Learning to Represent Image and Text with Denotation Graph

Bowen Zhang [*] [†]
University of Southern California
zhan734@usc.edu

Hexiang Hu [*] [†]
University of Southern California
hexiangh@usc.edu

Vihan Jain
Google Research
vihanjain@google.com

Eugene Ie
Google Research
eugeneie@google.com

Fei Sha [‡]
Google Research
fsha@google.com

Abstract

Learning to fuse vision and language information and representing them is an important research problem with many applications. Recent progresses have leveraged the ideas of pretraining (from language modeling) and attention layers in Transformers to learn representation from datasets containing images aligned with linguistic expressions that describe the images. In this paper, we propose learning representations from a set of implied, visually grounded expressions between image and text, automatically mined from those datasets. In particular, we use denotation graphs to represent how specific concepts (such as sentences describing images) can be linked to abstract and generic concepts (such as short phrases) that are also visually grounded. This type of generic-to-specific relations can be discovered using linguistic analysis tools. We propose methods to incorporate such relations into learning representation. We show that state-of-the-art multimodal learning models can be further improved by leveraging automatically harvested structural relations. The representations lead to stronger empirical results on downstream tasks of cross-modal image retrieval, referring expression, and compositional attribute-object recognition. Both our codes and the extracted denotation graphs on the Flickr30K and the COCO datasets are publically available on https://sha-lab.github.io/DG.

1 Introduction

There has been an abundant amount of aligned visual and language data such as text passages describing images, narrated videos, subtitles in movies, etc. Thus, learning how to represent visual and language information when they are semantically related has been a very actively studied topic. There are many *vision* + *language* applications: image retrieval with descriptive sentences or captions (Barnard and Forsyth, 2001; Barnard et al., 2003; Hodosh et al., 2013; Young et al., 2014), image captioning (Chen et al., 2015; Xu et al., 2015), visual question answering (Antol et al., 2015), visual navigation with language instructions (Anderson et al., 2018b), visual objects localization via short text phrases (Plummer et al., 2015), and others. A recurring theme is to learn the representation of these two streams of information so that they correspond to each other, highlighting the notion that many language expressions are visually grounded.

A standard approach is to embed the visual and the language information as points in a (joint) visual-semantic embedding space (Frome et al., 2013; Kiros et al., 2014; Faghri et al., 2018). One can then infer whether the visual information is aligned with the text information by checking how these points are distributed.

How do we embed visual and text information? Earlier approaches focus on embedding each stream of information independently, using models that are tailored to each modality. For example, for image, the embedding could be the features at the last fully-connected layer from a deep neural network trained for classifying the dominant objects in the image. For text, the embedding could be the last hidden outputs from a recurrent neural network.

Recent approaches, however, have introduced several innovations (Lu et al., 2019; Li et al., 2019a; Chen et al., 2019). The first is to contextualize

[*] Work done while at Google
[†] Authors Contributed Equally
[‡] On leave from USC (feisha@usc.edu)

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 823–839,
November 16–20, 2020. ©2020 Association for Computational Linguistics

the embeddings of one modality using information from the other one. This is achieved by using co-attention or cross-attention (in addition to self-attention) in Transformer layers. The second is to leverage the power of pre-training (Radford et al., 2019; Devlin et al., 2019): given a large number of parallel corpora of images and their descriptions, it is beneficial to identify pre-trained embeddings on these data such that they are useful for downstream *vision* + *language* tasks.

Despite such progress, there is a missed opportunity of learning stronger representations from those parallel corpora. As a motivating example, suppose we have two paired examples: one is an image $\boldsymbol{x}_1$ corresponding to the text $\boldsymbol{y}_1$ of TWO DOGS SAT IN FRONT OF PORCH and the other is an image $\boldsymbol{x}_2$ corresponding to the text $\boldsymbol{y}_2$ of TWO DOGS RUNNING ON THE GRASS. Existing approaches treat the two pairs independently and compute the embeddings for each pair without acknowledging that both texts share the common phrase $\boldsymbol{y}_1 \cap \boldsymbol{y}_2 =$ TWO DOGS and the images have the same visual categories of two dogs.

We hypothesize that learning the correspondence between the common phrase $\boldsymbol{y}_1 \cap \boldsymbol{y}_2$ and the set of images $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, though not explicitly annotated in the training data, is beneficial. Enforcing the alignment due to this *additionally constructed* pair introduces a form of structural constraint: the embeddings of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have to convey similar visual information that is congruent to the similar text information in the embeddings of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$.

In this paper, we validate this hypothesis and show that extracting additional and implied correspondences between the texts and the visual information, then using them for learning leads to better representation, which results in a stronger performance in downstream tasks. The additional alignment information forms a graph where the edges indicate how visually grounded concepts can be instantiated at both abstract levels (such as TWO DOGS) and specific levels (such as TWO DOGS SAT IN FRONT OF THE PORCH). These edges and the nodes that represent the concepts at different abstraction levels form a graph, known as denotation graph, previously studied in the NLP community (Young et al., 2014; Lai and Hockenmaier, 2017; Plummer et al., 2015) for grounding language expressions visually.

Our contributions are to propose creating visually-grounded denotation graphs to facilitate representation learning. Concretely, we apply the technique originally developed for the FLICKR30K dataset (Young et al., 2014) also to COCO dataset (Lin et al., 2014) to obtain denotation graphs that are grounded in each domain respectively (§ 3). We then show how the denotation graphs can be used to augment training samples for aligning text and image (§ 4). Finally, we show empirically that the representation learned with denotation graphs leads to stronger performance in downstream tasks (§ 5).

2 Related Work

Learning representation for image and text Single-stream methods learn each modality separately and align them together with a simple fusion model, often an inner product between the two representations. Frome *et al.* (Frome et al., 2013) learns the joint embedding space for images and labels and use the learned embeddings for zero-shot learning. Kiros *et al.* (Kiros et al., 2014) uses bi-directional LSTMs to encode sentences and then maps images and sentences into a joint embedding space for cross-modal retrieval and multi-modal language models. Li *et al.* (Li et al., 2019b) designs a high-level visual reasoning module to contextualize image entity features and obtain a more powerful image representation. Vendrov *et al.* (Vendrov et al., 2016) improves image retrieval performance by exploiting the hypernym relations among words. There is a large body of work that has been focusing on improving the visual or text embedding functions (Socher et al., 2014; Eisenschtat and Wolf, 2017; Nam et al., 2017; Huang et al., 2018; Gu et al., 2018).

Another line of work, referred to as cross-stream methods infer fine-grained alignments between local patterns of visual (*i.e.*, local regions) and linguistic inputs (*i.e.*, words) between a pair of image and text, then use them to derive the similarity between the image and the text. SCAN (Lee et al., 2018) uses cross-modal attention mechanism (Xu et al., 2015) to discover such latent alignments. Inspired by the success of BERT (Devlin et al., 2019), recent efforts have conducted visual-linguistic pre-training on large-scale datasets (Sharma et al., 2018), using a powerful sequence model such as

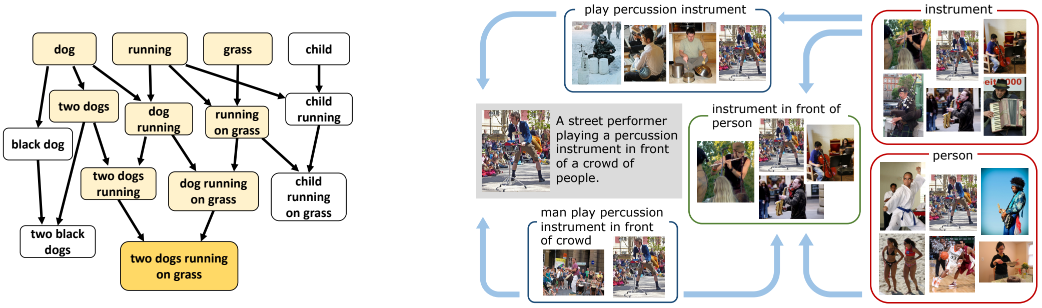

dog
running
grass
child
two dogs
dog running
running on grass
child running
black dog
two dogs running
dog running on grass
child running on grass
two black dogs
two dogs running on grass
play percussion instrument
instrument
A street performer playing a percussion instrument in front of a crowd of people.
instrument in front of person
person
man play percussion instrument in front of crowd

Figure 1: (Left) A schematic example of denotation graph showing the hierarchical organization of linguistic expression (adapted from `https://shannon.cs.illinois.edu/DenotationGraph/`) (Right) A random-subgraph from the denotation graph extracted from the FLICKR30K dataset, with images attached to concepts at different levels of hierarchy.

deep Transformers (Lu et al., 2019; Li et al., 2019a; Chen et al., 2019; Su et al., 2020; Li et al., 2019c). The pre-training strategies of these methods typically involve many self-supervised learning tasks, including the image-text matching (Lu et al., 2019), masked language modeling (Devlin et al., 2019; Lu et al., 2019) and masked region modeling (Chen et al., 2019).

In contrast to those work, we focus on exploiting additional correspondences between image and text that are not explicitly given in the many image and text datasets. By analyzing the linguistic structures of the texts in those datasets, we are able to discover more correspondences that can be used for learning representation. We show the learned representation is more powerful in downstream tasks.

***Vision* + *Language* Tasks** There has been a large collection of tasks combining vision and language, including *image captioning* (Chen and Lawrence Zitnick, 2015; Fang et al., 2015; Hodosh et al., 2013; Karpathy and Fei-Fei, 2015; Kulkarni et al., 2013), *visual QA* (Antol et al., 2015), *text-based image verification* (Suhr et al., 2017, 2018; Hu et al., 2019), *visual commonsense reasonin* (Zellers et al., 2019), and so on. In the context of this paper, we focus on studying *cross-modality retrieval* (Barnard et al., 2003; Barnard and Forsyth, 2001; Gong et al., 2014; Hodosh et al., 2013; Young et al., 2014; Zhang et al., 2018), as well as transfer learning on downstream tasks, including *compositional attribute-object recognition* (Isola et al., 2015; Misra et al., 2017) and *referring expressions* (Dale and Reiter, 1995; Kazemzadeh et al., 2014; Kong et al., 2014; Mitchell et al., 2012). Please refer to § 5 for explanation of these tasks.

3 Denotation Graph (DG)

Visually grounded text expressions denote the images (or videos) they describe. When examined together, these expressions reveal structural relations that do not exhibit when each expression is studied in isolation. In particular, through linguistic analysis, these expressions can be grouped and partially ordered and thus form a relation graph, representing how (visually grounded) concepts are shared among different expressions and how different concepts are related. This insight was explored by Young et al. (2014) and the resulting graph is referred to as a denotation graph, schematically shown in the top part of Fig. 1. In this work, we focus on constructing denotation graphs from the FLICKR30K and the COCO datasets, where the text expressions are sentences describing images.

Formally, a denotation graph $\mathcal{G}$ is a polytree where a node v_i in the graph corresponds to a pair of a linguistic expression $\boldsymbol{y}_i$ and a set of images $\boldsymbol{X}_i = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{n_i}\}$. A directed edge e_{ij} from a node v_i to its child v_j represents a subsumption relation between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$. Semantically, $\boldsymbol{y}_i$ is more abstract (generic) than $\boldsymbol{y}_j$, and the tokens in $\boldsymbol{y}_i$ can be a subset of $\boldsymbol{y}_j$'s. For example, TWO DOGS describes all the images which TWO DOGS ARE RUNNING describes, though less specifically. Note that the subsumption relation is defined on the semantics of these expressions. Thus, the tokens do not have to be exactly matched on their surface forms. For instance, IN FRONT OF PERSON or IN FRONT OF CROWD are also generic concepts that

Table 1: Key statistics of the two DGs: averaged over the all nodes in the graph, internal nodes and leaf nodes (formated as all/internal/leaf)

Dataset	DG-FLICKR30K	DG-COCO
# of edges	1.94M	4.57M
# of nodes	597K/452K/145K	1.41M/841K/566K
# of tokens/node	6.78/4.45/14.04	5.88/4.07/8.58
# of images/node	4.46/5.57/1.00	5.06/7.79/1.00

subsume IN FRONT OF A CROWD OF PEOPLE, see the right-hand side of Fig. 1 for another example.

More formally, the set of images that correspond to $\boldsymbol{v}_i$ is the union of all the images corresponding to $\boldsymbol{v}_i$'s children $\text{ch}(v_i)$: $\boldsymbol{X}_i = \bigcup_{v_j \in \text{ch}(v_i)} \boldsymbol{X}_j$. We also use $\text{pa}(v_j)$ to denote the set of v_j's parents.

Denotation graphs (DG) can be seen as a hierarchical organization of semantic knowledge among concepts and their visual groundings. In this sense, they generalize the tree-structured object hierarchies that have been often used in computer vision. The nodes in the DG are composite phrases that are semantically richer than object names and the relationship among them is also richer.

Constructing DG We used the publicly available tool[1], following Young *et al.* (Young et al., 2014). For details, please refer to the Appendix and the reference therein. Once the graph is constructed, we attach the images to the proper nodes by set-union images of each node's children, starting from the sentence-level node.

DG-FLICKR30K and DG-COCO[2] We regenerate a DG on the FLICKR30K dataset[3] (Young et al., 2014) and construct a new DG on the COCO (Lin et al., 2014) dataset. The two datasets come from different visual and text domains where the former contains more iconic social media photos and the latter focuses on photos with complex scenes and has more objects. Figure 1 shows a random subgraph of DG-FLICKR30K.

Table 1 lists the key statistics of the two DGs. We note that in both graphs, a large number of internal nodes (more abstract concepts or phrases) are introduced. For such concepts, the linguistic expressions are much shorter and the number of images they correspond to is also larger.

4 Learning with Denotation Graphs

The denotation graphs, as described in the previous section, provide rich structures for learning representations of text and image. In what follows, we describe three learning objectives, starting from the most obvious one that matches images and their descriptions (§ 4.1), followed by learning to discriminate between general and specialized concepts (§ 4.2) and learning to predict concept relatedness (§ 4.3). We perform ablation studies of those objectives in § 5.4.

4.1 Matching Texts with Images

We suppose the image $\boldsymbol{x}$ and the text $\boldsymbol{y}$ are represented by (a set of) vectors $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\psi}(\boldsymbol{y})$ respectively. A common choice for $\boldsymbol{\phi}(\cdot)$ is the last layer of a convolutional neural network (He et al., 2015; Xie et al., 2017) and for $\boldsymbol{\psi}(\cdot)$ the contextualized word embeddings from a Transformer network (Vaswani et al., 2017). The embedding of the *multimodal* pair is a vector-valued function over $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\psi}(\boldsymbol{y})$:

$$\boldsymbol{v}(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\psi}(\boldsymbol{y})) \tag{1}$$

There are many choices of $f(\cdot, \cdot)$. The simplest one is to concatenate the two arguments. We can also use the element-wise product between the two if they have the same embedding dimension (Kiros et al., 2014), or complex mappings parameterized by layers of attention networks and convolutions (Lu et al., 2019; Chen et al., 2019) – we experimented some of them in our empirical studies.

4.1.1 Matching Model

We use the following probabilistic model to characterize the joint distribution

$$p(\boldsymbol{x}, \boldsymbol{y}) \propto \exp(\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{v}(\boldsymbol{x}, \boldsymbol{y})) \tag{2}$$

where the exponent $s(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{v}$ is referred as the matching score. To estimate $\boldsymbol{\theta}$, we use the

[1] Available online at https://github.com/aylai/DenotationGraph

[2] Both DGs are made publically available at https://sha-lab.github.io/DG/

[3] The original DG, while publicly available at https://shannon.cs.illinois.edu/DenotationGraph/ contains 1.75 million nodes which are significantly less than ours, due to the difference in the version of the NLP toolkit.

maximum likelihood estimation

$$\boldsymbol{\theta}^* = \arg\max \sum\nolimits_{v_i} \sum\nolimits_k \log p(\boldsymbol{x}_{ik}, \boldsymbol{y}_i) \quad (3)$$

where $\boldsymbol{x}_{ik}$ is the kth element in the set $\boldsymbol{X}_i$. However, this probability is intractable to compute as it requires us to get all possible pairs of $(\boldsymbol{x}, \boldsymbol{y})$. To approximate, we use negative sampling.

4.1.2 Negative Sampling

For each (randomly selected) positive sample $(\boldsymbol{x}_{ij}, \boldsymbol{y}_i)$, we explore 4 types of negative examples and assemble them as a negative sample set $\mathcal{D}^-_{ik}$:

Visually mismatched pair We randomly sample an image $\boldsymbol{x}^- \notin \boldsymbol{X}_i$ to pair with $\boldsymbol{y}_i$, *i.e.*, $(\boldsymbol{x}^-, \boldsymbol{y}_i)$. Note that we automatically exclude the images from v_i's children.

Semantically mismatched pair We randomly sample a text $\boldsymbol{y}_j \neq \boldsymbol{y}_i$ to form the pair $(\boldsymbol{x}_{ik}, \boldsymbol{y}_j)$. Note that we constrain $\boldsymbol{y}_j$ not to include concepts that could be more abstract than $\boldsymbol{y}_i$ as the more abstract can certainly be used to describe the specific images $\boldsymbol{x}_{ik}$.

Semantically hard pair We randomly sample a text $\boldsymbol{y}_j$ that corresponds to an image $\boldsymbol{x}_j$ that is visually similar to $\boldsymbol{x}_{ik}$ to form $(\boldsymbol{x}_{ik}, \boldsymbol{y}_j)$. See (Lu et al., 2019) for details.

DG Hard Negatives We randomly sample a *sibling* (but not cousin) node v_j to v_i such that $\boldsymbol{x}_{ik} \notin \boldsymbol{X}_j$ to form $(\boldsymbol{x}_{ik}, \boldsymbol{y}_j)$

Note that the last 3 pairs have increasing degrees of semantic confusability. In particular, the 4*th* type of negative sampling is only possible with the help of a denotation graph. In that type of negative samples, $\boldsymbol{y}_j$ is semantically very close to $\boldsymbol{y}_i$ (from the construction) yet they denote different images. The "semantically hard pair", on the other end, is not as hard as the last type as $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ could be very different despite high visual similarity.

With the negative samples, we estimate $\boldsymbol{\theta}$ as the minimizer of the following negative log-likelihood

$$\ell_{\text{MATCH}} = -\sum_{v_i}\sum_{k} \log \frac{e^{s(\boldsymbol{x}_{ik}, \boldsymbol{y}_i)}}{\sum_{(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \sim \mathcal{D}_i} e^{s(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}} \quad (4)$$

where $\mathcal{D}_i = \mathcal{D}^-_{ik} \cup \{(\boldsymbol{x}_{ik}, \boldsymbol{y}_i)\}$ contains both the positive and negative examples.

4.2 Learning to Be More Specific

The hierarchy in the denotation graph introduces an opportunity for learning image and text representations that are sensitive to fine-grained distinctions. Concretely, consider a parent node $\boldsymbol{v}_i$ with an edge to the child node $\boldsymbol{v}_j$. While the description $\boldsymbol{y}_j$ matches any images in its children nodes, the parent node's description $\boldsymbol{y}_i$ on a higher level is more abstract. For example, the concepts INSTRUMENT and PLAY PERCUSSION INSTRUMENT in Fig 1 is a pair of examples showing the latter more accurately describes the image(s) at the lower-level.

To incorporate this modeling notion, we introduce

$$\ell_{\text{SPEC}} = \sum_{e_{ij}}\sum_{k} [s(\boldsymbol{x}_{jk}, \boldsymbol{y}_i) - s(\boldsymbol{x}_{jk}, \boldsymbol{y}_j)]_+ \quad (5)$$

as a specificity loss, where $[h]_+ = \max(0, h)$ denotes the hinge loss. The loss is to be minimized such that the matching score for the less specific description $\boldsymbol{y}_i$ is smaller than that for the more specific description $\boldsymbol{y}_j$.

4.3 Learning to Predict Structures

Given the graph structure of the denotation graph, we can also improve the accuracy of image and text representation by modeling high-order relationships. Specifically, for a pair of nodes v_i and v_j, we want to predict whether there is an edge from v_i to v_j, based on each node's corresponding embedding of a pair of image and text. Concretely, this is achieved by minimizing the following negated likelihood

$$\ell_{\text{EDGE}} = -\sum\nolimits_{e_{ij}} \sum\nolimits_{k,k'} \log p(e_{ij} = 1 | \\ \boldsymbol{v}(\boldsymbol{x}_{ik}, \boldsymbol{y}_i), \boldsymbol{v}(\boldsymbol{x}_{jk'}, \boldsymbol{y}_j)) \quad (6)$$

We use a multi-layer perceptron with a binary output to parameterize the log-probability.

4.4 The Final Learning Objective

We combine the above loss functions as the final learning objective for learning on the DG

$$\ell_{\text{DG}} = \ell_{\text{MATCH}} + \lambda_1 \cdot \ell_{\text{SPEC}} + \lambda_2 \cdot \ell_{\text{EDGE}} \quad (7)$$

where λ_1, λ_2 are the hyper-parameters that trade-off different losses. Setting them to 1.0 seems to work well. The performance under different λ_1 and λ_2 are reported in Table 12 and Table 13. We study how each component could affect the learning of representation in § 5.4.

5 Experiments

We examine the effectiveness of using denotation graphs to learn image and text representations. We first describe the experimental setup and key implementation details (§ 5.1). We then describe key image-text matching results in § 5.2, followed by studies about the transfer capability of our learned representation (§ 5.3). Next, we present ablation studies over different components of our model (§ 5.4). Finally, we validate how well abstract concepts can be used to retrieve images, using our model (§ 5.5).

5.1 Experimental Setup

We list major details in the following to provide context, with the full details documented in the Appendix for reproducibility.

Embeddings and Matching Models Our aim is to show denotation graphs improve state-of-the-art methods. To this end, we experiment with two recently proposed state-of-the-art approaches and their variants for learning from multi-modal data: ViLBERT (Lu et al., 2019) and UNITER (Chen et al., 2019). The architecture diagrams and the implementation details are in the Appendix, with key elements summarized in the following.

Both the approaches start with an image encoder, which obtains a set of embeddings of image patches, and a text encoder which obtains a sequence of word (or word-piece) embeddings. For ViLBERT, text tokens are processed with Transformer layers and fused with the image information with 6 layers of co-attention Transformers. The output of each stream is then element-wise multiplied to give the fused embedding of both streams. For UNITER, both streams are fed into 12 Transformer layers with cross-modal attention. A special token CLS is used, and its embedding is regarded as the fused embedding of both streams.

For ablation studies, we use a smaller ViLBERT for rapid experimentation: ViLBERT (Reduced) where there are 3 Transformer layers and 2 co-attention Transformers for the text stream, and 1 Transformer layer for the image stream.

Constructing Denotation Graphs As described in §3, we construct denotation graphs DG-FLICKR30K and DG-COCO from the **FLICKR30K** (Young et al., 2014) and the **COCO** (Lin et al., 2014) datasets. FLICKR30K was originally developed for the tasks of image-based and text-based retrieval. It contains 29,000 images for training, 1,000 images for validation, and 1,000 images for testing. COCO is a significantly larger dataset, developed for the image captioning task. It contains 565,515 sentences with 113,103 images. We evaluate on both the 1,000 images testing split and the 5,000 images testing split (in the Appendix), following the setup in (Karpathy and Fei-Fei, 2015). Key characteristics for the two DGs are reported in Table 1.

Evaluation Tasks We evaluate the learned representations on three common *vision* + *language* tasks. In text-based image retrieval, we evaluate two settings: the text is either a sentence or a phrase from the test corpus. In the former setting, the sentence is a leaf node on the denotation graph, and in the latter case, the phrase is an inner node on the denotation graph, representing more general concepts. We evaluate the FLICKR30K and the COCO datasets, respectively. The main evaluation metrics we use are precisions at recall R@M where $M = 1, 5$ or 10 and RSUM which is the sum of the 3 precisions (Wu et al., 2019). Conversely, we also evaluate using the task of image-based text retrieval to retrieve the right descriptive text for an image.

In addition to the above cross-modal retrieval, we also consider two downstream evaluation tasks, *i.e.*, **Referring Expression** and **Compositional Attribute-Object Recognition**. (1) Referring Expression is a task where the goal is to localize the corresponding object in the image given an expression (Kazemzadeh et al., 2014). We evaluate on the dataset **REFCOCO+**, which contains 141,564 expressions with 19,992 images. We follow the previously established protocol to evaluate on the validation split, the TestA split, and the TestB split. We are primarily interested in zero-shot/few-shot learning performance. (2) Compositional Attribute-Object Recognition is a task that requires a model

Table 2: Text-based Image Retrieval (Higher is better)

Method	R@1	R@5	R@10	RSUM
FLICKR30K				
ViLBERT	59.1	85.7	92.0	236.7
ViLBERT + DG	63.8	87.3	92.2	243.3
UNITER	62.9	87.2	92.7	242.8
UNITER + DG	66.4	88.2	92.2	246.8
COCO 1K Test Split				
ViLBERT	62.3	89.5	95.0	246.8
ViLBERT + DG	65.9	91.4	95.5	252.7
UNITER	60.7	88.0	93.8	242.5
UNITER + DG	62.7	88.8	94.4	245.9
COCO 5K Test Split				
ViLBERT	38.6	68.2	79.0	185.7
ViLBERT + DG	41.8	71.5	81.5	194.8
UNITER	37.8	67.3	78.0	183.1
UNITER + DG	39.1	68.0	78.3	185.4

Table 3: Image-based Text Retrieval (Higher is better)

Method	R@1	R@5	R@10	RSUM
FLICKR30K				
ViLBERT	76.8	93.7	97.6	268.1
ViLBERT + DG	77.0	93.0	95.0	265.0
UNITER	78.3	93.3	96.5	268.1
UNITER + DG	78.2	93.0	95.9	267.1
COCO 1K Test Split				
ViLBERT	77.0	94.1	97.2	268.3
ViLBERT + DG	79.0	96.2	98.6	273.8
UNITER	74.4	93.9	97.1	265.4
UNITER + DG	77.7	95.0	97.5	270.2
COCO 5K Test Split				
ViLBERT	53.5	79.7	87.9	221.1
ViLBERT + DG	57.5	84.0	90.1	232.2
UNITER	52.8	79.7	87.8	220.3
UNITER + DG	51.4	78.7	87.0	217.1

to learn from images of SEEN (attribute, object) label pairs, such that it can generalize to recognize images of UNSEEN (attribute, object) label pairs. We evaluate this task on the **MIT-STATE** dataset (Isola et al., 2015), following the protocol by Misra et al. (2017). The training split contains 34,562 images from 1,262 SEEN labels, and the test split contains 19,191 images from 700 UNSEEN labels. We report the Top-1, 2, 3 accuracies on the UNSEEN test set as evaluation metrics.

Training Details Both ViLBERT and UNITER models are pre-trained on the Conceptual Caption dataset (Sharma et al., 2018) and the pre-trained models are released publicly[4]. On the DG-FLICKR30K, ViLBERT and UNITER are trained with a minibatch size of 64 and ViLBERT is trained for 17 epochs and UNITER for 15 epochs, with a learning rate of 0.00004. On the DG-COCO, ViLBERT is trained for 17 epochs and UNITER for 15 epochs with a minibatch size of 64 and a learning rate of 0.00004. The hyperparameters in Eq. (7) are set to 1.0, unless specified (see the Appendix).

5.2 Main Results

Table 2 and Table 3 report the performances on cross-modal retrieval. On both datasets, models trained with denotation graphs considerably outperform the corresponding ones which are not.

Table 4: Image Retrieval via Text (Transfer Learning)

SOURCE	FLICKR→COCO		COCO →FLICKR	
→TARGET	R@1	RSUM	R@1	RSUM
ViLBERT	43.5	199.5	49.0	209.0
+ SOURCE DG	44.9	200.5	52.8	218.2

For the image-based text retrieval task, ViLBERT and UNITER on FLICKR30K suffers a small drop in R@10 when DG is used. On the same task, UNITER on COCO 5K Test Split decreases more when DG is used. However, note that on both splits of COCO, ViLBERT is a noticeably stronger model, and using DG improves its performance.

5.3 Zero/Few-Shot and Transfer Learning

Transfer across Datasets Table 4 illustrates that the learned representations assisted by the DG have better transferability when applied to another dataset (TARGET DOMAIN) that is different from the SOURCE DOMAIN dataset which the DG is based on. Note that the representations are *not* fine-tuned on the TARGET DOMAIN. The improvement on the direction COCO →FLICKR30K is stronger than the reverse one, presumably because the COCO dataset is bigger than FLICKR30K. (R@5 and R@10 are reported in the Appendix.)

Zero/Few-shot Learning for Referring Expression We evaluate our model on the task of referring expression, a supervised learning task, in the setting of zero/few-shot transfer learning. In zero-shot learning, we didn't fine-tune the model on the referring expression dataset (*i.e.* REFCOCO+). Instead,

[4] The UNITER(Chen et al., 2019) model performs an additional online hard-negative mining (which we did not) during the training of image-text matching to improve their results. This is computationally very costly.

Table 5: Zero/Few-shot Learning for Referring Expression (Reported in R@1 on validation, TestA and TestB data)

Setting →	0% (Zero-shot)			25%			50%			100%		
Method	Val	TestA	TestB	Val	TestA	TestB	Val	TestA	TestB	Val	TestA	TestB
ViLBERT	35.7	41.8	29.5	67.2	74.0	57.1	68.8	75.6	59.4	71.0	76.8	61.1
ViLBERT + DG-COCO	36.1	43.3	29.6	67.4	74.5	57.3	69.3	76.6	59.3	71.0	77.0	60.8

Table 6: Image Recognition on UNSEEN Attribute-Object Pairs on the MIT-STATE Dataset

Method	Top-1	Top-2	Top-3
VisProd (Misra et al., 2017)	13.6	16.1	20.6
RedWine (Misra et al., 2017)	12.1	21.2	27.6
SymNet (Li et al., 2020)	19.9	28.2	33.8
ViLBERT pre-trained on			
N/A	16.2	26.3	33.3
COCO	17.9	28.8	36.2
DG-COCO	19.4	30.4	37.6

Table 7: Ablation Studies of Learning from DG

ViLBERT variants →	Reduced	Full
w/o DG	215.4	236.7
w/ DG		
$+\ \ell_{\text{MATCH}}$ $-$ DG HARD NEGATIVES	221.5	236.5
$+\ \ell_{\text{MATCH}}$	228.4	241.7
$+\ \ell_{\text{MATCH}} + \ell_{\text{SPEC}}$	228.8	242.6
$+\ \ell_{\text{MATCH}} + \ell_{\text{SPEC}} + \ell_{\text{EDGE}}$	231.2	243.3

we performed a "counterfactual" inference, where we measure the drop in the compatibility score (between a text describing the referring object and the image of all candidate regions) as we removed individual candidates results. The region that causes the biggest drop of compatibility score is selected. As a result, the selected region is most likely to correspond to the description. In the setting of few-shot learning, we fine-tune our COCO-pre-trained model on the task of referring expression in an end-to-end fashion on the referring expression dataset (*i.e.* REFCOCO+).

The results in Table 5 suggest that when the amount of labeled data is limited, training with DG performs better than training without. When the amount of data is sufficient for end-to-end training, the advantage of training with DG diminishes.

Compositional Attribute-Object Recognition We evaluate our model for supervised compositional attribute-object recognition (Misra et al., 2017), and report results on recognizing UNSEEN attribute-object labels on the MIT-STATE test data (Isola et al., 2015). Specifically, we treat the text of image labels (*i.e.*, attribute-object pairs as compound phrases) as the sentences to fine-tune the ViLBERT models, using the ℓ_{MATCH} objective. Table 6 reports the results (in top-K accuracies) of both prior methods and variants of ViLBERT, which are trained from scratch (N/A), pre-trained on COCO and DG-COCO, respectively. ViLBERT models pre-trained with parallel pairs of images and texts (*i.e.*, COCO and DG-COCO) improve significantly over the baseline that is trained on the MIT-STATE from scratch. The model pre-trained with DG-COCO achives the best results among ViLBERT variants. It performs on par with the previous state-of-the-art method in top-1 accuracy and outperforms them in top-2 and top-3 accuracies.

5.4 Ablation Studies

The rich structures encoded in the DGs give rise to several components that can be incorporated into learning representations. We study whether they are beneficial to the performances on the downstream task of text-based image retrieval. In the notions of §4, those components are: (1) remove "DG HARD NEGATIVES" from the ℓ_{MATCH} loss and only use the other 3 types of negative samples (§ 4.1); (2) align images with more specific text descriptions (§ 4.2); (3) predict the existences of edges between pairs of nodes (§ 4.3).

Table 7 shows the results from the ablation studies. We report results on two versions of ViLBERT: In ViLBERT (reduced), the number of parameters in the model is significantly reduced by making the model less deep, and thus faster for development. Instead of being pre-trained, they are trained on the FLICKR30K dataset directly for 15 epochs with a minibatch size of 96 and a learning rate of $4e^{-5}$. In ViLBERT (Full), we use the aforementioned settings. We report RSUM on the FLICKR30K dataset for the task of text-based image retrieval.

All models with DG perform better than the mod-

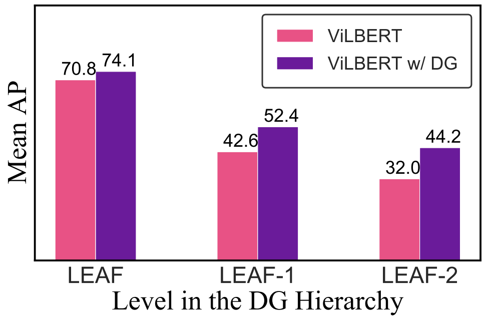


Figure 2: Image Retrieval using Mid-level Linguistic Expression on FLICKR30K Denotation Graph. The results are reported in Mean Average Precision (Mean AP).

els without DG. Secondly, the components of DG HARD NEGATIVES, ℓ_{SPEC}, and ℓ_{EDGE} contribute positively and their gains are cumulative.

5.5 Image Retrieval from Abstract Concepts

The leaf nodes in a DG correspond to complete sentences describing images. The inner nodes are shorter phrases that describe more abstract concepts and correspond to a broader set of images, refer to Table 2 for some key statistics in this aspect.

Fig. 2 contrasts how well abstract concepts can be used to retrieve images. The concepts are the language expressions corresponding to the leaf nodes, the nodes that are one level above (LEAF-1), or two levels above (LEAF-2) the leaf nodes from the DG-FLICKR30K. Since abstract concepts tend to correspond to multiple images, we use mean averaged precision (mAP) to measure the retrieval results. ViLBERT+DG outperforms ViLBERT significantly. The improvement is also stronger when the concepts are more abstract.

It is interesting to note that while the ℓ_{MATCH} used in ViLBERT w/ DG incorporates learning representations to align images at both specific and abstract levels, such learning benefits all levels. The improvement of retrieving at abstract levels does not sacrifice the retrieval at specific levels.

6 Conclusion

Image and text aligned data is rich in semantic correspondence. Besides treating text annotations as "categorical" labels, in this paper, we show that we can make full use of those labels. Concretely, denotation graphs (DGs) encode structural relations that can be automatically extracted from those texts with linguistic analysis tools. We proposed several ways to incorporate DGs into learning representation and validated the proposed approach on several tasks. We plan to investigate other automatic tools in curating more accurate denotation graphs with a complex composition of fine-grained concepts for future directions.

Acknowledgement

We appreciate the feedback from the reviewers. This work is partially supported by NSF Awards IIS-1513966/ 1632803/1833137, CCF-1139148, DARPA Award#: FA8750-18-2-0117, FA8750-19-1-0504 , DARPA-D3M - Award UCB-00009528, Google Research Awards, gifts from Facebook and Netflix, and ARO# W911NF-12-1-0241 and W911NF-15-1-0484. We particularly thank Haoshuo Huang for help in improving the efficiency of DG generation.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018a. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *ICCV*.

Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M Blei, and Michael I Jordan. 2003. Matching words and pictures. *JMLR*.

Kobus Barnard and David Forsyth. 2001. Learning the semantics of words and pictures. In *ICCV*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft COCO Captions: Data collection and evaluation server. *ArXiv 1504.00325*.

Xinlei Chen and C Lawrence Zitnick. 2015. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. UNITER: Learning universal image-text representations. *ArXiv 1909.11740*.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Aviv Eisenschtat and Lior Wolf. 2017. Linking image and text with 2-way nets. In *CVPR*.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving visual-semantic embeddings with hard negatives.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *CVPR*.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In *NeurIPS*, pages 2121–2129.

Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*.

Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. 2018. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *CVPR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CVPR*.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*.

Hexiang Hu, Ishan Misra, and Laurens van der Maaten. 2019. Evaluating text-to-image matching using binary image selection (BISON). In *ICCV workshop*.

Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. 2018. Learning semantic concepts and order for image and sentence matching. In *CVPR*.

Phillip Isola, Joseph J Lim, and Edward H Adelson. 2015. Discovering states and transformations in image collections. In *CVPR*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ICLR*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *NeurIPS Workshop Deep Learning*.

Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. 2014. What are you talking about? text-to-image coreference. In *CVPR*.

Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. 2013. Babytalk: Understanding and generating simple image descriptions. *T-PAMI*.

Alice Lai and Julia Hockenmaier. 2017. Learning to predict denotational probabilities for modeling entailment. In *ACL*.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. 2018. Stacked cross attention for image-text matching. In *ECCV*.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2019a. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. *ArXiv 1908.06066*.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. 2019b. Visual semantic reasoning for image-text matching. In *ICCV*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019c. VisualBERT: A simple and performant baseline for vision and language. *ArXiv 1908.03557*.

Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. 2020. Symmetry and group in attribute-object compositions. In *CVPR*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *ECCV*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ishan Misra, Abhinav Gupta, and Martial Hebert. 2017. From red wine to red tomato: Composition with context. In *CVPR*.

Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. 2012. Midge: Generating image descriptions from computer vision detections. In *EACL*.

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. 2017. Dual attention networks for multimodal reasoning and matching. In *CVPR*.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Technical Report*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*.

Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *T-ACL*, 2:207–218.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of generic visual-linguistic representations. *ICLR*.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *ACL*.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2018. A corpus for reasoning about natural language grounded in photographs.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. *ICLR*.

Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. 2019. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *CVPR*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv 1609.08144*.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. *CVPR*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *T-ACL*, 2:67–78.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Bowen Zhang, Hexiang Hu, and Fei Sha. 2018. Cross-modal and hierarchical modeling of video and text. In *ECCV*.

Appendix

In the Appendix, we provide details omitted from the main text due to the limited space, including:

- § A describes complete implementation details (cf. § 3 and § 5.1 of the main text).
- § B provides complete experimental results (cf. § 5.2 of the main text).
- § C visualizes the model's predictions on denotation graphs.

A Implementation Details

A.1 Constructing Denotation Graphs

We summarize the procedures used to extract DG from *vision + language* datasets. For details, please refer to (Young et al., 2014). We used the publicly available tool[5]. The analysis consists of several steps: (1) spell-checking; (2) tokenize the sentences into words; (3) tag the words with Part-of-Speech labels and chunk works into phrases; (4) abstract semantics by using the WordNet (Miller, 1995) to construct a hypernym lexicon table to replace the nouns with more generic terms; (5) apply 6 types of templated rules to create fine-to-coarse (*i.e.*, specific to generic) semantic concepts and connect the concepts with edges.

We set 3 as the maximum levels (counting from the sentence level) to extract abstract semantic concepts. This is due to the computation budget we can afford, as the final graphs can be huge in both the number of nodes and the edges. Specifically, without the maximum level constraint, we have 2.83M concept nodes in total for Flickr dataset. If the training is run on all these nodes, we will consume 19 times more iterations than training on the original dataset, which has 145K sentences (Young et al., 2014). As a result, much more time would be required for every experiment. With the 3 layers of DG from the leaf concepts, we have in 597K nodes. In this case, the training time would be cut down to 4.1 times of the original dataset.

Nonetheless, we experimented with more than 3 levels to train ViLBERT + DG-FLICKR30K with 5 and 7 maximum levels, respectively. The training hyper-parameters remain the same as ViLBERT + DG-FLICKR30K with 3 maximum layers. The aim is to check how much gain we could get from the additional annotations. We report the results in Table 8. It shows that actually, the model trained with 3 levels of DG achieves the best performance. This might be because those high-level layers of DG (counting from the sentences) contain very abstract text concepts, such as "entity" and "physical object", which is non-informative in learning the visual grounding.

Table 8: Text-based Image Retrieval Performance of ViLBERT trained with different number of DG levels

# of DG levels	R@1	R@5	R@10	RSUM
3 levels	65.9	91.4	95.5	252.7
5 levels	62.5	86.4	92.3	241.2
7 levels	62.8	86.3	91.6	240.7

Once the graph is constructed, we attach the images to the proper nodes by set-union images of each node's children, starting from the sentence-level node.

A.2 Model architectures of ViLBERT and UNITER

A comparison of these models is schematically illustrate in Fig. 3.

- ViLBERT. It has 6 basic Transformer layers for text and 8 layers for image. For all the Transformer layers on the text side, we use 12 attention heads and 256 feature dimensions, then linearly project down to 1024 feature dimensions. For all the Transformers on the image side, we use 8 attention heads and 128 feature dimensions, then combine into 1024 feature dimensions too.
- UNITER. All the Transformer layers have 12 heads and 256 feature dimensions.

The major difference between UNITER and ViLBERT is how attentions are used. In ViLBERT, one modality is used as a query, and the other is used as value and key. In UNITER, however, both are used as query, key, and value. Additionally, UNITER

[5] https://github.com/aylai/DenotationGraph

is similar to another model Unicoder-VL (Li et al., 2019a). However, the latter has not provided publicly available code for experimenting.

For ViLBERT model, each text and image co-attention Transformer layer contains 8 attention heads with 1024 dimensions in total. The text Transformer layer contains 12 attention heads with 3072 hidden dimensions in total. In contrast, the image Transformer layer has 8 attention heads with 1024 hidden dimensions in total. For UNITER model, each cross-attention Transformer layer contains 12 heads with 3072 hidden dimensions in total.

ViLBERT model contains 121 million parameters, while UNITER contains 111 million parameters.

A.3 Training Details

All models are optimized with the Adam optimizer (Kingma and Ba, 2015). The learning rate is initialized as $4e^{-5}$. Following ViLBERT (Lu et al., 2019), a warm-up training session is employed, during which we linearly increase the learning rate from 0 to $4e^{-5}$ in the first 1.5% part of the training epochs. The learning rate is dropped to $4e^{-6}$ and $4e^{-7}$ at the 10th and the 15th epochs, respectively. For ViLBERT (Reduced), we randomly initialized the model parameters in the image stream. The text stream is initialized from the first 3 layers of the pre-trained BERT model, and its co-attention Transformer layers are randomly initialized. For ViLBERT (Full) and UNITER (Chen et al., 2019), we load the model's weights pre-trained on the Conceptual Caption dataset to initialize them.

Training ViLBERT (Full) + DG with a minibatch size of 64 takes 2 to 3 days on an 8 TitanXp GPU server, or 1 day on TPU v2 cloud. The GPU server is equipped with Intel Xeon Gold 6154 CPU and 256G RAM.

A.4 Text Pre-processing

We follow BERT (Devlin et al., 2019) that uses WordPiece (Wu et al., 2016) tokenizer to tokenize the texts. For ViLBERT (Reduced) and ViLBERT (Full), we use the uncased tokenizer with a vocabulary size of 30,522. For UNITER, we use the cased tokenizer with a vocabulary size of 28,996. After tokenization, the tokens are transformed to 768 dimension features by a word embedding initialized from BERT pre-trained model. The 768-dimensional position features are included in the input to represent the position of each token.

A.5 Visual Pre-processing

For both ViLBERT and UNITER, we use the image patch features generated by the bottom-up attention features, as suggested by the original papers (Anderson et al., 2018a). The image patch features contain up to 100 image patches with their dimensions to be 2048. Besides this, a positional feature is used to represent the spatial location of bounding boxes for both ViLBERT and UNITER. Specifically, ViLBERT uses 5-dimensional position feature that encodes the normalized coordinates of the upper-left and lower-right corner for the bounding boxes, as well as one additional dimension encoding the normalized patch size. UNITER uses two additional spatial features that encode the normalized width and height of the object bounding box.

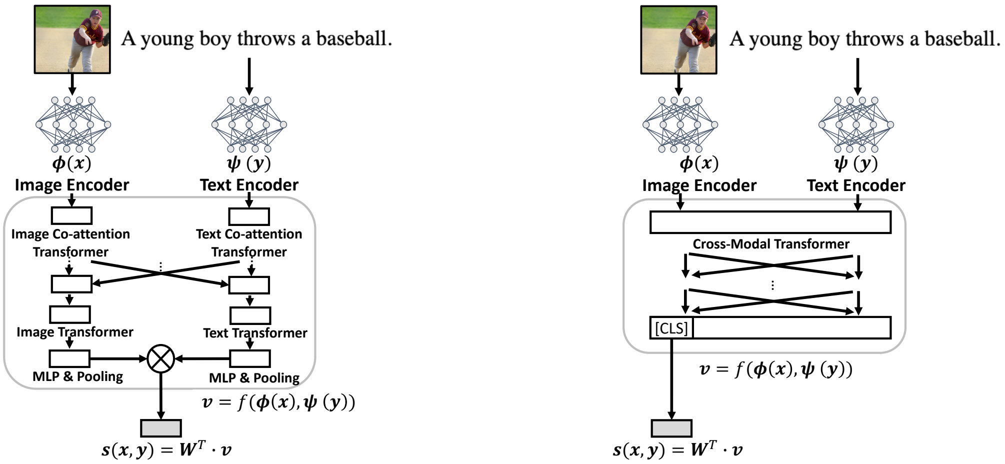
B Full Experimental Results

In this section, we include additional experimental results referred to by the main text. Specifically, we include results from a variety of models (*e.g.*, ViLBERT, ViLBERT + DG, UNITER, and UNITER + DG) on COCO dataset 5K test split (Karpathy and Fei-Fei, 2015) in § B.1. Then we provide a comprehensive ablation study on the impact of λ_1 and λ_2 of Eq. 7 in the main text in § B.3.

B.1 Complete Results on COCO Dataset

We report the full results on COCO dataset (1K test split and 5K test split) in Table 9 and Table 10. Additionally, we contrast to other existing approaches on these tasks. It could be seen that ViLBERT + DG and UNITER + DG improves the performance over the counterparts without DG by a significant margin on both COCO 1K and 5K test split – the only exception is that on the task of image-based text retrieval, UNITER performs better than UNITER+DG.

These results support our claim that training with DG helps the model to learn better visual and lin-



(a) **ViLBERT**

(b) **UNITER**

Figure 3: Architecture of (a) ViLBERT, (b) UNITER. The ⊗ means element-wise product. The [CLS] represents the embedding of [CLS] token in the last UNITER layer.

guistic features. Although ViLBERT and UNITER have different architectures, training with DG could improve the performance consistently.

B.2 Complete Results on FLICKR30K Dataset

We contrast to other existing approaches in Table 11 on the task of text-based image retrieval on the FLICKR30K dataset.

B.3 Ablation Study on λ_1 and λ_2

We conduct an ablation study on the impact of the two hyper-parameters λ_1 and λ_2 in Eq. 7 of the main text. We conduct the study with two ViLBERT variants: ViLBERT Reduced and ViLBERT. The results are reported in Table 12 and Table 13. As we have two hyper-parameters λ_1 and λ_2, we analyze their impacts on the final results by fixing one λ to be 1. Fixing the $\lambda_2 = 1$ and changing λ_1, we observe that ViLBERT prefers larger λ_1, while ViLBERT Reduced achieves slightly worse performance when λ_1 is smaller or larger. Fixing the $\lambda_1 = 1$ and changing λ_2, we observe that performance of both architectures slightly reduced when $\lambda_2 = 0.5$ and $\lambda_2 = 2$.

B.4 Full Results on Zero/Few-Shot and Transfer Learning

Implementation Details for Zero-shot Referring Expression Specifically, the learned ViLBERT and ViLBERT w/DG models are used first to produce a base matching score s_{BASE} between the expression to be referred and the whole image. We then compute the matching score s_{MASKED} between the expression and the image with each region feature being replaced by a random feature in turn. As the masked image region might be a noisy region, s_{MASKED} might be larger than s_{BASE}. Therefore, the model's prediction of which region the expression refers to is the masked region which causes the largest score in s_{REGION}, where

$$s_{\text{REGION}} = (s_{\text{BASE}} - s_{\text{MASKED}}) \cdot \mathbb{I}[s_{\text{MASKED}} > s_{\text{BASE}}].$$

Here $\mathbb{I}[\cdot]$ is an indicator function. Table 5 shows that ViLBERT + DG-COCO outperforms ViLBERT on this task.

Transfer Learning Results Table 14 reports the full set of evaluation metrics on transferring across datasets. Training with DG improves training without DG noticeably.

C Visualization of Model's Predictions on Denotation Graphs

We show several qualitative examples of both success and failure cases of ViLBERT + DG, when retrieving the text matched images, in Fig. 4 and Fig. 5. The image and text correspondence is generated by the Denotation Graph, which are derived from the caption and image alignment. We observe that in the Fig.4, the ViLBERT + DG successfully

Table 9: Results on Cross-Modal Retrieval on COCO dataset 1K test split (Higher is better)

Text-based Image Retrieval

Method	R@1	R@5	R@10	RSUM
Models ran or implemented by us				
ViLBERT	62.3	89.5	95.0	246.8
ViLBERT + DG	65.9	91.4	95.5	252.7
UNITER	60.7	88.0	93.8	242.5
UNITER + DG	62.7	88.8	94.4	245.9
Known results from literature				
VSE++(Faghri et al., 2018)	52.0	84.3	92.0	228.3
SCO(Huang et al., 2018)	56.7	87.5	94.8	239.0
SCAN(Lee et al., 2018)	58.8	88.4	94.8	242.0
VSRN(Li et al., 2019b)	62.8	89.7	95.1	247.6

Image-based Text Retrieval

Method	R@1	R@5	R@10	RSUM
Models ran or implemented by us				
ViLBERT	77.0	94.1	97.2	268.3
ViLBERT + DG	79.0	96.2	98.6	273.8
UNITER	74.4	93.9	97.1	265.4
UNITER + DG	77.7	95.0	97.5	270.2
Known results from literature				
VSE++(Faghri et al., 2018)	64.6	90.0	95.7	250.3
SCO(Huang et al., 2018)	69.9	92.9	97.5	260.3
SCAN(Lee et al., 2018)	72.7	94.8	98.4	265.9
VSRN(Li et al., 2019b)	76.2	94.8	98.2	269.2

Table 10: Results on Cross-Modal Retrieval on COCO dataset 5K test split (Higher is better)

Text-based Image Retrieval

Method	R@1	R@5	R@10	RSUM
Models ran or implemented by us				
ViLBERT	38.6	68.2	79.0	185.7
ViLBERT + DG	41.8	71.5	81.5	194.8
UNITER	37.8	67.3	78.0	183.1
UNITER + DG	39.1	68.0	78.3	185.4
Known results from literature				
VSE++(Faghri et al., 2018)	30.3	59.4	72.4	162.1
SCO(Huang et al., 2018)	33.1	62.9	75.5	171.5
SCAN(Lee et al., 2018)	38.6	69.3	80.4	188.3
VSRN(Li et al., 2019b)	40.5	70.6	81.1	192.2
UNITER(Chen et al., 2019)†	48.4	76.7	85.9	211.0

Image-based Text Retrieval

Method	R@1	R@5	R@10	RSUM
Models ran or implemented by us				
ViLBERT	53.5	79.7	87.9	221.1
ViLBERT + DG	57.5	84.0	90.1	232.2
UNITER	52.8	79.7	87.8	220.3
UNITER + DG	51.4	78.7	87.0	217.1
Known results from literature				
VSE++(Faghri et al., 2018)	41.3	71.1	81.2	193.6
SCO(Huang et al., 2018)	42.8	72.3	83.0	198.1
SCAN(Lee et al., 2018)	50.4	82.2	90.0	222.6
VSRN(Li et al., 2019b)	53.0	81.1	89.4	223.5
UNITER (Chen et al., 2019)†	63.3	87.0	93.1	243.4

†: The UNITER(Chen et al., 2019) model performs an additional online hard-negative mining (which we did not) during the training of image-text matching to improve their results, which is computationally very costly.

recognizes the images that are aligned with the text: "man wear reflective vest", while the ViLBERT fails to retrieve the matched image. In the failure case in Fig. 5, although ViLBERT + DG fails to retrieve the images that are exactly matched to the text, it still retrieves very relevant images given the query.

Table 11: Results on Text-based Image Retrieval on FLICKR30K test split (Higher is better)

Method	R@1	R@5	R@10	RSUM
Models ran or implemented by us				
ViLBERT	59.1	85.7	92.0	236.7
ViLBERT + DG	63.8	87.3	92.2	243.3
UNITER	62.9	87.2	92.7	242.8
UNITER + DG	66.4	88.2	92.2	246.8
Known results from literature				
VSE++(Faghri et al., 2018)	39.6	70.1	79.5	189.2
SCO(Huang et al., 2018)	41.1	70.5	80.1	191.7
SCAN(Lee et al., 2018)	48.6	77.7	85.2	211.5
VSRN(Li et al., 2019b)	54.7	81.8	88.2	224.7
ViLBERT(Lu et al., 2019)	58.2	84.9	91.5	234.6
UNITER(Chen et al., 2019)	71.5	91.2	95.2	257.9

Table 12: Ablation studies on the impact of λ_1 and λ_2 of ViLBERT Reduced on Text-based Image Retrieval on FLICKR30K dataset (Higher is better)

(a) Ablating λ_1

λ_1	λ_2	R@1	R@5	R@10	RSUM
0.5	1.0	57.7	83.1	88.5	229.2
1.0	1.0	58.7	83.3	89.3	231.2
2	1.0	56.5	82.6	88.6	227.7

(b) Ablating λ_2

λ_1	λ_2	R@1	R@5	R@10	RSUM
1.0	0.5	56.3	81.7	87.2	225.2
1.0	1.0	58.7	83.3	89.3	231.2
1.0	2	58.5	82.3	88.0	228.9

Table 13: Ablation studies on the impact of λ_1 and λ_2 of ViLBERT on Text-based Image Retrieval on FLICKR30K dataset (Higher is better)

(a) Ablating λ_1

λ_1	λ_2	R@1	R@5	R@10	RSUM
0.5	1.0	63.1	86.7	91.7	241.4
1.0	1.0	63.8	87.3	92.2	243.3
2	1.0	64.1	87.6	92.5	244.2

(b) Ablating λ_2

λ_1	λ_2	R@1	R@5	R@10	RSUM
1.0	0.5	63.7	87.0	92.4	243.2
1.0	1.0	63.8	87.3	92.2	243.3
1.0	2	63.1	86.6	91.9	241.6

Table 14: Transferrability of the learned representations

SOURCE→TARGET	FLICKR30K →COCO				COCO →FLICKR30K			
Model	R@1	R@5	R@10	RSUM	R@1	R@5	R@10	RSUM
ViLBERT	43.5	72.5	83.4	199.4	49.0	76.0	83.9	209.0
ViLBERT + SOURCE DG	44.9	72.7	83.0	200.5	52.8	79.2	86.2	218.2

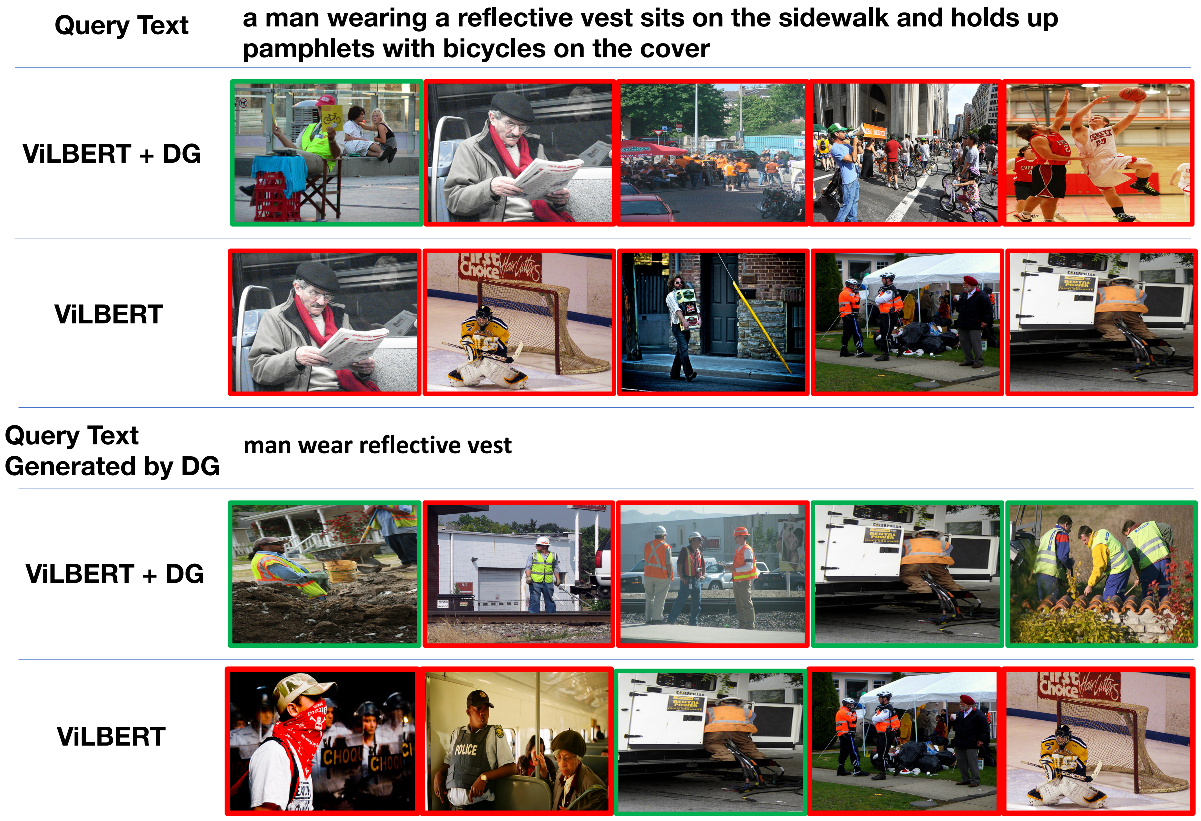

Query Text
a man wearing a reflective vest sits on the sidewalk and holds up pamphlets with bicycles on the cover
ViLBERT + DG
ViLBERT
Query Text Generated by DG
man wear reflective vest
ViLBERT + DG
ViLBERT

Figure 4: **FLICKR30K Denotation Graph: Given Text and Retrieve Image.** Qualitative example of ViLBERT + DG successfully retrieves the text matched images. We mark the correct sample in green and incorrect one in red.

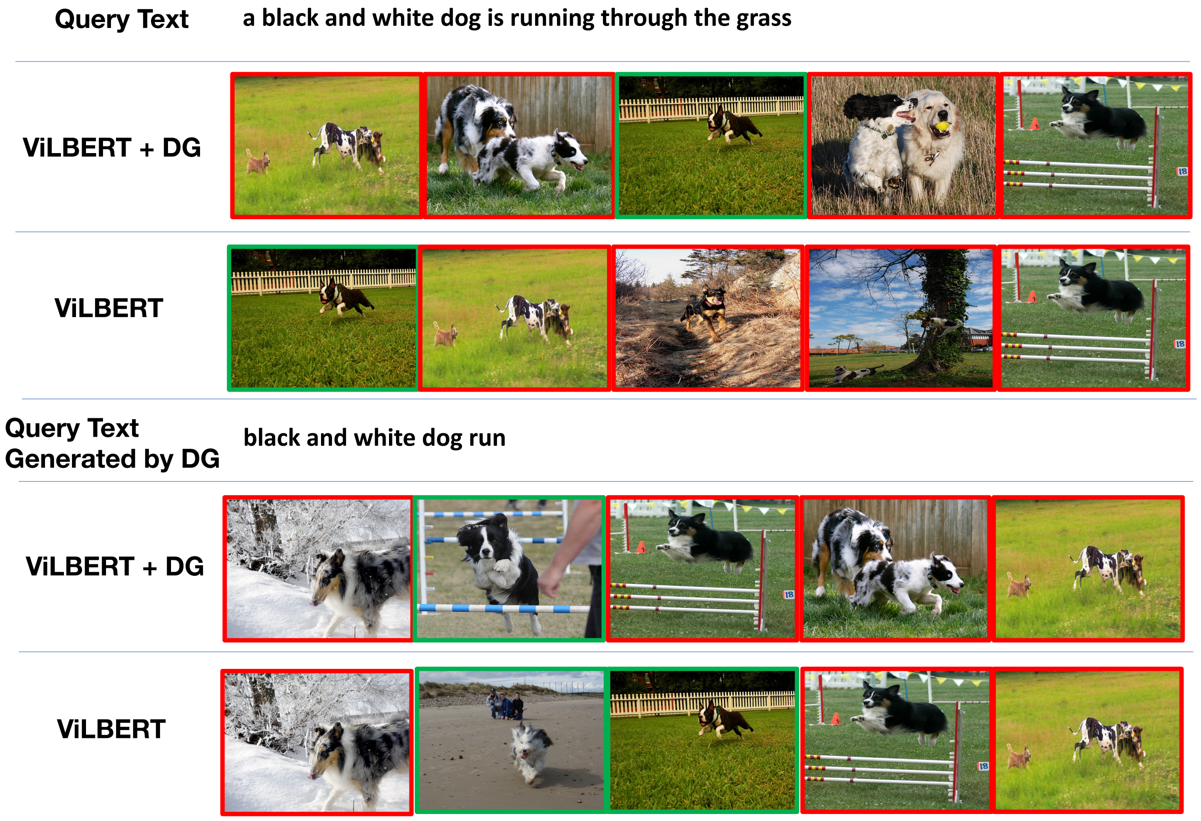

Query Text
a black and white dog is running through the grass
ViLBERT + DG
ViLBERT
Query Text Generated by DG
black and white dog run
ViLBERT + DG
ViLBERT

Figure 5: **FLICKR30K Denotation Graph: Given Text and Retrieve Image.** Qualitative example of ViLBERT + DG fails to retrieve the text matched images. We mark the correct sample in green and incorrect one in red.