Composing Object Relations and Attributes for Image-Text Matching

Khoi Pham[1] Chuong Huynh[1] Ser-Nam Lim[2] Abhinav Shrivastava[1]
[1]University of Maryland, College Park [2]University of Central Florida

Abstract

*We study the visual semantic embedding problem for image-text matching. Most existing work utilizes a tailored cross-attention mechanism to perform local alignment across the two image and text modalities. This is computationally expensive, even though it is more powerful than the unimodal dual-encoder approach. This work introduces a dual-encoder image-text matching model, leveraging a scene graph to represent captions with nodes for objects and attributes interconnected by relational edges. Utilizing a graph attention network, our model efficiently encodes object-attribute and object-object semantic relations, resulting in a robust and fast-performing system. Representing caption as a scene graph offers the ability to utilize the strong relational inductive bias of graph neural networks to learn object-attribute and object-object relations effectively. To train the model, we propose losses that align the image and caption both at the holistic level (image-caption) and the local level (image-object entity), which we show is key to the success of the model. Our model is termed **C**omposition model for **O**bject **R**elations and **A**ttributes, **CORA**. Experimental results on two prominent image-text retrieval benchmarks, Flickr30K and MSCOCO, demonstrate that CORA outperforms existing state-of-the-art computationally expensive cross-attention methods regarding recall score while achieving fast computation speed of the dual encoder. Our code is available at* https://github.com/vkhoi/cora_cvpr24

1. Introduction

Image-text matching is a fundamental computer vision problem that aims to measure the semantic correspondence between an image and a text. Such correspondence can be used for image retrieval given a text description, or text retrieval provided an image query, both of which are important in various computer vision applications (*e.g.*, weakly supervised problems [18, 19]). The problem is inherently challenging due to the ambiguous nature of the image and text modalities [6, 46]. For example, an image can depict a complicated situation that a multitude of different captions

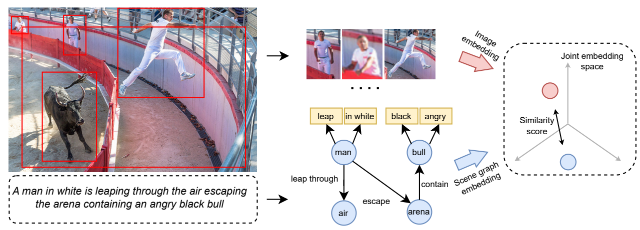

Figure 1. **Illustration of CORA.** CORA has a dual-encoder architecture, consisting of one encoder that embeds the input image and one encoder that embeds the text caption scene graph into a joint embedding space. (Best viewed in color and zoomed in.)

can describe, whereas a single caption is too abstract and can semantically apply to multiple images. Various studies have been proposed and can be categorized into two main directions: (1) the unimodal dual encoder and (2) the cross-attention approach.

In the dual-encoder framework, two modality-independent encoders embed the image and text caption separately into a joint embedding space. In this space, a similarity function such as a dot product can measure the image-text similarity. This strategy is also referred to as the global alignment approach, as the goal is to holistically represent an image (or text) as a single embedding. Due to their simplicity and low computational cost (*e.g.*, retrieving an image given a text query can be done via a vector-matrix multiplication with the cached embeddings), such methods are more widely adopted for real-world retrieval databases.

The second approach, cross-attention network, constitutes the majority of recent work. Instead of embedding each modality separately, cross-modality attention is adopted to locally align fine-grained visual cues of an image (image regions) with textual cues of a caption (word tokens), from which the overall correspondence score is aggregated. While this approach outperforms dual encoder in terms of power, it presents a substantial computational challenge. Upon receiving a text (or image) query, every image vs. text query pair must be processed through the cross-attention model to determine their similarity scores. This requirement renders the method impractical for retrieval systems managing large databases due to its extensive computational demands. This work focuses on the dual-encoder approach and shows that our dual-encoder proposal even outperforms the SOTA cross-attention networks.

Existing approaches use a text sequence model (*e.g.*, GRU [7], LSTM [15]) to encode the text caption. A text usually contains an extensive range of semantic information, such as object categories, attributes of objects, and relations between objects. Attributes describe appearance of objects [22, 36, 38, 39, 44], while relations describe how objects interact with one another [56]. Forcing a text sequence model to learn to parse a caption into different levels of semantics is challenging, especially in the low data regime. For example, by design, a sequence model that simply processes a caption from left to right (GRU, LSTM) may find it challenging to determine which attributes belong to an object and which objects participate in a relation. Numerous works have shown that Transformer-based text sequence models (BERT [8]) can produce good structural parsing of a sentence [14], however, these models must be trained on large amounts of data. Nevertheless, it has been shown in [3] that even the CLIP text encoder [42] in Stable Diffusion [40, 43] still exhibits incorrect object-attribute binding (*i.e.*, pair an attribute with the wrong object in the sentence) despite having been trained on large datasets. Therefore, it becomes desirable to have a text embedding model that can capture the semantic relations between concepts accurately.

In this work, instead of a sequence model, we propose representing a caption as a scene graph of object and attribute nodes connected by relation edges. An example of a scene graph is illustrated in Fig. 1, where we show that semantic structures such as object-attribute and object-object pairings are already organized. To this end, we propose our **C**omposition model for **O**bject **R**elations and **A**ttributes, **CORA**, a dual-encoder model for image-text matching. On the image side, we re-use GPO [4] which is a SOTA pooling operator for image-text matching to embed the image as a vector. On the text side, we propose to use a graph attention network [2, 48] with strong relational inductive bias to produce a holistic scene graph embedding for the caption. Scene graph-based approaches have been previously explored in [25, 28, 30, 51] for image-text matching, but they all employ expensive cross-attention. In addition to the margin-based triplet ranking loss [10] adopted by prior work, we propose a contrastive loss to guide CORA in making alignment at both the holistic image-caption level and the local image-object entity level. The proposed loss helps make training more stable and result in better downstream retrieval accuracy, as well as additionally acquires CORA with the image-object entity retrieval capability.

Our model is evaluated on two image-text retrieval benchmarks, Flickr30K and MS-COCO, where it outperforms SOTA dual-encoder and expensive cross-attention methods. Our paper makes the following contributions:

- We propose CORA, a dual encoder for image-text matching that uses a graph attention network instead of a sequence model to produce scene graph embedding for a caption.
- We propose using contrastive loss that trains the model to make global alignment (image-caption) and local alignment (image-object entity), resulting in more stable training, better retrieval accuracy, and image-object retrieval capability.
- Our model CORA achieves SOTA retrieval performance on Flickr30K and MS-COCO, two prominent benchmarks for image-text retrieval.

2. Related Work

Dual-encoder. This approach is dominant in earlier works [10, 11, 21, 24, 50] in image-text matching. The image and text captions are independently embedded in a joint metric space where matching image-caption pairs are located close to each other. Existing work in this paradigm often improves the joint embedding space by introducing new losses [6, 10], proposing new architecture for each modality encoder [24, 52, 54], or learning better pooling methods [4, 26]. For example, VSE++ [10] proposes a triplet loss with hard negative mining which has been adopted by all following image-text matching work. VSRN [24], DSRAN [52], SAEM [54] implement graph convolution and self-attention to improve the encoder architecture. GPO [4] achieves competitive results by designing a new pooling operator that can learn from data. Recently, MV-VSE [26] and SDE [20] propose using multiple embeddings per sample data, and HREM [12] presents a dual-encoder model that can be trained with a cross-modality matching loss for enhancing the embedding quality.

Cross-attention. In contrast to embedding the image and text independently, this approach considers the fine-grained local correspondence between image features and text tokens before computing the similarity. SCAN [23] is the first representative work that introduces this idea of using cross-attention between the two modalities to find their alignments. CAAN [58] later improves the idea by employing an additional intra-modal interaction step after the cross-modal interaction. SGARF [9] proposes to learn jointly from both the global and local alignment to highlight important image regions. Recently, NAAF [57] encourages the dissimilarity degrees between mismatched pairs of image region and word to boost the similarity matching, and CHAN [35] proposes a new cross-modal alignment method that can neglect the redundant misalignments.

Graph-based image-text matching. Among both dual-encoder and cross-attention methods, some have utilized scene graphs as part of their pipeline for more accurate image-text alignment [25, 28, 30, 51]. Frameworks based on this approach leverage the capacity of Graph Convolutional Networks (GCN) to capture the spatial and semantic relationships between visual regions and textual tokens. For example, SGM [51], GCN+DIST [25], GraDual [30] utilize off-the-shelf visual scene graph generator [56] to extract scene graph from images, then perform cross-modal alignment between the visual and textual graph. GSMN [28], on the other hand, uses a fully connected graph for the visual regions but additionally uses the regions' polar coordinates to encode their spatial relationships.

In our work, we build upon the scene graph representation of the caption to develop the text encoder for our dual-encoder model. Our model focuses on explicitly learning to compose objects with their attributes and all objects in the scene through their relationships to produce a single embedding vector for the text rich in semantic information. To the best of our knowledge, there has yet to be any previous dual-encoder work on explicitly capturing the object, attribute, and relation semantics through scene graphs for image-text matching. Our method is different from previous graph-based approaches in that we do not use external visual scene graph generator, which is prone to wrong prediction, and we carefully design a 2-step graph encoding approach trained with a contrastive loss to align at both the global image-text and local image-object level. Our network outperforms SOTA methods without the heavy cross-attention module.

3. Method

This section describes our Composition model for Object Relations and Attributes. We first describe the overall framework in Sec. 3.1, then present in Sec. 3.2 how we perform visual embedding on the input image, how we parse the text caption into a scene graph and extract text features for each node in the graph. In Sec. 3.3, we describe how we can embed this scene graph into the joint embedding space with the image using the graph attention network. Finally, training objectives are detailed in Sec. 3.4.

3.1. Overall Framework

We begin by describing the overall framework of CORA, which is illustrated in Fig. 2. The model consists of two encoders: a visual encoder $f^{\mathcal{V}}$ that takes in an input image $\mathbf{x}$ and produces the image embedding vector $v = f^{\mathcal{V}}(\mathbf{x}) \in \mathbb{R}^D$; and a text encoder $f^{\mathcal{T}}$ that takes in the text caption $\mathbf{y}$ and produces its embedding $t = f^{\mathcal{T}}(\mathbf{y}) \in \mathbb{R}^D$ in the joint D-dimensional embedding space. Instead of embedding the text caption directly, we first parse it into a scene graph using a parser ϕ^{SG}, then apply a graph attention network $f^{\mathcal{G}}$ to embed this scene graph. Our text embedding formulation therefore can be rewritten as $t = f^{\mathcal{G}}(\phi^{\mathrm{SG}}(\mathbf{y}))$.

The similarity score between the image and the text caption is defined as the cosine similarity between their embeddings v and t:

$$\mathrm{sim}(\mathbf{x}, \mathbf{y}) = \frac{v^{\mathrm{T}} t}{\|v\| \|t\|}. \tag{1}$$

The dual-encoder is efficient for image-text retrieval. In the context of image retrieval, all image embeddings can be computed and cached in advance. When a text query arrives, it only needs to be embedded with $f^{\mathcal{G}}(\phi^{\mathrm{SG}}(.))$, then a simple vector-matrix multiplication is sufficient to retrieve all nearest neighbor images of the query.

3.2. Feature Extraction

Visual feature extractor. Given an input image $\mathbf{x}$, we follow convention from prior work to use the pre-trained bottom-up detection model BUTD [1]. With this model, the top-36 most confident salient regions in $\mathbf{x}$ are detected, along with their visual features $\{x_k \in \mathbb{R}^{2048}\}_{k=1}^{N_{\mathcal{V}}}, N_{\mathcal{V}} = 36$. The detection model used here is a Faster R-CNN with ResNet-101 backbone [13], pre-trained on Visual Genome [22]. We also transform the region features with an FC layer so that they have the same dimensions as the joint embedding space: $x_k \in \mathbb{R}^D$. Furthermore, we also apply multi-head self-attention to contextualize the region features against one another. Then, in order to perform feature aggregation on this set to obtain a holistic representation for the input image $v = f^{\mathcal{V}}(\mathbf{x}) \in \mathbb{R}^D$, we implement $f^{\mathcal{V}}$ using GPO [4] which is a SOTA pooling operator for image-text matching. Essentially, GPO learns to generate the best pooling coefficient for every visual region, which is better than naively applying mean pooling over the visual feature set.

Scene graph parser. Formally, we implement a textual scene graph parser that can construct a graph $G = (V, E)$ given a text caption $\mathbf{y}$, where $V = O \cup A$ denotes the set of object nodes O and attribute nodes A, and $E = E_{\mathrm{OA}} \cup E_{\mathrm{OO}}$ represents the set of object-attribute edges E_{OA} and object-object relation edges E_{OO}. Example of a scene graph is illustrated in Fig. 1. We implement a scene graph parser based on [45, 53], using the syntactical dependency parser from the spaCy library [16]. We develop rules to extract object nouns (*e.g.*, *construction worker*), adjective and verb attributes (*e.g.*, *salmon-colored, sitting*), verb relations (*e.g.*, *person-jump over-fence, dog-wear-costume*), and preposition relations (*e.g.*, *flag-above-building*). Existing scene graph parsers [45, 53] are developed upon inferior language toolkits, thus often misdetect concepts (*e.g.*, those consisting of multiple word tokens are not detected). The implementation of our parser is made publicly available.

Semantic concept encoder. We denote the set of object nodes $O = \{o_i\}$, attribute nodes $A = \{a_i\}$, and object-object relation edges $E_{\mathrm{OO}} = \{r_{ij}\}$. These concepts are still in text format that need to be encoded into vector representation. As these concepts often consist of multiple word tokens (*e.g.*, *pair of shoes, jump over*), we use a text se-

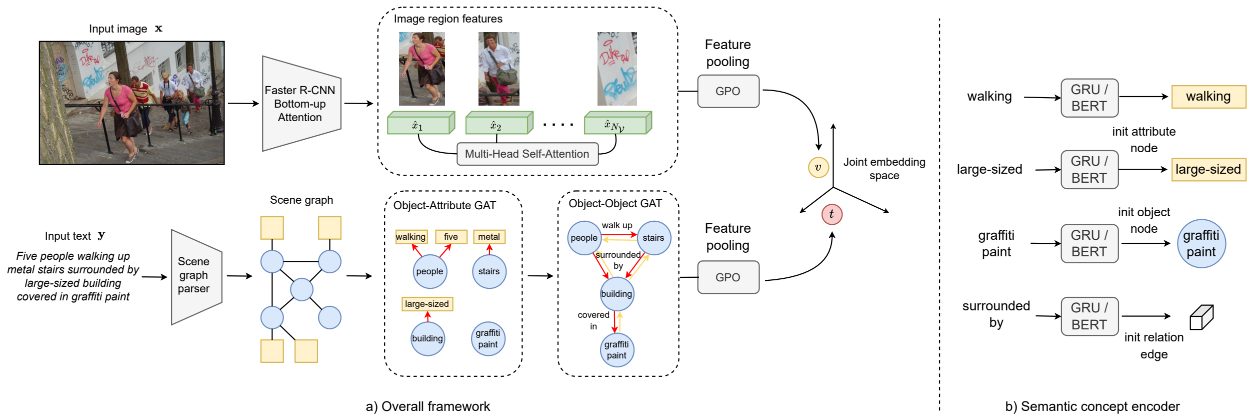

Figure 2. **Overview of CORA.** a) CORA consists of (1) an image encoder that detects and extracts the salient regions' features from the input image, contextualizes them through a multi-head self-attention, then aggregates them into a single image embedding through the GPO [4] pooling operator, (2) a text encoder that first parses the input text into a scene graph where all semantic information is readily organized, then two graph attention networks Object-Attribute GAT and Object-Object GAT are used to encode this graph into the same joint space with the image. The red arrow denotes the edge of the active role, while the yellow arrow is for the passive role in the relation (refer to Sec. 3.3.2). b) The semantic concept encoder that uses GRU or BERT to encode each semantic concept in the graph corresponding to the object, attribute nodes and relation edges.

quence model as a phrase encoder to encode all semantic concepts. To demonstrate the generalizability of our method across different language features, we implement this semantic concept encoder using Bi-GRU [7] and BERT [8]. For Bi-GRU, given an L-word semantic concept, we use the GloVe [37] word embedding of each word to obtain a sequence of L 300-dimensional vectors. Next, we employ a Bi-GRU and take the final hidden states as the representation for the concept $c \in \mathbb{R}^{300}$. For BERT, we use the average of the output hidden states of all tokens at the last layer to represent the concept $c \in \mathbb{R}^{768}$. For both types of features, we then use an FC layer to transform the concept embedding to have the same dimension D as the joint embedding space. These concept embeddings are used to initialize the node features for $\{o_i\}$ and $\{a_i\}$ and the edge features for $\{r_{ij}\}$ in the scene graph.

3.3. Scene Graph Embedding

After obtaining the graph structure from the parser and the initialized features for all nodes and edges in the graph, we continue to elaborate on our scene graph embedding method as follows. The core idea of our method is that the scene semantics should be composed at two levels in a bottom-up manner, where we use a separate graph attention network (GAT) [2, 48] for each level. At the bottom level, a GAT models the relations between an object and its associated attributes. At the top level, another GAT is used to model the relations between solely the objects, compose them together and produce the final scene embedding.

GAT Preliminaries. GAT is among the most popular graph neural network methods, with SOTA results in graph representation learning. We follow the implementation of GATv2 [2], which is an improved version of the original GAT [48]. We provide a brief description of GATv2 here. Given a directed graph $G = (V, E)$, containing nodes $V = \{1, ..., N\}$ and $E \subseteq V \times V$ where $(j, i) \in E$ denotes an edge from node j to i. For each node i, we also have its initial representation denoted as $h_i \in \mathbb{R}^d$. In a message passing step, to update features for node i, we first compute the importance value of neighbor node j w.r.t. i as following

$$e(h_i, h_j) = a^{\mathrm{T}}\mathrm{LeakyReLU}(W \cdot [h_i \| h_j]), \tag{2}$$

where $\|$ denotes vector concatenation, $W \in \mathbb{R}^{d \times 2d}, a \in \mathbb{R}^{d \times 1}$. Followed by softmax, normalized attention coefficients of all neighbors $j \in \mathcal{N}_i$ can be obtained: $\alpha_{i,j} = \mathrm{softmax}(e(h_i, h_j))$. Then, new representation h_i for node i is aggregated by

$$h_i' = \mathrm{ReLU}(\sum_{j \in \mathcal{N}_i} \alpha_{i,j} W h_j). \tag{3}$$

Formally, the output of one GAT layer on a graph G is

$$\{h_i'\} = \mathrm{GAT}(\{h_i\}, G). \tag{4}$$

3.3.1 Object-Attribute GAT

At the bottom level, we care about how the semantic representation of an object is modified by its connected attributes in the graph. These attributes are modifiers that alter the visual appearance of the object. Because an attribute of one object should in no way alter the appearance of another object, in this step, we apply GAT only on the subgraph $G_{\text{OA}} = (V, E_{\text{OA}})$ consists of only edges between the object and attribute nodes.

We denote $\{h_i\}_{i=1}^{|V|}, h_i \in \mathbb{R}^D$ as the initial representations for all nodes in the graph. These representations are initialized from the aforementioned semantic concept embedding step. We train a graph attention network, which we name $\text{GAT}_{\text{Obj-Att}}$ to perform message passing in graph G_{OA}. The updated representation of all nodes is therefore

$$\{h'_i\} = \text{GAT}_{\text{Obj-Att}}(\{h_i\}, G_{\text{OA}}). \tag{5}$$

At the output, we are only interested in the updated representation of the set of object nodes. Since these objects have been composed with their corresponding attributes, we name them as **entities** and denote them as $\{e_i\}_{i=1}^{|O|}$, which will be used in one of our proposed losses.

3.3.2 Object-Object Relation GAT

At the top level, after acquiring the entity embeddings $\{e_i\}_{i=1}^{|O|}$ for all object nodes, we continue to apply another GAT, which we name $\text{GAT}_{\text{Obj-Obj}}$ on the subgraph $G_{\text{OO}} = (O, E_{\text{OO}})$ consisting of only object nodes and edges between them. Because these object nodes are connected with object-object relation edges $\{r_{ij}\}$, our first step before applying GAT is to contextualize the entity embeddings with their corresponding edges.

Edge features. Consider a directed relation edge r_{ij}. In this relation, node i plays the subject (active) role while node j plays the object (passive) role. For example, in the relation *man-hold-cup*, *man* is the subject while *cup* is the object. To obtain the edge features for this relation, we concatenate its semantic encoding r_{ij} with the embedding of the entity that plays the passive role e_j as follows: $r'_{ij} = [r_{ij} \| e_j]$. While existing work [34] often concatenates r_{ij} with both the subject and object entity, in our work, we find that it is empirically better to characterize a relation with only the passive object entity. This is intuitively reasonable since the meaning of a relation such as *hold-cup, use-computer* does not depend on what kind of subject is involved.

Edge-contextualized entity features. Consider object node i, we define $\text{Active}(i) = \{j | r_{ij} \in E_{\text{OO}}\}$ consisting all nodes that node i has a subject (active) relation with. Vice-versa, we define $\text{Passive}(i) = \{j | r_{ji} \in E_{\text{OO}}\}$ which is all nodes that node i has an object (passive) relation. We contextualize the embedding of entity i with its edges as

$$e'_i = e_i + \frac{\sum_{j \in \text{Active}(i)} W_A r'_{ij}}{|\text{Active}(i)|} + \frac{\sum_{j \in \text{Passive}(i)} W_P r'_{ji}}{|\text{Passive}(i)|}, \tag{6}$$

where W_A and W_P are two learnable matrices mapping edge features to have the same dimension with entity embeddings.

Scene graph embedding. With $\{e'_i\}_{i=1}^{|O|}$ as the initial representation for all object nodes. We train a $\text{GAT}_{\text{Obj-Obj}}$ on graph G_{OO}. The updated representation for all nodes is

$$\{\hat{e}_i\} = \text{GAT}_{\text{Obj-Obj}}(\{e'_i\}, G_{\text{OO}}). \tag{7}$$

In order to pool the whole graph into one single embedding vector, we also use GPO [4] similar to our visual feature extraction step. We take the output representation that is pooled from GPO as the scene embedding t to represent the original input text caption in the joint embedding space.

3.4. Training Objectives

Let $B = \{(v_i, t_i, \{e_{ik}\}_{k=1}^{|O_i|})\}_{i=1}^{N}$ be the training batch of output image embedding v_i of the i-th image, output text embedding t_i of the i-th text caption from $\text{GAT}_{\text{Obj-Obj}}$, and set of output entity embeddings $\{e_{ik}\}_{k=1}^{|O_i|}$ of the i-th text caption from $\text{GAT}_{\text{Obj-Att}}$. It is reminded that these entities $\{e_{ik}\}$ are embeddings of the object nodes in the scene graph of t_i. We train our model CORA with the following losses. For brevity, we denote $s(v, t) = v^{\text{T}} t / (\|v\| \|t\|)$ to be the cosine similarity between v and t.

Triplet loss with hardest negatives. Following prior work in image-text retrieval [4, 10], we also adopt the hinge-based triplet loss with hardest negative mining,

$$\mathcal{L}_{\text{HARD}} = \sum_i \max_j [\alpha + s(v_i, t_j) - s(v_i, t_i)]_+ \tag{8}$$

$$+ \max_j [\alpha + s(v_j, t_i) - s(v_i, t_i)]_+. \tag{9}$$

Essentially, for every matching image-caption v_i and t_i in the training batch, this loss looks for the negative caption t_j that is closest to v_i, and the negative image v_j that is closest to t_i in the embedding space. t_j and v_j are the hardest negatives in the training batch and help provide a strong discriminative learning signal to the model.

Contrastive loss. As observed by previous work [4], the hardest triplet loss above results in unstable learning during early training epochs. We find that applying a contrastive loss that encourages the model to align the output representations of all matching image, text, and object entity together results in more stable training and better final results. Because the entity embeddings $\{e_{ik}\}_{k=1}^{|O_i|}$ are also involved in the equation here, our model CORA is also trained to perform image retrieval given an object entity (*e.g.*, image searching for *straw hat*). The loss is formulated as follows

$$\mathcal{L}_{\text{CON}} = - \sum_i \sum_u \log \frac{\exp(s(v_i, u))}{\sum_{u' \in \mathcal{N}_i} \exp(s(v_i, u'))} \tag{10}$$

$$- \sum_i \sum_u \log \frac{\exp(s(v_i, u))}{\sum_{v' \in \mathcal{N}_u} \exp(s(v', u))}, \tag{11}$$

where $u \in \{t_i\} \cup \{e_{ik}\}_{k=1}^{|O_i|}$ is the semantic embedding of either the text or an object entity corresponding to image i, $\mathcal{N}_i$ is the negative set of semantic concepts that do not correspond to image i, and similarly $\mathcal{N}_u$ is the negative set of images that do not contain semantic concept u.

Specificity loss. The contrastive loss above aligns the embeddings of image, text, and entity together in the joint space. In addition, we would like to impose some structure in this space such that the similarity between an image v_i and text t_i should be larger than between v_i and all entities $\{e_{ik}\}$. The reason is that a caption always depicts more semantic information than an entity alone, hence t_i should be more specific w.r.t. v_i and exhibits a larger similarity score. The loss takes the form of a hinge-based triplet loss

$$\mathcal{L}_{\text{SPEC}} = \sum_i \sum_k [\alpha + s(v_i, e_{ik}) - s(v_i, t_i)]_+. \tag{12}$$

Our overall loss is therefore a weighted sum of all losses:

$$\mathcal{L} = \mathcal{L}_{\text{HARD}} + \lambda_{\text{CON}} \mathcal{L}_{\text{CON}} + \lambda_{\text{SPEC}} \mathcal{L}_{\text{SPEC}}. \tag{13}$$

4. Experiments

We describe our experiments to validate the effectiveness of CORA. We describe the datasets in Sec 4.1 and analyze the results in Sec 4.2. To validate design choices, we present ablations in Sec 4.3. We refer to supplementary for implementation, qualitative results, and inference time analysis.

4.1. Dataset and Evaluation Metrics

Datasets. We perform experiments on two standard benchmarks, Flickr30K [41] and MS-COCO [27], on the image-to-text retrieval (I2T) and text-to-image retrieval (T2I) tasks. In both datasets, every image is annotated with five text descriptions. As in prior work [4], we follow the splits convention on both datasets. Flickr30K contains 31K images, of which 29K images are for training, 1K for validation, and 1K for testing. MS-COCO provides 123,287 images and is split into 113,287 images for training, 5000 images for validation, and 5000 images for testing.

Metrics. We report the commonly used Recall@K (R@K), where K $\in \{1, 5, 10\}$. This metric computes the percentage of queries where the correct match appears in the top-K retrievals. To summarize performance, we report RSUM which is the sum of R@K at all values of K $\in \{1, 5, 10\}$ on I2T and T2I tasks. For MS-COCO, by convention, the results are reported in two settings: 5K setting, and 1K setting where the results are averaged over five 1K data folds.

4.2. Quantitative Results

We summarize our results compared with SOTA methods on Flickr30K and MS-COCO in Tab. 1 and Tab. 2. The methods are denoted with whether they are cross-attention

Table 1. **Our framework achieves the best and second-best on the Flickr30K dataset with two different encoders.** Without the CA - "cross-attention", our method still has competitive results to other baselines. † denotes methods that use ensembling of multiple models, and we highlight **the highest** and <u>second-highest</u> RSUM for each section.

Method	Venue	CA	Image → Text			Text → Image			RSUM
			R@1	R@5	R@10	R@1	R@5	R@10	
Faster R-CNN + Bi-GRU									
SCAN† [23]	ECCV'18	✓	67.4	90.3	95.8	48.6	77.7	85.2	465.0
VSRN [24]	ICCV'19		71.3	90.6	96.0	54.7	81.8	88.2	482.6
SGM [51]	WACV'20	✓	71.8	91.7	95.5	53.5	79.6	86.5	478.6
GCN+DIST [25]	CVPR'20	✓	70.8	92.7	96.0	60.9	86.1	91.0	497.5
GSMN† [28]	CVPR'20	✓	76.4	94.3	97.3	57.4	82.3	89.0	496.8
CAAN [58]	CVPR'20	✓	70.1	91.6	97.2	52.8	79.0	87.9	478.6
VSE_{∞} [4]	CVPR'21		76.5	94.2	97.7	56.4	83.4	89.9	498.1
VSRN_{++} [24]	PAMI'22		79.2	94.6	97.5	60.6	85.6	91.4	508.9
SGARF† [9]	AAAI'23	✓	77.8	94.1	97.4	58.5	83.0	88.8	499.6
CODER [49]	ECCV'22	✓	79.4	94.9	97.7	59.0	85.2	91.0	507.2
MV-VSE† [26]	IJCAI'22		79.0	94.9	97.7	59.1	84.6	90.6	505.8
GraDual† [30]	WACV'23	✓	78.3	96.0	98.0	60.4	86.7	92.0	511.4
CHAN [35]	CVPR'23	✓	79.7	94.5	97.3	60.2	85.3	90.7	507.7
NAAF† [57]	CVPR'23	✓	81.9	96.1	98.3	61.0	85.3	90.6	513.2
SDE† [20]	CVPR'23		80.9	94.7	97.6	59.4	85.6	91.1	509.3
HREM† [12]	CVPR'23		81.4	96.5	98.5	60.9	85.6	91.3	514.2
Ours			82.2	95.6	97.7	61.8	86.5	92.0	<u>515.8</u>
Ours†			82.3	96.1	98.0	63.0	87.4	92.8	**519.6**
Faster R-CNN + BERT									
VSE_{∞} [4]	CVPR'21		81.7	95.4	97.6	61.4	85.9	91.5	513.5
CODER [49]	ECCV'22	✓	83.2	96.5	98.0	63.1	87.1	93.0	520.9
MV-VSE† [26]	IJCAI'22		82.1	95.8	97.9	63.1	86.7	92.3	517.5
CHAN [35]	CVPR'23	✓	80.6	96.1	97.8	63.9	87.5	92.6	518.5
HREM† [12]	CVPR'23		84.0	96.1	98.6	64.4	88.0	93.1	**524.2**
Ours			83.7	96.6	98.3	62.3	87.1	92.6	520.1
Ours†			83.4	95.9	98.6	64.1	88.1	93.1	<u>523.3</u>

or dual-encoder approaches, and are divided into groups depending on the textual backbone used (Bi-GRU vs. BERT). Following previous work [12, 20, 57], we also report the ensemble results which are obtained by averaging the similarities from two checkpoints trained with different seeds.

Comparisons with state-of-the-art methods. When using Bi-GRU as the semantic concept encoder, our method CORA outperforms all state-of-the-art methods by an impressive margin. CORA achieves $+5.4$ RSUM absolute improvement over HREM on Flickr30K, and $+13.7$ RSUM over NAAF on MS-COCO 5K. Note that NAAF is among the SOTA cross-attention methods (CHAN, GraDual, CODER, SGARF) which are more computationally expensive but having more learning capacity advantage over dual encoders, however CORA is still able to surpass them. The non-ensemble version of CORA also outperforms all non-ensemble methods while even exceeding the ensemble ones (SDE, GraDual).

When using BERT for encoding semantic concepts, CORA achieves second best RSUM score on Flickr30K and MS-COCO and is only inferior to the recent SOTA HREM. HREM is also a dual encoder, but is trained with a cross-modality mechanism (which is later discarded at inference) to enhance each modality embedding for matching. The same idea of HREM can be applied to CORA to boost the performance even further, but is out of the scope of our work. Switching from Bi-GRU to using BERT, our method

Table 2. **Our method yields competitive results on the MS-COCO dataset.** Our performance is competitive in all test schema with previous works, especially on the simple Bi-GRU architecture. † denotes methods that use ensembling of multiple models. **Bold** and <u>underline</u> highlight the best and second-best performance.

Method	Venue	Cross-Attention	MS-COCO 5-fold 1K Test							MS-COCO 5K Test						
			Image → Text			Text → Image			RSUM	Image → Text			Text → Image			RSUM
			R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10	
Faster R-CNN + Bi-GRU																
SCAN† [23]	ECCV'18	✓	72.7	94.8	98.4	58.8	88.4	94.8	507.9	50.4	82.2	90.0	38.6	69.3	80.4	410.9
VSRN [24]	ICCV'19		76.2	94.8	98.2	62.8	89.7	95.1	516.8	53.0	81.1	89.4	40.5	70.6	81.1	415.7
SGM [51]	WACV'20	✓	73.4	93.8	97.8	57.5	87.3	94.3	504.1	50.0	79.3	87.9	35.3	64.9	76.5	393.9
CAAN [58]	CVPR'20	✓	75.5	95.4	98.5	61.3	89.7	95.2	515.6	52.5	83.3	90.9	41.2	70.3	82.9	421.1
VSE_∞ [4]	CVPR'21		78.5	96.0	98.7	61.7	90.3	95.6	520.8	56.6	83.6	91.4	39.3	69.9	81.1	421.9
SGARF† [9]	AAAI'23	✓	79.6	96.2	98.5	63.2	90.7	96.1	524.3	57.8	-	91.6	41.9	-	81.3	-
CODER [49]	ECCV'22	✓	78.9	95.6	98.6	62.5	90.3	95.7	521.6	58.5	84.3	91.5	40.9	70.8	81.4	427.4
MV-VSE† [26]	IJCAI'22		78.7	95.7	98.7	62.7	90.4	95.7	521.9	56.7	84.1	91.4	40.3	70.6	81.6	424.6
GraDual† [30]	WACV'23	✓	77.0	96.4	98.6	65.3	91.9	96.4	525.6	-	-	-	-	-	-	-
CHAN [35]	CVPR'23	✓	79.7	96.7	98.7	63.8	90.4	95.8	525.1	60.2	85.9	92.4	41.7	71.5	81.7	433.4
NAAF† [57]	CVPR'23	✓	80.5	96.5	98.8	64.1	90.7	96.5	527.2	58.9	85.2	92.0	42.5	70.9	81.4	430.9
SDE† [20]	CVPR'23		80.6	96.3	98.8	64.7	91.4	96.2	528.0	60.4	86.2	92.4	42.6	73.1	83.1	437.8
HREM† [12]	CVPR'23		81.2	96.5	98.9	63.7	90.7	96.0	527.0	60.6	86.4	92.5	41.3	71.9	82.4	435.1
Ours			80.9	96.3	98.8	64.9	91.3	96.4	<u>528.5</u>	61.4	85.6	92.4	43.3	72.9	83.3	<u>438.9</u>
Ours†			81.7	96.7	99.0	66.0	92.0	96.7	**532.1**	63.0	86.8	92.7	44.2	73.9	84.0	**444.6**
Faster R-CNN + BERT																
VSE_∞ [4]	CVPR'21		79.7	96.4	98.9	64.8	91.4	96.3	527.5	58.3	85.3	92.3	42.4	72.7	83.2	434.2
CODER [49]	ECCV'22	✓	82.1	96.6	98.8	65.5	91.5	96.2	530.7	62.6	86.6	93.1	42.5	73.1	83.3	441.2
MV-VSE† [26]	IJCAI'22		80.4	96.6	99.0	64.9	91.2	96.0	528.1	59.1	86.3	92.5	42.5	72.8	83.1	436.3
CHAN [35]	CVPR'23	✓	81.4	96.9	98.9	66.5	92.1	96.7	532.5	59.8	87.2	93.3	44.9	74.5	84.2	443.9
HREM† [12]	CVPR'23		82.9	96.9	99.0	67.1	92.0	96.6	<u>534.5</u>	64.0	88.5	93.7	45.4	75.1	84.3	**451.0**
Ours			82.4	96.8	98.8	66.2	91.9	96.6	532.7	62.4	86.8	92.6	44.2	73.6	83.9	443.6
Ours†			82.8	97.3	99.0	67.3	92.4	96.9	**535.6**	64.3	87.5	93.6	45.4	74.7	84.6	<u>450.1</u>

enjoys a smaller RSUM improvement compared to other work (+5.5 RSUM for CORA vs. +15.9 for HREM and +10.5 for CHAN on MS-COCO 5K). This is due to BERT being more suitable for encoding long text while CORA is using BERT to encode short phrases (*e.g.*, *construction worker, sitting*). A text encoder that is more suitable for encoding short phrases is therefore more desirable and we leave this as future work. Similarly, when using BERT, CORA surpasses the performance of SOTA cross-attention methods CODER and CHAN. This shows the ability to generalize across different feature extractors of CORA.

Comparisons with scene graph-based approaches. CORA outperforms all scene-graph based methods, which includes SGM [51], GCN+DIST [25], GSMN [28], and GraDual [30]. These methods all employ an additional off-the-shelf visual scene graph generator [56] (except GSMN) to produce a scene graph for an input image, and use cross-attention to exchange information between the textual and the visual graph, but still achieve inferior results to CORA. This further shows that CORA is very effective at encoding scene graphs. These methods all embed a whole scene graph holistically (unlike CORA which separates it into two object-attribute and object-object steps), are trained with a holistic loss to align image and text (unlike CORA that has loss terms to additionally align image and local object entity), and use visual scene graph generator [56] which is susceptible to making wrong predictions and has been reported to misdetect rare object relationships [47].

4.3. Ablation Studies

We perform a series of ablation studies to explore the impact of our graph attention network design and how the losses affect the final performance. All experiments in this section use Bi-GRU for the semantic encoder and are performed on the Flickr30K dataset. The results are reported in Tab. 3.

Number of layers in GAT. The experiments show that having 1 layer for $\text{GAT}_{\text{Obj-Att}}$ and 2 layers for $\text{GAT}_{\text{Obj-Obj}}$ achieves the best accuracy. For the object-attribute graph, 1 layer is sufficient to propagate the attribute information to their corresponding object node. For the object-object relation graph, using only 1 layer is not enough to aggregate information from the whole graph, while increasing to 3 layers starts to give diminishing returns.

Graph structure. We study whether our 2-step scene graph encoding step is beneficial to the final performance. We refer to *Joint* as the model that uses a single GAT on the whole graph at once, *FC* as the variant that uses fully connected graph instead of the structure parsed from the scene graph parser, and *Obj-Att & Obj-Obj* as our proposed 2-step scene graph encoding model. Note that having a separate object-attribute encoding step allows our model to produce individual entity embeddings (see Sec. 3.3.1) that are later used in the contrastive loss to align an image with each of

Table 3. **Ablation studies** for the number of layers in GAT, the graph structure whether encoding scene graph jointly or in 2 separate steps is beneficial, and the impact of losses. **Bold** and underline highlight the best and second-best performance.

			Image → Text			Text → Image			RSUM
			R@1	R@5	R@10	R@1	R@5	R@10	
Number of GAT layers									
$n_{\text{Obj-Att}}$	$n_{\text{Obj-Obj}}$								
1	1		79.8	95.3	97.1	60.5	85.4	91.0	509.1
1	2		82.2	95.6	97.7	61.8	86.5	92.0	**515.8**
1	3		81.6	95.8	97.6	61.3	86.2	91.9	514.4
2	1		79.9	95.1	97.5	60.8	85.5	91.3	510.1
2	2		81.7	95.3	96.8	61.6	87.0	92.1	514.5
2	3		80.2	95.3	96.5	60.9	85.9	91.8	510.6
Graph structure									
Joint	Obj-Att&Obj-Obj	FC							
✓			78.9	93.5	96.3	59.6	85.8	90.4	504.8
✓		✓	77.6	93.0	96.0	59.4	85.1	90.5	501.6
	✓		82.2	95.6	97.7	61.8	86.5	92.0	**515.8**
	✓	✓	81.2	95.1	96.9	61.3	86.9	91.8	513.2
Losses									
$\mathcal{L}_{\text{HARD}}$	$\mathcal{L}_{\text{CON}}$	$\mathcal{L}_{\text{SPEC}}$							
✓			71.6	92.2	95.9	53.9	83.4	89.9	486.9
	✓		75.0	94.0	96.4	57.9	84.4	90.1	497.8
✓	✓		78.9	95.2	97.3	59.1	85.8	91.0	507.3
✓	✓	✓	82.2	95.6	97.7	61.8	86.5	92.0	**515.8**

its objects independently. This is not possible with the *Joint* model because after the GAT step, all object nodes are already contextualized. The results indeed show that our proposed 2-step encoding step is superior, and using the graph structure parsed from the scene graph parser is better than connecting all nodes together in a fully connected manner.

Losses. Similar to prior work, we first explore using the triplet loss with hardest negatives $\mathcal{L}_{\text{HARD}}$ and find that using it alone is insufficient, since it can only be used on the image-text level and not at the image-object entity level. Using solely the contrastive loss $\mathcal{L}_{\text{CON}}$ gives an accuracy boost, but is still far from optimal. By combining $\mathcal{L}_{\text{HARD}}$ and $\mathcal{L}_{\text{CON}}$, we achieve a significant accuracy improvement. Since the contrastive loss treats the graph embedding and entity embedding equally, by imposing the structure that a whole text caption should depict more information than an entity alone through the loss $\mathcal{L}_{\text{SPEC}}$, CORA achieves SOTA results.

4.4. Qualitative Results & Text-to-Entity Retrieval

Fig. 3 illustrates how CORA can perform image-to-text and image-to-object entity retrieval. More qualitative results can be found in the supplementary.

In addition, we also experiment with using the image-entity score for re-ranking the image-text matching. We employ the idea that if an object entity in the text is not closely matched with the image, then the image-text matching score should be lower. Formally, we utilize the following formula

$$\hat{s}(v_i, t_i) = \beta \cdot s(v_i, t_i) + (1 - \beta) \cdot \min_k s(v_i, e_{ik}), \quad (14)$$

Table 4. **Reranking results on MS-COCO 5K** after ensembling with image-entity score.

	Image → Text			Text → Image			RSUM
	R@1	R@5	R@10	R@1	R@5	R@10	
Faster R-CNN + Bi-GRU							
CORA	63.0	86.8	92.7	44.2	73.9	84.0	444.6
CORA + reranking	63.2	86.8	92.7	44.3	74.1	84.0	445.1
Faster R-CNN + BERT							
CORA	64.3	87.5	93.6	45.4	74.7	84.6	450.1
CORA + reranking	64.2	87.6	93.8	45.5	74.8	84.7	450.6

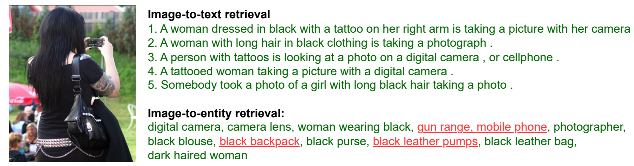

Figure 3. **Qualitative result** demonstrates how CORA can perform image-to-text and image-to-entity retrieval. Green denotes correct retrieval while red denotes incorrect ones.

where $\beta \in [0, 1]$ is a hyperparameter that we select on the validation set. The results on MS-COCO are reported in Tab. 4, where we achieve a slight accuracy improvement with this simple strategy. One potential future direction is to explore smarter mechanism to combine the image-text and image-object entity embedding alignment score.

5. Conclusion

Limitation. Despite achieving new SOTA results, CORA still faces some limitations. CORA is strongly dependent on the scene graph quality from the parser. If the parser fails to extract a scene graph from the input text, CORA also fails to encode the text. This happens seldomly in MS-COCO, where there are captions that are just exclamatory sentences uttered by the annotator, *e.g*., "I am so happy to see this view", "There are so many things to see here." On the other hand, text sequence model is still able to capture the nuances of these text descriptions.

In this paper, we propose a dual-encoder model CORA for image-text matching that is based on scene graph. CORA achieves new SOTA results, outperforms all SOTA computationally expensive cross-attention methods. We show a promising future direction for image-text matching that, by representing a caption as a scene graph of object and attribute nodes connected by relation edges, we can utilize the strong relational inductive bias of graph neural network to compose objects, relations, and their attributes into a scene graph embedding that is effective for image-text retrieval.

Acknowledgements. This project was partially funded by NSF CAREER Award (#2238769) to AS.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 3, 1

[2] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 2, 4

[3] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023. 2

[4] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *CVPR*, 2021. 2, 3, 4, 5, 6, 7, 1

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*. Springer, 2020. 2

[6] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *CVPR*, 2021. 1, 2

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2, 4

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 4, 1

[9] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *AAAI*, 2021. 2, 6, 7, 1, 3

[10] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017. 2, 5

[11] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *NeurIPS*, 26, 2013. 2

[12] Zheren Fu, Zhendong Mao, Yan Song, and Yongdong Zhang. Learning semantic relationship among instances for image-text matching. In *CVPR*, 2023. 2, 6, 7, 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[14] John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. 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 4129–4138, 2019. 2

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[16] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. 3

[17] Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. *arXiv preprint arXiv:1909.00964*, 2019. 2

[18] Chuong Huynh, Yuqian Zhou, Zhe Lin, Connelly Barnes, Eli Shechtman, Sohrab Amirghodsi, and Abhinav Shrivastava. Simpson: Simplifying photo cleanup with single-click distracting object segmentation network. In *CVPR*, 2023. 1

[19] Chuong Huynh, Seoung Wug Oh, , Abhinav Shrivastava, and Joon-Young Lee. Maggie: Masked guided gradual human instance matting. In *CVPR*, 2024. 1

[20] Dongwon Kim, Namyup Kim, and Suha Kwak. Improving cross-modal retrieval with set of diverse embeddings. In *CVPR*, 2023. 2, 6, 7

[21] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 2

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 2, 3

[23] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *ECCV*, 2018. 2, 6, 7, 1, 3

[24] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *ICCV*, pages 4654–4662, 2019. 2, 6, 7, 3

[25] Yongzhi Li, Duo Zhang, and Yadong Mu. Visual-semantic matching by exploring high-order attention and distraction. In *CVPR*, 2020. 2, 3, 6, 7

[26] Zheng Li, Caili Guo, Zerun Feng, Jenq-Neng Hwang, and Xijun Xue. Multi-view visual semantic embedding. In *IJCAI*, page 7, 2022. 2, 6, 7, 3

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[28] Chunxiao Liu, Zhendong Mao, Tianzhu Zhang, Hongtao Xie, Bin Wang, and Yongdong Zhang. Graph structured network for image-text matching. In *CVPR*, 2020. 2, 3, 6, 7

[29] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 1, 2, 3

[30] Siqu Long, Soyeon Caren Han, Xiaojun Wan, and Josiah Poon. Gradual: Graph-based dual-modal representation for image-text matching. In *WACV*, 2022. 2, 3, 6, 7

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[32] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 32, 2019. 2

[33] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, pages 181–196, 2018. 2

[34] Kien Nguyen, Subarna Tripathi, Bang Du, Tanaya Guha, and Truong Q Nguyen. In defense of scene graphs for image captioning. In *ICCV*, 2021. 5

[35] Zhengxin Pan, Fangyu Wu, and Bailing Zhang. Fine-grained image-text matching by cross-modal hard aligning network. In *CVPR*, 2023. 2, 6, 7

[36] Genevieve Patterson and James Hays. Coco attributes: Attributes for people, animals, and objects. In *ECCV*, pages 85–100. Springer, 2016. 2

[37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014. 4, 2

[38] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, pages 13018–13028, 2021. 2

[39] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Improving closed and open-vocabulary attribute prediction using transformers. In *ECCV*, 2022. 2

[40] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. In *CVPR*, 2024. 2

[41] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 6

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 1

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[44] Nirat Saini, Khoi Pham, and Abhinav Shrivastava. Disentangling visual embeddings for attributes and objects. In *CVPR*, pages 13658–13667, 2022. 2

[45] 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 *Workshop on Vision and Language (VL15)*, Lisbon, Portugal, 2015. Association for Computational Linguistics. 3

[46] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *CVPR*, pages 1979–1988, 2019. 1

[47] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 7

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 2, 4

[49] Haoran Wang, Dongliang He, Wenhao Wu, Boyang Xia, Min Yang, Fu Li, Yunlong Yu, Zhong Ji, Errui Ding, and Jingdong Wang. Coder: Coupled diversity-sensitive momentum contrastive learning for image-text retrieval. In *ECCV*, 2022. 6, 7, 2

[50] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 2

[51] Sijin Wang, Ruiping Wang, Ziwei Yao, Shiguang Shan, and Xilin Chen. Cross-modal scene graph matching for relationship-aware image-text retrieval. In *WACV*, 2020. 2, 3, 6, 7

[52] Keyu Wen, Xiaodong Gu, and Qingrong Cheng. Learning dual semantic relations with graph attention for image-text matching. *IEEE transactions on circuits and systems for video technology*, 31(7):2866–2879, 2020. 2

[53] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *CVPR*, 2019. 3

[54] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. Learning fragment self-attention embeddings for image-text matching. In *ACM MM*, 2019. 2

[55] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 2, 3

[56] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 2, 3, 7

[57] Kun Zhang, Zhendong Mao, Quan Wang, and Yongdong Zhang. Negative-aware attention framework for image-text matching. In *CVPR*, 2022. 2, 6, 7, 1

[58] Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z Li. Context-aware attention network for image-text retrieval. In *CVPR*, 2020. 2, 6, 7, 3