables)**.** Let $\mathcal{X}$ and $\mathcal{Y}$ be the sample spaces for X and Y, f be any function: $(\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$, and $\mathcal{P}$ and $\mathcal{Q}$ be the probability measures on $\mathcal{X} \times \mathcal{Y}$. Then,

$$D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big) = \sup_f \mathbb{E}_{(x,y)\sim\mathcal{P}}[f(x,y)] - \mathbb{E}_{(x,y)\sim\mathcal{Q}}[e^{f(x,y)}] + 1.$$

Proof. The second-order functional derivative of the objective is $-e^{f(x,y)} \cdot d\mathcal{Q}$, which is always negative. The negative second-order functional derivative implies the objective has a supreme value. Then, take the first-order functional derivative and set it to zero:

$$d\mathcal{P} - e^{f(x,y)} \cdot d\mathcal{Q} = 0.$$

We then get the optimal $f^*(x,y) = \log \frac{d\mathcal{P}}{d\mathcal{Q}}$. Plug in $f^*(x,y)$ into the objective, we obtain

$$\mathbb{E}_{\mathcal{P}}[f^*(x,y)] - \mathbb{E}_{\mathcal{Q}}[e^{f^*(x,y)}] + 1 = \mathbb{E}_{\mathcal{P}}[\log \frac{d\mathcal{P}}{d\mathcal{Q}}] = D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big).$$

■

Lemma 17 (Nguyen et al. [2010] with three variables)**.** Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be the sample spaces for X, Y, and Y, f be any function: $(\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}) \to \mathbb{R}$, and $\mathcal{P}$ and $\mathcal{Q}$ be the probability measures on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Then,

$$D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big) = \sup_f \mathbb{E}_{(x,y,z)\sim\mathcal{P}}[f(x,y,z)] - \mathbb{E}_{(x,y,z)\sim\mathcal{Q}}[e^{f(x,y,z)}] + 1.$$

Proof. The second-order functional derivative of the objective is $-e^{f(x,y,z)} \cdot d\mathcal{Q}$, which is always negative. The negative second-order functional derivative implies the objective has a supreme value. Then, take the first-order functional derivative and set it to zero:

$$d\mathcal{P} - e^{f(x,y,z)} \cdot d\mathcal{Q} = 0.$$

We then get the optimal $f^*(x,y,z) = \log \frac{d\mathcal{P}}{d\mathcal{Q}}$. Plug in $f^*(x,y,z)$ into the objective, we obtain

$$\mathbb{E}_{\mathcal{P}}[f^*(x,y,z)] - \mathbb{E}_{\mathcal{Q}}[e^{f^*(x,y,z)}] + 1 = \mathbb{E}_{\mathcal{P}}[\log \frac{d\mathcal{P}}{d\mathcal{Q}}] = D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big).$$

■

Immediate results following Lemma 16.
Lemma 18.

$$\begin{aligned}\mathrm{Weak}-\mathrm{MI}\,(X;Y|Z) &= D_{\mathrm{KL}}\Big(P_{X,Y} \,\|\, \mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]\Big) \\ &= \sup_f \mathbb{E}_{(x,y)\sim P_{X,Y}}[f(x,y)] - \mathbb{E}_{(x,y)\sim \mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]}[e^{f(x,y)}] + 1.\end{aligned}$$

Proof. Let $\mathcal{P}$ be $P_{X,Y}$ and $\mathcal{Q}$ be $\mathbb{E}_{P_Z}\big[P_{X|Z}P_{Y|Z}\big]$ in Lemma 16. ■

Lemma 19. $\sup_f \mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\log \frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right] \leq D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big)$.

Proof. $\forall f$, we have

$$\begin{aligned}D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big) &= \mathbb{E}_{(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big)\right] \\ &\geq \mathbb{E}_{(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\mathbb{E}_{(x,y_1)\sim\mathcal{P}}\left[\log \frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right] - \mathbb{E}_{(x,y_1)\sim\mathcal{Q}}\left[\frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right] + 1\right] \\ &= \mathbb{E}_{(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\mathbb{E}_{(x,y_1)\sim\mathcal{P}}\left[\log \frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right] - 1 + 1\right] \\ &= \mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\log \frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\right].\end{aligned}$$

The first line comes from the fact that $D_{\mathrm{KL}}\Big(\mathcal{P} \,\|\, \mathcal{Q}\Big)$ is a constant. The second line comes from Lemma 16. The third line comes from the fact that (x,y_1) and $(x,y_{2:n})$ are interchangeable when they are all sampled from $\mathcal{Q}$.

To conclude, since the inequality works for all f, and hence

$$\sup_f \mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\Big[\log\frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x,y_j)}}\Big]\leq D_{\text{KL}}\Big(\mathcal{P}\,\|\,\mathcal{Q}\Big).$$

■

Note that Lemma 19 does not require $n\to\infty$, which is a much more practical setting compared to the analysis made only when $n\to\infty$. And a remark is that the equality holds in Lemma 19 when $n\to\infty$.

Immediate results following Lemma 17.
Lemma 20.

$$\begin{aligned}\text{MI}\left(X;Y|Z\right)&=\mathbb{E}_{P_Z}\Big[D_{\text{KL}}\left(P_{X,Y|Z}\,\|\,P_{X|Z}P_{Y|Z}\right)\Big]\\&=D_{\text{KL}}\left(P_{X,Y,Z}\,\|\,P_ZP_{X|Z}P_{Y|Z}\right)\\&=\sup_f\mathbb{E}_{(x,y,z)\sim P_{X,Y,Z}}[f(x,y,z)]-\mathbb{E}_{(x,y,z)\sim P_ZP_{X|Z}P_{Y|Z}}[e^{f(x,y,z)}]+1.\end{aligned}$$

Proof. Let $\mathcal{P}$ be $P_{X,Y,Z}$ and $\mathcal{Q}$ be $P_ZP_{X|Z}P_{Y|Z}$ in Lemma 17. ■

Showing $\text{Weak}-\text{MI}\left(X;Y|Z\right)\leq\text{MI}\left(X;Y|Z\right)$**.**
Proposition 16. $\text{Weak}-\text{MI}\left(X;Y|Z\right)\leq\text{MI}\left(X;Y|Z\right)$.

Proof. According to Lemma 18,

$$\begin{aligned}\text{Weak}-\text{MI}\left(X;Y|Z\right)&=\sup_f\mathbb{E}_{(x,y)\sim P_{X,Y}}[f(x,y)]-\mathbb{E}_{(x,y)\sim\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]}[e^{f(x,y)}]+1\\&=\sup_f\mathbb{E}_{(x,y,z)\sim P_{X,Y,Z}}[f(x,y)]-\mathbb{E}_{(x,y,z)\sim P_ZP_{X|Z}P_{Y|Z}}[e^{f(x,y)}]+1.\end{aligned}$$

Let $f_1^*(x,y)$ be the function when the equality for $\text{Weak}-\text{MI}\left(X;Y|Z\right)$ holds, and let $f_2^*(x,y,z)=f_1^*(x,y)$ ($f_2^*(x,y,z)$ will not change $\forall z\sim P_Z$):

$$\begin{aligned}\text{Weak}-\text{MI}\left(X;Y|Z\right)&=\mathbb{E}_{(x,y,z)\sim P_{X,Y,Z}}[f_1^*(x,y)]-\mathbb{E}_{(x,y,z)\sim P_ZP_{X|Z}P_{Y|Z}}[e^{f_1^*(x,y)}]+1\\&=\mathbb{E}_{(x,y,z)\sim P_{X,Y,Z}}[f_2^*(x,y,z)]-\mathbb{E}_{(x,y,z)\sim P_ZP_{X|Z}P_{Y|Z}}[e^{f_2^*(x,y,z)}]+1.\end{aligned}$$

Comparing the above equation to Lemma 20,

$$\text{MI}\left(X;Y|Z\right)=\sup_f\mathbb{E}_{(x,y,z)\sim P_{X,Y,Z}}[f(x,y,z)]-\mathbb{E}_{(x,y,z)\sim P_ZP_{X|Z}P_{Y|Z}}[e^{f(x,y,z)}]+1,$$

we conclude $\text{Weak}-\text{MI}\left(X;Y|Z\right)\leq\text{MI}\left(X;Y|Z\right)$. ■

8.5.1.2 Proof of Proposition 14 in the Main Text

Proposition 17 (Conditional InfoNCE (C-InfoNCE) for conditional contrastive learning, restating Proposition 14 in the main text)**.**

$$\begin{aligned}\mathrm{C-InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right]\right] \\ \leq \mathbb{E}_{P_Z}\Big[D_{\mathrm{KL}}\left(P_{X,Y|Z}\,\|\,P_{X|Z}P_{Y|Z}\right)\Big] = \mathrm{MI}\left(X;Y|Z\right),\end{aligned}$$

Proof. Given a $z \sim P_Z$, we let $\mathcal{P} = P_{X,Y|Z=z}$ and $\mathcal{Q} = P_{X|Z=z}P_{Y|Z=z}$. Then,

$$\mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\log\frac{e^{f(x,y_1,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x,y_j,z)}}\right] = \mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right].$$

The only variables in the above equation are X and Y with Z being fixed at z, and hence the following can be obtained via Lemma 19:

$$\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right] \leq D_{\mathrm{KL}}\left(\mathcal{P}\,\|\,\mathcal{Q}\right) = D_{\mathrm{KL}}\left(P_{X,Y|Z=z}\,\|\,P_{X|Z=z}P_{Y|Z=z}\right).$$

The above inequality works for any function $f(\cdot,\cdot,\cdot)$ and any $z \sim P_Z$, and hence

$$\sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right]\right] \leq \mathbb{E}_{P_Z}\Big[D_{\mathrm{KL}}\left(P_{X,Y|Z}\,\|\,P_{X|Z}P_{Y|Z}\right)\Big].$$

■

8.5.1.3 Proof of Proposition 15 in the Main Text

Proposition 18 (Weak-Conditional InfoNCE (WeaC-InfoNCE) for conditional contrastive learning, restating Proposition 15 in the main text)**.**

$$\begin{aligned}\mathrm{WeaC-InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right]\right] \\ \leq D_{\mathrm{KL}}\Big(P_{X,Y}\,\|\,\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]\Big) = \mathrm{Weak-MI}\left(X;Y|Z\right) \leq \mathrm{MI}\left(X;Y|Z\right).\end{aligned}$$

Proof. By defining $\mathcal{P} = P_{X,Y}$ and $\mathcal{Q} = \mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]$, we have

$$\mathbb{E}_{(x,y_1)\sim\mathcal{P},(x,y_{2:n})\sim\mathcal{Q}^{\otimes(n-1)}}\left[\log\frac{e^{f(x,y_1)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x,y_j)}}\right] = \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right]\right].$$

Via Lemma 19, we have

$$\sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right]\right] \leq D_{\mathrm{KL}}\Big(P_{X,Y}\,\|\,\mathbb{E}_{P_Z}\left[P_{X|Z}P_{Y|Z}\right]\Big).$$

Combing with Proposition 16 that $\mathrm{Weak-MI}\left(X;Y|Z\right) \leq \mathrm{MI}\left(X;Y|Z\right)$, we conclude the proof. ■

8.5.1.4 Showing WeaC-InfoNCE is a lower bound of C-InfoNCE

Proposition 19.

$$\begin{aligned}
&\mathrm{WeaC-InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}{}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right]\right] \\
\leq \quad &\mathrm{C-InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}{}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right]\right].
\end{aligned}$$

Proof. Let $f_1^*(x,y)$ be the function when the equality holds in WeaC-InfoNCE, and let $f_2^*(x,y,z) = f_1^*(x,y)$ $\Big(f_2^*(x,y,z)$ will not change $\forall z \sim P_Z\Big)$:

$$\mathrm{WeaC-InfoNCE} := \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}{}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f_2^*(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f_2^*(x_i,y_j,z)}}\right]\right].$$

Since the equality holds with the supreme function in C-InfoNCE, and hence

$$\mathrm{WeaC-InfoNCE} \leq \mathrm{C-InfoNCE}.$$

■

8.5.2 Experimental setup

This section provides the experimental setup details of Section 8.2 in the main text, including hyperparameters, optimizers, code snippets, the total amount of computational resource, as well as more experimental results. Code, data, and instructions needed to reproduce the results are included in the released code under this anonymous link `https://anonymous.4open.science/r/conditional_infonce-4232`.

8.5.2.1 Speech Representation Learning

In this subsection, we provide experimental details of speech representation learning in Section 8.2.1 in the main text.

Dataset, Splits, and License. We use the Librispeech [Panayotov et al., 2015] dataset. The dataset is available at the link: `https://www.openslr.org/12`. It is a corpus of approximately 1,000 hours on English speech with a sampling rate of 16 kHz. There are three training splits, containing 100, 360, 500 hours of speech sequences, respectively. We use the 100 hour training split. There are also separate evaluation and test sets provided. The license of the dataset is Creative Commons Attribution 4.0 International. The dataset does not contain identifiable personality information.

Training Setups and Baseline Model. We use the 100-hour split from Librispeech to pretrain and finetune the models, and we use the predefined test set in the dataset to evaluate the models. We follow the baseline implemented in Rivière et al. [2020], an implementation of InfoNCE on speech. In particular, given an input signal $x_{1:T}$ with T being the time steps, we first pass it through an encoder ϕ_θ parametrized by θ to produce a sequence of hidden representations $\{h_{1:T}\}$ where $h_t = \phi_\theta(x_t)$. Then, we obtain the contextual representation c_t at time step t with a sequential model ψ_ρ parametrized by ρ: $\mathbf{c}_t = \psi_\rho(h_1, \ldots, h_t)$, where c_t contains context information before time step t.

For unsupervised pretraining, we select a multi-layer convolutional network as the encoder ϕ_θ, and we select a two-layer transformer with hidden dimension 256 as the sequential model ψ_ρ. Here, the positive pair is (h_{t+k}, c_t) where k is the number of time steps ahead, and the negative pairs are (h_i, c_t), where h_i hidden representations of a batch of random hidden representations assumed to be unrelated to c_t. The scoring function f based on Equation (8.1) in the main text at step t with k steps ahead is $f_k = f_k(h, c_t) = \exp((h)^\top W_k c_t)$, where W_k is a learnable linear transformation defined separately for each $k \in \{1, ..., K\}$ and K is predetermined as 12 time steps. The loss will then be formulated as:

$$\ell_t^{\text{InfoNCE}} = -\frac{1}{K}\sum_{k=1}^{K} [\log \frac{\exp(f_k(h_{t+k}, c_t))}{\sum_{h_i \in \mathcal{N}} \exp(f_k(h_i, c_t)))}] \tag{8.5}$$

After the pretraining step, we then evaluate the network by the following: we first fix the pretrained model and add one additional linear classifier on top. We then fine-tune the linear classifier with samples from the training split, but this time with labels. After fine-tuning, we fix both the pretrained model and the fine-tuned classifier and report the top-1 accuracy on the corresponding evaluation set (which in this case would be the "test-clean" split in Librispeeh.)

C-InfoNCE and WeaC-InfoNCE. To implement the Conditional InfoNCE (C-InfoNCE) and the Weak-Conditional InfoNCE (WeaC-InfoNCE), for each update of the objective function, we sample a batch of sequences that comes from the same outcome of the conditioned variable Z to calculate the loss function using C-InfoNCE or WeaC-InfoNCE. For instance, if we condition on speaker ID being Speaker 1, we first find all sequences that come from Speaker 1, and sample a batch of sequences from Speaker 1 to perform the calculation of C-InfoNCE or WeaC-InfoNCE. All positive and negative pairs would then be from Speaker 1. After we calculate C-InfoNCE or WeaC-InfoNCE and update the network parameters, we condition on a new outcome of speaker ID, say Speaker 2, and repeat the steps above. Which sequences are coming from which speakers are known as meta-data in the dataset, and the mapping from sequences to speakers is established at the beginning of training. For details, please refer to Line 361 to Line 408 in this file: https://github.com/facebookresearch/CPC_audio/blob/master/cpc/dataset.py.

In C-InfoNCE, we also need to include z, the speaker ID (or the sequence ID) in the network $f(\cdot)$. Since speaker or sequence IDs are indices, we convert the indices into an eight-dimension vector containing either 0 or 1 in each position for the speaker ID, or a sixteen-dimension vector for the sequence ID. Essentially, we convert the digital number of each speaker ID or sequence ID into its binary form and treat that as the z vector. We then replace $f_k(h_{t+k}, c_t)$ with $f_k(h_{t+k}W_{hz}z, c_tW_{cz}z)$, and $f_k(h_{t_i}, c_t)$ with $f_k(h_{t+k}W_{hz}z, c_tW_{cz}z)$ in Equation 8.5, which results in the following formulation:

$$\ell_t^{\text{C}-\text{InfoNCE}} = -\frac{1}{K}\sum_{k=1}^{K} [\log \frac{\exp(f_k(h_{t+k}W_{hz}z, c_tW_{cz}z)))}{\sum_{h_i \in \mathcal{N}} \exp(f_k(h_iW_{hz}z, c_tW_{cz}z)))}] \tag{8.6}$$

where W_{hz} and W_{cz} are two learnable linear transformations. For WC-InfoNCE, on the other hand, it follows the same loss function in Equation 8.5, as it does not require z for $f(\cdot)$:

$$\ell_t^{\text{WeaC}-\text{InfoNCE}} = -\frac{1}{K}\sum_{k=1}^{K} [\log \frac{\exp(f_k(h_{t+k}, c_t))}{\sum_{h_i \in \mathcal{N}} \exp(f_k(h_i, c_t)))}] \tag{8.7}$$

Hyper-parameters and Optimization. We pretrain the network using the sequences in the $100-$hour training set for 200 epochs. We set the batch size per GPU as 16, and sample 128 negative samples in each batch. We use the Adam optimizer [Kingma and Ba, 2015], with a learning rate of $2e-4$, $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 1e-8$. We use a learning rate warm-up of 10. We fix all setups, including architecture, learning rate, and optimizer for InfoNCE, C-InfoNCE and WeaC-InfoNCE. For evaluation, we run 100 epochs using the pretrained model and the training sequences with labels; and we evaluate the fine-tuned model on the test split of Librispeech.

Computational Resource. The models are trained and evaluated on 4 RTX-2080Ti GPUs. 200 epochs of pretraining take 2 days.

8.5.2.2 Fair Representation Learning

In this subsection, we provide experimental details of fair representation learning in Section 8.2.2 in the main text.

Dataset, Splits, and License. We train our models on UCI German [Dua and Graff, 2017], UCI Adult [Dua and Graff, 2017] and Health Heritage [kag] datasets, where we do not report the results for the third dataset in the main text.

Health Heritage [1] comprises $60,000$ patient samples and over 20 attributes. We consider the Cartesian product of nine age values and two gender values (thus eighteen groups in total) as sensitive attributes, and the task is to predict whether an index of patient mortality is positive or negative as the downstream task. We split the Health dataset into an 80% part for training and a 20% part for testing, following Song et al. [2019]. It grants entrants of competition a right to use for his/her/its own patient management or other internal business purposes, but may not grant or otherwise transfer to any third party (which is not applicable in our case, since we will not publicize the dataset).

UCI German[2] has a total of $10,00$ samples. We follow the split in Song et al. [2019], where there are 900 samples in the training set and 100 samples in the test set. It has the Database Contents License v1.0.

UCI Adult[3] has a total of $48,842$ samples, with a pre-determined training split of $32,561$ samples and a test split of $16,281$ samples. It has the CC0: Public Domain License.

For all three datasets, no personally identifiable information is available.

Training Setups and Baseline Methods. In the fair representation learning experiment, we first train models without labels by using contrastive self-supervised objectives: MIFR [Song et al., 2019], L-MIFR [Song et al., 2019], InfoNCE [Oord et al., 2018], C-InfoNCE and WeaC-InfoNCE. In this section, we first briefly introduce how MIFR and L-MIFR work.

MIFR and L-MIFR aim to maximize the expressiveness of representations while satisfying certain fairness constraints. This is done by $\max \mathrm{MI}\left(X;Y|Z\right)$ s.t. $\mathrm{MI}\left(Y;Z\right) < \epsilon$. $\max \mathrm{MI}\left(X;Y|Z\right)$ aims to learn an expressive representation Y that maximally preserves information in X; and by conditioning on Z, information in X that is correlated with Z will be discarded [Song et al., 2019]. On the other hand, $\mathrm{MI}\left(Y;Z\right) < \epsilon$ controls the maximum amount of the mutual information between Y and Z (to be ϵ), to ensure a controllable level of fairness.

[1] https://www.kaggle.com/c/hhp

[2] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[3] https://archive.ics.uci.edu/ml/datasets/adult

For the optimization process for MIFR and L-MIFR, Song et al. [2019] optimize $\max \mathrm{MI}\,(X;Y|Z)$ via its variational lower bound $\max \mathbb{E}_{q_\phi(X,Y,Z)}\left[\log p_\theta(X \mid Y,Z)\right]$, where q_ϕ and p_θ are two neural networks parameterized by $\phi \in \Phi$ and $\theta \in \Theta$. q_ϕ is used to approximate $P_{X,Y,Z}$ and p_θ is used to parameterize $P_{X|Y,Z}$. Furthermore, to ensure the optimization satisfies the constraint $MI\,(Y;Z) < \epsilon$, Song et al. [2019] performs Lagrangian dual relaxation, where MIFR and L-MIFR consider different approaches to search for the Langrangian multipliers. The detailed discussion of this optimization is out of the scope in this chapter, and readers can refer to the original paper for more clarification.

C-InfoNCE and WeaC-InfoNCE. The difference between the proposed C-InfoNCE/WeaC-InfoNCE and MIFR/L-MIFR from Song et al. [2019] is the maximization of $\mathrm{MI}\,(X;Y|Z)$. Unlike MIFR or L-MIFR Song et al. [2019] which maximize $\mathrm{MI}\,(X;Y|Z)$ by $\mathbb{E}_{q_\phi(X,Y,Z)}\left[\log p_\theta(X \mid Y,Z)\right]$, C-InfoNCE maximizes $\mathrm{MI}\,(X;Y|Z)$ by $\mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}{}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^n \log \frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x_i,y_j,z)}}\right]\right]$, and WeaC-InfoNCE maximizes $\mathrm{MI}\,(X;Y|Z)$ by $\mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim P_{X,Y|z}{}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^n \log \frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^n e^{f(x_i,y_j)}}\right]\right]$, respectively. In specific, for C-InfoNCE and WeaC-InfoNCE, we consider the same Lagrangian dual optimization process as L-MIFR, and we only change how we maximize $\mathrm{MI}\,(X;Y|Z)$.

An implementation difference between C-InfoNCE and WeaC-InfoNCE is that the function $f(\cdot)$ in C-InfoNCE considers Z as an input, while the function $f(\cdot)$ in WeaC-InfoNCE does not consider Z as an input. Now, we discuss how we represent Z. For the UCI German dataset, Z is a binary age indicator, and therefore $Z \in \{0,1\}$. For the UCI Adult dataset, Z is an indicator of male and female, and therefore $Z \in \{0,1\}$. For the Health Heritage dataset, Z is the Cartesian product of 9 age values and 2 genders, which in total we have 18 discrete values for Z. We use the binary representations for Z, and therefore $Z \in \{0,1\}^5$ (a 5-dimensional vector).

Hyper-parameters and Optimization. We assume the model does not have access to labels during training; instead, it takes in the input and sensitive attributes. We follow Song et al. [2019], where we consider maximizing the conditional mutual information given the fairness constraint. All neural networks for approximating distributions in MIFR and L-MIFR are two-layer neural networks. The $f(\cdot)$s in C-InfoNCE and WeaC-InfoNCE are also two-layer networks. After pretraining, we use logistic regression classifiers over the representation Y for prediction tasks. We use $\lambda_1 = \lambda_2 = 1.0$ for optimization in MIFR, initialize $\lambda_1 = \lambda_2 = 1.0$ for L-MIFR and allow a range of $(0.01, 100)$, and fix $\epsilon_1 = \epsilon_2 = 0.1$ for all experimental settings. We use the Adam optimizer with learning rate $1e-3$ and $\beta_1 = 0.5$ where the learning rate is multiplied by 0.98 every 1000 optimization iterations. For the Adult and the Health dataset, we optimize for 2000 epochs; we optimize for 10000 epochs for the German dataset.

Additional Results. Table 8.4 presents the new results for ROC-AUC on Health dataset. Note that we do not provide the Δ_{DP} results since Δ_{DP} is only defined for binary attributes, while the Health dataset considers 18 sensitive attributes. We find our methods (C-InfoNCE and WeaC-InfoNCE) work the best and C-InfoNCE outperforms WeaC-InfoNCE.

Next, on the UCI Adult dataset, we provide new results for other fairness criteria. Following Song et al. [2019], we consider three fairness criteria: Demographic Parity, Equalized Odds, and Equalized Opportunity. Specifically, we consider these notions in terms of mutual information measurements constructed by the corresponding definition of each notion. For example, Demographic Parity requires that the representation Y and sensitive attribute Z are independent. From a mutual information perspective, that means Y and

Objective	Health Heritage (ROC AUC ($\uparrow$))
Unconditional Self-supervised Learning $\Rightarrow$ *max* $\mathrm{MI}\,(X;Y)$ *s.t.* $\mathrm{MI}\,(Y;Z) < 0.1$	
InfoNCE [Oord et al., 2018]	0.57
Conditional Self-supervised Learning ($Z = Age \times Gender$) $\Rightarrow$ *max* $\mathrm{MI}\,(X;Y\|Z)$ *s.t.* $\mathrm{MI}\,(Y;Z) < 0.1$	
L-MIFR [Song et al., 2019]	0.63
MIFR [Song et al., 2019]	0.56
C-InfoNCE (ours)	**0.66**
WeaC-InfoNCE (ours)	0.65

Table 8.4: Results for the area under the ROC curve (ROC AUC, higher means better downstream performance) for fair representation learning on Health datasets. The conditioned variable (Z, the sensitive attributes) is the Cartesian product of two gender choices and nine age values. X are the input, and Y are the learned representations.

Z should have low mutual information, which is $I_{DP} = \mathrm{MI}\,(Y;Z)$. The second fairness criterion, the Equalized Odds, requires a classifier to predict labels equally well for all sensitive attribute values. In this case, the requirement is equivalent to Y and Z have low mutual information given the label, which is $I_{EO} = \mathrm{MI}\,(Y;Z|label)$. The third criteria, the Equalized Opportunity, considers $y = 1$ as a preferred label (a label that confers an advantage or benefit) and requires a classifier to predict the preferred label equally well for all sensitive attribute values. That is to say, we require that Y and Z have low mutual information given label= 1, which is $I_{EOpp} = \mathrm{MI}\,(Y;Z|label = 1)$. Readers can refer to [Song et al., 2019] for more details.

From Table 8.5, C-InfoNCE achieves the lowest level of mutual information measurements on different fairness criteria, suggesting the representation learned through C-InfoNCE satisfies different fairness criteria better than other baselines when we measure different criteria using mutual information. We also notice that in both cases, WeaC-InfoNCE performs close to C-InfoNCE, achieving competitive downstream performance while preserving almost the same level of fairness as C-InfoNCE.

Computational Resource. We use one RTX-2080Ti GPU for training these datasets, and training $10,000$ epochs on German, the longest among three datasets, takes six hours.

8.5.2.3 Multi-domain Visual Learning

In this subsection, we provide experimental details of multi-domain visual representation learning in Section 8.2.3 in the main text.

Dataset, Splits, and License. We train our models on CIFAR-10 [Krizhevsky et al., 2009], Tiny ImageNet [Le and Yang, 2015] and SUN 397 [Xiao et al., 2010]. CIFAR-10 [Krizhevsky et al., 2009] is an object detection dataset with $60,000$ 32×32 images in 10 classes. The test sets includes $10,000$ images. Tiny ImageNet [Le and Yang, 2015] is a scaled-down version of ImageNet dataset for object detection, with $100,000$ 64×64 images in 200 classes for training, and $10,000$ test images for evaluation. SUN 397 dataset [Xiao et al., 2010] is a scene understanding dataset with $108,753$ images of 397 categories. We randomly partition the dataset into $76,128$ images for training, $10,875$ images for validation, and $21,750$ images for testing. All three datasets are licensed under the Creative Commons Attribution 4.0 License.

Objective	UCI Adult		
	$I_{DP} = \text{MI}\,(Y;Z)$ (↓)	I_{EO} (↓)	I_{EOpp}, (↓)
Unconditional Self-supervised Learning $\Rightarrow$ *max* $\text{MI}\,(X;Y)$ *s.t.* $\text{MI}\,(Y;Z) < 0.1$			
InfoNCE [Oord et al., 2018]	0.09	0.10	0.07
Conditional Self-supervised Learning (Z = *Age or Gender*) $\Rightarrow$ *max* $\text{MI}\,(X;Y\|Z)$ *s.t.* $\text{MI}\,(Y;Z) < 0.1$			
L-MIFR [Song et al., 2019]	0.08	0.09	0.04
MIFR [Song et al., 2019]	0.13	0.11	0.09
C-InfoNCE (ours)	**0.06**	**0.08**	0.04
WeaC-InfoNCE (ours)	0.07	**0.08**	0.04

Table 8.5: Results for mutual information measurements of different fairness notions: Demographic Parity, Equalized Odds, and Equalized Opportunity (lower means better fairness) for fair representation learning on Adult datasets. The conditioned variable (Z, the sensitive attributes) is the gender attribute. X are the input, and Y are the learned representations.

Training Setups. We consider four different experimental settings: 1) uni-domain unconditional self-supervised learning using InfoNCE, 2) multi-domain unconditional self-supervised learning using InfoNCE, 3) multi-domain conditional self-supervised learning using C-InfoNCE, and 4) multi-domain conditional self-supervised learning using WeaC-InfoNCE. All the experiments are provided in `https://anonymous.4open.science/r/conditional_infonce-4232`.

1. Uni-domain Unconditional Self-supervised Learning using InfoNCE

   This setting considers the exact same setup as SimCLR [Chen et al., 2020a]. In particular, we perform pretraining on a single dataset and then evaluates the pretrained model on the same dataset. ResNet-50 [He et al., 2016] is chosen as the backbone model for performing the self-supervised pretraining. Note that we remove the last classifier layer in ResNet-50, and we consider 2048-2048-128 multi-layer perceptrons (MLPs) with ReLU non-linearity as the projection head in InfoNCE ($g(\cdot)$ in $f(\cdot)$ in InfoNCE).

   After the pretraining, we consider the linear evaluation protocol Chen et al. [2020a], which fixes the pretrained encoder, removes the projection head, and adopts a linear classifier on top of the pretrainined encoder. The linear classifier is fine-tuned with the downstream labels. Note that both the pretraining and the fine-tuning steps are performed on the training samples.

   For evaluation, we fix the pretrainined encoder as well as the linear classifier. We then report the evaluation accuracy on the test samples.

2. Multi-domain Unconditional Self-supervised Learning using InfoNCE

   This setting is similar to the previous setting with two differences: 1) the composition of the data batch and 2) the network designs.

   The composition of the data batch for input. Under the uni-domain setting, we consider the data only from a single dataset within a data batch. On the contrary, under the multi-domain setting, we consider the data from the three datasets within a data batch. Note that we ensure the same data batch size for both the uni-domain and the multi-domain setting. Particularly, for the uni-domain setting, we consider the data batch size 960. For the multi-domain setting, we consider the data batch size 320 for each dataset, resulting in 960 data batch size in total.

The network designs. We select the ResNet-50 as the feature encoder model. Nonetheless, under the multi-domain setting, images from different datasets can be of different sizes. Hence, for the few building blocks of the ResNet-50, we consider three separate blocks of $CONV - BN - RELU - MAXPOOL$ to handle various image sizes. The rest of the ResNet-50 model is shared for all three datasets.

Same as the uni-domain setting, the multi-domain setting considers the projection head on top of the multi-dataset 960-batched data after the feature encoder. The projection head is considering in the pretraining stage that uses InfoNCE. The fine-tuning stage considers different linear classifiers for different datasets.

3. Multi-domain Conditional Self-supervised Learning using C-InfoNCE

   We also make the discussions based on the composition of the data batch and the network designs.

   The composition of the data batch for input. The multi-domain conditional setting considers the domain specification (the dataset ID) as the conditioned variable, and hence each batch of the input data comes from the same conditioned value. In specific, the data within a data batch are always from the same dataset, not mixing the data from three datasets as in the multi-domain unconditional setting. In particular, we first sample the dataset ID (randomly choosing among CIFAR-10, Tiny ImageNet, and SUN 397), and then we sample 960 images from the selected dataset to form a data batch.

   The network designs. We consider the same design of the feature encoder model as the design under the multi-domain unconditional setting.

   The difference between the multi-domain unconditional setting and the multi-domain conditional setting using C-InfoNCE is the projection head. In particular, the function $f(\cdot)$ takes in representations x, y as the input under multi-domain unconditional setting, while the function $f(\cdot)$ takes in representations x, y as well as the conditional value z as input under multi-domain conditional setting using C-InfoNCE. For the latter setting, we design $f(x, y, z)$ as $f(x, y, z) = \text{cosine similarity with temperature}\Big(g(x, z), g(y, z)\Big) = \text{cosine similarity with temperature}\Big(g_z(x), g_z(y)\Big)$. $g_z(\cdot)$ represents the projection head considers for the conditioned value z (in our case Z is the dataset specification). Hence, we consider different projection heads for different datasets. The projection head is considering in the pretraining stage that uses InfoNCE. The fine-tuning stage considers different linear classifiers for different datasets.

4. Multi-domain Conditional Self-supervised Learning using WeaC-InfoNCE

   We also make the discussions based on the composition of the data batch and the network designs.

   The composition of the data batch for input. The composition of the data batch for input is exactly the same between multi-domain conditional setting using C-InfoNCE and WeaC-InfoNCE.

   The network designs. We consider the same design of the feature encoder model as the design under the multi-domain unconditional setting.

   Different from the the multi-domain conditional setting using C-InfoNCE, we share the same projection head among datasets for WeaC-InfoNCE. The reason is that the function $f(\cdot)$ takes in only the representations x, y as input for WeaC-InfoNCE. Hence, we design $f(x, y)$ as $f(x, y) = \text{cosine similarity with temperature}\Big(g(x), g(y)\Big)$. $g(\cdot)$ represents the projection head that is shared among all datasets. The projection head is considering in the pretraining stage that uses InfoNCE. The fine-tuning stage considers different linear classifiers for different datasets.

Hyper-parameters and Optimization. Following the settings in [Chen et al., 2020a]and [Tsai et al., 2021a], we deploy distributed parallel training, with a batch size of 960. We use the LARS optimizer [You et al., 2017] with momentum 0.9. The learning rate is set to 1.5. The projection heads (e.g., $g(\cdot)$) are two-layer MLP layers with hidden dimension $2,048$ and batch normalization. We train the model for 500 epochs. We only tune one hyper-parameter, the temperature parameter τ in the contrastive objectives by grid search, and the optimal value we found is 0.5.

Computationl Resource. We use four RTX-2080Ti GPUs for pretraining, and the setting with the slowest speed, the multi-domain conditional training via C-InfoNCE, takes 2.5 days to train for 500 epochs.

8.5.2.4 Conditional Mutual Information Estimation

In this section, we seek to understand if the proposed C-InfoNCE and WeaC-InfoNCE can estimate the conditional mutual information accordingly, as both of them are lower bounds of conditional mutual information. We compare the two with other different conditional mutual information estimators: classifier-based estimator [Mukherjee et al., 2020] and difference-based estimator [Mukherjee et al., 2020]. InfoNCE estimates $\mathrm{MI}\,(X;Z)$ but not the conditional mutual information, $\mathrm{MI}\,(X;Y|Z)$, thus we do not compare with it here. We base our implementation on prior work [Mukherjee et al., 2020, Tsai et al., 2020d].

Dataset, Splits, and License. Following [Mukherjee et al., 2020], we generate two separate datasets based on two linear models, with three random variables X, Y and Z, where X and Y is 1-dimensional while dimension d_Z can scale. The two datasets are the following:

$$\text{Dataset I: } X \sim \mathcal{N}(0,1); Z \sim \mathcal{U}(-0.5, 0.5)^{d_Z}; \epsilon \sim \mathcal{N}\left(Z_1, \sigma_\epsilon^2\right); Y \sim X + \epsilon$$

$$\text{Dataset II: } X \sim \mathcal{N}(0,1); Z \sim \mathcal{N}(0,1)^{d_Z}; U = w^T Z, \|w\|_1 = 1; \epsilon \sim \mathcal{N}\left(U, \sigma_\epsilon^2\right); Y \sim X + \epsilon$$

where $\mathcal{U}(-0.5, 0.5)^{d_Z}$ means each coordinate of Z is drawn i.i.d from a uniform distribution between -0.5 and 0.5. Z_1 is the first dimension of Z. We set $\sigma_\epsilon^2 = 0.1$ (the same as Mukherjee et al. [2020]) and obtain unit norm random vector w from $\mathcal{N}\left(0, I_{d_Z}\right)$ and keep it constant. In Dataset I, Y only depends on Z_1, while in Dataset II the variables Y depends on all dimensions in Z. For both setting we vary *either* the number of samples *or* dimension d_Z. For both these datasets, the sample size is varied as $n \in \{5000, 10000, 20000, 50000\}$ keeping d_z fixed at 20. We also vary $d_z \in \{1, 10, 20, 50, 100\}$, keeping sample size fixed at $n = 20000$. To split the dataset into the train set and the test set, the first two-thirds of the synthetic samples will be in the train set, and the rest will be in the test set. The dataset can be generated by the codebase we provide, and is open for public usage.

Training Setups and Baseline Models. We discuss briefly on how the two baselines, classifier-based and difference-based estimation work. Classifier-based estimation [Mukherjee et al., 2020] is to train a classifier that could distinguish points from different distributions. To start with, recall the definition of conditional mutual information:

$$\mathrm{MI}\,(X;Y|Z) := \int_{\mathcal{Z}} D_{\mathrm{KL}}(P_{X,Y|Z} \| P_{X|Z} P_{Y|Z})\, \mathrm{d}P_Z = \int_{\mathcal{Z}} D_{\mathrm{KL}}(P_{X,Y,Z} \| P_{X,Z} P_{Y|Z})\, \mathrm{d}P_Z \tag{8.8}$$

Classifier-based method use $\int_{\mathcal{Z}} D_{\mathrm{KL}}(P_{X,Y,Z} \| P_{X,Z} P_{Y|Z})\, \mathrm{d}P_Z$ to estimate conditional mutual information. To be specific, given n i.i.d samples $\{x_i, y_i, z_i\}_{i=1}^n, (x_i, y_i, z_i) \sim P_{X,Y,Z}$, Mukherjee et al. [2020] use the generative model GAN [Goodfellow et al., 2014] to model the conditional distribution $P(Y|Z)$. For notation simplicity, we refer the GAN model as $\hat{P}^{\mathrm{GAN}}(Y|Z)$. Given samples from the joint distribution,

$P_{X,Y,Z}$, and samples from $P_{X,Z}\hat{P}^{\text{GAN}}_{Y|Z}$, classifier-based method labels the points drawn from $P_{(X,Y,Z)}$ as $label = 1$ and the points from $\hat{P}_{X,Z}P^{\text{GAN}}_{Y|Z}$ as $label = 0$. Then, it trains a binary classifier for predicting the assigned binary label. Then the point-wise likelihood ratio $\frac{p(x,y,z)}{p(x,z)p(y|z)} \approx \frac{p(x,y,z)}{p(x,z)p^{\text{GAN}}(y|z)}$ of each data point (x_i, y_i, z_i) can be calculated by $\frac{Pr(label=1|(x_i,y_i,z_i))}{1-Pr(label=1|(x_i,y_i,z_i))}$, where $Pr(label = 1|(x_i, y_i, z_i)$ is the predicted probability of data point has $label = 1$ from the classifier. Using the point-wise likelihood, we can obtain $\int_{\mathcal{Z}} D_{\text{KL}}(P_{X,Y,Z} \| P_{X,Z}P_{Y|Z})\text{d}P_Z$ by plugging the point-wise likelihood into a lower bound of KL-divergence. Further discussions of this classifier-based estimation method is out of the scope of our discussion, and readers could refer to Mukherjee et al. [2020] for more details.

Mukherjee et al. [2020] further presents the difference-based method that represents the conditional mutual information as the difference between two mutual information quantities: $I(X;Y|Z) = I(X;Y,Z) - I(X;Z)$. Then, it considers the classifier-based estimation for both $I(X;Y,Z)$ and $I(X;Z)$.

C-InfoNCE and WeaC-InfoNCE. On the other hand, the proposed C-InfoNCE and WeaC-InfoNCE are different. Given the formulations of C-InfoNCE and WeaC-InfoNCE:

$$\text{C} - \text{InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i,z)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j,z)}}\right]\right] \leq \text{MI}\left(X;Y|Z\right), \tag{8.9}$$

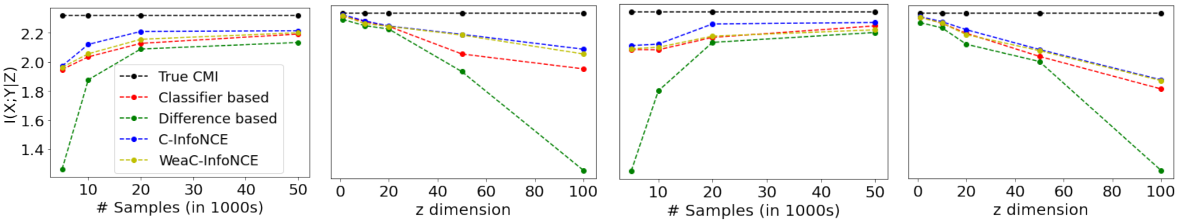
and

$$\text{WeaC} - \text{InfoNCE} := \sup_f \mathbb{E}_{z\sim P_Z}\left[\mathbb{E}_{(x_i,y_i)\sim {P_{X,Y|z}}^{\otimes n}}\left[\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{f(x_i,y_i)}}{\frac{1}{n}\sum_{j=1}^{n}e^{f(x_i,y_j)}}\right]\right] \leq \text{MI}\left(X;Y|Z\right) \tag{8.10}$$

Given samples (x, y, z) from the joint distribution $P_{X,Y,Z}$, we want to sample from the conditional distribution $P_{X,Y|Z}$ and from the product of conditional marginals $P_{X|Z}P_{Y|Z}$. To be able to sample from $P_{X,Y|Z}$ and $P_{X|Z}P_{Y|Z}$, we first cluster the value of $Z \in (-0.5, 0.5)$ to K clusters, $\{C_1, C_2, ..., C_k\}$ by performing K-mean clustering on it, with corresponding cluster centers $\{z_1, z_2, ..., z_k\}$. In our deployment, we set $K = 10$. We can then sample the data points $(x, y, z) \sim P_{X,Y|Z=z_m}$ by sampling from the set $\{(x, y, z)|z \text{ in cluster } C_m\}$. To sample from $P_{X|Z=z_m}P_{Y|Z=z_m}$, we sample from the set $\{(x_i, y_{j\neq i}, z)|z \text{ in cluster } C_m\}$ where x_i and y_j comes from different data point. For C-InfoNCE, we plug in the point (x, y, z) into Equation 9 using $f(x, y, z) = g_x(x)W_{xz}z \cdot g_y(y)W_{yz}z$ where W_{xz} and W_{yz} are learnable transformations, and $g_x(\cdot)$ and $g_y(\cdot)$ are two-layer fully connected neural networks with hidden dimension 64. For WeaC-InfoNCE, we plug in the point (x, y, z) into Equation 10 using $f(x, y) = g_x(x) \cdot g_y(y)$ as Z is not an input of WeaC-InfoNCE.

Hyper-parameters and Optimization. We use two-layer neural networks for the classifier in the classifier-based or difference-based methods and also for $g(\cdot)$ in C-InfoNCE and WeaC-InfoNCE. The hidden dimension is 64. We use a batch size of 64 and an Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.9$, with a learning rate of $1e - 4$. We train for 100 epochs and use ReLU as the activation function.

Results and Discussions. In Figure 8.1 we show the conditional mutual information estimation results. The proposed conditional methods can estimate mutual information better than classifier or difference-based methods under the same dimension of Z in sub-figure (a) and (c), and the estimations degrade less severely compared to other methods when we vary the dimension of Z to as large as 200 (sub-figure (b) and (d)). We show that C-InfoNCE and WeaC-InfoNCE can be good tools for conditional mutual information estimation.



(a) Dataset I: fix $d_Z = 20$.

(b) Dataset I: fix $n = 20k$.

(c) Dataset II: fix $d_Z = 20$.

(d) Dataset II: fix $n = 20k$.

Figure 8.1: Conditional mutual information estimation on two datasets generated by two linear models. The proposed C-InfoNCE and WeaC-InfoNCE estimates conditional mutual information better than other baselines on both datasets and on two settings, either by fixing sample size $n = 20,000$ and varying the dimension of the conditioned variable d_Z, or fixing the dimension $d_Z = 20$ and vary sample sizes.

Computationl Resource. We use one RTX-2080Ti GPU for training on the two datasets, and training 20 epochs take less than one hour.

Chapter 9

Conclusion and Limitations Discussion

In this thesis, we studied *cross-view learning with limited supervision*. We provided both empirical and theoretical analyses that how we can leverage the information across different data views to learn good representations when having access to only limited supervision signals (e.g., without supervised labels or with only auxiliary information of data). This chapter provides a succinct summary of the main contributions, and then we discuss certain limitations of our work. Studying these limitations helps us better understand the topic - cross-view learning with limited supervision - and hence attempting to address these limitations can be potential future research directions.

9.1 Summary of Thesis Contributions

This thesis contributes in three folds. Firstly, we presented approaches that take into account the heterogeneous structures across views when modeling multi-view data. In particular, Chapter 3 introduced the Multimodal Transformer that attends to interactions between views across distinct time steps and latently correlates the cross-view signals. Chapter 4 introduced the Factorized Multimodal Model that disentangles multi-view data into multi-view discriminative factors and view-specific generative factors. We showed that, when considering the cross-view heterogeneous structures, we can learn representations that achieve better data generation, discriminative performance (i.e., multi-view prediction), and interpretability (both local and global interpretability) for the model.

Secondly, we presented approaches to quantify the relationships between different data views. In particular, Chapter 5 introduced tractable and scalable estimators to quantify the mutual information between two variables, with each variable representing a view of data. We showed that quantifying the mutual information not only helps us have a nicer understanding of the multi-view data but also enables us to better develop multi-view representation learning algorithms and associate cross-view instances (e.g., pairing an image with its associated caption).

Thirdly, we presented methods to learn good data representations by leveraging only limited supervision (e.g., the commonly available cross-view information) but not the downstream task label information. In particular, Chapter 6 manifested how we can learn the representations that can perform well on downstream tasks by utilizing only the pairing information in multi-view data. Then, Chapter 7 and Chapter 8 extended Chapter 6 to learn representations that further include the auxiliary information (e.g., hashtags for Instagram images) and exclude the unwanted information (e.g., personal information for privacy-sensitive data). This line of research is also called *self-supervised learning* and it opens a new horizon of learning good data representations without the expensive process of annotating the downstream task labels. Nonetheless, these self-supervised learning methods require a much higher computation cost comparing to the supervised representation learning methods. The high computational cost hinders the practical application in areas that have only low computational powers, such as learning representations using mobile phones.

9.2 Limitations and Future Research Directions

A first observation about this thesis is that we considered multi-view data from mostly two or three different views (e.g., the human multi-modal utterance with visual, acoustic, and textual views). While this was an important stepping stone, data can often include a larger number of views, such as signals for aircraft sensors that track oil temperature, fuel pressure, air speed measurement, lightening detection, vibration detection, etc. Some of our proposed multi-view representation learning approaches may have trouble to scale up to larger number of views, and we acknowledge this potential limitation as *representation learning with a large number of views*. Second, our empirical and theoretical analysis on self-supervised learning lie mainly within visual modality. In particular, we considered augmented variants as different views of an image, and then we analyzed why the self-supervised learned representations can reach good downstream performance and the practical deployments of different self-supervised objectives. Although we manifested good results in visual modality, we have not yet shown that our approaches can generalize to other modality, such as audio or textual modalities. We identify this limitation as *self-supervised learning beyond visual modality*. Lastly, when studying multi-view representation learning, the thesis focuses primarily on the task of perception and less about action generation (e.g., action generation for navigation). The perception and action generation procedures are two important phases for an intelligent agent, where the perception phase receives multi-view signals and transfers them into high-level representations, and the action generation phase takes the internal representations and generates actions. Our thesis does not directly tackle the problem of action generation in multi-view representation learning, such as the movement of a robot after receiving the multi-view sensory signals (e.g., visual inputs from camera or distance measurements from ultrasonic sensors). We identify this limitation as *multi-view representation learning for action generation*. In the following sub-section, we discuss these potential limitations and point towards promising future research directions.

9.2.1 Representation Learning with a Large Number of Views

Multi-view data may contain a large number of views. For instance, physiological data can include EEG, ECG, EMG, blood pressure, skin conductance, etc. Another example is climate data, which contains wind speed and direction, air temperature, relative humidity, barometric pressure, solar radiation, etc. Learning representations from these applications is particular challenging as most of the multi-view representation learning approaches are not specifically designed for very large number of views. This limitation also exists in this thesis, where our discussed multi-view representation learning methods were primarily evaluated with data that contains up to three views (e.g., the human multi-modal utterance that contains visual, textual, and acoustic views).

One future work direction to address this limitation would be to focus on designing the multi-view representation learning algorithms that are invariant to the number of views. Examples of such algorithms can be the early fusion method that simply concatenates the encoded representations from all the views, or the late fusion method that weighted averages the predictions from each view. Although the early and the late fusion methods can work with the multi-view data with a large number of views, they fail to consider the fine-grained interactions between views, such as the cross-modal attentional mechanism described in Chapter 3 that attends to interactions between views across distinct time steps and latently correlates the cross-view signals. However, the cross-modal attentional mechanism (also for most of the recent multi-view representations learning approaches) considers the pairwise cross-view interactions, and hence the number of interactions grows quadratically with the number of views, where the quadratic computation leads to heavy computation. To reduce this heavy computation, we shall re-design multi-view representation learning algorithms to consider the number of the interactions that grows constantly with respect to the number of views.

A second future work to address this limitation will be studying self-supervised learning from multi-view data with a large number of views. In particular, Chapters 6, 7, and 8 presented empirical and theoretical analysis on how we can leverage cross-view information from bi-view data to learn good self-supervised representations. A key contribution in these chapters is that: we show that Mutual Information can quantify the cross-view relationships and it connects to lots of self-supervised representation learning methods. Nonetheless, the Mutual Information only works for two variables, and hence it cannot be used to study self-supervised learning for data with the number of views more than two. Here, we point out a potential path to address this limitation: we can study the usage of Interaction Information (in replacement of Mutual Information) for self-supervised learning, where the Interaction Information works to quantify the joint relationships between three variables.

9.2.2 Self-supervised Learning beyond Visual Modality

In Chapters 6, 7, and 8, we studied self-supervised learning from a multi-view perspective within the visual modality. In particular, we apply different image augmentations on an image, and then we treat the augmented variants of the same image as different views to each other. A core premise in our analysis is that different views provide approximately the same amount of task-relevant information (see Chapter 6). The premise holds true under our setup due to two common assumptions. The first assumption is that applying image augmentations will only affect the style of the image but not the content, and the second assumption is that what matters for the downstream tasks is the content but not the style. Unfortunately, the premise may not hold true for the self-supervised learning settings for other modalities, such as the textual and the acoustic modalities. For instance, the BERT model [Devlin et al., 2018] (one of the most famous self-supervised model for text) considers the masked and the non-masked words as different views of the textual data, and it is hardly true that the masked and the non-masked words contain the same amount of the task-relevant information. Moreover, compared to the development of visual self-supervised learning [Arora et al., 2019, Chen et al., 2020a,c, Grill et al., 2020, He et al., 2019, Oord et al., 2018, Tsai et al., 2020c, 2021c, Zbontar et al., 2021], the progress for textual self-supervised learning [Devlin et al., 2018, Peters et al., 2018, Yang et al., 2019] and acoustic self-supervised learning [Baevski et al., 2020, Hsu et al., 2021, Schneider et al., 2019] is relatively slow.

A future work is to provide theoretical understanding on why recent textual and acoustic self-supervised learning approaches work. Although we have no a complete answer now, we point out two recent theoretical work [Lee et al., 2020, Teng and Huang, 2021] that may provide some intuitions for this direction. First, Lee et al. [2020] showed that if the self-supervised objective is to predict a partial of the data from the rest portion of the data, then this objective may learn the representations that perform well on downstream tasks. This work can possibly explain the success of the BERT model [Devlin et al., 2018], which presented the objective to predict the masked words from the non-masked words in text corpora. Nonetheless, its analysis cannot be applied to the auto-regressive text pre-training as in GPT [Radford et al., 2018] or the permutation text pre-training as in BART [Lewis et al., 2019]. Second, Teng and Huang [2021] showed that if the self-supervised objectives is to perform context prediction that require high-level semantic understand of data, then the learned representation can achieve good performance on downstream tasks. This work can explain the success of the permutation text pre-training objective as in BART [Lewis et al., 2019], while it fails to explain the success for the masking text pre-training objective as in BERT [Devlin et al., 2018] or the auto-regressive text pre-training as in GPT [Radford et al., 2018]. Our goal is to provide a more general theoretical understanding for the success of textual and acoustic self-supervised learning approaches. The understanding can even lead us to design self-supervised learning methods for data from a wide range of modalities, such as physiological signals and 3D point clouds.

A second future work direction is to improve the existing textual and acoustic self-supervised learning algorithms. We can see that recent advances have improved visual self-supervised representation learning from different aspects. For example, Grill et al. [2020], He et al. [2019] presented to improve the robustness to the training batch size, Tsai et al. [2021c], Zbontar et al. [2021] presented to increase the training stability, Chapter 8 showed how we can exclude unwanted information, and Chapter 7 showed how we can include auxiliary information in the self-supervised representations. Unfortunately, all these successes are done for the visual self-supervised learning approaches. Our goal is to bring them to the textual and acoustic self-supervised learning approaches. Now, we discuss an example of adapting an existing trick from visual to textual self-supervised representation learning. He et al. [2019] introduced the momentum encoder that performs a momentum parameters update to improve the robustness to the training batch size of images. Inspired by this idea, we can thus consider a momentum parameters update for the encoder in the textual or acoustic self-supervised model and see how it can increase the robustness to the training batch size for textual and acoustic data.

9.2.3 Multi-view Representation Learning for Action Generation

An intelligent agent can perceive its environment and act on it. At a colloquial level, the perception phase encodes the real-world observations (i.e., multi-view observations) into internal representations, and the action generation phase decodes the internal representations into actions. So far, our thesis focuses primarily on the representation learning phase and do not consider generating the actions from multi-view data. Put it another way, we plan to move from *non-embodied* AI to *embodied* AI. Specifically, this thesis' focus belongs to non-embodied AI with the limitations that we do not take the interactions with the real world into account. For example, the methods we discussed can work well in controlled conditions (e.g., the multi-media signals captured by a location-fixed camera), yet they may fail to adapt to the constantly changing situations (e.g., the multi-media signals captured by a moving robots).

A future work is to extend the study on multi-view representation learning and action generation in interactive multi-view environments. Examples of such environments are the autonomous systems (that collect Lidar, Radar, and RGB signals), the household robots (that receive acoustic, visual, and tactile signals), and the AI assistant in medical surgeries (that receive different kinds of physiology data). Different from the work in this thesis that focuses on only a single prediction (e.g., predicting the sentiment from human multi-modal utterance), there contain multiple and sequential decision-making processes in these interactive multi-view environments. An action that is generated at a particular time step may affect the current state of the agent, and then the agent can possibly receive different multi-view signals. The interactive nature makes this future direction particularly challenging, yet there exists much broader applications. For instance, a popular topic in visual navigation is how we can leverage the information in simulated environment to improve the policy in the real world. Recent method [Li et al., 2020b] presents to leverage the RGB images in simulated environment to improve the navigation in the real-world. An extension is that we can further take the depth image (as an alternative view to the RGB images) into account for further improving the visual navigation.

9.2.4 Conclusion

In this chapter, we studied three potential limitations (representation learning with a large number of views, self-supervised learning beyond visual modality, and multi-view representation learning for action generation) of our thesis work on the topic - *cross-view learning with limited supervision*. We also discussed the potential future work to address these limitations. Nonetheless, in addition to the studied challenges (heterogeneous structures, relationship quantification, and learning with limited supervision) and the discussed limitations, there still remain lots of challenges and sub-challenges waiting for us to solve. To conclude, we hope that our thesis sheds light on advantages of leveraging information across different data views to learn good representations. We also believe that our work can potentially open up new horizons for learning representations when having access to limited supervision signals.

Bibliography

Heritage health prize. URL https://www.kaggle.com/c/hhp. 8.5.2.2

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 5.2.2, 5.7.3.1

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018. 6.1.2, 6.1.2, 6.5.1

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45, 2015. 5.4

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009. 5.7.2.3

Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017. 5.4, 6.1.1

Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018. 2.3

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019. 2.3, 2.3, 5.4, 6, 6.3, 8, 8.1.1, 9.2.2

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3.2.2

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019. (document), 5, 5.4, 5.2, 5.4, 5.7.3.4, 5.7.3.4, 6, 6.1.3, 6.2, 6.3, 6.5.6.2, 6.5.6.3, 8.1.1

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations (ICLR)*, 2019. 3.1

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020. 2.3, 7, 7.1, 9.2.2

Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in neural information processing systems*, pages 89–96, 2005. 6.3

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019. 2.1

David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003. 2.2, 2.2, 5.1

Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2): 525–536, 1998. 5.3, 5.7.2.3, 6.5.3

Ayanendranath Basu, Ian R Harris, Nils L Hjort, and MC Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998. 5.1

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018. 2.2, 2.2, 5, 5.1, 5.2.1, 5.3, 5.1, 5.3, 5.3, 5.7.1.1, 3, 4, 5.7.2.1, 5.7.2.1, 3, 6.5.3

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8), August 2013. 4.3

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 2.1

Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009. 2.2, 5.1, 5.2.1

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967. 5.1

Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324. URL `https://doi.org/10.1023/A:1010933404324`. 4.5.2

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018. 2.3

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth S. Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Journal of Language Resources and Evaluation*, 42(4):335–359, dec 2008a. doi: 10.1007/s10579-008-9076-6. 4.2.2

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335, 2008b. 3, 3.3.1

Flavio P Calmon, Dennis Wei, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized data pre-processing for discrimination prevention. *arXiv preprint arXiv:1704.03354*, 2017. 8.3

Cristian S Calude. *Information and randomness: an algorithmic perspective*. Springer Science & Business Media, 2013. 6.1.2

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 7.1

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. (document), 2.3, 7.1, 7.2.2, 7.5.4, **??**, 7.2

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. *arXiv* *preprint arXiv:1706.07230*, 2017. 4.3

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018a. 5

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, and Heewoo Jun. Generative pretraining from pixels. 6.1.1, 6, 6.3

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*, 2018b. 3.2.1

Minghai Chen, Sen Wang, Paul Pu Liang, Tadas Baltrušaitis, Amir Zadeh, and Louis-Philippe Morency. Multimodal sentiment analysis with word-level fusion and reinforcement learning. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, ICMI, 2017. 4.3

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020a. (document), 2.3, 2.3, 2.3, 5.4, 5.7.3.4, 6, 6.1.1, 6.1.1, 6.1.3, 6.1.3, 6.2, 6.3, 6.5.6.3, 6.6, 7, 7.1, 7.2.2, 7.2.2, 7.2.3, 7.2.3, 7.3.2, **??**, 7.4, 7.3.3, **??**, 7.3.4, **??**, 7.5.5, **??**, 8, 8.1, 8.1.1, 8.1.1, 8.3, 8.2.3, 8.3, 1, 8.5.2.3, 9.2.2

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b. 8.1, 8.3

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016. 4.1.2, 4.3, 5

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c. 9.2.2

Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. *arXiv preprint arXiv:1412.6583*, 2014. 4.3

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*. ACL, 2014. (document), 4.1.4, 4.2.2, 4.5.6, 4.5

Yu-An Chung and James Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *arXiv preprint arXiv:1803.08976*, 2018. 5.5, 5.7.4

Yu-An Chung and James Glass. Generative pre-training for speech with autoregressive predictive coding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3497–3501. IEEE, 2020. 2.3

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990. 2.2, 5.1

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999. 1.1, 2.2, 8.1.2

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 6, 6.5.2, 6.5.4

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013. 7.5.4

Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. A kernel for time series based on global alignments. In *Acoustics, Speech and Signal Processing, ICASSP 2007.*, 2007. 4.5.4

Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Language modeling with longer-term dependency. 2018. 3.1, 3.2.1

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 5

JJ Daudin. Partial association measures and an application to qualitative regression. *Biometrika*, 67(3): 581–590, 1980. 8.1.3

Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. Covarepa collaborative voice analysis repository for speech technologies. In *ICASSP*. IEEE, 2014. 3.3.1, 3.5.4, 4.2.2, 4.5.3

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 6.2

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2.1, 2.3, 2.3, 2.3, 2.3, 3.1, 3.2.1, 6, 6.1.1, 6.1.1, 6.1.3, 6.2, 7, 7.1, 8.3, 9.2.2

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2.3

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983. 2.2, 2.2, 5.1, 5.2.1, 5.1, 4

Jun Du, Charles X Ling, and Zhi-Hua Zhou. When does cotraining work in real data? *IEEE Transactions on Knowledge and Data Engineering*, 23(5):788–799, 2010. 6.3

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`. 8.2.2, 8.5.2.2

Sri Harsha Dumpala, Rupayan Chakraborty, and Sunil Kumar Kopparapu. Audio-visual fusion for sentiment classification using cross-modal autoencoder. *NIPS*, 2019. 3.1

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012. 8.3

Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992. 3.5.4, 4.5.3

Paul Ekman, Wallace V Freisen, and Sonia Ancoli. Facial signs of emotional experience. *Journal of personality and social psychology*, 39(6):1125, 1980. 3.5.4, 4.5.3

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006. 6.5.7

Meir Feder and Neri Merhav. Relations between entropy and error probability. *IEEE Transactions on Information Theory*, 40(1):259–266, 1994. 6.1.4, 6.5.4, 6.5.4

M Federici, A Dutta, P Forré, N Kushmann, and Z Akata. Learning robust representations via multi-view information bottleneck. International Conference on Learning Representation, 2020. 6.3, 6.5.1

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 4.3

Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99, 2004. 8.1.3

Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *NIPS*, volume 20, pages 489–496, 2007. 8.1.3

Kathleen R Gibson, Kathleen Rita Gibson, and Tim Ingold. *Tools, language and cognition in human evolution*. Cambridge University Press, 1994. 3

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2.3, 2.3, 6.1.1

Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011. 2.1

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5.1, 8.5.2.4

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011. 8.3

A. Graves, A. r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013. 4.5.2

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006. **??**, 3.3.2, **??**, **??**, 3.5.2

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005a. 2.2, 4.5.4

Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, and Bernhard Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6(Dec):2075–2129, 2005b. 8b

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012. 2.2, 4.1.2

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2.3, 9.2.2

Yue Gu, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. Multimodal affective analysis using hierarchical attention strategy with word-level alignment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2018. 3, 3.1

Sylvain Guimond and Wael Massrieh. Intricate correlation between body posture, personality trait and incidence of body pain: A cross-referential study report. *PLOS ONE*, 2012. 4.5.4

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *arXiv preprint* *arXiv:1610.02413*, 2016. 8.3

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR, 2020. 2.3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2.1, 6.2, 7.3.2, 8.2.3, 1

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2.3, 2.3, 2.3, 5.4, 6, 6.1.1, 6.1.3, 6.1.3, 6, 9.2.2

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. (document), 7, 7.1, 7.2.2, 7.2.3, **??**, 7.3.3, **??**, **??**, 8.1.1, 8.1.1, 8.3, 8.2.3, 8.3

Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 5.4, 5.4, 5.7.3.4, 6.2

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 4.1.2, 4.1.2, 4.3

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 5.1, 5.2.1, 5.4, 5.4, 5.7.3.4, 6.1.1, 6.1.3, 3, 6.2, 6.3

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2.1, 2.3, 4.1.4, 4.5.2

Robert V Hogg, Joseph McKean, and Allen T Craig. *Introduction to mathematical statistics*. Pearson Education, 2005. 2.2

K Hornik, M Stinchcombe, and H White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 5.3, 5.3, 4, 5.7.1.1, 5.7.1.1, 5.7.1.1, 5.7.1.2, 5.7.1.3, 5.7.1.4, 9, 6.5.3

Wei-Ning Hsu and James Glass. Disentangling by partitioning: A representation learning framework for multimodal sensory data. *arXiv preprint arXiv:1805.11264*, 2018. 4.3

Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: How much can a bad teacher benefit asr pre-training. In *Neural Information Processing Systems Workshop on Self-Supervised Learning for Speech and Audio Processing Workshop*, 2020. 7.1

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*, 2021. 9.2.2

iMotions. Facial expression analysis, 2017. URL `goo.gl/1rh1JN`. 3.3.1, 3.5.4, 4.2.2, 4.5.3

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021. 8.3

Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1413–1421, 2015. 5.4

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009. 5.2.2

Theofanis Karaletsos, Serge Belongie, and Gunnar Rätsch. Bayesian representation learning with oracle constraints. *arXiv preprint arXiv:1506.05011*, 2015. 4.3

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 7, 7.2.3, 7.2.3, **??**, 7.3.4, 7.5.5, **??**

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5.7.2.2, 5.7.4, 8.5.2.1

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4.1.2, 4.1.2, 4.3, 5.1

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019. 8.3

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019. 5.4, 5.7.3.4

Lingpeng Kong, Cyprien de Masson d'Autume, Wang Ling, Lei Yu, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. *arXiv preprint arXiv:1910.08350*, 2019. 5.4

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004. 5

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 5.4, 5.7.3.4, 6, 6.2, 8.2.3, 8.5.2.3

Patricia K. Kuhl. A new view of language acquisition. *Proceedings of the National Academy of Sciences*, 2000. doi: 10.1073/pnas.97.22.11850. 4.2.2

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, 2015. 4.3

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 6.2

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017. 4.3

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 8.3

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. Combining language and vision with a multimodal skip-gram model. *arXiv preprint arXiv:1501.02598*, 2015. 3.1

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7:7, 2015. 8.2.3, 8.5.2.3

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5.4, 5.7.3.4

Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 4.2.1

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017. 2.3

Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *arXiv preprint arXiv:2008.01064*, 2020. 2.3, 6, 6.3, 9.2.2

Sergey Levine, Christian Theobalt, and Vladlen Koltun. Real-time prosody-driven synthesis of body language. *ACM Trans. Graph.*, 28(5):172:1–172:10, December 2009. 4.5.4

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014. 2.2, 5.1

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019. 2.3, 9.2.2

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015. 5.1

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020a. 7, 7.1, 7.3.3, **??**, 7.5.4, **??**

Shangda Li, Devendra Singh Chaplot, Yao-Hung Hubert Tsai, Yue Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Unsupervised domain adaptation for visual navigation. *arXiv preprint arXiv:2010.14543*, 2020b. 9.2.3

Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *IEEE transactions on knowledge and data engineering*, 31(10):1863–1883, 2018. 2.1

Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *European Conference on Computer Vision*, 2016. 7, 7.3.1, 7.5.3.2

Paul Pu Liang, Ziyin Liu, Amir Zadeh, and Louis-Philippe Morency. Multimodal language analysis with recurrent multistage fusion. *EMNLP*, 2018a. 3.1, 3.2.3, 3.3, 3.3.1, **??**, **??**

Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Multimodal local-global ranking fusion for emotion recognition. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI, 2018b. 4.3

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6.2

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017. 3.1

Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. 6.3

Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *ACL*, 2018. 4.3

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 5

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 8.3

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pages 3384–3393. PMLR, 2018. 8, 8.2.2, 8.3

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 7.1, 7.2.3

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Prismatic Inc, Steven J. Bethard, and David Mcclosky. The stanford corenlp natural language processing toolkit. In *In ACL, System Demonstrations*, 2014. 3

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884, 2020. 2.2, 2.2, 6.2

Daniel McNamara, Cheng Soon Ong, and Robert C Williamson. Costs and benefits of fair representation learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 263–270, 2019. 8.3

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019. 8

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2.3, 5.5, 5.7.4

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 7, 7.2.1, 7.3.5

Rudy Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal processing*, 16(3):233–248, 1989. 2.2

Gelareh Mohammadi, Alessandro Vinciarelli, and Marcello Mortillaro. The voice of personality: Mapping nonverbal vocal behavior into trait attributions. In *Proceedings of ACM Multimedia Workshop on Social Signal Processing*, 0 2010. 4.5.4

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019. 5.7.4

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020. 2.2

Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR*. IEEE, 2007. 4.5.2

Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 169–176. ACM, 2011. 4.1.4, 4.2.2, 4.2.2, 4.5.2

Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. Ccmi: Classifier based conditional mutual information estimation. In *Uncertainty in Artificial Intelligence*, pages 1083–1093. PMLR, 2020. 6.5.1, 8.5.2.4, 8.5.2.4, 8.5.2.4, 8.5.2.4

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4.2.1

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, 2011. 3.1, 4, 4.3

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11): 5847–5861, 2010. 2.2, 2.2, 5.1, 5.1, 3, 5.7.2.1, 5.7.2.1, 16, 17

Behnaz Nojavanasghari, Deepak Gopinath, Jayanth Koushik, Tadas Baltrušaitis, and Louis-Philippe Morency. Deep multimodal fusion for persuasiveness prediction. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ICMI 2016. ACM, 2016. (document), 4.1.4, 4.2.1, 4.5.2, 4.5.6, 4.5

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2.3, 2.3, 6.1.1

Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017. 2.3, 2.3

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016. 5.2.1, 5.1, 5.7.2.1

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2.2, 2.3, 5.3, 5.1, 5.3, 5.4, 5.4, 5.4, 5.7.2, 9, 5.7.2.1, 5.7.2.2, 5.7.3.3, 5.7.3.4, 5.7.4, 6, 6.1.3, 6.1.3, 3, 6.2, 6.3, 6.5.6.2, 7, 7.1, 7.2.2, 7.2.3, 8, 8.1, 8.1.1, 4, **??**, 8.2.1, 8.2.1, **??**, **??**, **??**, 8.2.3, 8.3, 8.5.2.2, **??**, **??**, 9.2.2

Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018. 2.3

Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. In *Advances in Neural Information Processing Systems*, pages 15578–15588, 2019. 5.4, 5.4, 7.1

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010. 8

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015. 8.2.1, 8.5.2.1

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016. 3.1

Sunghyun Park, Han Suk Shim, Moitreya Chatterjee, Kenji Sagae, and Louis-Philippe Morency. Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, pages 50–57, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2885-2. doi: 10.1145/2663204.2663260. URL `http://doi.acm.org/10.1145/2663204.2663260`. 4.2.2

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 5.2.2, 5.7.3.1

Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 8.3

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 6.1.3, 6.3

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017. 2.3

Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005. 5

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 2.3, 3.3.1, 3.5.4, 4.2.2, 4.5.3

Veronica Perez-Rosas, Rada Mihalcea, and Louis-Philippe Morency. Utterance-Level Multimodal Sentiment Analysis. In *Association for Computational Linguistics (ACL)*, August 2013. 4.2.2

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018. 2.3, 2.3, 6.1.3, 7, 7.1, 9.2.2

Hai Pham, Thomas Manzini, Paul Pu Liang, and Barnabas Poczos. Seq2seq2sentiment: Multimodal sequence to sequence models for sentiment analysis. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*. ACL, 2018. 4.3

Hai Pham, Paul Pu Liang, Thomas Manzini, Louis-Philippe Morency, and Barnabas Poczos. Found in translation: Learning robust joint representations by cyclic translations between modalities. *AAAI*, 2019. 3, 3.1, 3.2.3, 3.3, 3.3.1, 3.3.1, **??**, **??**, 3.3.2, 3.3.3, **??**, **??**, **??**, **??**, 3.3.3

Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *European Conference on Computer Vision*, pages 3–18. Springer, 2016. 2.3

Barnabás Póczos and Jeff Schneider. Nonparametric estimation of conditional information and divergences. In *Artificial Intelligence and Statistics*, pages 914–923. PMLR, 2012. 6.2

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019. 2.2, 2.2, 5, 5.1, 5.2.1, 5.2.1, 5.2.2, 5.3, 5.1, 5.3, 5.3, 5.4, 5.7.1.1, 5, 5.7.1.1, 5.7.2.1, 5.7.2.1, 5.7.2.2, 5.7.4, 6.1.3, 3, 8.1.1

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *ACL*, 2017a. 4.5.2

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 873–883, 2017b. 3, 3.3.1, 3.3.1

Ariadna Quattoni, Sybor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007. 4.5.2

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 2.3, 2.3, 9.2.2

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish

Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2.3, 7.1

Shyam Sundar Rajagopalan, Louis-Philippe Morency, Tadas Baltrušaitis, and Goecke Roland. Extending long short-term memory for multi-view structured learning. In *ECCV*, 2016. 2.1, 4.5.2

Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, 2014. 4.3

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. Unsupervised pretraining transfers well across languages. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7414–7418. IEEE, 2020. 8.2.1, 8.2.1, 8.3, 8.5.2.1

Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of wasserstein autoencoders. *arXiv preprint arXiv:1802.03761*, 2018. 4.1.2, 4.1.2, 4.3

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 7, 7.3.1, 7.5.3.4, 7.5.5

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019. 2.3, 7, 7.1, 9.2.2

Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. 2002. 2.1

M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11), November 1997. 4.5.2

George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012. 2.2

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 6.5.3

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. 2018. 3.2.1

Imran Sheikh, Sri Harsha Dumpala, Rupayan Chakraborty, and Sunil Kumar Kopparapu. Sentiment analysis using imperfect views from spoken language and acoustic modalities. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*, pages 35–39, 2018. 3.3.3

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017. 8.3

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. 4.3

Kihyuk Sohn, Wenling Shang, and Honglak Lee. Improved multimodal deep learning with variation of information. In *Advances in Neural Information Processing Systems*, pages 2141–2149, 2014. 4.3

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019. 2.2, 2.2, 5, 5.1, 5.2.1, 5.2.2, 5.3, 5.1, 5.3, 5.3, 5.4, 5.7.2.1, 5.7.2.1, 5.7.2.1, 5.7.2.1, 5.7.2.2, 5.7.3.2, 5.7.4, 6.1.3, 6.2

Jiaming Song and Stefano Ermon. Multi-label contrastive predictive coding. *arXiv preprint arXiv:2007.09852*, 2020. 11

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2164–2173. PMLR, 2019. 8, 8.2.2, **??**, **??**, 8.3, 8.5.2.2, 8.5.2.2, 8.5.2.2, 8.5.2.2, 8.5.2.2, **??**, **??**, **??**, **??**

Yale Song, Louis-Philippe Morency, and Randall Davis. Multi-view latent variable discriminative models for action recognition. In *CVPR*, 2012. 4.5.2

Yale Song, Louis-Philippe Morency, and Randall Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, 2013. 4.5.2

Mohammad S Sorower. A literature survey on algorithms for multi-label learning. 6.5.7

Karthik Sridharan and Sham M Kakade. An information theoretic framework for multi-view learning. 2008. 6, 6.1.1, 5

Nitish Srivastava and Russ R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012. 2.1, 3.1, 4, 4.2.2, 4.3

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/D18-1548`. 3.1

Masashi Sugiyama and Makoto Yamada. On kernel parameter selection in hilbert-schmidt independence criterion. *IEICE TRANSACTIONS on Information and Systems*, 2012. 4.5.4

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008. 5.2.1

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012a. 5, 5.1

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012b. 5, 5.1

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 7.1

Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016. 4.3

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. Why self-attention? a targeted evaluation of neural machine translation architectures. *arXiv preprint arXiv:1808.08946*, 2018. 3.1

Jiaye Teng and Weiran Huang. Can pretext-based self-supervised learning be boosted by downstream data? a theoretical analysis. *arXiv preprint arXiv:2103.03568*, 2021. 2.3, 9.2.2

Bruce Thompson. *Canonical correlation analysis: Uses and interpretation*. Number 47. Sage, 1984. 2.1

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 5.4, 5.4, 5.7.3.4, 6, 6.1.3, 6.2, 6.5.6.3

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 6.1.2, 6.3

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv* *preprint physics/0004057*, 2000. 6.1.2, 6.1.2, 6.5.1

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017. 4.1.2, 4.1.2, 4.5.1.1

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. *arXiv preprint arXiv:2008.10150*, 2020. 2.3, 6, 6.3

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021. 8.1.1

Yao-Hung Hubert Tsai and Ruslan Salakhutdinov. Improving one-shot learning through fusing side information. *NeurIPS LLD*, 2017. 1, 1b, 1d, 4.3

Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Learning cross-domain landmarks for heterogeneous domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5081–5090, 2016. 2.1

Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3571–3580, 2017a. 1, 1a, 1b, 1c

Yao-Hung Hubert Tsai, Han Zhao, Ruslan Salakhutdinov, and Nebojsa Jojic. Learning markov chain in unordered dataset. *NeurIPS TSW 2017*, 2017b. 2, 2a, 2b, 2c, 4.3

Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis-Philippe Morency, and Ruslan Salakhutdinov. Learning factorized multimodal representations. *arXiv preprint arXiv:1806.06176*, 2018. 3, 3.1, 3.2.3, 3.3, 3.3.1, 3.3.1, 3.3.1, **??**, 3.3.3, **??**, 5

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019a. 1a, 2.1

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: A unified understanding for transformer's attention via the lens of kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4335–4344, 2019b. 4, 4a, 4b

Yao-Hung Hubert Tsai, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi. Video relationship reasoning using gated spatio-temporal energy graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10424–10433, 2019c. 3, 3a, 3b, 3c

Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis-Philippe Morency, and Ruslan Salakhutdinov. Learning factorized multimodal representations. In *International Conference on Representation Learning*, 2019d. 2a

Yao-Hung Hubert Tsai, Han Zhao, Russ R Salakhutdinov, and Geoffrey J Gordon. Learning neural networks with adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 11393–11404, 2019e. 5, 5a, 5b, 5c

Yao-Hung Hubert Tsai, Martin Q Ma, Muqiao Yang, Ruslan Salakhutdinov, and Louis-Philippe Morency. Interpretable multimodal routing for human multimodal language. *arXiv preprint arXiv:2004.14198*, 2020a. 6, 6b

Yao-Hung Hubert Tsai, Nitish Srivastava, Hanlin Goh, and Ruslan Salakhutdinov. Capsules with inverted dot-product attention routing. In *International Conference on Learning Representations*, 2020b. 6, 6a

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Demystifying self-supervised learning: An information-theoretical framework. *arXiv preprint arXiv:2006.05576*, 2020c. 4a, 5.4, 9.2.2

Yao-Hung Hubert Tsai, Han Zhao, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Neural methods for point-wise dependency estimation. *arXiv preprint arXiv:2006.05553*, 2020d. 3a, 10, 6.5.3, 8.5.2.4

Yao-Hung Hubert Tsai, Shaojie Bai, Louis-Philippe Morency, and Ruslan Salakhutdinov. A note on connecting barlow twins with negative-sample-free contrastive learning. *arXiv preprint arXiv:2104.13712*, 2021a. 8, 8b, 8.5.2.3

Yao-Hung Hubert Tsai, Tianqin Li, Weixin Liu, Peiyuan Liao, Ruslan Salakhutdinov, and Louis-Philippe Morency. Integrating auxiliary information in self-supervised learning. *arXiv preprint arXiv:2106.02869*, 2021b. 5a

Yao-Hung Hubert Tsai, Martin Q Ma, Muqiao Yang, Han Zhao, Louis-Philippe Morency, and Ruslan Salakhutdinov. Self-supervised representation learning with relative predictive coding. *arXiv preprint arXiv:2103.11275*, 2021c. 7, 7a, 7.1, 8.1.1, 9.2.2

Yao-Hung Hubert Tsai, Martin Q Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning: Removing undesirable information in self-supervised representations. *arXiv preprint arXiv:2106.02866*, 2021d. 6a

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. In *ICLR*, 2021e. 7.2.3, 8, 8.1.1, 8.1.1, 8.3

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019. (document), 2.2, 5, 5.1, 5.2, 5.4, 5.4, 5.7.3.1, 5.7.3.4, 5.7.3.4, 6.1.2, 3, 6.2, 8, 8.1, 8.1.1

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 6, 6.1.3, 5

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 8

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000. 5.7.5

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2.3, 3, 3.1, 3.2.1, 3.2.1, 3.2.2, 3.2.2, 3.2.2, 3.5.1, 5

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018. 2.3

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 6, 6.1.3, 6.3

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 7, 7.3.1, 7.5.3.3

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 8.3

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 3.1

Yansen Wang, Ying Shen, Zhun Liu, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Words can shift: Dynamically adjusting word representations using nonverbal behaviors. *AAAI*, 2019. 3.1, 3.2.3, 3.3.1, 3.3.1, **??**, **??**, 3.3.2, 3.3.3, **??**, **??**, **??**, **??**

Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision*, pages 616–634. Springer, 2016. 8

Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013. 2.2

Yandong Wen, Mahmoud Al Ismail, Weiyang Liu, Bhiksha Raj, and Rita Singh. Disjoint mapping network for cross-modal matching of voices and faces. *arXiv preprint arXiv:1807.04836*, 2018. 7.1

Martin Wöllmer, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun, Kenji Sagae, and Louis-Philippe Morency. Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, 28(3):46–53, 2013. 4.2.2

Denny Wu, Yixiu Zhao, Yao-Hung Hubert Tsai, Makoto Yamada, and Ruslan Salakhutdinov. " dependency bottleneck" in auto-encoding architectures: an empirical study. *arXiv preprint arXiv:1802.05408*, 2018a. 4.5.4

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018b. 2.3

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 8.2.3, 8.5.2.3

Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013. 2.1, 6.1.1

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015. 2.1

Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NIPS*, 2015. 4.3

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019. 2.3, 9.2.2

Sho Yokoi, Sosuke Kobayashi, Kenji Fukumizu, Jun Suzuki, and Kentaro Inui. Pointwise hsic: A linear-time kernelized co-occurrence norm for sparse linguistic expressions. *arXiv preprint arXiv:1809.00800*, 2018. 5.1

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 8.5.2.3

Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 7, 7.3.1, 7.5.3.1

Dong Yu and Li Deng. *AUTOMATIC SPEECH RECOGNITION.* Springer, 2016. 2.1

Lei Yu, Jan Buys, and Phil Blunsom. Online segment to segment neural transduction. *arXiv preprint arXiv:1609.08194*, 2016. 3.2.3

Jiahong Yuan and Mark Liberman. Speaker identification on the scotus corpus. *Journal of the Acoustical Society of America*, 123(5):3878, 2008. 3.3.1, 4.5.3

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88, 2016. 3, 3.3.1, 4.2.2, 4.5.2

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. In *EMNLP*, 2017. 4, 4.1.4, 4.2.2, 4.2.2, 4.3, 4.5.2

Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Memory fusion network for multi-view sequential learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018a. (document), 3, 3.3.1, 3.3.1, 4.1.4, 4.2.2, 4.5.2, 4.5.3, 4.5.6, 4.5

Amir Zadeh, Paul Pu Liang, Soujanya Poria, Prateek Vij, Erik Cambria, and Louis-Philippe Morency. Multi-attention recurrent network for human communication comprehension. *AAAI*, 2018b. 4.5.2

AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *ACL*, 2018c. 3, 3.2.3, 3.3.1, **??**

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 8a, 2.3, 9.2.2

Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, 2010. 4.5.5

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013. 8

Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018. 2.3

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827. PMLR, 2013. 8.3

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2.3, 2.3, 2.3, 6, 6.1.1, 6.1.3, 6.3

Xiujun Zhang, Xing-Ming Zhao, Kun He, Le Lu, Yongwei Cao, Jingdong Liu, Jin-Kao Hao, Zhi-Ping Liu, and Luonan Chen. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics*, 28(1):98–104, 2012. 5

Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *arXiv preprint arXiv:1906.08386*, 2019. 8

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018. 2.3

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017. 4.1.2

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015. 6.2

Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *NIPS*, 2014. 4.3

Fuzhen Zhuang, Xiaohu Cheng, Ping Luo, Sinno Jialin Pan, and Qing He. Supervised representation learning: Transfer learning with deep autoencoders. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 8.3