Deep Variation-structured Reinforcement Learning for Visual Relationship and Attribute Detection

Lisa Lee, DAP Fall 2018
Joint work with Xiaodan Liang & Eric Xing

Abstract

Despite progress in visual perception tasks such as image classification and detection, computers still struggle to understand the interdependency of objects in the scene as a whole, e.g., relations between objects or their attributes. Existing methods often ignore global context cues capturing the interactions among different object instances, and can only recognize a handful of types by exhaustively training individual detectors for all possible relationships. To capture such global interdependency, we propose a deep Variation-structured Reinforcement Learning (VRL) framework to sequentially discover object relationships and attributes in the whole image. First, a directed semantic action graph is built using language priors to provide a rich and compact representation of semantic correlations between object categories, predicates, and attributes. Next, we use a variation-structured traversal over the action graph to construct a small, adaptive action set for each step based on the current state and historical actions. In particular, an ambiguity-aware object mining scheme is used to resolve semantic ambiguity among object categories that the object detector fails to distinguish. We then make sequential predictions using a deep RL framework, incorporating global context cues and semantic embeddings of previously extracted phrases in the state vector. Our experiments on the Visual Relationship Detection (VRD) dataset and the large-scale Visual Genome dataset validate the superiority of VRL, which can achieve significantly better detection results on datasets involving thousands of relationship and attribute types. We also demonstrate that VRL is able to predict unseen types embedded in our action graph by learning correlations on shared graph nodes.

1 Introduction

Although much progress has been made in image classification [7], detection [20] and segmentation [15], we are still far from reaching the goal of holistic scene understanding—that is, a model capable of recognizing the interactions and relationships between objects, and describing their attributes. While objects are the core building blocks of an image, it is often the relationships and attributes that determine the holistic interpretation of the scene. For example in Fig. 1, the left image can be understood as "a man standing on a yellow and green skateboard", and the right image as "a woman wearing a blue wet suit and kneeling on a surfboard". Being able to extract and exploit such visual information would benefit many real-world applications such as image search [19], question answering [1, 9], and fine-grained recognition [27, 4].

Visual relationships are a pair of localized objects connected via a predicate; for example, predicates can be actions ("kick"), comparative ("smaller than"), spatial ("near to"), verbs ("wear"), or prepositions ("with"). Attributes describe a localized object, e.g., with color ("yellow") or state ("standing"). Detecting relationships and attributes is more challenging than traditional object detection [20] due to the following reasons: (1) There are a massive number of possible relationship and attribute types (e.g., 13,894 relationship types in Visual Genome [13]), resulting in a greater skew of rare and infrequent types. (2) Each object can be associated with many relationships and attributes, making it inefficient to exhaustively search all possible relationships for each pair of objects. (3) A global, holistic perspective of the image is essential to resolve semantic ambiguities (e.g., "woman wearing wetsuit" vs. "woman wearing shirt"). Existing approaches [22, 10, 16] only predict a limited set of relationship types (e.g., 13 in Visual Phrase [22]) and ignore semantic interdependencies between relationships and attributes by evaluating each region within a

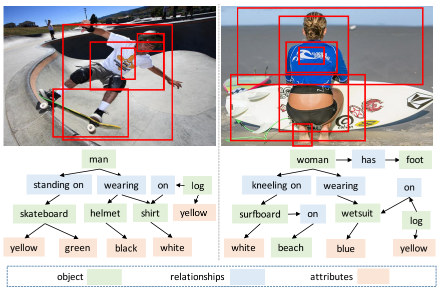

Figure 1: In each example (left and right), we show the bounding boxes of objects in the image (top), and the relationships and attributes recognized by our proposed VRL framework (bottom). Only the top few results are illustrated for clarity.

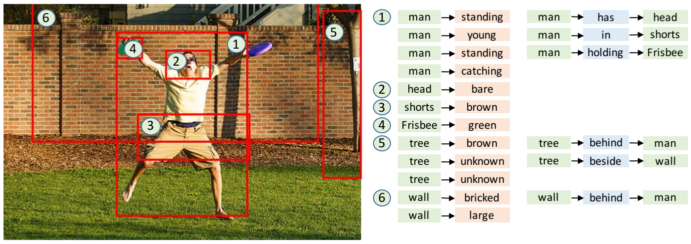

Figure 2: The VRL does a sequential breadth-first search, predicting all relationships and attributes with respect to the current subject instance before moving onto the next instance.

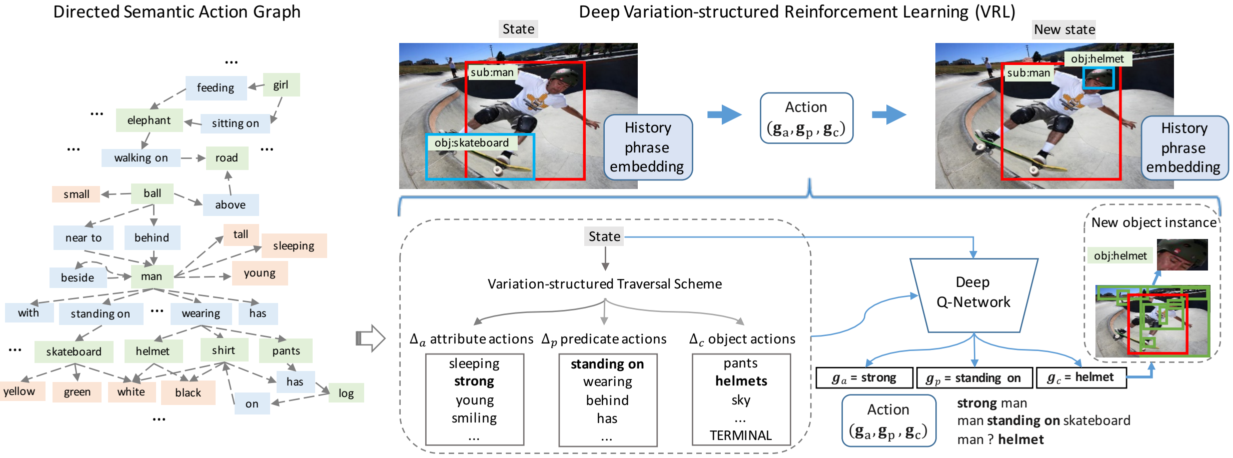

Figure 3: An overview of the VRL framework that sequentially detects relationships ("subject-predicate-object") and attributes ("subject-attribute"). First, we build a directed semantic action graph $\mathcal{G}$ to configure the whole action space. In each step, the input state consists of the current subject and object instances ("sub:man", "obj:skateboard") and a history phrase embedding, which captures the search paths that have already been traversed by the agent. A variation-structured traversal scheme over $\mathcal{G}$ dynamically constructs three small action sets Δ_a, Δ_p, Δ_c. The agent predicts three actions: (1) $\mathbf{g}_a \in \Delta_a$, an attribute of the subject; (2) $\mathbf{g}_p \in \Delta_p$, the predicate between the subject and object; and (3) $\mathbf{g}_c \in \Delta_c$, the next object category of interest ("obj:helmet"). The new state consists of the new subject/object instances ("sub:man", "obj:helmet") and an updated history phrase embedding.

scene separately [16]. It is impractical to exhaustively search all possibilities for each region, and also deviates from human perception. Therefore, it is preferable to have a more principled decision-making framework, which can discover all relevant relationships and attributes within a small number of search steps. To address the aforementioned issues, we propose a deep Variation-structured Reinforcement Learning (VRL) framework which sequentially detects relationship and attribute instances by exploiting global context cues.

First, we use language priors to build a directed semantic action graph $\mathcal{G}$, where the nodes are nouns, attributes, and predicates, connected by directed edges that represent semantic correlations (see Fig. 3). This graph provides a highly-informative, compact representation that enables the model to learn rare relationships and attributes from frequent ones using shared graph nodes. For example, the semantic meaning of "riding" learned from "person-riding-bicycle" can help predict the rare phrase "child-riding-elephant". This generalizing ability allows VRL to handle a considerable number of possible relationship types.

Second, existing deep reinforcement learning (RL) models [24] often require several costly episodes of trial and error to converge, even with a small action space, and our large action space would exacerbate this problem. To efficiently discover all relationships and attributes in a small number of steps, we introduce a novel variation-structured traversal scheme over the action graph which constructs small, adaptive action sets $\Delta_a, \Delta_p, \Delta_c$ for each step based on the current state and historical actions: Δ_a contains candidate attributes to describe an object; Δ_p contains candidate predicates for relating a pair of objects; and Δ_c contains new object instances to mine in the next step. Since an object instance may belong to multiple object categories which the object detector cannot distinguish, we introduce an ambiguity-aware object mining scheme to assign each object with the most appropriate category given the global scene context. Our variation-structured traversal scheme offers a very promising technique for extending the applications of deep RL to complex real-world tasks.

Third, to incorporate global context cues for better reasoning, we explicitly encode the semantic embeddings of previously extracted phrases in the state vector. It makes a better tradeoff between increasing the input dimension and utilizing more historical context, compared to appending history frames [28] or binary action vectors [2] as in previous RL methods.

Extensive experiments on the Visual Relationship Detection (VRD) dataset [16] and Visual Genome dataset [13] demonstrate that the proposed VRL outperforms state-of-the-art methods for both relationship and attribute detection, and also has good generalization capabilities for predicting unseen types.

2 Related Works

Visual relationship and attribute detection. There has been an increased interest in the problem of visual relationship detection [22, 21, 13]. However, most existing approaches [22] [13] can detect only a handful of pre-defined, frequent types by training individual detectors for each relationship. Recently, Lu et al. [16] leveraged word embeddings to handle large-scale relationships. However, their model still ignores the structured correlations between objects and relationships. Furthermore, some methods [10, 23, 14] organized predictions into a scene graph which can provide a structured representation for describing the objects, their attributes and relationships in each image. In particular, Johnson et al. [10] introduced a conditional random field model for reasoning about possible groundings of scene graphs while Schuster et al. [23] proposed a rule-based and classifier-based scene graph parser. In contrast, the proposed VRL makes the first attempt to sequentially discover objects, relationships and attributes by fully exploiting global interdependency.

Deep reinforcement learning. Integrating deep learning methods with reinforcement learning (RL) [11] has recently shown very promising results on decision-making problems. For example, Mnih et al. [18] proposed using deep Q-networks to play ATARI games. Silver et al. [24] proposed a new search algorithm based on the integration of Monte-Carlo tree search with deep RL, which beat the world champion in the game of Go. Other efforts applied deep RL to various real-world tasks, e.g., robotic manipulation [6], indoor navigation [28], and object proposal generation [2]. Our work deals with real-world scenes that are much more complex than ATARI games or images taken in some constrained scenarios, and investigates how to make decisions over a larger action space (e.g., thousands of attribute types). To handle such a large action

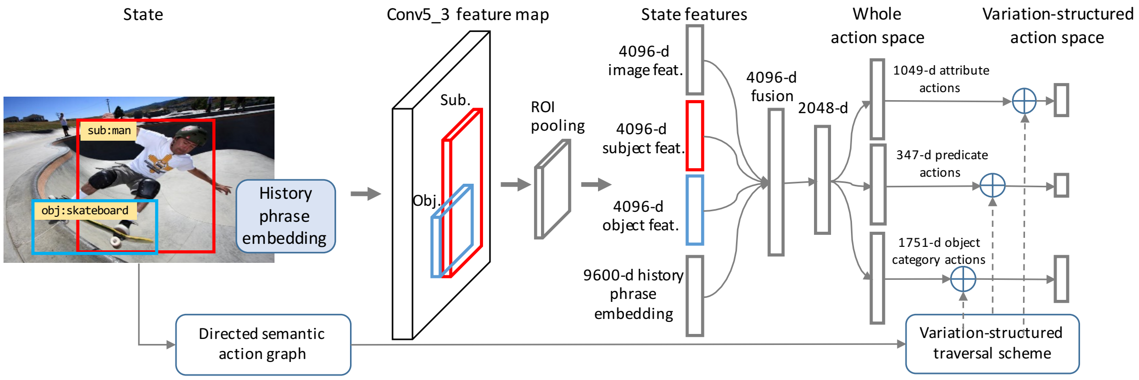

Figure 4: Network architecture of deep VRL. The state vector **f** is a concatenation of (1) a 4096-dim feature of the whole image, taken from the fc6 layer of the pre-trained VGG-16 ImageNet model [25]; (2) two 4096-dim features of the subject s and object s' instances, taken from the conv5_3 layer of the trained Faster R-CNN object detector; and (3) a 9600-dim history phrase embedding, which is created by concatenating four 2400-dim semantic embeddings from a Skip-thought language model [12] of the last two relationship phrases (relating s and s') and the last two attribute phrases (describing s) that were predicted by VRL. A variation-structured traversal scheme over the directed semantic action graph produces a smaller action space from the whole action space, which originally consists of $|\mathcal{A}| = 1049$ attributes, $|\mathcal{P}| = 347$ predicates, and $|\mathcal{C}| = 1750$ object categories plus one terminal trigger. From this variation-structured action space, the model selects actions with the highest predicted Q-values in state **f**.

space, we propose a variation-structured traversal scheme over the whole action graph to decrease the number of possible actions in each step, which substantially reduces the number of trials and thus speeds up the convergence.

3 Dataset

We conduct our experiments on the Visual Relationship Detection (VRD) dataset [16] and the Visual Genome dataset [13]. Both datasets consist of images annotated with bounding boxes of objects in the image, as well as predicates and attributes describing the objects.

VRD [16] contains 5000 images (4000 for training, 1000 for testing) with 100 object categories and 70 predicates. In total, the dataset contains 37,993 relationship instances with 6,672 relationship types, out of which 1,877 relationships occur only in the test set and not in the training set. For the Visual Genome Dataset [13], we experiment on 87,398 images (out of which 5000 are held out for validation, and 5000 for testing), containing 703,839 relationship instances with 13,823 relationship types and 1,464,446 attribute instances with 8,561 attribute types. There are 2,015 relationship types that occur in the test set but not in the training set, which allows us to evaluate VRL on zero-shot learning.

4 Deep Variation-structured Reinforcement Learning

We propose a novel VRL framework which formulates the problem of detecting visual relationships and attributes as a sequential decision-making process. An overview is provided in Fig. 3. The key components of VRL, including the directed semantic action graph, the variation-structured traversal scheme, the state space, and the reward function, are detailed in the following sections.

4.1 Directed Semantic Action Graph

We build a directed semantic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to organize all possible object nouns, attributes, and relationships into a compact and semantically meaningful representation (see Fig. 3). The nodes $\mathcal{V}$ consist of the set of all candidate object categories $\mathcal{C}$, attributes $\mathcal{A}$, and predicates $\mathcal{P}$. Object categories in $\mathcal{C}$ are nouns, and may be people, places, or parts of objects. Attributes in $\mathcal{A}$ can describe color, shape, or pose. Relationships are directional, i.e. they relate a subject noun and an object noun via a predicate. Predicates in $\mathcal{P}$ can be spatial (e.g., "inside of"), compositional (e.g. "part of") or action (e.g., "swinging").

The directed edges $\mathcal{E}$ consist of attribute phrases $\mathcal{E}_{\mathcal{A}} \subseteq \mathcal{C} \times \mathcal{A}$ and predicate phrases $\mathcal{E}_{\mathcal{P}} \subseteq \mathcal{C} \times \mathcal{P} \times \mathcal{C}$. An attribute phrase $(c, a) \in \mathcal{E}_{\mathcal{A}}$ represents an attribute $a \in A$ belonging to a noun $c \in \mathcal{C}$. For example, the attribute phrase "young girl" can be represented by ("girl", "young") $\in \mathcal{E}_{\mathcal{A}}$. A predicate phrase $(c, p, c') \in \mathcal{E}_{\mathcal{P}}$ represents a subject noun $c \in \mathcal{C}$ and an object noun $c' \in \mathcal{C}$ related by a predicate $p \in P$. For example, the predicate phrase "a man is swinging a bat" can be represented by ("man", "swinging", "bat") $\in \mathcal{E}_{\mathcal{P}}$.

The recently released Visual Genome dataset [13] provides a large-scale annotation of images containing 18,136 unique object categories, 13,041 unique attributes, and 13,894 unique relationships. We then select the types that appear at least 30 times in Visual Genome dataset, resulting in 1,750 object-, 8,561 attribute-, and 13,823 relationship-types. From these attribute and relationship types, we build a directed semantic action graph by extracting all unique object category words, attribute words, and predicate words as the graph nodes. Our directed action graph thus contains $|\mathcal{C}| = 1750$ object nodes, $|\mathcal{A}| = 1049$ attribute nodes, and $|\mathcal{P}| = 347$ predicate nodes. On average, each object word is connected to 5 attribute words and 15 predicate words. This semantic action graph serves as the action space for VRL, as we will see in the next section.

4.2 Variation-structured RL

Instead of learning in the entire action space as in traditional deep RL [18, 28], we propose a novel variation-structured traversal scheme over the semantic action graph that dynamically constructs small action sets for each step.

First, VRL uses an object detector to get a set $\mathcal{S}$ of candidate object instances, and then sequentially assigns relationships and attributes to each instance $s \in \mathcal{S}$. For our experiments, we used state-of-the-art Faster R-CNN [20] as the object detector, where the network parameters were initialized using the pre-trained VGG-16 ImageNet model [25].

Since subject instances in an image often have multiple relationships and attributes, we do a breadth-first search: we predict all relationships and attributes with respect to the current subject instance of interest, and then move onto the next instance. We start from the subject instance with the most confident classification score. To prevent the agent from being trapped in a single search path (e.g., in a small local region), the agent selects a new starting subject instance if it has traversed through 5 neighboring objects in the breadth-first search.

The same object in multiple scenarios may be described by different, semantically ambiguous noun categories that cannot be distinguished by the object detector. To address this semantic ambiguity, we introduce an ambiguity-aware object mining scheme which leverages scene contexts captured by extracted relationships and attributes to help determine the most appropriate object category.

Variation-structured action space. The directed semantic graph $\mathcal{G}$ serves as the action space for VRL. For any object instance $s \in \mathcal{S}$ in an image, denote its object category by $s_c \in \mathcal{C}$ and its bounding box by $B(s) = (s_x, s_y, s_w, s_h)$ where (s_x, s_y) is the center coordinate, s_w is the width, and s_h is the height. Given the current subject instance s and object instance s', we select three actions $\mathbf{g}_a \in \mathcal{A}$, $\mathbf{g}_p \in \mathcal{P}$, $\mathbf{g}_c \in \mathcal{C}$ according to the VRL network as follows:

(1) Select an attribute $\mathbf{g}_a$ describing s from the set $\Delta_a = \{a : (s_c, a) \in \mathcal{E}_{\mathcal{A}} \backslash \mathcal{H}_{\mathcal{A}}(s)\}$, where $\mathcal{H}_{\mathcal{A}}(s)$ denotes the set of previously mined attribute phrases for s.

(2) Select a predicate $\mathbf{g}_p$ relating the subject noun s_c and object noun s'_c from $\Delta_p = \{p : (s_c, p, s'_c) \in \mathcal{E}_\mathcal{P}\}$.

(3) To select the next object instance $\tilde{s} \in \mathcal{S}$ in the image, we select its corresponding object category $\mathbf{g}_c$ from a set $\Delta_c \subseteq \mathcal{C}$, which is constructed using an ambiguity-aware object mining scheme as follows (also illustrated in Fig. 5). Let $N(s) \subseteq \mathcal{S}$ be the set of objects neighboring s, where a neighbor of s is defined to be any object $\tilde{s} \in \mathcal{S}$ such that $|\tilde{s}_x - s_x| < 0.5(\tilde{s}_w + s_w)$ and $|\tilde{s}_y - s_y| < 0.5(\tilde{s}_h + s_h)$. For each object $\tilde{s}$, let $C(\tilde{s}) \subseteq \mathcal{C}$ be the set of object categories of $\tilde{s}$ whose confidence scores are at most 0.1 less than that of the most confident category. Let $\Delta_c = \bigcup_{\tilde{s} \in N(s) \setminus \mathcal{H}_\mathcal{S}} C(\tilde{s}) \cup \{\texttt{Terminal}\}$, where $\mathcal{H}_\mathcal{S}$ is the set of previously extracted object instances and `Terminal` is a terminal trigger indicating the end of the object mining scheme for this subject instance. If $N(s) \setminus \mathcal{H}^s$ is empty or the terminal trigger is activated, then we select a new subject instance following the breadth-first scheme. The terminal trigger allows the number of object mining steps for each subject instance to be dynamically specified and limited to a small number.

In each step, the VRL selects actions from the adaptive action sets Δ_a, Δ_p, and Δ_c, which we call the variation-structured action space due to their dynamic structure.

State space. A detailed overview of the state feature extraction process is shown in Fig. 4. Given the current subject s and object s' instances in each time step, the state vector $\mathbf{f}$ is a concatenation of (1) the feature vectors of s and s'; (2) the feature vector of the whole image; and (3) a history phrase embedding vector, which is created by concatenating the semantic embeddings of the last two relationship phrases (relating s and s') and the last two attribute phrases (describing s) that were mined via the variation-structured traversal scheme. More specifically, each phrase (e.g., "person riding bicycle") is embedded into a 2400-dim vector using a pre-trained Skip-thought language model [12], thus resulting in a 9600-dim history phrase embedding.

The feature vector of the whole image provides global context cues which not only help in recognizing relationships and attributes, but also allow the agent to be aware of other uncovered objects. The history phrase embedding captures the search paths and scene contexts that have already been traversed by the agent.

Rewards: Suppose we have groundtruth labels, which consist of the set $\hat{\mathcal{S}}$ of object instances in the image, and attribute phrases $\hat{\mathcal{E}}^\mathcal{A}$ and predicate phrases $\hat{\mathcal{E}}^\mathcal{P}$ describing the objects in $\hat{\mathcal{S}}$. Given a predicted object instance $s \in \mathcal{S}$, we say that a groundtruth object $\hat{s} \in \hat{\mathcal{S}}$ overlaps with s if they have the same object category (i.e., $s_c = \hat{s}_c \in \mathcal{C}$), and their bounding boxes have at least 0.5 Intersection-over-Union (IoU) overlap.

We define the following reward functions to reflect the detection accuracy of taking action $(\mathbf{g}_a, \mathbf{g}_p, \mathbf{g}_c)$ in state $\mathbf{f}$, where the current subject and object instances are s and s', respectively:

(1) $\mathcal{R}_a(\mathbf{f}, \mathbf{g}_a)$ returns +1 if there exists a groundtruth object $\hat{s} \in \hat{\mathcal{S}}$ that overlaps with s, and the predicted attribute relationship $(s_c, \mathbf{g}_a)$ is in the groundtruth set $\hat{\mathcal{E}}^\mathcal{A}$. Otherwise, it returns -1.

(2) $\mathcal{R}_p(\mathbf{f}, \mathbf{g}_p)$ returns +1 if there exists $\hat{s}, \hat{s}' \in \hat{\mathcal{S}}$ that overlap with s and s' respectively, and $(s_c, \mathbf{g}_p, s'_c) \in \hat{\mathcal{E}}^\mathcal{P}$. Otherwise, it returns -1.

(3) $\mathcal{R}_c(\mathbf{f}, \mathbf{g}_c)$ returns +5 if the next object instance $\tilde{s} \in \mathcal{S}$ corresponding to category $\mathbf{g}_c \in \mathcal{C}$ overlaps with a new groundtruth object $\hat{s} \in \mathcal{S}$. Otherwise, it returns -1. Thus, it encourages faster exploration over all objects in the image.

4.3 Deep Variation-structured RL

We optimize three policies to select three actions for each state by maximizing the sum of discounted rewards, which can be formulated as a decision-making process in the deep RL framework. Due to the high-dimensional continuous image data and a model-free environment, we resort to the deep Q-Network (DQN) framework proposed by [17, 18], which generalizes well to unseen inputs. The detailed architecture of our Q-network is illustrated in Fig. 4. Specifically, we use DQN to estimate three Q-value sets, parametrized by network weights θ_a, θ_p, θ_c, which correspond to the action sets $\mathcal{A}, \mathcal{P}, \mathcal{C}$. In each training episode, we use an ϵ-greedy strategy to select actions $\mathbf{g}_a, \mathbf{g}_p, \mathbf{g}_c$ in the variation-structured action space $\Delta_a, \Delta_p, \Delta_c$, where the agent selects random actions with probability ϵ, and selects actions with the highest estimated Q-values

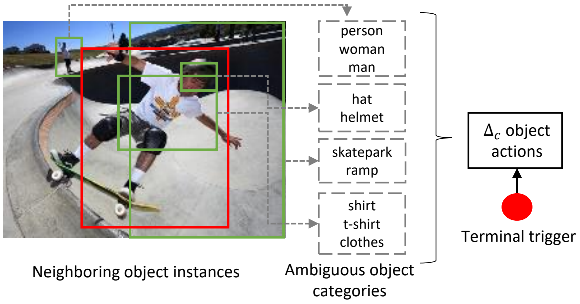
person
woman
man
hat
helmet
skatepark
ramp
shirt
t-shirt
clothes
Δ_c object actions
Terminal trigger
Neighboring object instances
Ambiguous object categories

Figure 5: Illustration of ambiguity-aware object mining. The image on the left shows the subject instance (red box) and its neighboring object instances (green boxes). The action set Δ_c contains candidate object categories of each neighboring object which the object detector cannot distinguish (e.g., "hat" vs. "helmet"), and a terminal trigger indicating the end of the object mining scheme for this subject instance.

with probability $1 - \epsilon$. During testing, we directly select the best actions with highest estimated Q-values in $\Delta_a, \Delta_p, \Delta_c$. The agent sequentially determines the best actions to discover objects, relationships, and attributes in the given image, until either the maximum search step is reached or there are no remaining uncovered object instances.

We also utilize a replay memory to store experience from past episodes. In each step, we draw a random mini-batch from the replay memory to perform the Q-learning update. The replay memory helps stabilize the training by smoothing the training distribution over past experiences and reducing correlation between training samples [17, 18]. Given a transition sample $(\mathbf{f}, \mathbf{f}', \mathbf{g}_a, \mathbf{g}_p, \mathbf{g}_c, \mathcal{R}_a, \mathcal{R}_p, \mathcal{R}_c)$, the network weights $\theta_a^{(t)}$, $\theta_p^{(t)}$, $\theta_c^{(t)}$ are updated as follows:

$$\begin{aligned}
\theta_a^{(t+1)} =& \theta_a^{(t)} + \alpha(\mathcal{R}_a + \lambda \max_{\mathbf{g}_{a'}} Q(\mathbf{f}', \mathbf{g}_{a'}; \theta_a^{(t)-}) - Q(\mathbf{f}, \mathbf{g}_a; \theta_a^{(t)})) \bigtriangledown_{\theta_a^{(t)}} Q(\mathbf{f}, \mathbf{g}_a; \theta_a^{(t)}), \\
\theta_p^{(t+1)} =& \theta_p^{(t)} + \alpha(\mathcal{R}_p + \lambda \max_{\mathbf{g}_{p'}} Q(\mathbf{f}', \mathbf{g}_{p'}; \theta_p^{(t)-}) - Q(\mathbf{f}, \mathbf{g}_p; \theta_p^{(t)})) \bigtriangledown_{\theta_p^{(t)}} Q(\mathbf{f}, \mathbf{g}_p; \theta_p^{(t)}), \\
\theta_c^{(t+1)} =& \theta_c^{(t)} + \alpha(\mathcal{R}_c + \lambda \max_{\mathbf{g}_{c'}} Q(\mathbf{f}', \mathbf{g}_{c'}; \theta_c^{(t)-}) - Q(\mathbf{f}, \mathbf{g}_c; \theta_c^{(t)})) \bigtriangledown_{\theta_c^{(t)}} Q(\mathbf{f}, \mathbf{g}_c; \theta_c^{(t)}),
\end{aligned} \tag{1}$$

where $\mathbf{g}_{a'}$, $\mathbf{g}_{p'}$, $\mathbf{g}_{c'}$ represent the actions that can be taken in state $\mathbf{f}'$, α is the learning rate, and λ is the discount factor. The target network weights $\theta_a^{(t)-}$, $\theta_p^{(t)-}$, $\theta_c^{(t)-}$ are copied every τ steps from the online network, and kept fixed in all other steps.

5 Experiments

Implementation Details. We train a deep Q-network for 60 epochs with a shared RMSProp optimizer [26]. Each epoch ends after performing an episode on all training images. We use a mini-batch size of 64 images. The maximum search step for each image is empirically set to 300. During ϵ-greedy training, ϵ is annealed linearly from 1 to 0.1 over the first 20 epochs, and is fixed to 0.1 in the remaining epochs. The discount factor λ is set to 0.9, and the network parameters $\theta_a^{(t)-}$, $\theta_p^{(t)-}$ and $\theta_c^{(t)-}$ are copied after every $\tau = 10000$ steps. The learning rate α is initialized to 0.0007 and decreased by a factor of 10 after every 10 epochs. Only the top 100 candidate object instances, ranked by objectness confidence scores by the trained object detector, are selected for mining relationships and attributes in an image, in order to balance efficiency and effectiveness. On VRD [16], VRL takes about 8 hours to train an object detector with 100 object categories, and two days to converge. On the Visual Genome dataset [13], VRL takes between 4 to 5 days to train an object detector with 1750 object categories, and one week to converge. On average, it takes 300ms to feed-forward one image into VRL. More details about the dataset are provided in Sec. 5. The implementations are based on the publicly available Torch7 platform on a single NVIDIA GeForce GTX 1080.

Table 1: Results for relationship phrase detection (Phr.) and relationship detection (Rel.) on the VRD dataset. R@100 and R@50 are abbreviations for Recall@100 and Recall@50.

Method	Phr. R@100	Phr. R@50	Rel. R@100	Rel. R@50
Visual Phrases [22]	0.07	0.04	-	-
Joint CNN+R-CNN [25]	0.09	0.07	0.09	0.07
Joint CNN+RPN [25]	2.18	2.13	1.17	1.15
Lu et al. V only [16]	2.61	2.24	1.85	1.58
Faster R-CNN [20]	3.31	3.24	-	-
Joint CNN+Trained RPN [20]	3.51	3.17	2.22	1.98
Faster R-CNN V only [20]	6.13	5.61	5.90	4.26
Lu et al. [16]	17.03	16.17	14.70	13.86
Our VRL	**22.60**	**21.37**	**20.79**	**18.19**
Lu et al. [16] (zero-shot)	3.76	3.36	3.28	3.13
Our VRL (zero-shot)	10.31	9.17	8.52	7.94

Evaluation. Following [16], we use recall@100 and recall@50 as our evaluation metrics. Recall@x computes the fraction of times the correct relationship or attribute instance is covered in the top x confident predictions, which are ranked by the product of objectness confidence scores for the relevant object instances (i.e., confidence scores of the object detector) and Q-values of the selected predicates or attributes. As discussed in [16], we do not use the mean average precision (mAP), which is a pessimistic evaluation metric because the dataset cannot exhaustively annotate all possible relationships and attributes in an image.

Following [16], we evaluate on three tasks: (1) In relationship phrase detection [22], the goal is to predict a "subject-predicate-object" phrase, where the localization of the entire relationship has at least 0.5 overlap with a groundtruth bounding box. (2) In relationship detection, the goal is to predict a "subject-predicate-object" phrase, where the localizations of the subject and object instances have at least 0.5 overlap with their corresponding groundtruth boxes. (3) In attribute detection, the goal is to predict a "subject-attribute" phrase, where the subject's localization has at least 0.5 overlap with a groundtruth box.

Baseline models. First, we compare our model with state-of-the-art approaches, **Visual Phrases [22]**, **Joint CNN+R-CNN [25]** and **Lu et al. [16]**. Note that the latter two methods use R-CNN [5] to extract object proposals. Their results on VRD are reported in [16], and we also experiment their methods on the Visual Genome dataset. **Lu et al. V only [16]** trains individual detectors for object and predicate categories separately, and then combines their confidences to generate a relationship prediction. Furthermore, we train and compare with the following models: "**Faster R-CNN [20]**" directly detects each unique relationship or attribute type, following Visual Phrases [22]. "**Faster R-CNN V only [20]**" model is similar to Lu et al. V only [16], with the only difference being that Faster R-CNN is used for object detection. "**Joint CNN+RPN [25]**" extracts proposals using the pre-trained RPN [20] model on VOC 2012 [3] and then performs the classification. "**Joint CNN+Trained RPN [20]**" trains a separate RPN model on our dataset to generate proposals.

5.1 Comparison with State-of-the-art Models

Comparison of results with baseline methods on VRD and Visual Genome are reported in Tables 1, 2, 3.

Shared Detectors vs Individual Detectors. The compared models can be categorized into two classes: (1) Models that train individual detectors for each predicate or attribute type, i.e., Visual Phrases [22], Joint CNN+R-CNN [25], Joint CNN+RPN [25], Faster R-CNN [20], Joint CNN+Trained RPN [20]. (2) Models that train shared detectors for predicate or attribute types, and then combine their results with object detectors to generate the final prediction, i.e., Lu et al. V only [16], Faster R-CNN V only [20], Lu

Table 2: Results for relationship detection on Visual Genome.

Method	Phr. R@100	Phr. R@50	Rel. R@100	Rel. R@50
Joint CNN+R-CNN [25]	0.13	0.10	0.11	0.08
Joint CNN+RPN [25]	1.39	1.34	1.22	1.18
Lu et al. V only [16]	1.66	1.54	1.48	1.20
Faster R-CNN [20]	2.25	2.19	-	-
Joint CNN+Trained RPN [20]	2.52	2.44	2.37	2.23
Faster R-CNN V only [20]	5.79	5.22	4.87	4.36
Lu et al. [16]	10.23	9.55	7.96	6.01
Our VRL	**16.09**	**14.36**	**13.34**	**12.57**
Lu et al. [16] (zero-shot)	1.20	1.08	1.13	0.97
Our VRL (zero-shot)	7.98	6.53	7.14	6.27

Table 3: Results for attribute detection on Visual Genome.

Method	Attribute Recall@100	Attribute Recall@50
Joint CNN+R-CNN [25]	2.38	1.97
Joint CNN+RPN [25]	3.48	2.63
Faster R-CNN [20]	7.36	5.22
Joint CNN+Trained RPN [20]	9.77	8.35
Our VRL	**26.43**	**24.87**

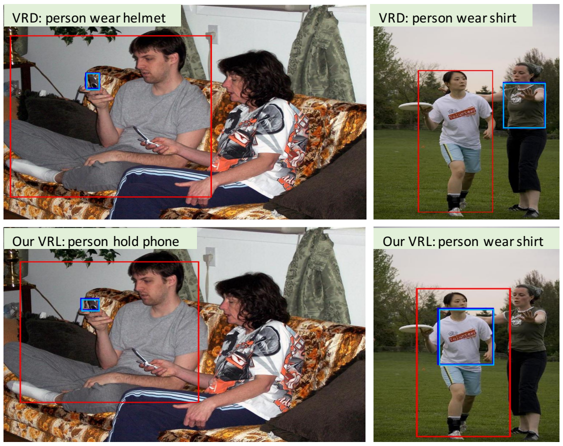

Figure 6: Qualitative comparison between VRD [16] and our VRL.

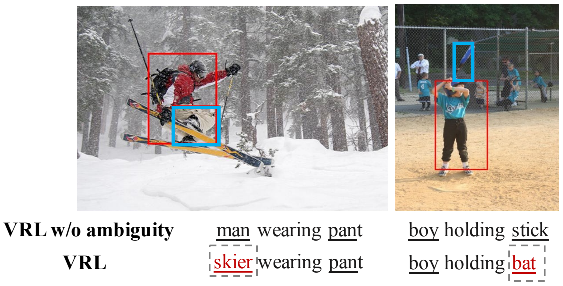

Figure 7: Comparison between "VRL w/o ambiguity" and VRL. Using ambiguity-aware object mining, VRL successfully resolves vague predictions into more concrete ones ("man" → "skier" and "stick" → "bat").

et al. [16] and our VRL. Since the space of all possible relationships and attributes is often large, there are insufficient training examples for infrequent relationships, leading to poor average performance of the models that use individual detectors.

RPN vs R-CNN. In all cases, we obtain performance improvements using RPN network [20] over R-CNN [5] for proposal generation. Additionally, training the proposal network on VRD and VG datasets can also increase the recalls over the pre-trained networks on other datasets.

Language Priors. Unlike baselines that simply train classifiers from visual cues, VRL and Lu et al. [16]

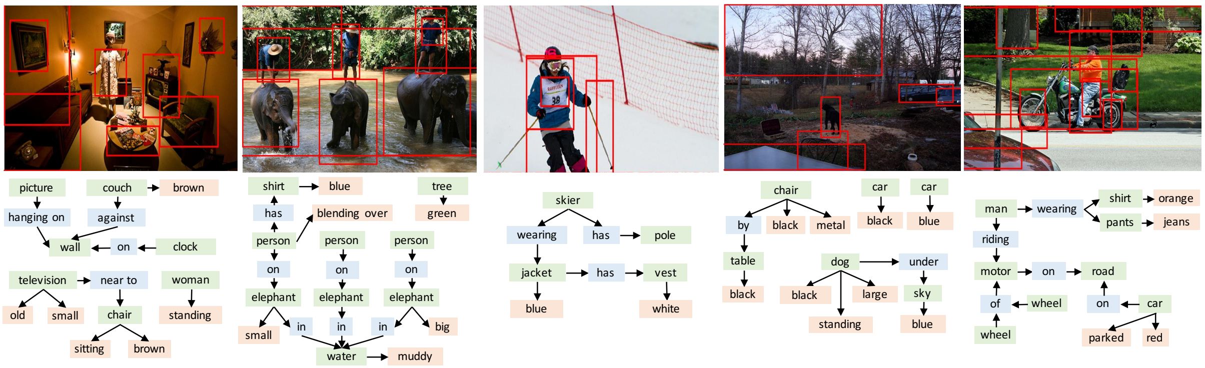

Figure 8: Examples of relationship and attribute detection results generated by VRL on the Visual Genome dataset. We show the top predictions for each image: the localized objects (top) and a semantic graph describing their relationships and attributes (bottom).

leverage language priors to facilitate prediction. Lu et al. [16] uses semantic word embeddings to finetune the likelihood of a predicted relationship, while VRL follows a variational-structured traversal scheme over a directed semantic action graph built from language priors. Both VRL and Lu et al. [16] achieve significantly better performance than other baselines, which demonstrates the necessity of language priors for relationship and attribute detection. Moreover, VRL still shows substantial improvements comparison to Lu et al. [16]. Therefore, VRL's directed semantic action graph provides a more compact and rich representation of semantic correlations than the word embeddings used in Lu et al. [16]. The significant performance improvement is also due to the sequential reasoning of RL.

Qualitative Comparisons We show some qualitative comparison with Lu et al. [16] in Fig. 6, and more detection results of VRL in Fig. 8. Our VRL generates a rich understanding of the image, including the localization and recognition of objects, and the detection of object relationships and attributes. For instance, VRL can correctly detect interactions ("person on elephant", "man riding motor"), spatial layouts ("picture hanging on wall", "car on road"), parts of objects ("person has shirt", "wheel of motor"), and attribute descriptions ("television old", "woman standing").

5.2 Discussion

We give further analysis on the key components of VRL, and report the results in Table 4.

Reinforcement Learning vs. Random Walk. The variant "**RL**" is a standard deep RL model that selects three actions over the entire action space instead of the variation-structured action space. We compare RL with a simple "**Random Walk**" traversal scheme where in each step, the agent randomly selects one object instance in the image, and predicts relationships and attributions for the two most-recently selected instances. "Random Walk" only achieves slightly better results than "Joint+Trained RPN [20]" and performs much worse than the remaining variants, again demonstrating the benefit of sequential mining in RL.

Variation-structured traversal scheme. VRL achieves a remarkably higher recall compared to RL (e.g., 13.34% vs 6.23% on relationship detection and 26.43% vs 12.47% on attribute detection, in terms of Recall@100). Thus, we conclude that using a variation-structured traversal scheme to dynamically configure the small action set for each state can accelerate and stabilize the learning procedure, by dramatically decreasing the number of possible actions. For example, the number of predicate actions (347) can be dropped to 15 on average.

History phrase embedding. To validate the effectiveness of history phrase embeddings, we evaluate two variants of VRL: (1) "**VRL w/o history phrase**" does not incorporate history phrase embedding into state features. This variant causes the recall to drop by over 4% compared to the original VRL. Thus, leveraging history phrase embeddings can help inform the current state what has happened in the past and stabilize

Table 4: Performance of VRL and its variants on Visual Genome.

Method	Rel. R@100	Rel. R@50	Attr. R@100	Attr. R@50
Joint CNN+Trained RPN [20]	2.37	2.23	9.77	8.35
Random Walk	3.67	3.09	10.21	8.59
RL	6.23	5.10	12.47	10.09
VRL w/o history phrase	9.05	8.12	20.09	19.45
VRL w/ directional actions	10.66	9.85	20.31	18.62
VRL w/ historical actions	11.98	10.01	23.02	22.15
VRL w/o ambiguity	12.01	11.20	24.78	22.46
Our VRL	**13.34**	**12.57**	**26.43**	**24.87**
VRL w/ LSTM	13.86	13.07	25.98	25.01

search trajectories that might get stuck in repetitive cycles. (2) "**VRL w/ historical actions**" directly stores a historical action vector in the state [2]. Each historical action vector is the concatenation of four ($|\mathcal{C}| + |\mathcal{A}| + |\mathcal{P}|$)-dim action vectors corresponding to the last four actions taken, where each action vector is zero in all elements except the indices corresponding to the three actions taken in $\mathcal{C}$, $\mathcal{A}$, $\mathcal{P}$. This variant still causes the recall to drop, demonstrating that semantic phrase embeddings learned by language models can capture richer history cues (e.g., relationship similarity).

Ambiguity-aware object mining. "**VRL w/o ambiguity**" only considers the top-1 predicted category of each object for the action set Δ_c. It obtains lower recall than VRL, suggesting that incorporating semantically ambiguous categories into Δ_c can help identify a more appropriate category for each object under different scene contexts. Fig. 7 illustrates two examples where VRL successfully resolves vague predictions of "VRL w/o ambiguity" into more concrete ones ("man"→"skier" and "stick"→"bat").

Spatial actions. Similar to [2], we experiment using spatial actions in the deep RL setting to sequentially extract object instances. The variant "**VRL w/ directional actions**" replaces the 1751-dim object category action vector with a 9-dim action vector indexed by directions (N, NE, E, SE, S, SW, W, NW) plus one terminal trigger. In each step, the agent selects a neighboring object instance with the highest confidence whose center lies in one of the eight directions w.r.t. that of the subject instance. The diverse spatial layouts of object instances across different images make it difficult to learn a spatial action policy, and causes this variant to perform poorly.

Long Short-Term Memory "**VRL w/ LSTM**" is a variant where all fully-connected layers in Fig. 4 are replaced with LSTM [8] layers, which have shown promising results in capturing long-term dependencies. However, "VRL w/ LSTM" has no noticeable performance improvements over VRL, while requiring much more training time. This shows that history phrase embeddings can sufficiently model historical context for sequential prediction.

5.3 Zero-shot Learning

We also compare VRL with Lu et al. [16] in the zero-shot learning setting (see Tables 1 and 2). A promising model should be capable of predicting unseen relationships, since the training data will not cover all possible relationship types. Lu et al. [16] uses word embeddings to project similar relationships onto unseen ones, while our VRL uses a large semantic action graph to learn similar relationships on shared graph nodes. Our VRL achieves $> 5\%$ performance improvements over Lu et al. [16] on both datasets.

6 Conclusion and Future Work

We proposed a novel deep variation-structured reinforcement learning framework for detecting visual relationships and attributes. The VRL sequentially discovers the relationship and attribute instances following a variation-structured traversal scheme on a directed semantic action graph. It incorporates global interdependency to facilitate predictions in local regions. Our experiments on VRD and the Visual Genome dataset demonstrate the power and efficiency of our model over baselines. As future work, a larger directed action graph can be built using natural language sentences. Additionally, VRL can be generalized into an unsupervised learning framework to learn from a massive number of unlabeled images.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1

[2] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2488–2496, 2015. 3, 11

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 8

[4] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1778–1785, 2009. 1

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 8, 9

[6] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation. *arXiv preprint arXiv:1610.00633*, 2016. 3

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 11

[9] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[10] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 1, 3

[11] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 3

[12] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015. 4, 6

[13] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 1, 3, 4, 5, 7

[14] W. Liao, M. Y. Yang, H. Ackermann, and B. Rosenhahn. On support relations and semantic scene graphs. *arXiv preprint arXiv:1609.05834*, 2016. 3

[15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[16] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016. 1, 3, 4, 7, 8, 9, 10, 11

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. In *Deep Learning, Neural Information Processing Systems Workshop*, 2013. 6, 7

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 3, 5, 6, 7

[19] D. Qin, C. Wengert, and L. Van Gool. Query adaptive similarity for large scale object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1610–1617, 2013. 1

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 5, 8, 9, 10, 11

[21] F. Sadeghi, S. K. Divvala, and A. Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1456–1464, 2015. 3

[22] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1745–1752, 2011. 1, 3, 8

[23] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 70–80. Citeseer, 2015. 3

[24] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. 3

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4, 5, 8, 9

[26] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012. 7

[27] Y. Zhu, A. Fathi, and L. Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *European Conference on Computer Vision*, pages 408–424. Springer, 2014. 1

[28] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *arXiv preprint arXiv:1609.05143*, 2016. 3, 5