Improving Cross-Modal Retrieval with Set of Diverse Embeddings

Dongwon Kim[1]   Namyup Kim[1]   Suha Kwak[1,2]
[1]Dept. of CSE, POSTECH   [2] Graduate School of AI, POSTECH
https://cvlab.postech.ac.kr/research/DivE/

Abstract

Cross-modal retrieval across image and text modalities is a challenging task due to its inherent ambiguity: An image often exhibits various situations, and a caption can be coupled with diverse images. Set-based embedding has been studied as a solution to this problem. It seeks to encode a sample into a set of different embedding vectors that capture different semantics of the sample. In this paper, we present a novel set-based embedding method, which is distinct from previous work in two aspects. First, we present a new similarity function called smooth-Chamfer similarity, which is designed to alleviate the side effects of existing similarity functions for set-based embedding. Second, we propose a novel set prediction module to produce a set of embedding vectors that effectively captures diverse semantics of input by the slot attention mechanism. Our method is evaluated on the COCO and Flickr30K datasets across different visual backbones, where it outperforms existing methods including ones that demand substantially larger computation at inference.

1. Introduction

Cross-modal retrieval is the task of searching for data relevant to a query from a database when the query and database have different modalities. While it has been studied for various pairs of modalities such as video-text [3, 7, 19] and audio-text [5, 16], the most representative setting for the task is the retrieval across image and text modalities [10, 29, 37, 45]. A naïve solution to cross-modal retrieval is a straightforward extension of the conventional unimodal retrieval framework [17, 18, 26], *i.e.*, learning a joint embedding space of the different modalities with known ranking losses (*e.g.*, contrastive loss [9] and triplet loss [43]). In this framework, each sample is represented as a single embedding vector and the task reduces to neighbor search on the joint embedding space.

However, this naïve approach has trouble in handling the inherent ambiguity of the cross-modal retrieval across image and text modalities [10, 45, 46]. A cause of the am-

"Boys wearing helmets carry a bicycle up a ramp at a skate park."

"Small children stand near bicycles at a skate park."

"A group of young children riding bikes and skateboards."

Figure 1. An example of the ambiguity problem introduced in the cross-modal retrieval task; an image region and a word corresponding to each other are highlighted in the same color. This example demonstrates that a single image can be coupled with multiple heterogeneous captions.

biguity is the fact that even a single image often contains various situations and contexts. Consider an image in Figure 1, which illustrates a group of children in a skate park. One of the captions coupled with it could be about children carrying up a bike, while another may describe the child riding a skateboard. Indeed, different local features of the image are matched to different captions. Similarly, visual manifestations of a caption could vary significantly as text descriptions are highly abstract. This ambiguity issue suggests that a sample should be embedded while reflecting its varying semantics in cross-modal retrieval. Embedding models dedicated to the uni-modal retrieval do not meet this requirement since they represent a sample as a single embedding vector.

Various methods have been studied to mitigate the ambiguity issue of cross-modal retrieval. Most of them adopt cross-attention networks that directly predict the similarity of an input image-caption pair [12, 14, 25, 30, 37, 38, 49, 50, 53]. These models successfully address the ambiguity since they explicitly infer relations across the modalities by drawing attentions on both modalities at once. However, they inevitably impose a large computation burden on retrieval systems since they demand both image and caption to be processed together for computing their similarity; all data in a database thus have to be reprocessed whenever a query arrives. In contrast, methods using separate textual and visual encoders [17, 21, 24, 29] seek to find sam-

ples relevant to query through nearest-neighbor search on pre-computed embedding vectors, which are more suitable to nowadays retrieval systems working on huge databases. However, most of the previous arts in this direction cannot address the ambiguity issue since their encoders return a single embedding vector for a given input.

Recent studies [10, 45] have advanced embedding model architectures to tackle the ambiguity issue even with separate encoders for the two modalities. To be specific, their models compute a set of heterogeneous embedding vectors for a given sample using self-attention layers on top of visual or textual features; such a set of embedding vectors is called *embedding set* in the remainder of this paper. Then the ambiguity issue is addressed through elements of the embedding set that encode diverse semantics of input.

Although the set-based embedding models enable a retrieval system to be powerful yet efficient, however, similarity functions used for their training do not consider the ambiguity of the data. Hence, training the models with the similarity functions often causes the following two side effects: (1) *Sparse supervision*–An embedding set most of whose elements remain untrained, or (2) *Set collapsing*–An embedding set with a small variance where elements do not encode sufficient ambiguity. Further, self-attention modules used for set prediction in the previous work do not explicitly consider disentanglement between set elements. These limitations lead to an embedding set whose elements encode redundant semantics of input, which also causes the *set collapsing* and degrades the capability of learned embedding models.

To address the aforementioned limitations of previous work, we propose a novel set-based embedding method for cross-modal retrieval. The proposed method is distinct from previous work in mainly two aspects. First, we design a novel similarity function for sets, called smooth-Chamfer similarity, that is employed for both training and evaluation of our model. In particular, our loss based on the smooth-Chamfer similarity addresses both limitations of the existing similarity functions, *i.e.*, sparse supervision and set collapsing. Second, we propose a model with a novel set prediction module motivated by slot attention [33]. In the proposed module, learnable embeddings called element slots compete with each other for aggregating input data while being transformed into an embedding set by progressive update. Therefore, our model captures the diverse semantic ambiguity of input successfully, with little redundancy between elements of the embedding set.

The proposed method is evaluated and compared with previous work on two realistic cross-modal retrieval benchmarks, COCO [32] and Flickr30K [41], where it outperforms the previous state of the art in most settings. In summary, our contribution is three-fold as follows:

- We address issues on previous set-based embedding

methods by proposing a novel similarity function for sets, named smooth-Chamfer similarity.

- We introduce a slot attention based set prediction module where elements of embedding set iteratively compete with each other for aggregating input data, which can capture semantic ambiguity of input without redundancy.

- Our model achieved state-of-the-art performance on the COCO and Flickr30K datasets, two standard benchmarks for cross-modal retrieval.

2. Related Work

Cross-modal retrieval: Previous work can be categorized into two classes, methods using two separate encoders and those using a cross-attention network. Models of the first class consist of disjoint visual and textual encoders and learn an embedding space where embedding vectors of matching pairs are located nearby. Retrieval is then performed by finding nearest neighbors of query on the embedding space, which is efficient since their embedding vectors are pre-computed. To improve the quality of the embedding space, loss functions [10, 17], model architectures [24, 27, 31], and embedding methods [10, 45] have been proposed. Though the separate encoders enable simple and fast retrieval, they often failed to handle the inherent ambiguity of cross-modal retrieval, leading to inferior retrieval performance. On the other hand, models of the other class adopt a cross-attention network that directly predicts the similarity score between an image-caption pair [14, 30, 49, 50, 52, 53]. Though these methods make their models explicitly consider relations across modalities, they are not suitable for real-world retrieval scenarios due to the large computation burden imposed during evaluation. We thus focus on improving the separate encoders in this paper.

Embedding beyond vector representation: Most of existing work projects each sample into a single embedding vector [17, 21, 24]. However, it has been empirically proven that such single vector representation is not sufficient to deal with inherent ambiguity of data [44, 48]. To mitigate this issue for the cross-modal retrieval task, PVSE [45] and PCME [10] introduce novel embedding methods. To be specific, PVSE represents each sample as a set of embedding vectors and PCME introduces probabilistic embedding where each sample is represented as a set of vectors sampled from a normal distribution.

Slot attention: Recently, slot attention [33] is proposed to learn object-centric representation, which is especially beneficial for tasks that require perception of objectness, such as object discovery and set property prediction. Slots, which are embedding vectors sampled from a random distribution, compete with each other for explaining the input
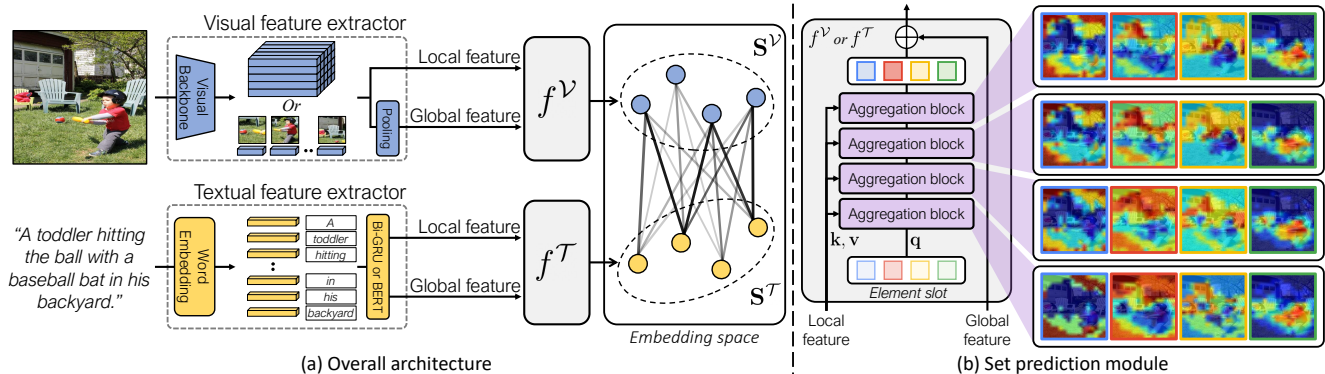
Figure 2. An overview of our model. (a) The overall framework of our model. The model consists of three parts: visual feature extractor, textual feature extractor, and set-prediction modules $f^{\mathcal{V}}$ and $f^{\mathcal{T}}$. First, the feature extractors of each modality extract local and global features from input samples. Then, the features are fed to the set prediction modules to produce embedding sets $\mathbf{S}^{\mathcal{V}}$ and $\mathbf{S}^{\mathcal{T}}$. The model is trained with the loss using our smooth-Chamfer similarity. (b) Details of our set prediction module and attention maps that slots of each iteration produce. A set prediction module consists of multiple aggregation blocks that share their weights. Note that $f^{\mathcal{V}}$ and $f^{\mathcal{T}}$ have the same model architecture.

in an iterative manner. This attention mechanism enables the final outputs to encode heterogeneous semantics that appear in the input without any explicit supervision. To produce informative embedding sets with sufficient within-set variance, our set prediction module employs the slot attention mechanism. However, the ability of slot attention to discover individual objects is only verified on synthetic datasets [28] and is known to fail on real-world images [4]. We make three architectural modifications to resolve this issue: (1) using learnable embeddings for initial element slots instead of random vectors, (2) replacing GRU [8] with a residual sum, and (3) adding a global feature to the final element slots. We will detail these differences in Section 3.2. Without them, we observed that the loss does not converge, and thus training fails.

3. Proposed Method

This section first describes the overall model architecture and elaborates on the proposed set prediction module. Then we introduce our smooth-Chamfer similarity and illustrate how it differs from existing set similarity functions. Finally, details of the training and inference of our model are provided.

3.1. Overall Model Architecture

The overall framework of our method is illustrated in Figure 2. Our model architecture consists of a visual feature extractor, a textual feature extractor, and a set prediction module of each modality: $f^{\mathcal{V}}$ and $f^{\mathcal{T}}$. The feature extractors have two branches that compute local features and global features from the input sample, respectively. The extracted features are given as input to the set prediction modules, each of which fuses local and global features to encode

an embedding set. For the visual and textual feature extractors, we followed the conventional setting of the previous work [10, 27, 45].

Visual feature extractor: We consider two different types of visual feature extractors for fair comparisons with previous work. One employs a flattened convolutional feature map as local features and their average pooled feature as the global feature. The other uses ROI features of a pretrained object detector as local features and their max pooled feature as the global feature. In any of these cases, the local and global features are denoted by $\psi^{\mathcal{V}}(\mathbf{x}) \in \mathbb{R}^{N \times D}$ and $\phi^{\mathcal{V}}(\mathbf{x}) \in \mathbb{R}^{D}$, respectively, where $\mathbf{x}$ is input image.

Textual feature extractor: For the textual feature, we also employ two different types of extractors: bi-GRU [8] and BERT [13]. When using bi-GRU, for L-words input caption $\mathbf{y}$, we take GloVe [40] word embedding of each word as a local feature $\psi^{\mathcal{T}}(\mathbf{y}) \in \mathbb{R}^{L \times 300}$. Then we apply bi-GRU with D dimensional hidden states on the top of the $\psi^{\mathcal{T}}(\mathbf{y})$. The last hidden state is used as the global feature $\phi^{\mathcal{T}}(\mathbf{y}) \in \mathbb{R}^{D}$. Similarly, when using BERT, output hidden states of BERT and their max pooled feature are used as a $\psi^{\mathcal{T}}(\mathbf{y}) \in \mathbb{R}^{L \times D}$ and $\phi^{\mathcal{T}}(\mathbf{y}) \in \mathbb{R}^{D}$, respectively.

3.2. Set Prediction Module

Elements of an embedding set should encode heterogeneous semantics that appear in the input data. Otherwise, it will degenerate into a set with small variations that cannot handle the ambiguity of the input.

Inspired by slot attention [33], we introduce an aggregation block, where *element slots* compete with each other for aggregating input data. The slots are progressively updated to capture various semantics of the input through multiple iterations of the aggregation block, and then the slots

are fused with a global feature to be used as elements of the output embedding set. In this manner, the proposed module produces an embedding set whose elements encode substantially different semantics, meanwhile preserving the global context of the input data. In this section, we only present how the visual set prediction module $f^{\mathcal{V}}$ works; $f^{\mathcal{T}}$ has the same model architecture and works in the same manner with $f^{\mathcal{V}}$.

As shown in Figure 2(b), the proposed set prediction module consists of multiple aggregation blocks, which share their weights. We define the initial element slots as learnable embedding vectors $\mathbf{E}^0 \in \mathbb{R}^{K \times D}$, where K is the cardinality of the embedding set. Then, the element slots of the t-th iteration $\mathbf{E}^t$ are computed by

$$\mathbf{E}^t = \text{AggBlock}(\psi^{\mathcal{V}}(\mathbf{x}); \mathbf{E}^{t-1}) \in \mathbb{R}^{K \times D}. \quad (1)$$

The aggregation block first layer-normalizes [2] inputs and then linearly projects input local feature $\psi^{\mathcal{V}}(\mathbf{x})$ to $\mathbf{k} \in \mathbb{R}^{N \times D_h}$ and $\mathbf{v} \in \mathbb{R}^{N \times D_h}$, and $\mathbf{E}^{t-1}$ to $\mathbf{q} \in \mathbb{R}^{K \times D_h}$. [1] Then, the attention map A between $\psi^{\mathcal{V}}(\mathbf{x})$ and $\mathbf{E}^{t-1}$ is obtained via

$$A_{n,k} = \frac{e^{M_{n,k}}}{\sum_{i=1}^{K} e^{M_{n,i}}}, \text{ where } M = \frac{\mathbf{k}\mathbf{q}^{\top}}{\sqrt{D_h}}. \quad (2)$$

It is worth noting that the attention map is normalized over *slots*, not *keys* as in the transformer [47]. Since this way of normalization lets slots compete with each other, each slot attends to nearly disjoint sets of local features, and these sets will correspond to the distinctive semantics of the input.

Using the obtained attention map, we update the element slot with the weighted mean of local features and then feed it to a multi-layer perceptron (MLP) with layer normalization, residual connection, and GELU [23] activation:

$$\hat{A}_{n,k} = \frac{A_{n,k}}{\sum_{i=1}^{N} A_{i,k}}, \overline{\mathbf{E}^t} = \hat{A}^{\top}\mathbf{v}W_o + \mathbf{E}^{t-1}, \quad (3)$$

$$\mathbf{E}^t = \text{AggBlock}(\psi^{\mathcal{V}}(\mathbf{x}); \mathbf{E}^{t-1}) = \text{MLP}(\overline{\mathbf{E}^t}) + \overline{\mathbf{E}^t}, \quad (4)$$

where $W_0 \in \mathbb{R}^{D_h \times D}$ is a learnable linear projection. The output of the aggregation block is used as an element slot for the t-th iteration. As shown in Figure 2, attention maps that slots of early stage produce are sparse and noisy, but refined to aggregate local features with distinctive semantics as proceeds.

Finally, at T-th iteration, the model predicts an embedding set $\mathbf{S}$ by adding the global feature $\phi^{\mathcal{V}}(\mathbf{x})$ to each element slot $\mathbf{E}^T$:

$$\mathbf{S} = \text{LN}(\mathbf{E}^T) + \left[\text{LN}(\phi^{\mathcal{V}}(\mathbf{x})) \overset{\times K}{\cdots} \right], \quad (5)$$

---

[1] When using convolutional features for visual feature extraction, we add sinusoidal positional encoding before projecting $\psi^{\mathcal{V}}(\mathbf{x})$ to $\mathbf{k}$.

(a) MIL similarity

(b) Match probability
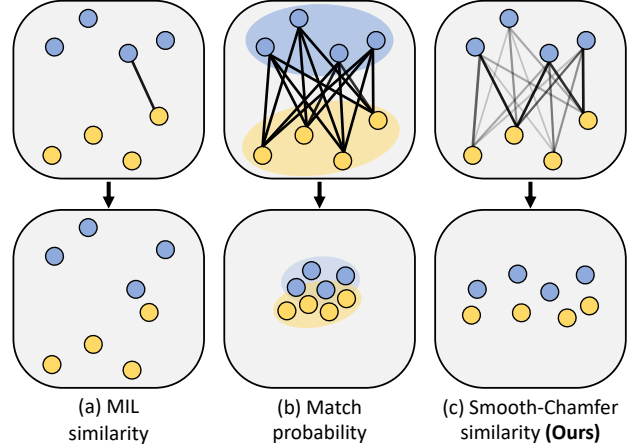
(c) Smooth-Chamfer similarity **(Ours)**

Figure 3. Comparison between our similarity function and existing ones used for embedding set. Illustrations of embedding space before and after optimizing each similarity function are presented. Presented two sets are matching pairs, and thus optimized to maximize their similarity. Lines indicate the association that similarity functions consider, where their intensities represent the weights given to each association.

where LN is a layer normalization, and $\left[\phi^{\mathcal{V}}(\mathbf{x}) \overset{\times K}{\cdots} \right] \in \mathbb{R}^{K \times D}$ is K repetitions of $\phi^{\mathcal{V}}(\mathbf{x})$. The module benefits from the global feature when treating samples with little ambiguity since the global feature allows them to be represented as an embedding set of small within-set variance.

3.3. Smooth-Chamfer Similarity

Our smooth-Chamfer similarity is proposed to resolve the drawbacks of the existing set-based embedding approaches. Before presenting smooth-Chamfer similarity, we first review the similarity functions used in PVSE [45] and PCME [10]. Figure 3 illustrates embedding spaces trained with these set similarity functions. Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be the embedding sets, which are the set of vectors. $c(x, y)$ denotes the cosine similarity between vectors x and y.

PVSE adopts the multiple instance learning (MIL) framework [15] during training and inference. Its similarity function, which we call MIL similarity, is given by $s_{\text{MIL}}(\mathbf{S}_1, \mathbf{S}_2) = \max_{x \in \mathbf{S}_1, y \in \mathbf{S}_2} c(x, y)$. MIL similarity only takes account of the closest pair of elements, as shown in Figure 3(a). While this behavior simplifies similarity measurement, MIL similarity suffers from the *sparse supervision* problem. In other words, the majority of elements are not sampled as the closest pair and thus remain untrained.

On the other hand, PCME uses the match probability (MP) as a similarity function, which is defined as $s_{\text{MP}}(\mathbf{S}_1, \mathbf{S}_2) = \sum_{x \in \mathbf{S}_1, y \in \mathbf{S}_2} \sigma(\alpha c(x, y) + \beta)$, where α and β are learnable parameters and σ is the sigmoid function. MP takes average on every pairwise distance between elements. Though MP resolves the sparse supervision, it introduces the *set collapsing* problem. As presented in Fig-

ure 3(b), directly optimizing MP leads to a collapsed embedding set since elements of embedding sets pull each other without the consideration of their relative proximity. To mitigate this issue, MP requires a specialized loss function [10, 39], which makes MP incompatible with other losses or frameworks.

The aforementioned problems stem from the way how similarity functions associate set elements. Associating only the nearest pair of elements like MIL leads to insufficient supervisory signals, while equally taking account of every possible association like MP results in a collapsed set. Unlike these similarity functions, our smooth-Chamfer similarity associates every possible pair with different degrees of weight, as illustrated in Figure 3(c). Smooth-Chamfer similarity is formulated as

$$
s_{\mathrm{SC}}(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{2\alpha|\mathbf{S}_1|} \sum_{x \in \mathbf{S}_1} \log \left(\sum_{y \in \mathbf{S}_2} e^{\alpha c(x,y)} \right)
$$
$$
+ \frac{1}{2\alpha|\mathbf{S}_2|} \sum_{y \in \mathbf{S}_2} \log \left(\sum_{x \in \mathbf{S}_1} e^{\alpha c(x,y)} \right),
\tag{6}
$$

where $|\cdot|$ is the set cardinality and $\alpha > 0$ is a scaling parameter. Using Log-Sum-Exp, it can be reformulated in a more interpretable form as shown below:

$$
s_{\mathrm{SC}}(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{2\alpha|\mathbf{S}_1|} \sum_{x \in \mathbf{S}_1} \underset{y \in \mathbf{S}_2}{\mathrm{LSE}} \left(\alpha c(x,y) \right)
$$
$$
+ \frac{1}{2\alpha|\mathbf{S}_2|} \sum_{y \in \mathbf{S}_2} \underset{x \in \mathbf{S}_1}{\mathrm{LSE}} \left(\alpha c(x,y) \right).
\tag{7}
$$

LSE indicates Log-Sum-Exp, which is the smooth approximation of the max function. Thanks to the property of LSE, smooth-Chamfer similarity softly assigns elements in one set to those in the other set, where the weight for a pair of elements is determined by their relative proximity. Hence, smooth-Chamfer similarity can provide dense supervision like MP but without the collapsing issue.

By replacing LSE with the max function, we can consider a *non-smooth* version of smooth-Chamfer similarity, which we refer to as Chamfer similarity. Chamfer similarity clears up the problem of set collapsing by assigning individual elements of $\mathbf{S}_1$ to their closest neighbor in $\mathbf{S}_2$, and vice versa. However, unlike smooth-Chamfer similarity, it does not provide supervision that is dense as MP since most associations are not considered.

The behavior of smooth-Chamfer similarity can be clearly demonstrated by its gradient with respect to $c(x, y)$, which is given by

$$
\frac{\partial s_{\mathrm{SC}}(\mathbf{S}_1, \mathbf{S}_2)}{\partial c(x', y')} = \frac{1}{2|\mathbf{S}_1|} \frac{e^{\alpha c(x', y')}}{\sum_{y \in \mathbf{S}_2} e^{\alpha c(x', y)}}
$$
$$
+ \frac{1}{2|\mathbf{S}_2|} \frac{e^{\alpha c(x', y')}}{\sum_{x \in \mathbf{S}_1} e^{\alpha c(x, y')}},
\tag{8}
$$

where x' and y' are elements of $\mathbf{S}_1$ and $\mathbf{S}_2$, respectively. One can see that the gradient is the sum of two relative similarity scores, suggesting that the gradient for $c(x', y')$ is emphasized when x' and y' are close to each other. Using the weighting scheme based on the relative proximity, we can give denser supervision during training, while preserving sufficient within-set variance.

3.4. Training and Inference

Training: Our model is trained with the objective functions presented in [45], which consists of the triplet loss, diversity regularizer, and Maximum Mean Discrepancy (MMD) [20] regularizer. Following previous work [17, 45, 50], we adopt the triplet loss with hard negative mining. Let $B = \{(S_i^{\mathcal{V}}, S_i^{\mathcal{T}})\}_{i=1}^N$ be a batch of embedding sets. Then the triplet loss is given by

$$
\mathcal{L}_{\mathrm{tri}}(B) = \sum_{i=1}^N \max_j [\delta + s_{\mathrm{SC}}(\mathbf{S}_i^{\mathcal{V}}, \mathbf{S}_j^{\mathcal{T}}) - s_{\mathrm{SC}}(\mathbf{S}_i^{\mathcal{V}}, \mathbf{S}_i^{\mathcal{T}})]_+
$$
$$
+ \sum_{i=1}^N \max_j [\delta + s_{\mathrm{SC}}(\mathbf{S}_i^{\mathcal{T}}, \mathbf{S}_j^{\mathcal{V}}) - s_{\mathrm{SC}}(\mathbf{S}_i^{\mathcal{T}}, \mathbf{S}_i^{\mathcal{V}})]_+,
\tag{9}
$$

where $[\cdot]_+$ indicates the hinge function and $\delta > 0$ is a margin. The MMD regularizer minimizes MMD between embedding sets of image and text. It prevents embeddings of different modalities from diverging at the early stage of training. The diversity regularizer penalizes similar element slots, which helps the model produce a set of diverse embeddings. The two regularizers are formulated as $\mathcal{L}_{\mathrm{mmd}} = \mathrm{MMD}(B^{\mathcal{V}}, B^{\mathcal{T}})$ and $\mathcal{L}_{\mathrm{div}} = \sum_{x,x' \in \mathbf{E}^T} e^{-2||x-x'||_2^2}$, where $B^{\mathcal{V}}$ and $B^{\mathcal{T}}$ denote subsets of batch B consisting of each modality embeddings.

Inference: We pre-compute the embedding set of size K for every sample in the database. Then, the sample most relevant to a query is retrieved via nearest-neighbor search on embedding sets, using smooth-Chamfer similarity.

4. Experiments

4.1. Datasets and Evaluation Metric

We validate the effectiveness of our method on COCO [32] and Flickr30K [41] datasets. In both datasets, we follow the split proposed by [29]. COCO dataset consists of a train split of 113,287 images, a validation split of 5,000 images, and a test split of 5,000 images. Retrieval results of our method on 1K test setting and 5K test setting are both reported, following [29]. In the 1K test setting, the average retrieval performance on the 5-fold test split is reported. For the Flickr30K dataset, we use 28,000 images for training, 1,000 images for validation, and 1,000 images for testing. We report the retrieval results on the Flick30K

Method	CA	1K Test Images							5K Test Images						
		Image-to-Text			Text-to-Image			RSUM	Image-to-Text			Text-to-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	
ResNet-152 + Bi-GRU															
VSE++ [17]	✗	64.6	90.0	95.7	52.0	84.3	92.0	478.6	41.3	71.1	81.2	30.3	59.4	72.4	355.7
PVSE [45]	✗	69.2	91.6	96.6	55.2	86.5	93.7	492.8	45.2	74.3	84.5	32.4	63.0	75.0	374.4
PCME [10]	✗	68.8	-	-	54.6	-	-	-	44.2	-	-	31.9	-	-	-
Ours	✗	70.3	91.5	96.3	56.0	85.8	93.3	**493.2**	47.2	74.8	84.1	33.8	63.1	74.7	**377.7**
Faster R-CNN + Bi-GRU															
SCAN† [30]	✓	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† [31]	✗	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
CAAN [53]	✓	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
IMRAM† [6]	✓	76.7	95.6	98.5	61.7	89.1	95.0	516.6	53.7	83.2	91.0	39.7	69.1	79.8	416.5
SGRAF† [14]	✓	79.6	96.2	98.5	63.2	90.7	96.1	524.3	57.8	-	91.6	41.9	-	81.3	-
VSE∞ [27]	✗	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
NAAF† [52]	✓	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
Ours	✗	79.8	96.2	98.6	63.6	90.7	95.7	524.6	58.8	84.9	91.5	41.1	72.0	82.4	430.7
Ours†	✗	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**
ResNeXt-101 + BERT															
VSE∞ [27]	✗	84.5	98.1	99.4	72.0	93.9	97.5	545.4	66.4	89.3	94.6	51.6	79.3	87.6	468.9
VSE∞ † [27]	✗	85.6	98.0	99.4	73.1	94.3	97.7	548.1	68.1	90.2	95.2	52.7	80.2	88.3	474.8
Ours	✗	86.3	97.8	99.4	72.4	94.0	97.6	547.5	69.1	90.7	95.6	52.1	79.6	87.8	474.9
Ours†	✗	86.6	98.2	99.4	73.4	94.5	97.8	**549.9**	71.0	91.8	96.3	53.4	80.9	88.6	**482.0**

Table 1. Recall@K (%) and RSUM on the COCO dataset. Evaluation results on both 1K test setting (average of 5-fold test dataset) and 5K test setting are presented. The best RSUM scores are marked in bold. CA and † indicate models using cross-attention and ensemble models of two hypotheses, respectively.

Method	CA	Image-to-text			Text-to-image			RSUM
		R@1	R@5	R@10	R@1	R@5	R@10	
ResNet-152 + Bi-GRU								
VSE++	✗	52.9	80.5	87.2	39.6	70.1	79.5	409.8
PVSE*	✗	59.1	84.5	91.0	43.4	73.1	81.5	432.6
PCME*	✗	58.5	81.4	89.3	44.3	72.7	81.9	428.1
Ours	✗	61.8	85.5	91.1	46.1	74.8	83.3	**442.6**
Faster R-CNN + Bi-GRU								
SCAN†	✓	67.4	90.3	95.8	48.6	77.7	85.2	465.0
VSRN†	✗	71.3	90.6	96.0	54.7	81.8	88.2	482.6
CAAN	✓	70.1	91.6	97.2	52.8	79.0	87.9	478.6
IMRAM†	✓	74.1	93.0	96.6	53.9	79.4	87.2	484.2
SGRAF†	✓	77.8	94.1	97.4	58.5	83.0	88.8	499.6
VSE∞	✗	76.5	94.2	97.7	56.4	83.4	89.9	498.1
NAAF†	✓	81.9	96.1	98.3	61.0	85.3	90.6	**513.2**
Ours	✗	77.8	94.0	97.5	57.5	84.0	90.0	500.8
Ours†	✗	80.9	94.7	97.6	59.4	85.6	91.1	509.3
ResNeXt-101 + BERT								
VSE∞	✗	88.4	98.3	99.5	74.2	93.7	96.8	550.9
VSE∞ †	✗	88.7	98.9	99.8	76.1	94.5	97.1	555.1
Ours	✗	88.8	98.5	99.6	74.3	94.0	96.7	551.9
Ours†	✗	90.6	99.0	99.6	75.9	94.7	97.3	**557.1**

Table 2. Recall@K(%) and RSUM on the Flickr30K dataset. CA, †, and * indicate models using cross-attention, ensemble models of two hypotheses, and models we reproduce, respectively.

using a test split of 1,000 images. In both datasets, each image is given with five matching captions. For evaluation, we use the Recall@K, which is the percentage of the queries that have matching samples among top-K retrieval results. Following [27], We also report the RSUM, which is the sum of the Recall@K at $K \in \{1, 5, 10\}$ in the image-to-text and text-to-image retrieval settings.

4.2. Implementation Details

Feature extractor: For the visual feature extractor, convolutional visual features are obtained by applying 1×1 convolution to the last feature map of CNN. We obtain ROI visual features by feeding the pre-extracted features [1] from a Faster R-CNN [42] to the 2-layer MLP with residual connection, following [27]. In every model, we set D to 1024 and K to 4.

Set prediction module: In every experiment, we set T to 4. When using convolutional visual features, D_h is set to 1024 and otherwise set to 2048.

Similarity and loss function: For smooth-Chamfer similarity, we set the scaling parameter α to 16. In $\mathcal{L}_{\text{tri}}$, we use margin δ of 0.1 for convolutional visual feature and 0.2 for ROI visual feature. We multiply the factor of 0.01 to $\mathcal{L}_{\text{div}}$ and $\mathcal{L}_{\text{mmd}}$.

Training: The model is trained with the AdamW optimizer [35]. We construct the input batch with 200 images and all of their captions. When using ROI visual features, the model is trained for 80 epochs with the initial learning rate of 1e-3, which is scheduled with cosine annealing [34]. For the models employing convolutional visual features, we follow the training setting of previous work [27, 45], which will be discussed in detail in the supplementary material.

4.3. Comparisons with Other Methods

Results on COCO and Flickr30K are summarized in Table 1 and Table 2, respectively. For fair comparisons with

Figure 4. For each element of the image embedding set, we present its attention map and the caption nearest to the element in the embedding space. Matching captions are colored in green. Entities corresponding to the attention maps are underlined.

previous work, our method is evaluated under three different visual extractors: ResNet-152 [22] pretrained on ImageNet [11], ROI features [1] pre-extracted by Faster R-CNN [42], and ResNeXt-101 [51] pretrained on the Instagram [36] dataset. For ResNet-152 and ResNeXt-101, input image resolution is set to respectively 224×224 and 512×512, following previous work [27, 45]. For the textual feature extractor, we use either bi-GRU or BERT. The ensemble results are obtained by averaging the similarity scores of two models trained with different random seeds.

Our method outperforms previous methods in terms of RSUM in every setting except for Flickr30K with ROI visual features. Even in this setting, ours achieves the second-best. Note that NAAF, currently the best in this setting, requires two orders of magnitude heavier computations than ours at inference since it relies on cross-attention, as will be shown in the next section. The ensemble of ours with ResNeXt-101 and BERT clearly outperforms all existing records, improving the previous best RSUM by 7.2%p and 2.0%p on COCO 5K and Flickr30K. Comparisons with PVSE and PCME demonstrate the superiority of our set-based embedding framework. Our model outperforms both of them in every case. Specifically, our model improves RSUM by 3.3%p on COCO 5K and 10.0%p on Flickr30K, compared to PVSE. Also, our method shows better results than models involving cross-attention networks [6, 14, 30], which impose substantial computation as noted. We finally emphasize that ours using a single model often outperforms previous work using ensemble [6, 14, 30, 31].

4.4. Computation Cost Analysis

To demonstrate the efficiency of our model during evaluation time, we measure FLOPs required for computing a similarity score between representations of an image and a caption. Ours demands 16.4K FLOPs, while SCAN [30], a representative cross-attention method, requires 1.24M FLOPs. This result shows that our method demands about 80 times less floating-point operations than SCAN since the cross-attention requires re-processing of the image or caption representations for each query. Also, it is worth noting that our method outperforms SCAN in terms of RSUM, as shown in Table 1 and 2.

4.5. Analysis of Attention

In Figure 4, attentions from the last aggregation block in the $f^{\mathcal{V}}$ are visualized. Each attention is used to encode individual elements of the image embedding set. For each element, its nearest caption in the embedding space is presented together. Thanks to slot attention, the set prediction module produces heterogeneous attention maps that focus on individual objects or contexts, which enable retrieving diverse captions. In particular, on the top-right, elements are matched with the captions that describe different semantics (helmet, children, park, and bicycle). Interestingly, we observed that one of the element slots attends to regions left out after other slots capture distinctive semantics, which wholly charges for retrieving related but highly abstract captions.

4.6. Ablation Study

We perform ablation studies to investigate the contributions of the proposed similarity function and set prediction module, in terms of RSUM and within-set variance. For all ablation experiments, we use the models employing ROI visual features on Flickr30K.

Importance of set similarity function: In the left of Ta-

Similarity	Arch.	RSUM
MIL	Ours	491.7
MP	Ours	490.5
Ours (Chamfer)	Ours	499.6
Ours (S-Chamfer)	PIE-Net	483.3
Ours (S-Chamfer)	Ours	**500.8**

Setting	log(Var.)	RSUM
PIE-Net	-7.35	483.3
Ours \w MP	-5.27	490.5
Transformer	-2.27	496.1
Ours	-2.13	**500.8**

Table 3. Ablation studies of the proposed similarity function and set prediction module.

K	1	2	3	4	5	6
RSUM	492.6	495.5	497.4	**500.8**	498.4	499.3

T	1	2	3	4	5	6
RSUM	492.6	497.8	499.3	**500.8**	499.4	499.1

α	1	2	4	8	16	32	64
RSUM	495.2	497.2	497.7	498.9	**500.8**	499.3	499.0

Table 4. Impact of hyperparameters for the proposed similarity function and set prediction module.

ble 3, we ablate the proposed similarity function and replace it with the MP, MIL, and Chamfer similarity. The results show that the models trained with smooth-Chamfer similarity surpass those trained with MIL or MP similarity under the same set prediction architecture. Though Chamfer similarity improves the performance by resolving set collapsing, smooth-Chamfer similarity further improves performance by providing dense supervision during training.

Importance of set prediction module: We also ablate the proposed set prediction module and substitute it with the PIE-Net, which is the baseline set prediction architecture of the previous set-based embedding methods, PVSE and PCME. The results in the left of Table 3 suggest that the impact of our set prediction module is significant.

Verification of set collapsing: In the right of Table 3, we report the average circular variance of embedding set. Formally, the circular variance of embedding set $\mathbf{S}$ is denoted as $1 - \|\sum_{x \in \mathbf{S}} x/|\mathbf{S}|\|_2$. As discussed earlier, results show that employing PIE-Net or MP leads to set collapsing. Moreover, we report the result when aggregation block is replaced to transformer. This variant results in a 13% lower variance on a linear scale and inferior performance since it does not guarantee disentanglement between elements.

4.7. Impact of Hyperparameters

In Table 4, we report the RSUM of the model while varying the cardinality of embedding sets K, the total number of iterations T, and the scaling parameter α. Results demonstrate that the model shows consistently high RSUM when $K > 3$, $T > 2$, and $\alpha > 4$. Specifically, we notice that when $K = 1$, which is identical to using a single embedding vector, the model shows significantly lower accuracy compared to others. It suggests that embedding sets enable a more accurate retrieval by addressing the semantic ambiguity. When $T = 1$, we observe substantial degradation of accuracy, which underpins that progressive refinement done

Evaluation				RSUM
$\mathbf{S}^{\mathcal{V}}(1)$	$\mathbf{S}^{\mathcal{V}}(2)$	$\mathbf{S}^{\mathcal{V}}(3)$	$\mathbf{S}^{\mathcal{V}}(4)$	
✓	✓	✓	✓	**500.8**
✓				491.1
	✓			309.6
		✓		484.9
			✓	486.0
✓		✓	✓	500.2

Evaluation				RSUM
$\mathbf{S}^{\mathcal{T}}(1)$	$\mathbf{S}^{\mathcal{T}}(2)$	$\mathbf{S}^{\mathcal{T}}(3)$	$\mathbf{S}^{\mathcal{T}}(4)$	
✓	✓	✓	✓	**500.8**
✓				481.9
	✓			483.0
		✓		481.7
			✓	497.2

Table 5. RSUM on the Flickr30K dataset when elements of the $\mathbf{S}^{\mathcal{V}}$(*left*) and $\mathbf{S}^{\mathcal{T}}$(*right*) are ablated during evaluation.

by multiple aggregation blocks helps element slots aggregate semantic entities, as shown in Figure 2.

4.8. Analysis of Embedding Set Elements

For further analysis of the embedding set elements, we use only one of the elements during the evaluation of the model. Let $\mathbf{S}(i)$ be an element of $\mathbf{S}$ produced by the i-th element slot. Table 5 summarizes the accuracy of the model in terms of RSUM when elements of $\mathbf{S}^{\mathcal{V}}$ and $\mathbf{S}^{\mathcal{T}}$ are ablated during evaluation. Results present that using only one of the elements degrades accuracy. Specifically, using only $\mathbf{S}^{\mathcal{V}}(2)$ during evaluation leads to far lower accuracy compared to using the other elements. To check whether $\mathbf{S}^{\mathcal{V}}(2)$ is just a noisy element that hinders final performance, we also report RSUM when only $\mathbf{S}^{\mathcal{V}}(2)$ is ablated. However, interestingly, we still observed the degraded accuracy compared to using a complete embedding set. Though one of the elements often retrieves non-matching samples, the results demonstrate that using them together during evaluation helps the model find accurate matching samples.

We hypothesize that $\mathbf{S}^{\mathcal{V}}(2)$ is trained to capture highly ambiguous semantics. Therefore, using them alone during evaluation leads to less accurate retrievals, while it improves performance when used together by successfully representing ambiguous situations and context. A similar tendency could be observed in Figure 4, where one of the slots often encodes an element that is located nearby to a related but highly ambiguous caption in the embedding space.

5. Conclusion

We propose the novel set embedding framework for cross-modal retrieval, consisting of the set prediction module and smooth-Chamfer similarity. The proposed set prediction module outputs an embedding set that successfully captures ambiguity, while smooth-Chamfer similarity resolves undesirable effects of the existing similarity functions. As a result, our model surpasses most of the previous methods, including methods involving higher computation costs, on COCO and Flickr30K datasets. We will extend our model for the different modalities and tasks in the future.

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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6, 7

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[4] Michael Chang, Thomas L. Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative inference algorithms with implicit differentiation. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. 3

[5] Gal Chechik, Eugene Ie, Martin Rehn, Samy Bengio, and Dick Lyon. Large-scale content-based audio retrieval from text queries. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, 2008. 1

[6] Hui Chen, Guiguang Ding, Xudong Liu, Zijia Lin, Ji Liu, and Jungong Han. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7

[7] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 3

[9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 1

[10] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 4, 5, 6

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 7

[12] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. North Ameri-*

[14] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 1, 2, 6, 7

[15] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. 4

[16] Benjamin Elizalde, Shuayb Zarar, and Bhiksha Raj. Cross modal audio search and retrieval with joint embeddings based on text and audio. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019. 1

[17] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *Proc. British Machine Vision Conference (BMVC)*, 2018. 1, 2, 5, 6

[18] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2013. 1

[19] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 1

[20] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2006. 5

[21] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7181–7189, 2018. 1, 2

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[23] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[24] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[25] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. In *arXiv preprint arXiv:2004.00849*, 2020. 1

[26] Boseung Jeong, Jicheol Park, and Suha Kwak. Asmr: Learning attribute-based person search with adaptive semantic margin regularizer. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 12016–12025, 2021. 1

[27] Chen Jiacheng, Hu Hexiang, Wu Hao, Jiang Yuning, and Wang Changhu. Learning the best pooling strategy for visual semantic embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3, 6,

7

[28] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[29] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 5

[30] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 6, 7

[31] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 6, 7

[32] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2014. 2, 5

[33] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3

[34] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proc. International Conference on Learning Representations (ICLR)*, 2017. 6

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2019. 6

[36] 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 *Proc. European Conference on Computer Vision (ECCV)*, 2018. 7

[37] Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[38] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[39] Seong Joon Oh, Andrew C. Gallagher, Kevin P. Murphy, Florian Schroff, Jiyan Pan, and Joseph Roth. Modeling uncertainty with hedged instance embeddings. In *International Conference on Learning Representations*, 2019. 5

[40] 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. 3

[41] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazeb-nik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 5

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 6, 7

[43] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[44] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[45] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4, 5, 6, 7

[46] Christopher Thomas and Adriana Kovashka. Emphasizing complementary samples for non-literal cross-modal retrieval. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022. 1

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2017. 4

[48] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 2

[49] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. Camp: Cross-modal adaptive message passing for text-image retrieval. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[50] Jiwei Wei, Xing Xu, Yang Yang, Yanli Ji, Zheng Wang, and Heng Tao Shen. Universal weighting metric learning for cross-modal matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 5

[51] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[52] Kun Zhang, Zhendong Mao, Quan Wang, and Yongdong Zhang. Negative-aware attention framework for image-text matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6

[53] Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z Li. Context-aware attention network for image-text retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3536–3545, 2020. 1, 2, 6