Unsupervised Scene Graph Induction from Language Supervision

Yanpeng Zhao[ε]
[ε]ILCC, University of Edinburgh
yanp.zhao@ed.ac.uk

Ivan Titov[εæ]
[æ]ILLC, University of Amsterdam
ititov@inf.ed.ac.uk

Abstract

Inducing scene graphs from images has been challenging even with supervised learning, largely because of sparse and unbalanced relation annotations. In this work, we propose to learn scene graph generation models directly from natural language supervision. Our idea is to use image captions as the source of supervision. Captions describe semantically diverse relations among visual objects and are abundantly available on the web, so we can avoid costly and biased scene graph annotations. We design a Visually Grounded Masked Language Model (VG~MLM) that is capable of inducing mappings from entities in the text to their visual counterparts (i.e., objects) in images and from predicates to object pairs. We propose automatic evaluation metrics to quantify model performance. On CLEVR-TV, an artificial image-captioning dataset, VG~MLM demonstrates decent performance in object labeling and relation prediction when using symbolic object representations. But we also find that VG~MLM is sensitive to hyperparameters and hard to optimize. Nevertheless, our work represents the first attempt at learning structured image representations via image-text pre-training.

1. Introduction

A scene graph represents relations between objects. Given an image, scene graph generation models detect objects (including localizing and classifying objects) and predict relations among them. The resulting scene graph representations abstract away low-level image features and represent image contents with high-level concepts that are expressed in language (see Figure 1). There has been a substantial body of work showing that scene graph representations are useful in a variety of vision tasks, including image retrieval [24, 44], image captioning [56, 29, 14], image synthesis [22, 9], and visual question answering [46, 20, 19].

Most approaches to scene graph generation adopt a supervised learning paradigm and thus require scene graph annotations. Apart from the high cost of obtaining manual annotations, the annotated scene graphs tend to be biased.

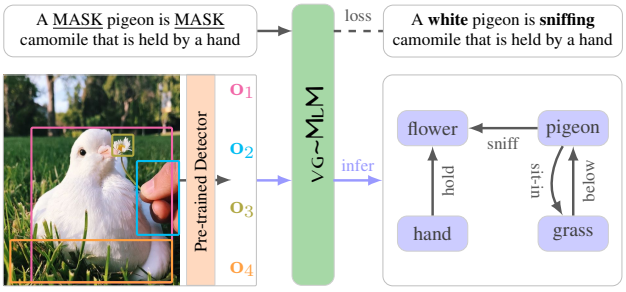

Figure 1. Visually Grounded Masked Language Model (VG~MLM) for unsupervised scene graph induction. VG~MLM is trained on image-text pairs and learns directly from natural language supervision. At test time, it infers scene graphs from images, without access to the aligned text.

Specifically, the distribution of annotated relations is heavily uneven and many relations appear only a handful number of times in the annotated data [26, 57]. Learning from biased data further leads to biased models. While weakly-supervised approaches have been studied, most of them mitigate only the annotation cost issue by using image-level labels, without requiring gold bounding boxes [65, 39], so the label bias issue remains. A possible solution to the issue is to use distant supervision, e.g., mining pseudo labels from large image-captioning datasets via an external parser [44], but pipeline models are potentially error-prone [66, 58].

In this work, we propose to learn scene graph induction models by using free-form captions as direct supervision, without relying on linguistic preprocessing, e.g., converting them into structured forms such as ⟨subject, predicate, object⟩ triplets. We are partly inspired by the recent phenomenal progress in learning visual representations from natural language supervision, which is usually in the form of image-text pairs. Image-text data is abundantly available on the web (e.g., online posts and tweets usually contain images and associated text) and can be curated via automatic tools, without requiring intensive human labor [45]. In the context of image-text pre-training, though continuous visual representations have been the main focus [35, 30, 40], it

has recently been shown that more symbolic representations (e.g., via semantic segmentation and object detection) can also be learned [54, 11]. Motivated by this line of work, we study structured image representation (i.e., scene graphs) learning via image-text pre-training.

The challenges of inducing scene graphs from image-text pairs are two-fold: inducing object representations and aligning each object (pair) to a textual concept that is expressed as a word. Object representations are usually produced by using a pre-trained object detector, but it is also possible to obtain them via unsupervised semantic segmentation [2, 33]. Given object representations, the next step[1] is to classify individual objects and assign relations to pairs of objects, i.e., aligning objects and object pairs to words that best describe them. In annotated scene graphs, two main clusters of words have been distinguished: entities for referring to objects and predicates for referring to visual relations. Since we learn directly from captions, we do not use prior knowledge about word clusters in learning and thus work in a more challenging setting.

We propose VG~MLM, short for Visually Grounded Masked Language Model, for object and relation classification. VG~MLM has an encoder-decoder architecture. Given object embeddings, the encoder produces contextualized object representations. The decoder implements masked language modeling and conditions on the outputs of the encoder. We design a special computational mechanism such that (1) at training time, VG~MLM predicts target words conditioning on both visual and textual contexts; and (2) at test time, VG~MLM is able to make predictions for each object (pair), without access to the aligned captions (see Figure 2).

To study our model, we create CLEVR-TV, an artificial image-captioning dataset built off CLEVR [23]. CLEVR-TV consists of descriptions of relations among abstract 3D shapes. By using abstract objects, we are able to focus on visual relation induction in isolation. We predict an object category for each object and assign a relation to each pair of objects. We propose automatic evaluation metrics to quantify model performance in terms of object and relation classification accuracy. By experimenting with different methods for visual object encoding, we find that symbolic object representation is important for VG~MLM to achieve decent performance. Our experiments also suggest that VG~MLM is hard to optimize, e.g, VG~MLM is sensitive to hyperparameters such as the learning rate.

2. Related Work

Visual Relationship Detection. Visual relations capture interactions among objects in an image and are important for representing and understanding detailed visual semantics. Early approaches to modeling visual relations have focused on spatial relations, which are generally overly generic, e.g., “above", “near", and “around" [10, 13, 27]. More complex visual relations have been studied in the literature of human-object interactions [7, 3, 41], referential expression comprehension [36, 59, 18], and visual phrase detection [43]. To automatically induce diverse relations between arbitrary objects, previous work has formulated a more general task called visual relationship detection (VRD; [34]). VRD aims to predict triplets of the form ⟨subject, predicate, object⟩. It involves localizing pairs of objects (i.e., subject and object) as bounding boxes, labeling objects, and assigning a relation (i.e., predicate) to each pair. Differently from VRD, which targets pairwise relation detection, scene graph generation (SGG; [53]) is introduced to induce scene graphs as a whole. In essence, SGG is simply a redefinition of VRD, but from a modeling perspective, it is proposed to emphasize the aspect of reasoning with surrounding contexts [53], while previous approaches to VRD make predictions for each object pair independently [34].

(Weakely) Supervised Scene Graph Generation. Supervised approaches to scene graph generation have been dominant [34, 64, 60, 32, 37, 55, 31, 51], presumably because of the availability of human-annotated scene graphs such as the Scene Graph dataset [24], the Visual Relationship Detection dataset [34], and the Visual Genome dataset [26]. Among them, Visual Genome has been widely used, but it has been found that the scene graphs in Visual Genome have noisy and sparse annotations [53] and exhibit strong structural regularities [63], presenting a great obstacle to learning reliable and generalizable supervised models. To tackle these challenges, previous work has resorted to different strategies, such as data refinement [53], multitask learning [32], regularized learning with linguistic knowledge [34, 60], exploiting correlations between relations and object labels [6, 63, 4], learning from commonsense knowledge [15, 61], and debiasing based on counterfactual analysis [50], to name but a few.

Despite the impressive development of supervised learning approaches, they are inherently limited due to the reliance on expensive human annotations. A popular strategy for mitigating the need for labeled data is to use weakly-supervised learning. Unlike supervised approaches, which assume localized scene graphs and thus require bounding box annotations, most weakly-supervised methods relax the assumption by assuming unlocalized scene graphs (i.e., image-level object and relation labels) and thus avoid costly manually-annotated bounding boxes. To obtain object proposals, some of the prior weakly-supervised approaches rely on pre-trained detectors [39, 1, 62, 47], and others jointly learn an object proposal module and a rela-

[1] In principle, inducing object representations and aligning them to words can be formulated as multitask learning and learned jointly.

tion detector [65].

Learning Visual Representations from Natural Language Supervision. In the area of visual representation learning, a recent breakthrough has been to learn general-purpose visual representations from natural language supervision [35, 67, 49, 30]. Natural language supervision has been primarily in the form of parallel images and text, which are abundant on the web and thus have led to the development of large-scale image-text pre-training [40, 21]. Apart from learning continuous visual representations, natural language supervision has also been shown to be helpful for learning more symbolic representations via, for example, object detection [43, 25] and segmentation [28, 54]. But little work has been carried out to learn more complex structured visual representations (e.g., scene graphs) from natural language supervision. Those that have used image-text pairs in scene graph generation usually require preprocessing text. For example, Yao *et al*. [57] and Zhong *et al*. [66] use a rule-based parser [44] to extract ⟨subject, predicate, object⟩ triplets from image captions, and curate pseudo labels by aligning the noisy triplets with gold object annotations. Ye and Kovashka [58] use the same parser to parse captions into graph-based semantic structures, which are further served as supervision. While we also use the captions that are associated with images, we directly learn from captions, without converting them into structured forms.

3. Problem Statement

Scene Graph. A scene graph is a directed graph where each node represents a visual object and each directed edge represents the relation between an object pair (see Figure 1). Formally, we define a scene graph as a 3-tuple $\mathcal{G} = (\mathcal{O}, \mathcal{P}, \mathcal{R})$, where

- $\mathcal{O}$ is a finite set of objects in a given image. Each object is labeled with a category such as “pigeon", “grass", and “flower", and is associated with a bounding box. There might be multiple objects labeled with the same category, e.g., “man.01" and “man.02". They can be distinguished from each other by their bounding boxes.
- $\mathcal{P}$ is a finite set of predicates such as “sniff", “under", and ‘sit-in‘. The special predicate "null" $\in \mathcal{P}$ indicates no relation;
- R is a finite set of triplets in the form of (o, p, o'), where $o, o' \in \mathcal{O}$ and $p \in \mathcal{P}$. Each triplet indicates that one object o is related to the other object o' via the relation p. For example, (pigeon, sniff, flower).

Problem Formulation. Assuming a dataset $\mathcal{D} = \{(v^{(i)}, t^{(i)}) | 1 \leq i \leq N\}$ consisting of N pairs of image v and caption t, our goal is to learn a scene graph induction model from D. We consider a novel unsupervised learning setting and contrast it with previous learning settings in the following two important respects:

- We provide object representations. To obtain object representations, we assume that a pre-trained object detector is available. The assumption is practical because object detection has been widely studied [12, 42, 16] and there are off-the-shelf performant detectors [52]. Most weakly-supervised learning approaches also use a pre-trained detector, but some of them use extra object label distributions predicted by the detector [39, 1], while we do not.
- We do not use image-level object labels and relation labels. Image-level captions are the only source of supervision. Unlike previous work, which uses unlocalized gold scene graphs or parses captions to create image-level pseudo labels, we learn directly from captions. At inference time, a model should make predictions (object and relation classification) conditioning on only images, without access to the aligned captions.

4. Scene Graph Induction Model

4.1. Image-Conditioned Masked Language Model

Conceptually, a scene graph induction model predicts object categories for individual objects and assigns relations to object pairs, so it is desirable to have two separate sets of labels for objects and relations, respectively. But, in our setting, we aim to learn directly from captions, so we assume that all the labels are contained in captions,[2] but we do not use the prior knowledge about these labels during learning. While this assumption poses a challenge for inference, it leads to the same setting as that used for image-text pre-training and allows for tapping into a large body of work in that area. Specifically, we draw inspiration from masked multimodal learning [35, 67]. Observing that the object and relation classification can be formulated as predicting a target word given certain visual objects, we propose VG~MLM[3] for unsupervised scene graph induction.

[2]For example, the caption “the pigeon sniffs the flower" contains two object labels “pigeon" and “flower", and a relation label “sniffs". At inference time, we may post-process inferred labels and use their base forms, e.g., “sniffs" will be replaced by “sniff".

[3]Visual-image-conditioned *causal* language modeling is another option, but the left-side contexts of a token do not necessarily contain all the relevant information needed for predicting the token, especially for English that tends to have a subject-verb-object word order, while masked language modeling does not have this limitation. Take the caption “the pigeon sniffs the flower", when predicting “sniffs", it is desirable to know both the left-side entity “pigeon" and the right-side entity “flower" because this not only narrows down possible targets but also guides the model to attend to relevant visual objects, i.e., “pigeon" and “flower". Nevertheless, it is possible to jointly perform masked language modeling and causal language modeling as in [67].

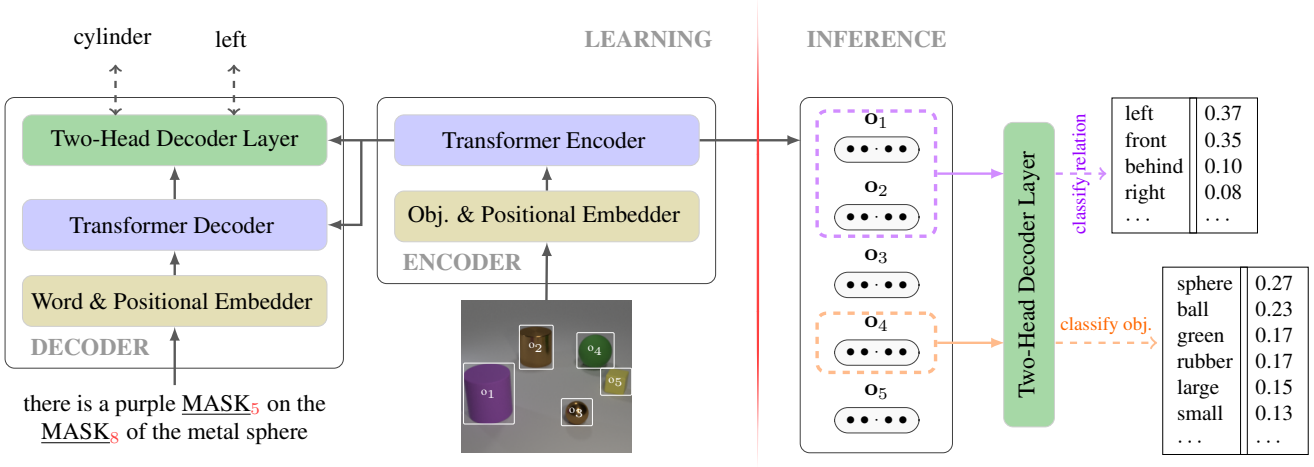

Figure 2. Visually Grounded Masked Language Model (VG~MLM). **Left:** VG~MLM has an encoder-decoder architecture. We customize a two-head Transformer decoder layer (see Figure 3), which stacks above the standard Transformer decoder. **Right:** At inference time, VG~MLM classifies individual objects and assigns relations to individual pairs, without access to the aligned text.

Though both encoder-decoder and decoder-only architectures would suffice, we adopt a Transformer encoder-decoder architecture because separating the encoder from the decoder lets us inject different architectural biases into them. As we will expand on in Section 4.2, these architectural biases are the key to enabling classifying objects and relations conditioning on solely visual objects. We learn VG~MLM by optimizing a masked language modeling objective. Formally, given a training set $\mathcal{D} = \{(v^{(i)}, t^{(i)}) | 1 \leq i \leq N\}$, we maximize the following conditional log-likelihood:

$$\mathcal{L}(\mathcal{D}; \theta) = \sum_{i=1}^{N} \sum_{j \in \mathcal{M}^{(i)}} \log p(t_j^{(i)} | v^{(i)}, t_{-\mathcal{M}}^{(i)}; \theta) \,, \quad (1)$$

where $\mathcal{M}$ represents a set of random token indices with $\max(\mathcal{M}) \leq |t|$ (the length of the caption t), t_j denotes j-th token of t, and $t_{-\mathcal{M}}$ indicates the caption with t_j (for every $j \in \mathcal{M}$) masked out, i.e., replaced by a special symbol "MASK". For example, if the caption t is "the pigeon sniffs the flower" and $\mathcal{M} = \{2, 4\}$, $t_{-\mathcal{M}}$ will be "the MASK sniffs MASK flower". Intuitively, VG~MLM is trained to predict a target token t_j conditioning on both visual contexts v, which are encoded by the encoder, and textual contexts $t_{-\mathcal{M}}$, which are encoded by the decoder.

4.2. VG~MLM for Scene Graph Induction

One of our goals is to infer the most probable relation for a pair of objects, without access to captions. Suppose an L-layer Transformer encoder outputs n_o contextualized object representations $\mathbf{o}_1^{L+1}, \mathbf{o}_2^{L+1}, \ldots, \mathbf{o}_{n_o}^{L+1} \in \mathbb{R}^{d_m}$. At inference time, we solve the following task:

$$\arg\max_r p(r | \mathbf{o}_i^{L+1}, \mathbf{o}_j^{L+1}; \theta) \,. \quad (2)$$

But, during training, the only assumption we have made about VG~MLM is that it predicts a target word conditioning on both visual contexts (i.e., objects) and textual contexts. Let us assume a single random token with the index k in a caption t is masked out, the learning task with a single image-text pair is formalized as:

$$\arg\max_\theta \log p(t_k | t_{-k}, \mathbf{o}_{1:n_o}^{L+1}; \theta) \,, \quad (3)$$

where t_{-k} indicates the caption with the k-th token masked out, similarly to $t_{-\mathcal{M}}$.

Here the problem is that the inference model is inconsistent with the model defined by Equation 3. To solve the problem, we tailor VG~MLM to make it capable of (1) inferring a distribution over relations rather than over the whole vocabulary, (2) making inferences conditioning on individual pairs of objects rather than on all the individual objects, and (3) inferring relation distributions conditioning on only visual objects rather than on both visual objects and textual contexts. Below we expand on our solutions to achieving these goals at inference time.

Inferring Relation Distributions. We assume that the vocabulary subsumes all the object and relation labels. But, since we do not distinguish them during learning, VG~MLM always predicts distributions over the whole vocabulary. At inference time, to focus on a specific set of words, e.g., relation words in relation classification, we simply reset the logits that correspond to non-relation words to "$-\inf$", i.e.,

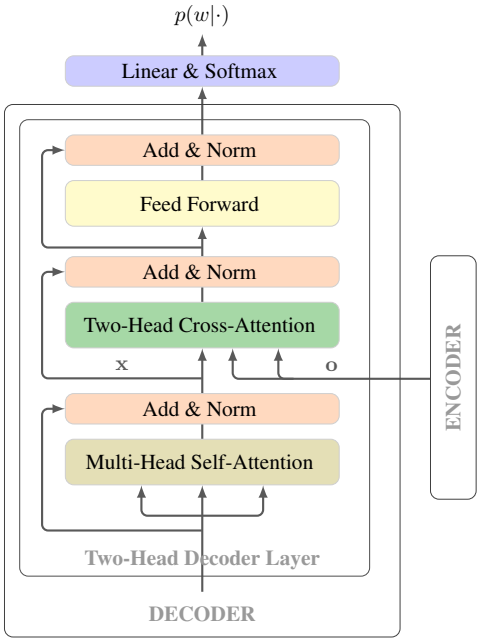

Figure 3. Two-head decoder layer. It is the same as the standard Transformer decoder layer except that we customize a two-head cross-attention module (see Figure 4).

ignoring all the non-relation words. In doing so, we need to identify all relation words in the vocabulary. Practically, we detect all the predicates and treat them as relation words.

Conditioning on Object Pairs. To keep consistent with the inference setting, where VG~MLM predicts relations conditioning on individual pairs of objects, we first construct VG~MLM to condition on all the individual pairs of objects during training. A simple way to represent object pairs is to concatenate the vector representations of the two objects for each pair. But, usually, not all the pairs are equally predictive of a target word, so it is desirable to prioritize object pairs by assigning a predictiveness score to each pair. Moreover, since the semantic roles of two objects determine the relations between them,[4] it is desirable to learn two sets of object representations to indicate the "subject" role and the "object" role, respectively.

We realize all the desiderata by using a two-head cross-attention mechanism. Following the standard attention mechanism, each attention head computes query $\mathbf{q}$, key $\mathbf{k}$, and value $\mathbf{v}$:

$$\mathbf{q}_k = \mathbf{W}^Q \mathbf{x}_k\,, \quad \mathbf{k}_i = \mathbf{W}^K \mathbf{o}_i^{L+1}\,, \quad \mathbf{v}_i = \mathbf{W}^V \mathbf{o}_i^{L+1}\,, \tag{4}$$

where $\mathbf{x}_k \in \mathbb{R}^{d_m}$ is output by the self-attention module of the two-head decoder layer and indicates the contextualized representation at the position k of t_{-k} (see Figure 3). $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_h \times d_m}$ (with $d_h = \frac{d_m}{2}$) are learnable parameters and head-specific. Given $\mathbf{q}$ and $\mathbf{k}$, the attention scores are computed as:

$$c_i = \frac{{\mathbf{q}_k}^T \cdot \mathbf{k}_i}{\sqrt{d_h}} \quad \text{with } 1 \le i \le n_o\,. \tag{5}$$

We use the value representations $\{\mathbf{v}_i^s\}$ from one head as the "subject" representations of objects, and the attention scores $\{c_i^s\}$ as the confidence of assigning the "subject" role to the corresponding objects. From the other head, we obtain the "object" representations $\{\mathbf{v}_j^o\}$ and the associated attention socres $\{c_j^o\}$. Finally, the representation $\mathbf{u}^{(i,j)}$ of an object pair (i,j) and the associated predictiveness score $c^{(i,j)}$ are given by:

$$\mathbf{u}^{(i,j)} = [\mathbf{v}_i^s : \mathbf{v}_j^o]\,, \quad c^{(i,j)} = c_i^s + c_j^o\,, \tag{6}$$

where $i, j \in [1, n_o]$ and $[:]$ indicates vector concatenation.[5]

Our decoder architecture follows that of the standard Transformer decoder except that, in the final layer, we replace the standard cross-attention mechanism with our customized two-head cross-attention mechanism (see Figure 2). Intuitively, we are implicitly assuming that, from the textual contexts of t_k and visual contexts $\mathbf{o}_{1:n_o}^{L+1}$, the model should be able to infer object pairs that are predictive of t_k and assign high scores to them.

Decoupling Object Pairs. Given representations of object pairs $\{\mathbf{u}^{(i,j)}\}$ and the associated predictiveness scores $\{c^{(i,j)}\}$ $(1 \le i, j \le n_o)$, by analogy to the standard attention mechanism, we would summarize visual contexts by averaging $\{\mathbf{u}^{(i,j)}\}$ according to the normalized predictiveness scores: $\hat{c}^{(i,j)} = \exp\left(c^{(i,j)}\right) / \sum_{i,j} \exp\left(c^{(i,j)}\right)$, merge textual contexts $\mathbf{x}_k$ with the summarized visual contexts, and infer a distribution over the vocabulary. Formally,

$$p(w|\mathbf{x}_k, \mathbf{o}_{1:n_o}^{L+1}; \theta) = h\left(\mathbf{x}_k + \sum_{i,j} \hat{c}^{(i,j)} \cdot \mathbf{u}^{(i,j)}\right)\,, \tag{7}$$

where $h(\cdot)$ is implemented as a residual layer followed by the $\mathrm{softmax}$ activation function. But this couples all the pairs of objects. A simple solution to this issue is to move the sum operator outside of $h(\cdot)$:

$$p(w|\mathbf{x}_k, \mathbf{o}_{1:n_o}^{L+1}; \theta) = \sum_{i,j} \hat{c}^{(i,j)} \cdot h\left(\mathbf{x}_k + \mathbf{u}^{(i,j)}\right)\,. \tag{8}$$

[4] For example, assuming "cube" is the subject, and "sphere" is the object, and the relation between them is "in_front_of", switching the roles of the two objects will change the relation into "behind".

[5] Note that $i = j$ implies that the pair is composed of two same objects and thus is equivalent to an individual object. This will be useful for object classification, which is conditioned on individual objects.

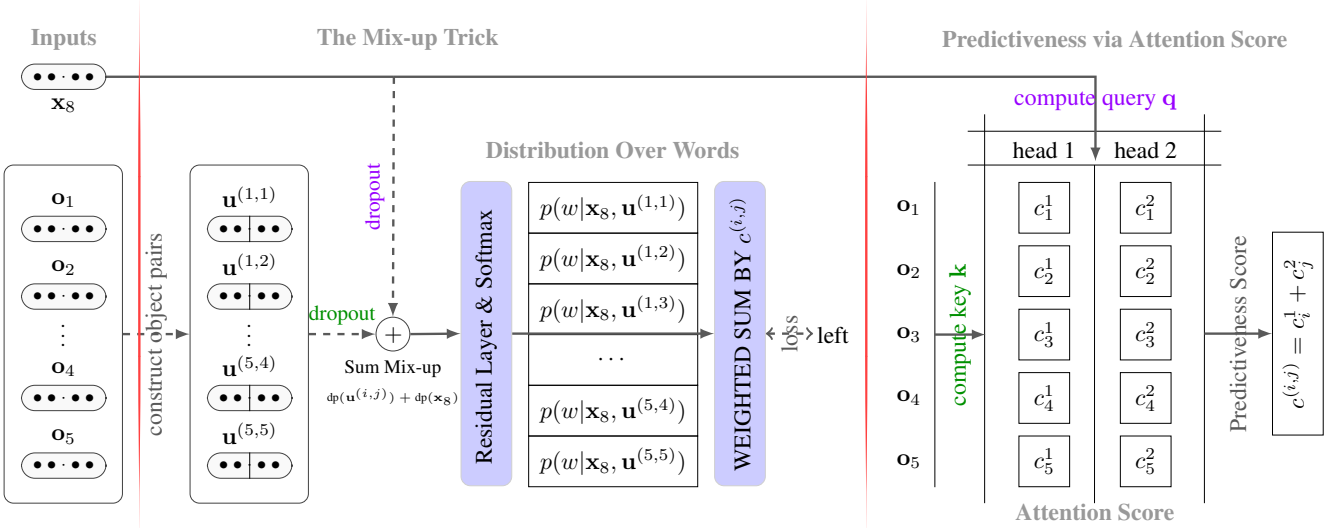

Figure 4. The two-head cross-attention module takes a contextualized word representation $\mathbf{x}_5$ and contextualized object representations $\mathbf{o}_{1:5}$ as the inputs. It uses $\mathbf{x}$ to compute the query and $\mathbf{o}$'s to compute the key and value (see Equation 4). Unlike the standard attention mechanism, which summarizes all the individual objects as a single vector, our module uses object pairs $\mathbf{u}^{(i,j)}$, the concatenation of the representations of two objects i and j (see Equation 6); each pair (i, j) is associated with a predictiveness score $c^{(i,j)}$ and independently performs a prediction $p(w|\mathbf{x}_8, \mathbf{u}^{(i,j)})$, we instead summarize the predictions by using the predictiveness scores (see Equation 8). The final part of the diagram illustrates how we compute predictiveness scores from the attention scores of the two attention heads (see Equation 6).

Intuitively, for each pair $\mathbf{u}^{(i,j)}$, we merge it with the textual contexts $\mathbf{x}_k$ and infer a distribution over the vocabulary, then we average all the inferred distributions according to the normalized predictiveness scores $\{\hat{c}^{(i,j)}\}$.

Conditioning on Only Visual Objects. At inference time, while Equation 8 enables inference conditioning on a given pair of objects:

$$p(w|\mathbf{x}_k, \mathbf{o}_i^{L+1}, \mathbf{o}_j^{L+1}; \theta) = \hat{c}^{(i,j)} \cdot h\left(\mathbf{x}_k + \mathbf{u}^{(i,j)}\right), \quad (9)$$

it relies on the textual contexts $\mathbf{x}_k$ in two different ways: (1) for the outer scalar $\hat{c}^{(i,j)}$, which is computed by using $\mathbf{x}_k$, we can simply drop it; and (2) for the inner $\mathbf{x}_k$, since it is merged with the pair via addition, we can also drop it. This leads to an inference procedure that is not conditioned on textual contexts:

$$p(w|\mathbf{o}_i^{L+1}, \mathbf{o}_j^{L+1}; \theta) = h\left(\mathbf{u}^{(i,j)}\right). \quad (10)$$

But the textual contexts of t_k, which are encoded in $\mathbf{x}_k$, are predictive of t_k in general, dropping $\mathbf{x}_k$ at inference time is likely to lead to a less accurate estimate of the word distribution for a given pair. A possible strategy for retaining the informative textual contexts encoded in $\mathbf{x}_k$ is to distill them into the representations of object pairs. Specifically, we randomly mix-up $\mathbf{x}_k$ and $\mathbf{u}^{(i,j)}$ during training:

$$p(w|\mathbf{x}_k, \mathbf{o}_{1:n_o}^{L+1}; \theta) = \sum_{i,j} \hat{c}^{(i,j)} \cdot h\left(f_t^{dp}(\mathbf{x}_k)\mathbb{1}_{[\text{train}]} + f_v^{dp}(\mathbf{u}^{(i,j)})\right), \quad (11)$$

where $f_t^{dp}(\cdot)$ and $f_v^{dp}(\cdot)$ are dropout functions applied to textual contexts and visual contexts, respectively. $\mathbb{1}_{[\text{train}]}$ is an indicator function and evaluates to 1 only at training time. Intuitively, when part of $\mathbf{x}_k$ that is predictive of a target is masked out, to maintain accurate predictions, $\mathbf{u}$ has to fill in the missing part.

4.3. Encoding Objects

We have so far discussed the decoder of VG~MLM, and specifically, the tailored cross-attention mechanism, which relies on contextualized object representations output by the encoder of VG~MLM. In this section, we elaborate on the encoder (Section 4.3.1), and describe the ways of representing object positions (Section 4.3.2) and visual objects (Section 4.3.3).

4.3.1 Contextualized Object Representations

Suppose n_o objects are initially embedded as d_o-dimensional continuous vectors $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_{n_o}$, and each object is associated with a positional embedding $\mathbf{e} \in \mathbb{R}^{d_b}$. For each object, we concatenate $\mathbf{o}_i$ and $\mathbf{e}_i$, then we apply

a linear map $f : \mathbb{R}^{d_o+d_e} \rightarrow \mathbb{R}^{d_m}$ and input the resultant object representations to an L-layer Transformer encoder. Following the multi-head attention mechanism, in the l-th Transformer layer, given an object representation $\mathbf{o}_i^l \in \mathbb{R}^{d_m}$ and for every $\mathbf{o}_j^l \in \mathbb{R}^{d_m}$ $(i, j \in [1, n_o])$, an attention head estimates the importance of o_j^l to o_i^l as:

$$s_{i,j} = \frac{\mathbf{q}_i{}^T \cdot \mathbf{k}_j}{\sqrt{d_h}} \quad \text{with} \quad \mathbf{q}_i = \mathbf{W}_l^Q \mathbf{o}_i^l, \mathbf{k}_j = \mathbf{W}_l^K \mathbf{o}_j^l, \tag{12}$$

from which the i-th context-aware object representation is computed as:

$$\mathbf{o}_i^{l+1} = \sum_{j=1}^{n_o} \hat{s}_{i,j} \cdot \mathbf{v}_j^l \quad \text{with} \quad \mathbf{v}_j^l = \mathbf{W}_l^V \mathbf{o}_j^l, \quad \hat{s}_{i,j} = \frac{\exp(s_{i,j})}{\sum_{j=1}^{n_o} \exp(s_{i,j})}. \tag{13}$$

In the above formulations, we have used head- and layer-specific learnable parameters $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d_h \times d_m}$ to transform object embeddings into query $\mathbf{q}$, key $\mathbf{k}$, and value $\mathbf{v}$. Suppose there are n_h heads in a Transformer encoder layer and $d_h = d_m/n_h$, the Transformer encoder layer will output $\mathbf{o}_i^{l+1} = [\mathbf{o}_{i,1}^l; \mathbf{o}_{i,2}^l; \ldots; \mathbf{o}_{i,n_h}^l]$, which is the concatenation of the i-th object representations from the n_h heads. A Transformer encoder may have multiple layers. In this case, the outputs $\mathbf{o}^{l+1}$ from the last layer l will be input to the next layer.

4.3.2 Positional Representations

We represent object positions as normalized bounding boxes, which are 4-dimensional vectors, e.g., $(x_1/W, y_1/H, x_2/W, y_2/H)$, where (W, H) is the image size of the form (width, height), and (x_1, y_1), (x_2, y_2) are the upper left and bottom right coordinates of a bounding box, respectively. Each 4-dimensional vector is further transformed into d_e-dimensional positional embedding $\mathbf{e}$ via a learnable linear map: $f : \mathbb{R}^4 \rightarrow \mathbb{R}^{d_b}$.

4.3.3 Object Representations

An important concept of scene graphs is symbolic object modeling, i.e., we abstract away detailed visual features of objects and represent them as symbolic units, i.e., object labels such as “dog" and “bird". Following the common practice, we would use a pre-trained detector to encode visual objects as continuous representations, while generally performant in terms of detection accuracy, since the detected bounding boxes usually do not contain exact objects, extracting object features from rectangular regions inevitably results in noisy object representations. Moreover, since an object usually has different appearances in different images, the extracted visual representations are generally specific to an image. Thus, object embeddings obtained from a detector defy the reusability concept of symbolic representations. To study our model, we instead consider two alternatives for object embedding. These alternatives are compatible with CLEVR images [23], an artificial image set we use for model study.

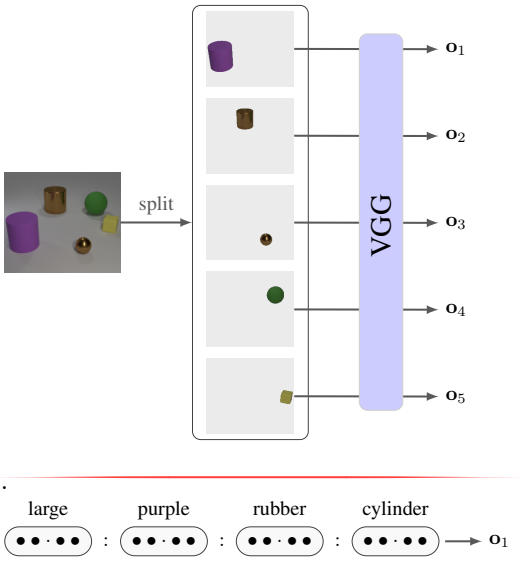

Figure 5. Object embedder. The above illustrates visual object representations via a pre-trained VGG model. Below is an example of symbolic object representations (“:" indicates vector concatenation).

Symbolic Object Representations. Prior to training with visual object representations, we also consider a simpler setting: learning VG~MLM with symbolic object representations. To this end, we represent objects in such a way that symbolic object representations are ensured. Our idea is to represent an object with four attribute values since each object can be characterized by four attributes. Specifically, we create an object template “⟨size⟩ ⟨color⟩ ⟨material⟩ ⟨shape⟩"; substituting the attribute variables with the attribute values of an object gives rise to a symbolic object representation, e.g., “*small red rubber sphere*." To encode objects into continuous representations, we learn a finite set of attribute value embeddings. These embeddings are shared across objects and images and thus meet the goal of reusability. We concatenate the embeddings of the four attribute values of an object to obtain its symbolic representation. By using symbolic object representations, we essentially obtain an upper bound on the performance of our model.

Visual Object Representations. Visual object representations encoded by a pre-trained detector are inevitably noisy. One possible way to reduce the noise is to use an

object segmenter. A segmenter produces object regions that roughly encapsulate exact objects and contain less noisy pixels than bounding boxes. For CLEVR images, we can actually obtain gold object segmentation; encoding each object segment via a pre-trained image encoder presumably gives rise to less noisy object embeddings. Practically, for each object segment, we first create a canvas that has the same size as the original image, then we copy the object to the canvas and ensure that it is in the same position as in the original image. Finally, we use a pre-trained image encoder to encode individual objects (see Figure 5).

5. CLEVR-TV: An Image-Captioning Dataset

Prior to applying VG~MLM to natural images, we would like to learn and test it on artificial data, which helps validate the effectiveness of our model design. In doing so, we propose CLEVR-TV, an artificial dataset for learning scene graph induction models from language supervision. CLEVR-TV consists of image-text pairs and builds upon CLEVR, which is a diagnostic dataset for evaluating visual reasoning capabilities of visual question-answering systems [23]. The text in CLEVR-TV describes relations between visual objects. The images in CLEVR-TV are composed of abstract 3D shapes. By focusing on abstract objects, we try to isolate relational reasoning from visual regularities, e.g., the co-occurrence of two objects "man" and "horse" is likely to entail the "riding" relation, while abstract objects minimize regularities of this kind.

Attribute	Value
Shape	cylinder, sphere
Size	large, small
Material	metal, rubber
Color	gray, red, blue, cyan, green, brown, purple, yellow
Relation	front, behind, right, left

Table 1. Object attributes and relationships in CLEVR-TV.

5.1. Image Generation

Following CLEVR [23], we render images from randomly sampled scene graphs by using Blender [5]. A scene graph represents objects as nodes and relations as edges. Each object is annotated with shape, size, material, and color and is related to other objects via four spatial relations, i.e., "front", "behind", "left", and "right" (see Table 1). A scene graph contains all the information necessary for rendering an image.

5.2. Caption Generation

We are interested in automatically generating diverse relational descriptions. In doing so, we draw inspiration from CLEVR and generate captions from the corresponding functional programs, which can be executed on scene graphs. A functional program is composed of elementary building blocks such as *counting*, *querying*, and *comparing* functions. Suppose the following functional program that requires a *shape* variable ⟨S⟩ as the input and produces a *shape* output ⟨A⟩:

$$<\text{A}> := \text{query\_shape}(\text{left\_of}(\text{filter\_shape}(<\text{S}>, \text{scene}()))),$$

where scene() returns the scene graph representation of an image, and the elementary functions filter_shape(), left_of(), and query_shape() return a list of objects. There are multiple ways of instantiating the program such as (1) "*there is a ⟨A⟩ on the left of the ⟨S⟩*," and (2) "*there is a ⟨S⟩, the ⟨A⟩ is on the left of it*." These instantiations are called caption templates. To generate captions, we simply replace the variables ⟨S⟩ and ⟨A⟩ with valid assignments. For example, the two caption templates may lead to the following captions: "*there is a cylinder on the left of the cube*" and "*there is a cube, a cylinder is on the left of it.*" [6]

We generate captions by roughly following the same procedure as was used for generating CLEVR question-answer pairs. Specifically, given the gold scene graph of an image, we select a functional program and execute it on the scene graph to obtain groups of valid variable assignments (e.g., ⟨S⟩:= *cube* and ⟨A⟩:= *cylinder* in the aforementioned captions are a group of valid assignments). To generate captions, we randomly select a group of assignments and a caption template that corresponds to the program, then we substitute the variables in the caption template with the assignments. To increase caption diversity, CLEVR defines a set of synonyms for some attribute values, e.g., "metal" is associated with {metallic, metal, shiny}, and the assignment bound to a variable will be randomly replaced by one of its synonyms.

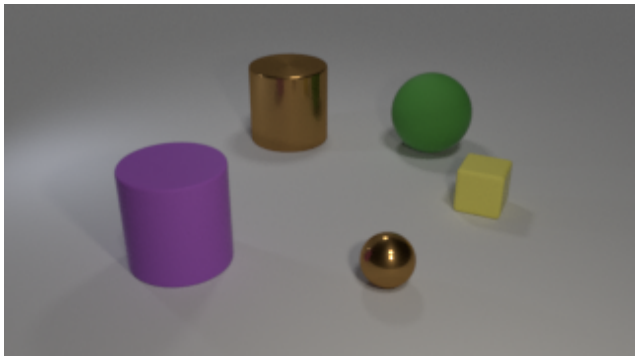

Figure 6. An example CLEVR-TV image. We only consider captions that describe the relations between two objects, e.g., "there is a large cylinder on the left of the green ball."

[6] https://github.com/zhaoyanpeng/clevr-ed.

6. Experiments

6.1. Evaluation Metrics

A scene graph induction model labels individual objects and assigns relations to object pairs. Thus, in evaluation, we are interested in precisions of object labeling and relation prediction.

• the precision that VG~MLM predicts an attribute of a given visual object.

At inference time, VG~MLM requires an object pair as the input (see Equation 10). To enable inference conditioning on individual objects, we create an object pair by concatenating the representations of an object and its copy.

We compute per-attribute precision. Conceptually, for a given attribute A, we only focus on the distribution (i.e., $\mathbf{s}^A$) of its admissible assignments (i.e., $\mathcal{H}(A)$) and find the most probable assignment from the admissible assignments ($\arg\max_k \mathbf{s}^A_k$). Since an attribute value can be described in different ways (e.g., "metal" can be described as "metal", "metallic", and "shiny"), we count it as a correct prediction as long as the prediction is one of the synonyms (i.e., $\hat{A}(\mathrm{o})$) of the value of the attribute A. For example, suppose the material of an object is "metal", a prediction "metallic" is considered correct because it is a synonym of "metal".

Formally, for each attribute $A \in \{\text{shape}, \text{size}, \text{material}, \text{color}\}$, we denote the set of values that can be assigned to A by $\mathcal{H}(A)$ (e.g., $\mathcal{H}(A) = \{\text{large}, \text{tiny}, \text{big}, \text{small}\}$ with $A =$ "shape"). Given an object, suppose the logits (i.e., unnormalized log probabilities) of the inferred categorical distribution over the vocabulary $\mathcal{V}$ is $\mathbf{s} \in \mathbb{R}^{|\mathcal{V}|}$. For a given attribute A, we focus on only its valid assignments $\mathcal{H}(A)$, so we reset the logits that correspond to the words that are not in $\mathcal{H}(A)$ to "$-\inf$", and indicate the resultant logits as $\mathbf{s}^A$. Then the attribute-specific precision over N objects is computed as:

$$p_A^{\text{same}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}^A_k \in \hat{A}(\mathrm{o}_i)]}\,, \tag{14}$$

where $\hat{A}(\mathrm{o})$ accepts an input object o and returns the indices of all the synonyms of the value of the object's attribute A. For example, suppose $A =$ "material" and an object's *material* is "metal", $\hat{A}(\mathrm{o})$ will return the indices of "metal", "metallic", and "shiny", which are the synonyms of "metal".

We can further generalize the metric to any pair of objects o_i and o_j:

$$p_A = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}^A_k \in \hat{A}(\mathrm{o}_i)\cup\hat{A}(\mathrm{o}_j)]}\,. \tag{15}$$

• the precision that VG~MLM predicts the relation between two visual objects.

We compute the precision of relation prediction in a similar way as we compute the precision of attribute prediction. Note that the four spatial relations in CLEVR-TV are composed of two pairs of opposite relations: ("left", "right") and ("front", "behind"). Given an object pair, only one relation is valid when considering two opposite relations (e.g., either "left" or "right"). But, if we consider the four relations together, there will be two valid relations, which come from the two pairs, respectively. For example, an object can be in *front* and to the *left* of another object at the same time, i.e., there are two gold relation labels for an object pair, but we only predict a single label, which is the most probable label. To resolve this issue, we compute precisions for the two pairs of relations, respectively.

A remaining problem is that relations are sensitive to the roles of the participating objects. When developing our model, we specifically assume the first object and the second object are assigned the "subject" role and the "object" role, respectively (see Equation 6), but since we are working with unsupervised learning, a learned model may switch the assumed role assignments and reverse the relation, i.e., the model may assign the "subject" role to the second object and the "object" role to the first object. Consequently, the relation between the two objects will also be reversed, e.g., $(o_1^s, \text{left\_of}, o_2^0) \Leftrightarrow (o_2^s, \text{right\_of}, o_1^o)$. Thus, we need to consider both cases in evaluation. Specifically, we first hypothesize default role assignments (defined by Equation 6), then we compute a precision p_R^{null} provided that the hypothesis is true and a precision p_R^{reject} given that the hypothesis is false.

$$p_R^{\text{null}} = \frac{1}{|\mathcal{H}(R)|}\sum_{r\in\mathcal{H}(R)} \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}^R_k \in \hat{R}(r)]}\,, \tag{16}$$

$$p_R^{\text{reject}} = \frac{1}{|\mathcal{H}(R)|}\sum_{r\in\mathcal{H}(R)} \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{1}_{[\arg\max_k \mathbf{s}^R_k \in \hat{R}(\bar{r})]}\,, \tag{17}$$

where $R \in \{\text{LR}, \text{FB}\}$ with $\mathcal{H}(\text{LR}) = \{\text{left}, \text{right}\}$ and $\mathcal{H}(\text{FB}) = \{\text{front}, \text{behind}\}$, $r \in \mathcal{H}(R)$ and $\bar{r}$ is opposite to r (e.g., $r =$ "left" and $\bar{r} =$ "right"), and $\hat{R}(r)$ returns the indices of the synonyms of r. Finally, the precision for a pair of opposite relations is computed as:

$$P_R = \max\{P_R^{\text{null}}, P_R^{\text{reject}}\}\,. \tag{18}$$

Intuitively, for a given pair (i, j) with the gold label r, the model should predict r if the hypothesis is true; otherwise, it should predict the opposite relation $\bar{r}$ because switching role assignments reverses the relation. For each case, we estimate the model's prediction precision and use the higher precision as the quantification of model performance.

6.2. Datasets and Baselines

Dataset. The training set of CLEVR-TV consists of 32102 images and 96064 captions, and the test set of CLEVR-TV consists of 1000 images. The distribution of four relations, i.e., “left", “right", “front", and “behind", is roughly uniform.

Baseline. Since our evaluation metrics are conditioned on attributes/relations, we consider a baseline model that relies on conditional sampling. Specifically, given a pair of objects, for each attribute, we randomly sample one of the values that can be assigned to the attribute; for each pair of opposite relations, we randomly sample a relation from the pair of opposite relations.

6.3. Settings and Hyperparameters

Standard Transformer Encoder Layers. The encoder and the decoder have 1 and 2 standard Transformer encoder layers, respectively. We set the number of attention heads $n_h = 4$ and the input feature dimension $d_m = 512$. We use the GeLU activation function [17] and disable dropout in the standard Transformer encoder layers.

Decoder Inputs. Both word embeddings and learnable positional embeddings are 256-dimensional. We concatenate word embeddings and the corresponding positional embeddings as the inputs to the decoder. Following the convention [8], we randomly mask out 15% of tokens.

Encoder Inputs. When using symbolic object embeddings, we only need gold bounding boxes and object names (e.g., “small yellow rubber cube"). We embed each attribute value (e.g., “yellow") as a 64-dimensional vector, so each object embedding will be 256-dimensional. To obtain visual object embeddings, we use gold object segmentation and encode each object into a 4096-dimensional vector by using a pre-trained VGG[7] model [48]. We set the dimension of objects' positional embeddings $d_e = 256$.

Two-head Crossmodal Layer. We empirically set all the dropout rates $f_s^{dp} = f_s^{dp} = f^{dp} = 0.25$.

Learning. We optimize VG~MLM with Adam, where the learning rate is 5×10^{-5}, the weight decay is 10^{-8}, and $\beta_1 = 0.9$, $\beta_2 = 0.999$. We use the MulTiStepLR learning rate scheduler, where milestones are $[15, 36, 45, 50]$ and $\gamma = 0.5$. We train VG~MLM for 100 epochs with a batch size of 50 and evaluate the final checkpoint.[8]

[7] VGG-19_BN: https://pytorch.org/vision/stable/models/vgg.html

[8] https://github.com/zhaoyanpeng/sgi.

Evaluation. For each setting, we run 5 times with different random seeds and report the mean and standard deviation of precision values.

6.4. Experimental Design

The goal of our experiments is to validate the effectiveness of our model architecture. We set out to investigate different ways of embedding objects and representing object pairs, and the strength of language supervision because these are major factors that affect model performance.

Object Representations. We consider symbolic object embeddings and visual object embeddings (see Section 4.3.3). Symbolic embeddings are used to estimate an upper bound on model performance, while visual embeddings are more practical. We indicate models that use symbolic and visual object embeddings by affixes “w/ S" and “w/ V", respectively.

Object Pair Representations. We have proposed to learn two sets of object representations for the subject role and the object role, respectively (see Equation 6), but a single set of object representations would also suffice. Recall that we represent a pair as the concatenation of the representations of the two objects: $\mathbf{u}^{(i,j)} = [\mathbf{o}_i : \mathbf{o}_j]$. With a single set of object representations, we need to additionally assume that the object (i.e., o_i) on the left-hand side of “:" and the object (i.e., o_j) on the right-hand side of “:" are assigned the subject role and the object role, respectively, i.e., role assignments are tied to the concatenation operator “:" rather than object representations. For example, the object o_i in $\mathbf{u}^{(i,j)} = [\mathbf{o}_i : \mathbf{o}_j]$ and $\mathbf{u}^{(j,i)} = [\mathbf{o}_j : \mathbf{o}_i]$ has the same representation but is assigned a “subject" role and a “object" role, respectively. In our experiments, we will use a single set of object representations by default because this simplifies our model.

Strength of Language Supervision. Language supervision is generally weaker compared to direct supervision in the form of the ⟨subject, predicate, object⟩ triplets. To study how it influences the learning of our model, we vary the strength of language supervision by using different ways of referring to objects.

- *Ambiguous Captions.* The automatically synthesized captions are ambiguous by design. Specifically, to mimic natural language, which is ambiguous to some extent, when synthesizing captions, we introduce ambiguities by randomly dropping some attributes of each object. Take the image in Figure 6, a synthesized caption could be “*there is a large cylinder on the left of the yellow cube*," where the “large cylinder" may refer to either “large brown metal cylinder" or “large purple

Model	Relation		Pairs of Same Objects				All Object Pairs			
	p_{LR}	p_{FB}	p_{shape}	p_{color}	p_{size}	p_{material}	p_{shape}	p_{color}	p_{size}	p_{material}
Baseline	$^*50.00$	$^*50.00$	34.50	11.62	49.63	49.69	52.50	21.58	71.01	71.66
VC-MLM with Symbolic Object Embeddings (w/ S)										
Ambiguous	$50.14_{\pm 0.1}$	$51.06_{\pm 0.7}$	$99.85_{\pm 0.3}$	$100.00_{\pm 0.0}$	$99.98_{\pm 0.0}$	$99.96_{\pm 0.1}$	$98.48_{\pm 2.4}$	$97.69_{\pm 1.3}$	$99.98_{\pm 0.0}$	$99.97_{\pm 0.1}$
FULL†	88.75	90.66	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
FULL	$50.40_{\pm 0.2}$	$78.47_{\pm 5.4}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$99.64_{\pm 0.6}$	$100.00_{\pm 0.0}$	$99.95_{\pm 0.1}$	$99.89_{\pm 0.2}$
≤ 4	$50.13_{\pm 0.1}$	$53.32_{\pm 3.2}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$95.85_{\pm 3.2}$	$95.22_{\pm 2.8}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$
≤ 3	$50.07_{\pm 0.1}$	$53.79_{\pm 5.6}$	$100.00_{\pm 0.0}$	$99.99_{\pm 0.0}$	$99.98_{\pm 0.0}$	$99.96_{\pm 0.1}$	$93.28_{\pm 3.5}$	$96.76_{\pm 1.8}$	$99.88_{\pm 0.2}$	$99.97_{\pm 0.1}$
≤ 2	$50.20_{\pm 0.1}$	$51.66_{\pm 0.9}$	$100.00_{\pm 0.0}$	$97.60_{\pm 3.4}$	$99.38_{\pm 1.2}$	$99.74_{\pm 0.4}$	$95.11_{\pm 3.9}$	$94.62_{\pm 3.6}$	$99.19_{\pm 1.5}$	$99.78_{\pm 0.3}$
≤ 1	$50.56_{\pm 0.4}$	$53.20_{\pm 1.1}$	$44.39_{\pm 21.2}$	$0.00_{\pm 0.0}$	$0.00_{\pm 0.0}$	$0.00_{\pm 0.0}$	$62.46_{\pm 18.0}$	$7.70_{\pm 6.6}$	$41.55_{\pm 0.0}$	$41.26_{\pm 0.0}$
VC-MLM (w/ S) + Causal Language Modeling (w/ CLM)										
FULL	$50.31_{\pm 0.1}$	$71.58_{\pm 9.1}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$	$99.99_{\pm 0.0}$	$100.00_{\pm 0.0}$	$100.00_{\pm 0.0}$
VC-MLM with Visual Object Embeddings (w/ V)										
FULL	$50.45_{\pm 0.5}$	$55.76_{\pm 4.6}$	$99.14_{\pm 0.1}$	$94.76_{\pm 0.4}$	$99.95_{\pm 0.0}$	$96.78_{\pm 0.2}$	$99.30_{\pm 0.2}$	$81.84_{\pm 1.1}$	$99.98_{\pm 0.0}$	$98.81_{\pm 0.5}$

Table 2. * denotes theoretical performance. † indicates the best run. "w/ S" and "w/ V" indicate that VG~MLM uses symbolic and visual object embeddings, respectively. FULL indicates that all the four attributes are used to refer to objects when possible and "$\leq n$" ($1 \leq n \leq 4$) indicates that up to n attributes are used to refer to objects (see Section 6.4)

rubber cylinder". Though similar to natural language, these ambiguities make learning difficult.

- *Unambiguous Captions.* We further consider five types of unambiguous captions and group them according to how specifically the objects are referred to. (1) FULL indicates that we use all the four attributes to refer to objects when possible. For example, the above ambiguous caption can be disambiguated as "*there is a large purple rubber cylinder on the left of the small yellow rubber cube*," while in captions like "*the purple rubber cylinder on the left of the small yellow rubber cube is large*," we use three attributes to refer to "large purple rubber cylinder" because we would like models to infer "large" from the contexts; and (2) $\leq n$ (where $n \in \{1, 2, 3, 4\}$) indicates that we use up to n attributes to refer to objects. As n decreases, there are fewer captions that are unambiguous.

6.5. Main Results

Unambiguous captions are helpful. Compared to "Ambiguous", unambiguous FULL achieves perfect or nearly perfect object classification performance and demonstrates a decent relation prediction precision for the front-behind pair (i.e., 78.5%). Thus, unambiguous language descriptions make learning easier.

The more specific object descriptions, the better. We vary the strength of language supervision by using different maximum numbers of attributes to refer to objects. For relation prediction, considering the variance, using more attributes does not improve the model further, e.g., "≤ 4" and "≤ 1" have a similar mean precision (i.e., 53% for the front-behind pair), though "≤ 1" only uses around 60% of the training examples that "≤ 4" uses. But, for object classification, using more attributes lead to higher precision. This is unsurprising because more attributes mean stronger supervision.

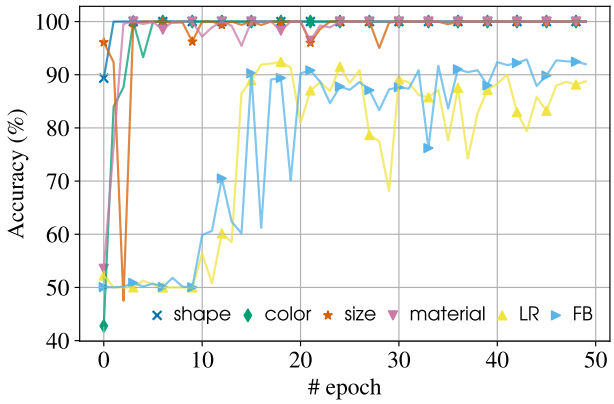

Figure 7. Per-attribute classification and relation prediction accuracy on the development set. Results are from the best run: unambiguous FULL†. LR and FB indicate relation prediction for the left-right pair and the front-behind pair, respectively.

Relation prediction relies on object classification. Figure 7 illustrates object classification and relation prediction accuracy as training proceeds. We can see that only after object classification performance starts to stabilize from the 15-th epoch, does relation prediction performance tend to plateau. Intuitively, the model first learns to map objects

to the corresponding words, then it learns to infer relations between objects.

Symbolic object representations are helpful. Compared to VG~MLM (w/ V), VG~MLM (w/ S) performs better in general, which indicates that symbolic object representations are the key to achieving good performance. While generally worse than symbolic object representations, visual object representations (w/ V) result in nearly perfect object classification for all the attributes except "color" and "material", presumably because these two attributes have more valid assignments, i.e., 8 for "color" and 5 for "material", while other attributes have fewer than 5.

Causal language modeling does not necessarily help. We additionally optimize a causal language modeling (CLM) objective during training. Compared to VG~MLM (w/ S), though VG~MLM (w/ S+CLM) also achieves nearly perfect object classification, it does not improve relation prediction. As discussed before, masked language modeling alone should be adequate for learning mappings between words and their visual counterparts.

7. Conclusion and Future Work

We have presented a novel setting for unsupervised scene graph induction, where a scene graph induction model is trained on image-text pairs and learns from only image-level captions. We propose an image-conditioned masked language model (VG~MLM) to tackle the task. VG~MLM adopts a Transformer encoder-decoder architecture. We tailor a multi-head attention module to connect the object encoder and caption decoder. The architecture design of the crossmodal module enables VG~MLM to infer scene graphs from images without relying on text. We create CLEVR-TV, which is an artificial image-captioning dataset, to learn and study VG~MLM, and propose automatic evaluation metrics to quantify the performance of VG~MLM. Though we empirically find that VG~MLM can achieve good performance through only symbolic object representations, we also observe that VG~MLM is unstable and sensitive to hyperparameters.

In the future, we would like to improve the proposed VG~MLM in the following respects:

- Exploring alternative model architectures. The architectural biases, which are implemented in VG~MLM, might be inappropriate and account for the difficulties of optimization;

- Jointly classifying objects and predicting relations. We currently classify objects and assign relations to object pairs independently. When working with natural images, knowing object labels arguably helps with relation prediction. For example, given an object pair ("man", "book"), the relation between them is more likely to be "read"/"hold" rather than "eat"/"ride". This type of commonsense knowledge has been exploited in previous work [34, 61]. We would expect VG~MLM to be able to derive it from abundant text, without relying on external knowledge bases. For example, we may substitute the decoder of VG~MLM with a pre-trained masked language model [38].

Acknowledgments

We would like to thank Serhii Havrylov for the initial discussions that helped formulate the task and motivated the model proposed in this work.

References

[1] Federico Baldassarre, Kevin Smith, Josephine Sullivan, and Hossein Azizpour. Explanation-based weakly-supervised learning of visual relations with graph networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 612–630, Cham, 2020. Springer International Publishing. 2, 3

[2] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation, 2019. 2

[3] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. Hico: A benchmark for recognizing human-object interactions in images. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1017–1025, 2015. 2

[4] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6156–6164, 2019. 2

[5] Blender Online Community. Blender - a 3d modelling and rendering package, 2016. 8

[6] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3298–3308, 2017. 2

[7] Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for static human-object interactions. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 9–16, 2010. 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 10

[9] Helisa Dhamo, Azade Farshad, Iro Laina, Nassir Navab, Gregory D. Hager, Federico Tombari, and Christian Rupprecht. Semantic image manipulation using scene graphs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5212–5221, 2020. 1

[10] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[11] Shijie Geng, Jianbo Yuan, Yu Tian, Yuxiao Chen, and Yongfeng Zhang. HiCLIP: Contrastive language-image pretraining with hierarchy-aware attention. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1440–1448, USA, 2015. IEEE Computer Society. 3

[13] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *Int. J. Comput. Vision*, 80(3):300–316, dec 2008. 2

[14] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10322–10331, 2019. 1

[15] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1969–1978, 2019. 2

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020. 10

[18] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4418–4427, 2017. 2

[19] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[20] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. 1

[21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR, 18–24 Jul 2021. 3

[22] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1219–1228, 2018. 1

[23] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2017. 2, 7, 8

[24] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 1, 2

[25] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion. Mdetr - modulated detection for end-to-end multimodal understanding. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1760–1770, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. 3

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, May 2017. 1, 2

[27] Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR 2011*, pages 1601–1608, 2011. 2

[28] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 3

[29] Xiangyang Li and Shuqiang Jiang. Know more say less: Image captioning based on scene graphs. *IEEE Transactions on Multimedia*, 21(8):2117–2130, 2019. 1

[30] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 121–137, Cham, 2020. Springer International Publishing. 1, 3

[31] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 346–363, Cham, 2018. Springer International Publishing. 2

[32] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1270–1279, 2017. 2

[33] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020. 2

[34] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 852–869, Cham, 2016. Springer International Publishing. 2, 12

[35] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1, 3

[36] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2016. 2

[37] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[38] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 12

[39] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. Weakly-supervised learning of visual relations. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5189–5198, 2017. 1, 2, 3

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 1, 3

[41] Vignesh Ramanathan, Congcong Li, Jia Deng, Wei Han, Zhen Li, Kunlong Gu, Yang Song, Samy Bengio, Chuck Rossenberg, and Li Fei-Fei. Learning semantic relationships for better action retrieval in images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1100–1109, 2015. 2

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 3

[43] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *CVPR 2011*, pages 1745–1752, 2011. 2, 3

[44] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher 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, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. 1, 3

[45] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. 1

[46] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8368–8376, 2019. 1

[47] Jing Shi, Yiwu Zhong, Ning Xu, Yin Li, and Chenliang Xu. A simple baseline for weakly-supervised scene graph generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16373–16382, 2021. 2

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 10

[49] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. 3

[50] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3713–3722, 2020. 2

[51] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[52] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 3

[53] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3106, 2017. 2

[54] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18113–18123, 2022. 2, 3

[55] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 690–706, Cham, 2018. Springer International Publishing. 2

[56] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10677–10686, 2019. 1

[57] Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, and Maosong Sun. Visual distant supervision for scene graph generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15796–15806, 2021. 1, 3

[58] Keren Ye and Adriana Kovashka. Linguistic structures as weak supervision for visual scene graph generation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8285–8295, 2021. 1, 3

[59] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 69–85, Cham, 2016. Springer International Publishing. 2

[60] Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1068–1076, 2017. 2

[61] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, page 606–623, Berlin, Heidelberg, 2020. Springer-Verlag. 2, 12

[62] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3733–3742, 2020. 2

[63] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018. 2

[64] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3107–3115, 2017. 2

[65] Hanwang Zhang, Zawlin Kyaw, Jinyang Yu, and Shih-Fu Chang. Ppr-fcn: Weakly supervised visual relation detection via parallel pairwise r-fcn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4243–4251, 2017. 1, 3

[66] Yiwu Zhong, Jing Shi, Jianwei Yang, Chenliang Xu, and Yin Li. Learning to generate scene graph from natural language supervision. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1803–1814, 2021. 1, 3

[67] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13041–13049, Apr. 2020. 3