Unified Visual-Semantic Embeddings: Bridging Vision and Language with Structured Meaning Representations

Hao Wu1,3,4,6,*,†, Jiayuan Mao5,6,*,†, Yufeng Zhang2,6,†, Yuning Jiang6, Lei Li6, Weiwei Sun1,3,4, Wei-Ying Ma6

1School of Computer Science, 2School of Economics, Fudan University
3Systems and Shanghai Key Laboratory of Data Science, Fudan University
4Shanghai Institute of Intelligent Electronics & Systems
5ITCS, Institute for Interdisciplinary Information Sciences, Tsinghua University, 6Bytedance AI Lab
{wuhao5688, zhyf, wwsun}@fudan.edu.cn, mjyl14@mails.tsinghua.edu.cn,
{jiangyuning, lileilab, maweiying}@bytedance.com

Abstract

We propose the Unified Visual-Semantic Embeddings (Unified VSE) for learning a joint space of visual representation and textual semantics. The model unifies the embeddings of concepts at different levels: objects, attributes, relations, and full scenes. We view the sentential semantics as a combination of different semantic components such as objects and relations; their embeddings are aligned with different image regions. A contrastive learning approach is proposed for the effective learning of this fine-grained alignment from only image-caption pairs. We also present a simple yet effective approach that enforces the coverage of caption embeddings on the semantic components that appear in the sentence. We demonstrate that the Unified VSE outperforms baselines on cross-modal retrieval tasks; the enforcement of the semantic coverage improves the model’s robustness in defending text-domain adversarial attacks. Moreover, our model empowers the use of visual cues to accurately resolve word dependencies in novel sentences.

1. Introduction

We study the problem of establishing accurate and generalizable alignments between visual concepts and textual semantics efficiently, based upon rich but few, paired but noisy, or even biased visual-textual inputs (e.g., image-caption pairs). Consider the image-caption pair A shown in Fig. 1: “A white clock on the wall is above a wooden table.” The alignments are formed at multiple levels: This short sentence can be decomposed into a rich set of semantic components [3]:

*indicates equal contribution.
†Work was done when HW, JM and YZ were intern researchers at the Bytedance AI Lab.
cross-modal retrieval performance in the training set (e.g., in [49, 30, 40]). As reported by [40], because of the inevitable bias in the dataset (e.g., two objects may co-occur with each other in most cases, see the table and the wall in Fig. 1 as an example), the learned sentence encoders usually pay attention to only part of the sentence. As a result, they are vulnerable to text-domain adversarial attacks: Adversarial captions constructed from original captions by adding small perturbations (e.g., by changing wall to be shelf) can easily fool the model [40, 39].

We resolve the aforementioned challenges by a natural combination of two ideas: cross-situational learning and the enforcement of semantic coverage that regularizes the encoder. Cross-situational learning, or learning from contrastive examples [12], uses contrastive examples in the dataset to resolve the referential ambiguity of objects: Looking at both Pair A and B in Fig. 1, we know that Clock should refer to an object that occurs only in scene A but not B. Meanwhile, to alleviate the biases of datasets such as object co-occurrence, we present an effective approach that enforces the semantic coverage: The meaning of a caption is a composition of all semantic components in the sentence [3]. Reflectively, the embedding of a caption should have a coverage of all semantic components, while changing any of them should affect the global caption embedding.

Conceptually and empirically, Unified VSE makes the following three contributions.

First, the explicit factorization of the visual-semantic embedding space enables us to build a fine-grained correspondence between visual and textual data, which further benefits a set of downstream visual-textual tasks. We achieve this through a contrastive example mining technique that uniformly applies to different semantic components, in contrast to the sentence or image-level contrastive samples used by existing visual-semantic learning [49, 30, 11]. Unified VSE consistently outperforms pre-existing approaches on a diverse set of retrieval-based tasks.

Second, we propose a caption encoder that ensures a coverage of all semantic components appeared in the sentence. We show that this regularization helps our model to learn a robust semantic representation for captions. It effectively defends adversarial attacks on the text domain.

Furthermore, we show how our learned embeddings can provide visual cues to assist the parsing of novel sentences, including determining content word dependencies and labelling semantic roles for certain verbs. It ends up that our model can build reliable connections between vision and language using given semantic cues and in return, bootstrap the acquisition of language.

2. Related work

Visual semantic embedding. Visual semantic embedding [13] is a common technique for learning a joint representation of vision and language. The embedding space empowers a set of cross-modal tasks such as image captioning [43, 48, 8] and visual question answering [4, 47].

A fundamental technique proposed in [13] for aligning two modalities is to use the pairwise ranking to learn a distance metric from similar and dissimilar cross-modal pairs [44, 35, 23, 9, 28, 24]. As a representative, VSE++ [11] uses the online hard negative mining (OHEM) strategy [41] for data sampling and shows the performance gain. VSE-C [40], based on VSE++, enhances the robustness of the learned visual-semantic embeddings by incorporating rule-generated textual adversarial samples as hard negatives during training. In this paper, we present a contrastive learning approach based on semantic components.

There are multiple VSE approaches that also use linguistically-aware techniques for the sentence encoding and learning. Hierarchical multimodal LSTM (HM-LSTM) [33] and [46], as two examples, both leverage the constituency parsing tree. Multimodal-CNN (m-CNN) [30] and CSE [49] apply convolutional neural networks to the caption and extract the a hierarchical representation of sentences. Our model differs with them in two aspects. First, Unified VSE is built upon a factorized semantic space instead of the syntactic knowledge. Second, we employ a contrastive example mining approach that uniformly applies to different semantic components. It substantially improves the learned embeddings, while the related works use only sentence-level contrastive examples.

The learning of object-level alignment in unified VSE is also related to [19, 21, 36], where the authors incorporate pre-trained object detectors for the semantic alignment. [10] propose a selective pooling technique for the aggregation of object features. Compared with them, Unified VSE presents a more general approach that embeds concepts of different levels, while still requiring no extra supervisions.

Structured representation for vision and language. We connect visual and textual representations in a structured embedding space. The design of its structure is partially motivated by the papers on relational visual representations (scene graphs) [29, 18, 17], where a scene is represented by a set of objects and their relations. Compared with them, our
model does not rely on labelled graphs during training. Researchers have designed various types of representations [5, 32] as well as different models [26, 50] for translating natural language sentences into structured representations. In this paper, we present how the usage of such semantic parsing into visual-semantic embedding facilitates the learning of the embedding space. Moreover, we present how the learned VSE can, in return, helps the parser to resolve parsing ambiguities using visual cues.

3. Unified Visual-Semantic Embeddings

We now describe the overall architecture and training paradigm for the proposed Unified Visual-Semantic Embeddings. Shown in Fig. 3, given an image-caption pair, we first parse the caption into a structured meaning representation, composed by a set of semantic components: object nouns, prenominal modifiers, and relational dependencies. We encode different types of semantic components with type-specific encoders. A caption encoder combines the embedding of the semantic components into a caption semantic embedding. Jointly, we encode images with a convolutional neural network (CNN) into the same, unified VSE space. The distance between the image embedding and the sentential embedding measures the semantic similarity between the image and the caption.

We employ a multi-task learning approach for the joint learning of embeddings for semantic components (as the “basis” of the VSE space) as well as the caption encoder (as the combiner of semantic components).

3.1. Visual-Semantic Embedding: A Revisit

We begin the section with an introduction to the two-stream VSE approach. It jointly learns the embedding spaces of two modalities: vision and language, and aligns them using parallel image-text pairs (e.g., image and captions from the MS-COCO dataset [27]).

Let \( \mathbf{v} \in \mathbb{R}^d \) be the representation of the image and \( \mathbf{u} \in \mathbb{R}^d \) be the representation of a caption matching this image, both encoded by neural modules. To archive the alignment, a bidirectional margin-based ranking loss has been widely applied [11, 49, 15]. Formally, for an image (caption) embedding \( \mathbf{v} \) (\( \mathbf{u} \)), denote the embedding of its matched caption (image) as \( \mathbf{u}^+ \) (\( \mathbf{v}^+ \)). A negative (unmatched) caption (image) is sampled whose embedding is denoted as \( \mathbf{u}^- \) (\( \mathbf{v}^- \)). We define the bidirectional ranking loss \( \ell_{\text{sent}} \) between captions and images as:

\[
\ell_{\text{sent}} = \sum_{\mathbf{u}} F_{\mathbf{v}} - (|\delta + s(\mathbf{u}, \mathbf{v}^-) - s(\mathbf{u}, \mathbf{v}^+)|_+) \\
+ \sum_{\mathbf{v}} F_{\mathbf{u}} - (|\delta + s(\mathbf{u}^-, \mathbf{v}) - s(\mathbf{u}^+, \mathbf{v})|_+) \tag{1}
\]

where \( \delta \) is a predefined margin, \(|x|_+ = \max(x, 0)\) is the traditional ranking loss and \( F_x(\cdot) = \max_x(\cdot) \) denotes the hard negative mining strategy [11, 41]. \( s(\cdot, \cdot) \) is a similarity function between two embeddings and is usually implemented as cosine similarity [11, 40, 49].

3.2. Semantic Encodings

The encoding of a caption is made up of three steps. As an example, consider the caption shown in Fig. 3, “A white clock on the wall is above a wooden table”. 1) We extract a structured meaning representation as a collection of three types of semantic components: object (clock, wall, table), attribute-object dependencies (white clock, wooden table) and relational dependencies (clock above table, clock on wall). 2) We encode each component as well as the full sentence with type-specific encoders into the unified VSE space. 3) We compose the embedding of the caption by combining semantic components.

Semantic parsing. We implement a semantic parser \(^1\) of image captions based on [38]. Given the input sentence, the parser first performs a syntactic dependency parsing. A set of rules is applied to the dependency tree and extracts object entities appeared in the sentence, adjectives that modify the

\(^1\)https://github.com/vacancy/SceneGraphParser
object nouns, subjects/objects of the verbs and prepositional phrases. For simplicity, we consider only single-word nouns for objects and single-word adjectives for object attributes.

**Encoding objects and attributes.** We use an unified object encoder $\phi$ for nouns and adjective-noun pairs. For each word $w$ in the vocabulary, we initialize a basic semantic embedding $w^{(\text{basic})} \in \mathbb{R}^{d_{\text{basic}}}$ and a modifier semantic embedding $w^{(\text{modif})} \in \mathbb{R}^{d_{\text{modif}}}$.

For a single noun word $w_n$ (e.g., *clock*), we define its embedding $w_n$ as $w_n^{(\text{basic})} \oplus w_n^{(\text{modif})}$, where $\oplus$ means the concatenation of vectors. For an (adjective, noun) pair $(w_a, w_n)$ (e.g., *white, clock*), its embedding $w_{a,n}$ is defined as $w_n^{(\text{basic})} \oplus w_n^{(\text{modif})} = w_n^{(\text{basic})} \oplus w_n^{(\text{modif})}$, which takes the embeddings of word-level semantics encoded by $\psi$ and modif. $w_n^{(\text{modif})}$ encodes the attribute information. In implementation, the basic semantic embedding is initialized from GloVe [34]. The modifier semantic embeddings (both $w_n^{(\text{modif})}$ and $w_{a,n}^{(\text{modif})}$) are randomly initialized and jointly learned. $w_n^{(\text{modif})}$ can be regarded as an intrinsic modifier for each noun.

To fuse the embeddings of basic and modifier semantics, we employ a gated fusion function:

$$
\phi(w_n) = \text{Norm}(\sigma(W_1 w_n + b_1)) \tanh(W_2 w_n + b_2)),
$$

$$
\phi(w_{a,n}) = \text{Norm}(\sigma(W_1 w_{a,n} + b_1)) \tanh(W_2 w_{a,n} + b_2)).
$$

Throughout the text, $\sigma$ denotes the sigmoid function: $\sigma(x) = 1/(1 + \exp(-x))$, and $\text{Norm}$ denotes the L2 normalization, i.e., $\text{Norm}(w) = w/\|w\|_2$. One may interpret $\phi$ as a GRU cell [7] taking no historical state.

**Encoding relations and full sentence.** Since relations and sentences are the composed based on objects, we encode them with a neural combiner $\psi$, which takes the embeddings of word-level semantics encoded by $\phi$ as input. In practice, we implement $\psi$ as an uni-directional GRU [7], and pick the L2-normalized last state as the output.

To obtain a visual-semantic embedding for a relational triple $\{w_s, w_r, w_o\}$ (e.g., *clock, above, table*), we first extract the word embeddings for the subject, relational word and the object using $\phi$. We then feed the encoded word embeddings in the same order into $\psi$ and takes the L2-normalized last state of the GRU cell. Mathematically, $u_{\text{rel}} = \psi(w_s, w_r, w_o) = \psi(\{\phi(w_s), \phi(w_r), \phi(w_o)\})$.

The embedding of a sentence $u_{\text{sent}}$ is computed over the word sequence $w_1, w_2, \ldots, w_k$ of the caption:

$$
u_{\text{sent}} = \psi(\{\phi(w_1), \phi(w_2), \ldots, \phi(w_k)\}),$$

where for any word $x$, $\phi(w_x) = \phi(w_x^{(\text{basic})} \oplus w_x^{(\text{modif})})$

Note that we share the weights of the encoders $\psi$ and $\phi$ among the encoding processes of all semantic levels. This allows our encoders of various types of components to bootstrap the learning of each other.

**Combining all of the components.** A straight-forward implementation of the caption encoder is to directly use the sentence embedding $u_{\text{sent}}$, as it has already combined the semantics of components in a contextually-weighted manner [25]. However, it has been revealed in [40] that such combination is vulnerable to adversarial attacks: Because of the biases in the dataset, the combiner $\psi$ usually focuses on only a small set of semantic components appeared in the caption.

We alleviate such biases by enforcing the coverage of the semantic components appeared in the sentence. Specifically, to form the caption embedding $u_{\text{cap}}$, the sentence embedding $u_{\text{sent}}$ is combined with an explicit bag-of-components embedding $u_{\text{comp}}$, as illustrated in Fig. 3 (right). Mathematically, we define $u_{\text{comp}}$ is computed by the aggregation of all components in the sentence:

$$
u_{\text{comp}} = \text{Norm}(\Phi(\{u_{\text{obj}} \cup \{u_{\text{attr}} \cup \{u_{\text{rel}}\}})),$$

where $\Phi(\cdot)$ is the aggregation function of semantic components. Then the caption is encoded as: $u_{\text{cap}} = \alpha u_{\text{sent}} + (1 - \alpha) u_{\text{comp}}$, where $0 \leq \alpha \leq 1$ is a scalar weight. The presence of $u_{\text{comp}}$ disallows the ignorance of any of the components in the final caption embedding $u_{\text{cap}}$.

### 3.3. Image Encodings

We use CNN to encode the input RGB image into the unified VSE space. Specifically, we choose a ResNet-152 model [14] pretrained on ImageNet [37] as the image encoder. We apply a layer of $1 \times 1$ convolution on top of the last convolutional layer (i.e., $\text{conv5}_3$) and obtain a convolutional feature map of shape $7 \times 7 \times d$ for each image. $d$ denotes the dimension of the unified VSE space.

The feature map, denoted as $V \in \mathbb{R}^{7 \times 7 \times d}$, can be view as the embeddings of $7 \times 7$ local regions in the image. The embedding $V$ for the whole image is defined as the aggregation $\Psi(\cdot)$ of the embeddings at all regions through a global spatial pooling operator.

### 3.4. Learning Paradigm

In this section, we present how to align vision and language into the unified space using contrastive learning on different semantic levels. The training pipeline is illustrated in Fig. 3. We start from the generation of contrastive examples for different semantic components.

**Negative example sampling.** It has been discussed in [40] that to build a large compositional space of semantics, directly sampling negative captions from a human-built dataset (e.g., MS-COCO captions) is not sufficient. In this paper, instead of manually define rules that augment the training data as in [40], we address this problem by sampling contrastive negative examples in the explicitly factorized semantic space. The generation does not require manually labelled data, and can be easily applied to any datasets. For a specific caption, we generate the following four types of contrastive negative samples.
• **Nouns.** We sample negative noun words from all nouns that do not appear in the caption.

• **Attribute-noun pairs.** We sample negative pairs by randomly substituting the adjective by another adjective or substituting the noun.

• **Relational triples.** We sample negative triples by randomly substituting the subject, or the relation, or the object. Moreover, we also sample the whole relational triples of captions in the dataset which describe other images, as the negative triples.

• **Sentences.** We sample negative sentences from the whole dataset. Meanwhile, following [13, 11], we also sample negative images from the whole dataset as constrastive images.

The key motivation behind our visual-semantic alignment is that: an object appears in a local region of the image, while the aggregation of all local regions should be aligned with the full semantics of a caption.

**Local region-level alignment.** In detail, we propose a relevance-weighted alignment mechanism for linking textual object descriptors and local image regions. As shown in Fig. 4, consider the embedding of a positive textual object descriptor $u^+_n$, a negative textual object descriptor $u^-_n$, and the set image local region embeddings $V_i$ where $i \in 7 \times 7$ extracted from the image. We generate a relevance map $M \in \mathbb{R}^{7\times7}$ with $M_i$, $i \in 7 \times 7$ representing the relevance between $u^+_n$ and $V_i$, computed as as Eq. (2). We compute the loss for noun and (adjective, noun) pairs by:

$$M_i = \frac{\exp(s(u^+_n, V_i))}{\sum_j \exp(s(u^+_n, V_j))}$$ (2)

$$\ell_{obj} = \sum_{i \in 7 \times 7} \left( M_i \cdot \beta + s(u^-_n, V_i) - s(u^+_n, V_i) \right)_+$$ (3)

The intuition behind the definition is that, we explicitly try to align the embedding at each image region with $u^+_n$. The losses are weighted by the matching score, thus reinforce the correspondence between $u^+_n$ and the matched region. This technique is related to multi-instance learning [45].

**Global image-level alignment.** For relational triples $u_{rel}$, semantic components aggregations $u_{comp}$ and sentences $u_{sent}$, their semantics usually cover multiple objects. Thus, we align them with the full image embedding $v$ via bidirectional ranking losses as Eq. (1). The alignment loss is denoted as $\ell_{rel}$, $\ell_{comp}$ and $\ell_{sent}$, respectively.

We want to highlight that, during training, we separately align the two type of semantic representations of the caption, i.e., $u_{sent}$ and $u_{comp}$, with the image. This differs from the inference-time computation of the caption. Recall that $\alpha$ can be viewed as a factor that balances the training objective and the enforcement of semantic coverage. This allows us to flexibly adjust $\alpha$ during inference.

### 3.5. Implementation details

We use $d = 1024$ as the dimension of the unified VSE space like [11, 40, 49]. We train the model by minimizing the alignment losses in a multi-task learning way.

$$\ell = \ell_{sent} + \eta_c \ell_{comp} + \eta_o \ell_{obj} + \eta_a \ell_{attr} + \eta_r \ell_{rel}$$ (4)

In the first 2 epochs, we set $\eta_c$, $\eta_o$ and $\eta_a$ to 0.5 and $\eta_r$ to 0 for learning single-object level representations. Then we turn up $\eta_r$ to 1.0 to make the model learn relational semantics. To make the comparison with related works fair, we always fix the weights of the ResNet. We use the Adam [22] optimizer with learning rate at 0.001. For model details, please refer to our supplementary material.

### 4. Experiments

We evaluate our model on the MS-COCO [27] dataset. It contains 82,783 training images with each image annotated by 5 captions. We use the common 1K validation and test split from [19]. We also report the performance on a 5K test split for comparison with [49, 11, 42].

We begin this section with the evaluation of traditional cross-modal retrieval. Next, we validate the effectiveness of enforcing the semantic coverage of caption embeddings by comparing models on cross-modal retrieval tasks with adversarial examples. We then propose a unified text-to-image retrieval task to support the contrastive learning on various semantic components. We end this section with an application of using visual cues to facilitate the semantic parsing of novel sentences. Due to the limitation of the text length, for mode details on data processing, metrics and model implementation, we refer the readers to our supplementary material.

### 4.1. Overall Evaluation on Cross-Modal Retrieval.

We first show the performance of image-to-sentence and sentence-to-image retrieval tasks to evaluate learned visual-semantic embeddings. We report the R@1 (recall@1), R@5,
To summarize the performance, we compute \( r_{sum} \) as the summation of \( R@1, R@5, \) and \( R@10 \). We conduct experiments on the image-to-sentence retrieval task with text-domain adversarial attacks. Following [40], we first design several types of adversarial captions by adding perturbations to existing captions.

1. **Object attack**: Randomly replace / append by an irrelevant one in the original caption.

2. **Attribute attack**: Randomly replace / add an irrelevant attribute modifier for one object in the original caption.

3. **Relational attack**: 1) Randomly replace the subject/relation/object word by an irrelevant one. 2) Randomly select an entity as a subject/object and add an irrelevant relational word and object/subject.

We include VSE++ and VSE-C as the baselines and show the results in Table 2 where different columns represent different types of attacks. VSE++ performs worst as it is only optimized for the retrieval performance on the dataset. Its sentence encoder is insensitive to a small perturbation in the text. VSE-C explicitly generates the adversarial captions based on human-designed rules as hard negative examples during training, which makes it relatively robust to those

### Table 1. Results of cross-modal retrieval task on MS-COCO dataset (1K and 5K testing split).

<table>
<thead>
<tr>
<th>Metric</th>
<th>1K testing split (5,000 captions)</th>
<th>5K testing split (25,000 captions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Image-to-sentence Retrieval</td>
<td>Sentence-to-image Retrieval</td>
</tr>
<tr>
<td></td>
<td>R@1</td>
<td>R@5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m-RNN [31]</td>
<td>41.0</td>
<td>73.0</td>
</tr>
<tr>
<td>DVS [20]</td>
<td>38.4</td>
<td>69.9</td>
</tr>
<tr>
<td>MNLM [24]</td>
<td>43.4</td>
<td>75.7</td>
</tr>
<tr>
<td>m-CNN [30]</td>
<td>42.8</td>
<td>73.1</td>
</tr>
<tr>
<td>HM-LSTM [33]</td>
<td>43.9</td>
<td>-</td>
</tr>
<tr>
<td>Order-embedding [42]</td>
<td>46.7</td>
<td>-</td>
</tr>
<tr>
<td>VSE-C [40, 1]</td>
<td>48.0</td>
<td>81.0</td>
</tr>
<tr>
<td>DeepS [44]</td>
<td>50.1</td>
<td>79.7</td>
</tr>
<tr>
<td>2WayNet [9]</td>
<td>55.8</td>
<td>75.2</td>
</tr>
<tr>
<td>sm-LSTM [15]</td>
<td>53.2</td>
<td>83.1</td>
</tr>
<tr>
<td>RRF-Net [28]</td>
<td>56.4</td>
<td>85.3</td>
</tr>
<tr>
<td>VSE++ [11, 2]</td>
<td>57.7</td>
<td>86.0</td>
</tr>
<tr>
<td>CSE [49]</td>
<td>56.3</td>
<td>84.4</td>
</tr>
<tr>
<td>UniVSE (Ours)</td>
<td>64.3</td>
<td>89.2</td>
</tr>
</tbody>
</table>

Table 2. Results on image-to-sentence retrieval task with text-domain adversarial attacks. For each caption, we generate 5 adversarial fake captions which do not match the images. Thus, the models need to retrieve 5 positive captions from 30,000 candidate captions.

R@10, and the median retrieval rank as in [11, 40, 49, 15]. To summarize the performance, we compute \( r_{sum} \) as the summation of R@1, R@5, and R@10.

Shown in Table 1, Unified VSE outperforms other baselines with various model architecture and training techniques [11, 49, 28, 40, 15]. This validates the effectiveness learning visual-semantic embeddings in the explicitly factorized visual-semantic embedding space. We also include the results under more challenging 5K test split. The gap between Unified VSE and other models gets further enlarged across all metrics.

### 4.2. Retrieval under text-domain adversarial attack

Recent works [40, 39] have raised their concerns on the robustness of the learned visual-semantic embeddings. They show that existing models are vulnerable to text-domain adversarial attacks (i.e., using adversarial captions) and can be easily fooled. This is closely related to the bias in small datasets over a large, compositional semantic space [40]. To prove the robustness of the learned unified VSE, we further
adversarial attacks. Unified VSE shows strong robustness across all types of adversarial attacks.

It is worth noting that VSE-C shows inferior performances in the normal retrieval tasks without adversarial captions (see Table 1), even compared with VSE++. Considering that VSE-C shares the exactly the same model architecture as VSE++, we can conclude that directly adding adversarial captions during training, although improves models’ robustness, may sacrifice the performance on other tasks. In contrast, the ability of Unified VSE to defend adversarial texts comes almost for free: we present zero adversarial captions during training. Unified VSE builds fine-grained semantic alignments via the contrastive learning of semantic components. It uses the explicit aggregation of the components \( u_{\text{comp}} \) to alleviate the dataset biases.

**Ablation study: semantic components.** We now delve into the effectiveness of different semantic components by choosing different combinations of components for the caption embedding. Shown in Table 2, we use different subsets of the semantic components to form the bag-of-component embeddings \( u_{\text{comp}} \). For example, in UniVSE\( \text{obj} \), only object nouns are selected and aggregated as \( u_{\text{comp}} \).

The results demonstrate the effectiveness of the enforcement of semantic coverage: even if the semantic components have got fine-grained alignment with visual concepts, directly using \( u_{\text{sent}} \) as the caption encoding still degenerates the robustness against adversarial examples. Consistent with the intuition, enforcing of coverage of a certain type of components (e.g., objects) helps the model to defend the adversarial attacks of the same type (e.g., defending adversarial attacks of nouns). Combining all components leads to the best performance.

**Choice of the combination factor: \( \alpha \).** We study the choice of \( \alpha \) by conducting experiments on both normal retrieval tasks and the adversarial one. Fig 4.2 shows the R@1 performance under the normal/adversarial retrieval scenario w.r.t. different choices of \( \alpha \). We observe that the \( u_{\text{comp}} \) term contributes little on the normal retrieval tasks but largely on tasks with adversarial attacks. Recall that \( \alpha \) can be viewed as a factor that balances the training objective and the enforcement of semantic coverage. By choosing \( \alpha \) from a reasonable range (0.6 to 0.8), our model can effective defend adversarial attacks, with no sacrifice for the overall performance.

### 4.3. Unified Text-to-Image Retrieval

We extend the word-to-scene retrieval used by [40] into a general unified text-to-image retrieval task. In this task, models receive queries of different semantic levels, including single words (e.g., “Clock.”), noun phrases (e.g., “White clock.”), relational phrases (e.g., “Clocks on wall”) and full sentences. For all baselines, the texts of different types are treated as full sentences. The result is presented in Table 3.

We generate positive image-text pairs by randomly choosing an image and a semantic component from 5 matched captions with the chosen image. It is worth mention that the semantic components extracted from captions may not cover all visual concepts in the corresponding image, which makes the annotation noisy. To address this, we also leverage the MS-COCO detection annotations to facilitate the evaluation (see \( \text{obj}(\text{det}) \) column). We treat the labels for detection bounding boxes as the annotation of objects in the scene.

**Ablation study: contrastive learning of components.** We evaluate the effectiveness of using contrastive samples for different semantic components. Shown in Table 3, UniVSE\( \text{obj} \) denotes the model trained with only contrastive samples of noun components. The same notation applies to other models. The UniVSE trained with a certain type of contrastive examples (e.g., UniVSE\( \text{obj} \) with contrastive nouns) consistently improves the retrieval performance of the same type of queries (e.g., retrieving images from a single noun). UniVSE trained with all kinds of contrastive samples performs best in overall and shows a significant gap w.r.t. other baselines.

**Visualization of the semantic alignment.** We visualize the semantic-relevance map on an image w.r.t. a given query \( u_q \) for a qualitative evaluation of the alignment performance of various semantic components. The map \( M_i \) is computed as the similarity between each image region \( v_i \) and \( u_q \), in a similar way as Eq. (2). Shown as Fig. 6, this visualization helps to verify that our model successfully aligns different semantic components with the corresponding image regions.
We choose the combination with the highest matching score with the image to decide the subject/object dependencies of the relation eat. We use parsed semantic components as the ground-truth and report the accuracy, defined as the fraction of the number of correct dependency resolution and the total number of attributes/relations. Table 4 reports the results on assisting semantic parsing with visual cues, compared with other baselines. Fig. 7 shows a real case in which we successfully resolve the textual ambiguity.

5. Conclusion

We present a unified visual-semantic embedding approach that learns a joint representation space of vision and language in a factorized manner: Different levels of textual semantic components such as objects and relations get aligned with regions of images. A contrastive learning approach for semantic components is proposed for the efficient learning of the fine-grained alignment. We also introduce the enforcement of semantic coverage: each caption embedding should have a coverage of all semantic components in the sentence. Unified VSE shows superiority on multiple cross-modal retrieval tasks and can effectively defend text-domain adversarial attacks. We hope the proposed approach can empower machines that learn vision and language jointly, efficiently and robustly.

6. Acknowledgements

We thank Haoyue Shi for helpful discussions and suggestions. This research is supported in part by the National Key Research and Development Program of China under grant 2018YFB0505000 and the National Natural Science Foundation of China under grant 61772138.
References

[31] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep Captioning with Multimodal Recurrent Neural Net-
works (m-RNN). In Proceedings of International Conference on Learning Representations (ICLR), 2015. 6
Supplementary Materials for Unified Visual-Semantic Embeddings

This supplementary material is organized as follows. First, in Appendix A, we provide more details for the implementation of our model and the training method. Second, in Appendix B, we provide the experiment setups, metrics, baseline implementations, qualitative examples and analysis for each experiment we discussed in the main text. We end this section with the visualization of the learned unified VSE space of different semantic levels.

A. Implementation Details
A.1. Generating Negative samples

To generate negative samples in sentence level, we follow the sampling paradigm introduced by [1]: We sample negative examples from all other captions/images in the dataset in a training batch. Note that as [1] shown, the batch size will largely affect the models’ performance. For a fair comparison, we set the batch size as 128 which is the same as [1, 4]. In the rest of this section, we discuss in detail how we sample negative semantic components.

As for nouns, we sample 16 negative nouns from a fixed set of nouns: This noun set is extracted from nouns with frequency more than 100 (in total 1,205 nouns extracted in MS-COCO dataset).

As for attribute-noun pairs, we randomly sample 8 other attributes in a fixed attribute set and replace the original attribute in the pair, as negative examples. The attribute set is composed by the frequently appeared attributes in the MS-COCO dataset. In detail, we extract in total 37 attributes, i.e., white, black, red, green, brown, yellow, orange, pink, gray/grey, purple, young, wooden, old, snowy, grassy, cloudy, colorful, sunny, beautiful, bright, sandy, fresh, morden, cute, dry, dirty, clean, polar, crowded, silver, plastic, concrete, rocky, wooded, messy, square. We also randomly replace nouns in the pairs to generate another set of negative attribute-noun pairs. For each attribute-noun pair, we randomly draw 16 negative examples.

We separately compute the ranking loss corresponding to two types of negatives, denoted as \( \ell_{\text{negnoun}} \) and \( \ell_{\text{negattr}} \). Both of them are computed by a uni-directional ranking loss with negative examples drawn in text-domain. OHEM strategy is not applied on them. The final loss is the sum of them, i.e., \( \ell_{\text{attr}} = \ell_{\text{negnoun}} + \ell_{\text{negattr}} \).

Here we add a small note for the reproducibility. In cases with multiple modifiers on the nouns (e.g., old black dog), for simplicity, in our implementation, we always extract the first modifier of each noun phrases as its attribute (old dog in this case).

As for relational triples, we randomly sample 4 relational words and 2 negatives subjects (nouns) and 2 negative objects (nouns) to replace the corresponding parts in the triple, as negative examples. In total, we have 8 negative triples for each relational triple The choice of this small number of negative examples is attributed to the trade-off between the computational efficiency and stability of training. Empirically, we find that increasing the number of negative triples does not bring much improvement to the performance.

We also sample negative relational triples from other captions within the training batch. In detail, we sample 1 negative rational triple for each other caption within the batch. This results in at most 128-1 = 127 negative examples for each relational triple (“at most” means some captions may not contain relational phrases). Similar as attribute-noun pairs, we individually compute the ranking loss on each type of negatives and sum them together as the \( \ell_{\text{rel}} \). The losses are computed by uni-directional ranking loss without OHEM.

As for negative bag-of-components, we sample negative ones in a similar manner as we do for sentences: We draw them from the bag-of-components in other captions within the training batch. We also draw other images from batch as negative images. The loss \( \ell_{\text{comp}} \) is computed by bi-directional ranking loss with OHEM strategy.
A.2. Model settings and details

**Weight of $\eta_c$, $\eta_o$, $\eta_a$, $\eta_r$.** The choice of the 4 hyperparameters in Eq.(4) (i.e., $\eta_c$, $\eta_o$, $\eta_a$, $\eta_r$) in the main text actually has no significant influence on the model’s performance, as they all contribute to the better alignment between two modalities. To show this, we fix three of the $\eta$s and test 5 different values for the rest one (e.g., set $\eta_c \in \{0.1, 1, 2, 4, 8\}$). The bidirectional retrieval scores $\text{rsum}$ of all 20 models are within the range of 468.2 ± 2.

**Dependency on the semantic parser.** Recall that the semantic components are all extracted by the semantic parser and we evaluate the influence of the recall of the semantic parser on the model’s performance by randomly dropping 30% attributes from the parser’s output during training/test. Shown in Table 1, the recall of the parser on training captions has a small contribution to the performance. However, low recalls on test captions noticeably degenerate the performance on discriminating adversarial captions, because UniVSE relies on the parsed components to find unmatched components between images and texts. Thus, in Section 4.4, we show that UniVSE can facilitate the semantic parsing with visual cues.

**Spatial aggregation method.** For the spatial aggregation $\Psi(\cdot)$ of the $7 \times 7$ image feature maps, instead of using the max pooling which may drop most information in the feature map or the average pooling which tends to include noises, we adopt a specific pooling method called max-$k$ pooling. Max-$k$ pooling select $k$ largest values in the feature map and return the average value of these $k$ largest responses. Formally, for the feature map $V \in \mathbb{R}^{7 \times 7 \times d}$, denoting the $k$-th largest value in $i$-th channel of $V$ as $V_k[i]$, the max-$k$ pooled global image embedding $v$ can be formalized as $v[i] = \text{mean}([V(x, y, i)|V(x, y, i) \leq V_k[i]], x, y \in [7 \times 7])$. Obviously, max pooling is the specific form of max-$k$ pooling when $k = 1$ and average pooling can be regarded as max-49 pooling (for a $7 \times 7$ spatial resolution). In the experiments, we empirically set $k$ to 10, as a trade-off between removing useful information and retaining unimportant information. Table 2 shows the performance of different spatial pooling methods. We can observe that the proposed max-$k$ pooling achieves best performance among three pooling methods (i.e., average pooling, max pooling and max-$k$ pooling). Notice that, max-$k$ pooling will bring better performances compared with max/average pooling, however, the max-$k$ pooling has to be trained under UniVSE structure (i.e., trained with $\ell_{\text{comp}}, \ell_{\text{obj}}, \ell_{\text{attr}}, \ell_{\text{rel}}$), otherwise, the local correspondences will not be learned well. The results in Table 2 shows a significant performance drop on defending the adversarial captions, if UniVSE (with max-10 pooling) is trained without loss $\ell_{\text{comp}}$.

**Semantic aggregation method** For the aggregation function $\Phi(\cdot)$ for semantic components to generate $u_{\text{comp}}$, we have added some experiments. Specifically, we evaluate the performance of both hand-coded functions: average pooling, sum, and max pooling (by taking a channel-wise max of all components), as well as learnable functions: GRU and self-attentive pooling [2]. For the GRU alternative, we treat the set of components as a sequence (ordered randomly), encode it with a GRU module, and use the last hidden state as the $u_{\text{comp}}$. The results are summarized in Table 3. GRU performs slightly better than other methods on the standard retrieval task. However, it requires extra computation cost. Max pooling outperforms others in discriminating adversarial captions, suggesting that it makes $u_{\text{comp}}$ more sensitive to the presence of unmatched components than the “average” alternative. However, it shows slightly inferior results on the standard retrieval task. In the experiments, we adopt average pooling as the implementation of semantic aggregation function $\Phi(\cdot)$.

**Setting of $\alpha$.** One may argue that the combination coefficient $\alpha$ can also be learnable when training the model instead of being a fixed value (0.75 in the experiments). Informally, $u_{\text{comp}}$ imposes a prior that the caption embedding should cover all semantic components in the text. The hyperparameter $\alpha$ controls the strength of this prior (see Figure 8 in the main text for details). Directly learning $\alpha$ under the supervision of the standard retrieval task may encourage the model to focus on only part of the semantic components [4]. Our empirical results support this: $\alpha$ finally converges to 0.93 when treated as a learnable parameter, in contrast to the value of 0.75 suggested in our paper. Shown in Table 4, making $\alpha$ learnable does not affect the performance on the standard retrieval tasks. However, it shows a significant performance drop when there are adversarial captions. Thus, we treat $\alpha$ as a fixed value in UniVSE.

**A.3. Hyperparameters**

We set the dimension $d_{\text{basic}}$ of basic semantic embeddings as 300. The embeddings are initialized by GloVe word embeddings pre-trained on the Common Crawl dataset: [http://nlp.stanford.edu/data/glove.840B.300d.zip](http://nlp.stanford.edu/data/glove.840B.300d.zip). The dimension $d_{\text{modif}}$ of modifier semantic embeddings is set to 100. The embeddings are randomly initialized. During training, we fix the basic semantic embeddings of words $w^{(\text{basic})}$. The learning rate of the Adam optimizer is fixed to 0.001 at first 6 epochs and is exponentially decayed by 2 for each next epoch until it reaches 1e-5.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td>469.5</td>
<td>210.9</td>
<td>189.9</td>
<td>202.2</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>468.9</td>
<td><strong>213.7</strong></td>
<td><strong>191.5</strong></td>
<td>199.4</td>
</tr>
</tbody>
</table>

Table 1. We evaluate the performance of UniVSE on the standard bidirectional retrieval task and the retrieval tasks with adversarial captions (object-typed, attribute-typed and relation-typed). We use \( rsum \) as the evaluation metric.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>452.9</td>
<td>198.7</td>
<td>184.2</td>
<td>186.2</td>
</tr>
<tr>
<td>Max</td>
<td>462.5</td>
<td>209.6</td>
<td>184.1</td>
<td>193.6</td>
</tr>
<tr>
<td>Max-10</td>
<td><strong>469.5</strong></td>
<td><strong>210.9</strong></td>
<td><strong>189.9</strong></td>
<td><strong>202.2</strong></td>
</tr>
<tr>
<td>Max-10 (without ( \ell_{comp} ))</td>
<td>466.1</td>
<td>202.9</td>
<td>182.1</td>
<td>192.2</td>
</tr>
</tbody>
</table>

Table 2. We evaluate the performance of UniVSE under different spatial aggregation settings on the standard retrieval task, and the retrieval tasks with adversarial captions (we report the \( rsum \)s).

<table>
<thead>
<tr>
<th>Semantic Aggregation Methods</th>
<th>Avg</th>
<th>Sum</th>
<th>Max</th>
<th>Self-Att.</th>
<th>GRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard (( rsum ))</td>
<td>469.5</td>
<td>471.1</td>
<td>465.8</td>
<td>467.1</td>
<td><strong>472.0</strong></td>
</tr>
<tr>
<td>Adversarial (( rsum ))</td>
<td>603.0</td>
<td>604.3</td>
<td><strong>628.4</strong></td>
<td>599.5</td>
<td>603.7</td>
</tr>
</tbody>
</table>

Table 3. We evaluate the performance of UniVSE under different semantic aggregation settings on the standard retrieval task, and the retrieval tasks with adversarial captions (we report the sum of \( rsum \)s under three types of attacks).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed ( \alpha ) (0.75)</td>
<td><strong>469.5</strong></td>
<td><strong>210.9</strong></td>
<td><strong>189.9</strong></td>
<td><strong>202.2</strong></td>
</tr>
<tr>
<td>Learnable ( \alpha ) (0.93)</td>
<td>468.1</td>
<td>204.1</td>
<td>182.2</td>
<td>190.4</td>
</tr>
</tbody>
</table>

Table 4. The performance of UniVSE with a learnable or a fixed \( \alpha \) on the standard retrieval task, and the retrieval tasks with adversarial captions (we report the \( rsum \)s).

B. Experiment Details

B.1. Cross-modal Retrieval

Visualizations. We show a set of examples of the image-to-sentence retrieval in Fig. 1 and sentence-to-image retrieval in Fig. 2.

B.2. Retrieval under text-domain adversarial attack

Experiment setup. We use the 1K test split (including 5,000 captions) for generating adversarial attacks. For each caption, we generate five adversarial captions under one type of attack setting. The detailed settings of the three types of adversarial attack are listed below.

1. **Object attack**: We randomly replace / append by an irrelevant noun for both 50% probability. The replacing/appending place is randomly selected in nouns of the caption. For the case of appending extra noun, the word and is also added before the appended noun, e.g., A dog eats meat \( \rightarrow \) A dog eats meat and table. The irrelevant nouns are drawn from the set containing nouns with high concreteness (manually extracted).

2. **Attribute attack**: If a caption contains attribute-noun pairs. We randomly select one pair and replace the attribute by a negative one. If a caption does not contain any attributes, we randomly choose one noun in the caption and append an attribute on it. The negative attribute is generated from the attribute set excluding the attributes (and its similar attributes) in the caption. The similar attribute group is defined as the following. \{white, snowy, polar\}, \{red, pink\}, \{blue, cloudy\}, \{green, grassy\}, \{brown, sandy, yellow, orange\}, \{rocky, concrete\}.
3. **Relational attack**: For those captions containing relational phrases, we randomly select one relation triple and with equivalent probability to choose one in the triple to be replaced by an irrelevant one. *e.g.*, *A dog eats meat* → *A dog plays meat*. For those captions which do not have any relational phrases, we first randomly select one noun in the caption and regard it as a subject/object with 50% / 50% probability. Then we draw a relational word and an irrelevant noun as the object/subject to form a new fake relation. *e.g.*, *A dog is sleeping* → *A dog in sky is sleeping.*

**Baselines.** We train the VSE-C according to the setting in [4] with the officially open-sourced code. In the original VSE-C paper, The VSE-C is trained by generating either noun-typed/numeral-typed/relation-typed or all of these three types of adversarial samples. We use the setting of training under all types of adversarial samples as a comparable competitor in this evaluation. For the ablation of UniVSE (*u_{sent} + u_{attr}* (i.e., *use u_{attr} as u_{comp}*)) under attribute attack scenario, we additionally include *u_{obj} to u_{comp}*. The reason is that the attribute attack may add new attribute modifier on a sentence with *no* attributed phrases. *u_{comp}* is not defined for such sentence if we only use *u_{attr} as u_{comp}*, since it does not contain any attributed phrases. As a solution, we additionally include *u_{obj} to u_{comp}* (i.e., *u_{comp} = Φ({u_{attr}} ∪ {u_{obj}})* to ensure *u_{comp}* is well defined even there is no attributed phrases in the sentence.

**Visualizations.** We show a set of examples of image-to-text retrieval under text-domain adversarial attack in Fig. 3.

**B.3. Unified text-to-image retrieval**

**Experiment setup.** We use the 1K test split as the retrieval set. The queries are generated from frequent semantic components extracted by the semantic parser from the training set. We regard a query as a valid one if at least 3 images (5 for noun-level retrieval) in the test set contain the query. For the obj(det) queries, we directly use the class names of the MS-COCO object detection / segmentation annotations.

**Baselines.** For VSE++ and VSE-C, as they do not have an object-level encoder. For any query, we always regard it as a short sentence and feed it into the sentence encoder to get the embedding of the query text. For UniVSE, as it has the object-level encoder which means a noun/attribute-noun pair can be either encoded by the object encoder ϕ or by neural combiner ψ by regarding the query as a short sentence. We select the encoder having higher performance on a validation set and report the results.

**Visualizations.** We show a set of retrieved image by queries of various types in Fig. 4.

**B.4. Semantic Parsing**

**Experiment setup.** We also use the 1K test split for this experiment. For each caption, we first extract nouns, adjectives and relational words. We call adjective and relational words as content words. The model should recover the dependencies linked with them. We exclude some relational words whose lexical meanings are usually ambiguous, such as include, to, of, etc.

Given a content word (either an adjective or a relational word), we generate all possible dependencies among nouns in the sentence to form candidate dependencies. Each candidate dependency, which is either an adjective-noun pair or a subject-relation-object triple, will get a matching score w.r.t. the image (the visual cue). We select the dependency that has the highest score as the recovered dependency w.r.t. the chosen content word.

**Metrics.** We report the accuracy of the recovered semantic dependencies. In detail, for an attribute-noun dependency, the model gets a correct count if the dependency having the highest matching score is identical to the ground-truth. For the dependency of a relation, the model gets 0.5 correct counts if the subject/object of the answer is the same as the ground-truth. If both of them are the same as the ground-truth, the model gets 1 correct count. The reported accuracy computed as the fraction between total correct counts and the total number of dependencies.

**Visualizations and failure case study.** Shown in Fig. 5, we visualize some successful and failure cases in semantic parsing with visual cues. Error source analysis is also provided.

**B.5. Embedding Visualization**

We visualize the semantic space of different semantic levels by t-SNE [3]. The result can be found in Fig. B.5. Through the joint learning of vision and language, our unified VSE space successfully recovers the similarities between semantic components at various levels.
References


Figure 1. Examples showing the top-5 image-to-text retrieval results. We highlight the positive captions in blue. The score in the front of each sentence is the similarity score of the caption and image computed by different model. Best viewed in color.
(a) Query: A white chair, books and shelves and a TV on in this room.

(b) Query: A couple of people sitting on a bench next to a dog.

(c) Query: Window view from the inside of airplanes, baggage carrier and tarmac.

Figure 2. Examples showing the top-5 sentence-to-image retrieval results. We highlight the correct images in green box.
Figure 3. Examples showing the top-5 image-to-sentence retrieval results with the presence of adversarial samples. We highlight the positive captions in blue. Captions with red words are adversarial samples generated from the original captions. Words in red indicates the irrelevant words in the adversarial captions. Best viewed in color.
Figure 4. The top-20 retrieved image in the 1K test split set by queries different types: attribute-object pairs and relational triples.
Figure 5. Examples showing the result of semantic parsing based on visual cues. The first and second rows visualize the examples of corrected dependency resolution and the last two rows are the failure cases (dependency resolutions differs from the one by our semantic parser). Words in italic are the content words whose dependency is to be recovered, and the words in red are wrong predictions. Fig. (g) is a failure case of our semantic parser: the word `couple` does not refer to a specific object in the scene. In Fig. (h), both dependencies `cat-next-chair` and `cat-next-table` are actually valid based only on visual cue. Similarly, Fig. (i), (j) and (k) are all cases where only visual cues can not recover the dependency. The result in Fig. (i) shows that our model has the tendency of linking spatially closer objects. In Fig. (l), `hat` and `bananas` actually refers to the same object. Best viewed in color.
Figure 6. The visualization of the semantic embedding space of different semantic levels. The unified VSE space successfully recovers the similarities between semantic components at various levels.