Multi-View Visual Semantic Embedding

Zheng Li[1] , **Caili Guo**[1,2*] , **Zerun Feng**[1] , **Jenq-Neng Hwang**[3] and **Xijun Xue**[4]
[1]Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications
[2]Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications
[3]University of Washington
[4]China Telecom System Integration Co.,Ltd
{lizhengzachary, guocaili, fengzerun}@bupt.edu.cn, hwang@uw.edu, xuexj@chinatelecom.cn

Abstract

Visual Semantic Embedding (VSE) is a dominant method for vision-language retrieval. Its purpose is to learn an embedding space so that visual data can be embedded in a position close to the corresponding text description. However, there are large intra-class variations in the vision-language data. For example, multiple texts describing the same image may be described from different views, and the descriptions of different views are often dissimilar. The mainstream VSE method embeds samples from the same class in similar positions, which will suppress intra-class variations and lead to inferior generalization performance. This paper proposes a Multi-View Visual Semantic Embedding (MV-VSE) framework, which learns multiple embeddings for one visual data and explicitly models intra-class variations. To optimize MV-VSE, a multi-view upper bound loss is proposed, and the multi-view embeddings are jointly optimized while retaining intra-class variations. MV-VSE is plug-and-play and can be applied to various VSE models and loss functions without excessively increasing model complexity. Experimental results on the Flickr30K and MS-COCO datasets demonstrate the superior performance of our framework.

1 Introduction

Cross-modal vision-language retrieval task is formulated as retrieving relevant samples across different visual and textual modalities, which has a variety of applications such as image-text retrieval [Faghri *et al.*, 2018; Zhang *et al.*, 2022], video-text retrieval [Feng *et al.*, 2020; Li *et al.*, 2020], *etc*.

Visual Semantic Embedding (VSE) is a dominant method for vision-language retrieval. Its purpose is to learn an embedding space so that visual data can be embedded in a position close to the corresponding text description. Following [Chen *et al.*, 2021], we divide VSE into three steps:

Step 1. Use *feature extractors* to extract a set (or sequence) of features from visual (or textual) data.

Step 2. Aggregate the extracted feature set into a feature vector and project it into the joint embedding space using *feature aggregators*.

Step 3. Calculate the matching score between the embeddings with a similarity metric.

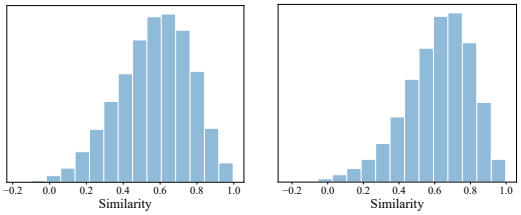
The embedding space learned by mainstream VSE methods is usually highly discriminative, which encourages small *intra-class variations*, that is, the embeddings of an image and its corresponding text description will be very close, and large *inter-class variations*, that is, unpaired images and texts embeddings are mapped to locations that are farther away.

Figure 1: Multi-view descriptions for a given image in Flickr30K.

However, there are large intra-class variations in the data used for vision-language retrieval. In vision-language datasets, a same image usually has multiple text descriptions, and these text descriptions may be described from different views. As shown in Figure 1, the two text descriptions of the same image have different concerns. View 1 pays more attention to people on the street, and view 2 only describes the man working a hotdog stand.

The intra-class variations of vision-language data are mainly caused by multi-view text descriptions. Therefore, we quantitatively analyze the intra-class variations of text descriptions on the two image-text retrieval benchmark datasets. We use Sentence-BERT [Reimers and Gurevych, 2019] to calculate the semantic similarities between different text descriptions of the same image and count the data distribution of the similarities. Sentence-BERT has nearly reached human performance on the sentence similarity task, and the statistical results are reliable. Figure 2 shows the data distribution of intra-class text similarities on Flickr30K [Young *et al.*, 2014] and MS-COCO [Lin *et al.*, 2014]. It can be seen that the sim-

*Corresponding author



(a) Flickr30K. (b) MS-COCO.

Figure 2: The distribution of intra-class text similarity on Flickr30K and MS-COCO.

ilarities are concentrated in $0.4 \sim 0.8$, and even a small part of similarities are less than 0. The intra-class variations of vision-language data are widespread, but most of the existing work ignores this problem.

Most VSE methods learn an embedding for each visual or textual data and map each visual data and corresponding text descriptions to similar locations, which will suppress intra-class variations and lead to inferior generalization performance. Several recent works [Xuan *et al.*, 2020; Zheng *et al.*, 2021a] have verified that even samples from the same class may show different patterns and characteristics. Suppressing the intraclass variations will undermine the ability of the learned model to generalize to unseen classes. In recent VSE works, [Qu *et al.*, 2020] propose context-aware multi-view summarization (CAMERA) network pays attention to multi-view issues. However, CAMERA only designs a multi-view summarization network structure for bottom-up attention features [Anderson *et al.*, 2018], and it is not easy to extend to other VSE models. In addition, the loss function of CAMERA only selects the view with the largest matching score for optimization, and a large number of view branches may not be optimized, and its optimization goal is easy to fall into the local minima.

To explicitly model intra-class variations and improve the generalization of the model, this paper proposes a Multi-View Visual Semantic Embedding (MV-VSE) framework, which uses multiple aggregators to learn multi-view embeddings for one visual data. MV-VSE shares the feature extractor, only a few additional feature aggregators need to be trained, which will not increase the complexity of the model excessively. To optimize multiple views at the same time and keep intra-class variations, a multi-view upper bound loss is proposed. MV-VSE is an extension of the existing VSE model and only requires additional training of multiple feature aggregators. Existing loss functions can also be rewritten into a multi-view upper bound version by our method. Therefore, MV-VSE is plug-and-play and can be applied to various VSE models and loss functions. The major contributions of this paper are summarized as follows:

- A novel Multi-View Visual Semantic Embedding (MV-VSE) framework is proposed to explicitly model intra-class variations and improve the generalization ability of the model, which can be applied to various mainstream VSE models.
- A multi-view upper bound loss is proposed, which can modify the existing losses so that the learned multiple embedding spaces maintain intra-class variations and is not easy to fall into local minima.
- We conduct extensive experiments on image-text retrieval. Experimental results demonstrate that MV-VSE yields compelling performance on the two widely used benchmark datasets: Flickr30K and MS-COCO, reflecting the successful modeling of intra-class variations and the improvement of the generalization of MV-VSE.

2 Related Works

Cross-modal vision-language retrieval has a variety of applications, such as image-text retrieval [Faghri *et al.*, 2018; Zhang *et al.*, 2022], video-text retrieval [Feng *et al.*, 2020; Li *et al.*, 2020], *etc*. The existing vision-language retrieval methods can be divided into two categories according to the cross-modal matching method, the Visual Semantic Embedding (VSE) method [Faghri *et al.*, 2018; Li *et al.*, 2019] and the cross-attention method [Lee *et al.*, 2018].

Visual Semantic Embedding. VSE method embeds the whole visual samples and text into a joint embedding space, and the matching score between samples can be calculated by a simple similarity metric (*e.g.* cosine similarity). [Frome *et al.*, 2013] propose the first VSE model DeViSE, which employs the CNN and Skip-Gram to project images and texts into a joint embedding space, and adopts a hinge-based triplet loss to optimize the model. [Faghri *et al.*, 2018] introduce online hard-negative mining in the triplet loss, which yields significant gains in retrieval performance. [Chen *et al.*, 2021] propose a Generalized Pooling Operator (GPO), which learns to automatically adapt itself to the best pooling strategy for different features. Although the VSE methods learn the inter-class variations well and achieve promising performance, the intra-class variations in the data are also not negligible.

Cross-Attention. The cross-attention method obtains the matching score by calculating the cross-attention between visual local features (*e.g.* bottom-up attention region features [Anderson *et al.*, 2018]) and text local features (*e.g.* word embeddings). [Lee *et al.*, 2018] propose a stacked cross attention network, which measures the image-text similarity by aligning image regions and words. Recently, a number of transformer-based methods [Lu *et al.*, 2019; Chen *et al.*, 2020; Kim *et al.*, 2021] use cross-modal attention to learn rich cross-modal interactions. The cross-attention method can preserve intra-class variations to a certain extent. However, in the inference stage, the cross-attention method needs to calculate cross-attention on all visual and textual data. The inference efficiency is several orders of magnitude lower than that of the VSE method, and it is not suitable for large-scale vision-language retrieval. MV-VSE proposed in this paper is an extension of existing VSE methods and has similar inference efficiency to VSE.

Intra-Class Variations Modeling. There are a number of works [Sanakoyeu *et al.*, 2019; Zheng *et al.*, 2021a; Zheng *et al.*, 2021b] to model intra-class variations through ensemble learning. Ensemble learning improves the generalization ability of the model by learning a set of sub-embeddings. But

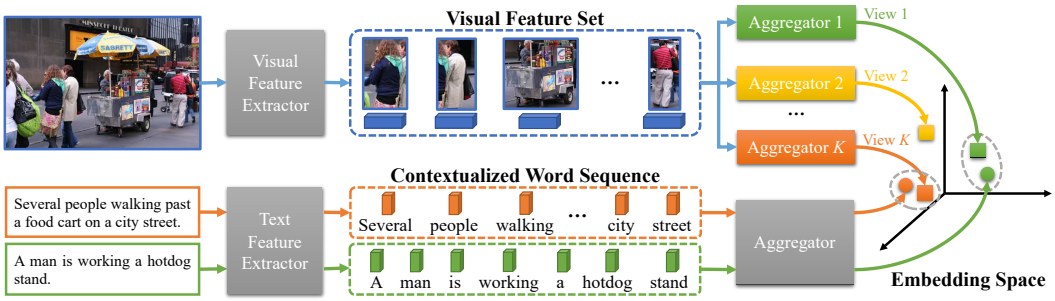


Figure 3: Illustration of the proposed Multi-View Visual Semantic Embedding (MV-VSE) Framework.

these works are concentrated in the field of unimodal image retrieval. In cross-modal retrieval, the modalities of query and candidate are different. Text is a local description with a subjective view, while visual data contain global information. The information between two modalities is asymmetry, and the ensemble learning method in unimodal retrieval cannot be used directly. In the field cross-modal retrieval, [Qu *et al.*, 2020] propose CAMERA network pays attention to multi-view issues. But CAMERA is not universal enough to be easily extended to other VSE models. In addition, the loss function of CAMERA does not make a targeted design for the multi-view problem, and its optimization goal is easy to fall into the local minima. This paper proposes a plug-and-play MV-VSE framework and specially designs a multi-view upper bound loss for the optimization of multi-view learning.

3 Multi-View Visual Semantic Embedding

3.1 Visual Semantic Embedding

We first briefly review the core steps of the VSE. A VSE model uses a visual feature extractor $\mathbf{\Phi}(\cdot)$ such as convolutional neural networks (*e.g.* ResNet [He *et al.*, 2016], Faster R-CNN [Ren *et al.*, 2015]) to extract a set of visual features from a visual data V, and a text feature extractor $\mathbf{\Psi}(\cdot)$ such as sequence models (*e.g.* GRU, Transformer [Vaswani *et al.*, 2017]) to extract a sequence of text features from a text T, respectively:

$$\{\boldsymbol{\phi}_n\}_{n=1}^N = \mathbf{\Phi}(V), \{\boldsymbol{\psi}_m\}_{m=1}^M = \mathbf{\Psi}(T), \tag{1}$$

where the visual feature set $\{\boldsymbol{\phi}_n\}_{n=1}^N$ has N elements of visual local representations with $\boldsymbol{\phi}_n \in \mathbb{R}^{d_1}$, and $\{\boldsymbol{\psi}_m\}_{m=1}^M$ denotes a sequence of M contextualized word token features out of a sequence model where M is the number of words and $\boldsymbol{\psi}_m \in \mathbb{R}^{d_2}$. Here d_1 and d_2 are the feature dimensions.

Then a visual aggregator $\mathbf{f}_v(\cdot)$ and a text aggregator $\mathbf{f}_t(\cdot)$ such as GPO [Chen *et al.*, 2021], aggregate $\{\boldsymbol{\phi}_n\}_{n=1}^N$ and $\{\boldsymbol{\psi}_m\}_{m=1}^M$ into visual and text embedding $\mathbf{v}, \mathbf{t} \in \mathbb{R}^{d_3}$, respectively:

$$\mathbf{v} = \mathbf{f}_v\left(\{\boldsymbol{\phi}_n\}_{n=1}^N\right), \mathbf{t} = \mathbf{f}_t\left(\{\boldsymbol{\psi}_m\}_{m=1}^M\right). \tag{2}$$

d_3 is the embedding dimensions.

Finally, the matching score between embeddings is calculated, usually using cosine similarity:

$$s(V,T) = s(\mathbf{v},\mathbf{t}) = \frac{\mathbf{v}^\top \mathbf{t}}{\|\mathbf{v}\| \cdot \|\mathbf{t}\|}. \tag{3}$$

In the inference stage, $s(V,T)$ is used to rank the candidates to get the retrieval results. VSE method has high inference efficiency, but it embeds the visual data and corresponding text descriptions in a similar position, which will suppress intra-class variations and lead to inferior generalization performance.

3.2 Multi-View Visual Semantic Embedding Framework

To explicitly model intra-class variations, we propose a Multi-View Visual Semantic Embedding (MV-VSE) framework, as shown in Figure 3. MV-VSE is an extension of the VSE. Its textual embeddings and visual feature extractors are common to VSE, so MV-VSE can be easily applied to existing VSE models. The special feature of MV-VSE is that it learns multiple embeddings for each visual data, thereby explicitly modeling intra-class variations.

After obtaining a visual feature set $\{\boldsymbol{\phi}_n\}_{n=1}^N$, a set of feature aggregators $\{\mathbf{f}_v^k(\cdot)\}_{k=1}^K$ are used to aggregate $\{\boldsymbol{\phi}_n\}_{n=1}^N$ into a set of visual embeddings $\{\mathbf{v}_k\}_{k=1}^K$:

$$\mathbf{v}_k = \mathbf{f}_v^k\left(\{\boldsymbol{\phi}_n\}_{n=1}^N\right), \tag{4}$$

where $\mathbf{v}_k \in \mathbb{R}^{d_3}$. Each $\mathbf{v}_k$ represents a view. K is the number of views. Each feature aggregator learns an embedding subspace for a visual data. MV-VSE learns K subspaces in total.

Then calculate the matching scores between the visual embeddings $\{\mathbf{v}_k\}_{k=1}^K$ of the K views and the text embedding $\mathbf{t}$:

$$s(\mathbf{v}_k,\mathbf{t}) = \frac{\mathbf{v}_k^\top \mathbf{t}}{\|\mathbf{v}_k\| \cdot \|\mathbf{t}\|}, \tag{5}$$

and take the largest score among the K views as the final matching score:

$$s^*(V,T) = \max_{k=1}^K s(\mathbf{v}_k,\mathbf{t}). \tag{6}$$

During the inference, $s^*(V,T)$ is used to rank the candidates to get the retrieval results. MV-VSE learns multiple embeddings for each visual data, retains intra-class variations, and also has the efficiency of VSE method inference. Compared with VSE, MV-VSE only needs to train $K-1$ additional feature aggregators. Both the text embedding network and the visual feature extractor are multiplexed, which will only increase the number of negligible parameters. Taking the

latest VSE model GPO [Chen *et al.*, 2021] as an example, the parameter amount of the feature aggregator is only 0.1 *M*, which is less than 1% of the entire model.

3.3 Multi-View Upper Bound Loss

Existing losses used in VSE models are designed for a single visual embedding, and these losses directly used in MV-VSE will suppress intra-class variations.

Take the most commonly used hinge-based triplet ranking loss with online hard negative mining [Faghri *et al.*, 2018] as an example. To learn a VSE model, the concrete optimization objective is defined by:

$$\begin{aligned}\mathcal{L}_{\text{Tri}} =&[\alpha - s(V,T) + s(V,\hat{T})]_+ \\ &+[\alpha - s(V,T) + s(\hat{V},T)]_+,\end{aligned} \tag{7}$$

where α is a fixed margin, (V,T) is a positive image-text pair. $\hat{T} = \arg\max_{T' \neq T} s(V,T')$ and $\hat{V} = \arg\max_{V' \neq V} s(V',T)$ denote as the hardest negative text and image samples measured by the VSE model within a mini-batch.

The triplet loss can be directly used for the optimization of K views:

$$\begin{aligned}\mathcal{L}_{\text{MV-Avg-Tri}} = \frac{1}{K}\sum_{k=1}^{K}\{&[\alpha - s(\mathbf{v}_k,\mathbf{t}) + s(\mathbf{v}_k,\hat{\mathbf{t}})]_+ \\ &+ [\alpha - s(\mathbf{v}_k,\mathbf{t}) + s(\hat{\mathbf{v}}_k,\mathbf{t})]_+\},\end{aligned} \tag{8}$$

where $(\mathbf{v}_k,\mathbf{t})$ is a positive image-text pair in k^{th} view. $\hat{\mathbf{t}} = \arg\max_{\mathbf{t}' \neq \mathbf{t}} s(\mathbf{v}_k,\mathbf{t}')$ and $\hat{\mathbf{v}}_k = \arg\max_{\mathbf{v}'_k \neq \mathbf{v}_k} s(\mathbf{v}'_k,\mathbf{t})$ denote as the hardest negative text and image samples measured by the MV-VSE model within a mini-batch in K^{th} view. However, $\mathcal{L}_{\text{MV-Avg-Tri}}$ will bring all visual embeddings $\{\mathbf{v}_k\}_{k=1}^K$ closer to $\mathbf{t}$, suppressing intra-class variations. In addition, each view is the same optimization goal, which will make the learned multiple visual embeddings lack diversity.

In order to optimize multi-view visual embeddings, [Qu *et al.*, 2020] directly substitute the maximum matching score $s^*(V,T)$ in K views into the triplet loss:

$$\begin{aligned}\mathcal{L}_{\text{MV-Max-Tri}} =&[\alpha - s^*(V,T) + s^*(V,\hat{T})]_+ \\ &+[\alpha - s^*(V,T) + s^*(\hat{V},T)]_+.\end{aligned} \tag{9}$$

$\mathcal{L}_{\text{MV-Max-Tri}}$ is consistent with the design of MV-VSE. However, taking $\mathcal{L}_{\text{MV-Max-Tri}}$ as the optimization target of MV-VSE, for a visual data V, only one feature aggregator will be optimized for each backpropagation, which is easy to fall into the local minima. Figure 4 (c) in Section 4 will verify this.

To simultaneously optimize all view branches and retain intra-class variations, we propose a multi-view upper bound loss. The specific definition of the loss is:

$$\begin{aligned}\mathcal{L}_{\text{MV-Up-Tri}} = \frac{1}{K}\sum_{k=1}^{K}\Big\{&[\alpha - s(\mathbf{v}_k,\mathbf{t}) + s^*(V,\hat{T})]_+ \cdot \mathbb{1}(V,\hat{T}) \\ &+ [\alpha - s(\mathbf{v}_k,\mathbf{t}) + s^*(\hat{V},T)]_+ \cdot \mathbb{1}(\hat{V},T)\Big\},\end{aligned} \tag{10}$$

where

$$\mathbb{1}(V,\hat{T}) = \begin{cases}1, & \forall \mathbf{v}_k, [\alpha - s(\mathbf{v}_k,\mathbf{t}) + s^*(V,\hat{T})] > 0, \\ 0, & \text{otherwise},\end{cases} \tag{11}$$

$$\mathbb{1}(\hat{V},T) = \begin{cases}1, & \forall \mathbf{v}_k, [\alpha - s(\mathbf{v}_k,\mathbf{t}) + s^*(\hat{V},T)] > 0, \\ 0, & \text{otherwise}.\end{cases} \tag{12}$$

$\mathcal{L}_{\text{MV-Up-Tri}}$ is a strict upper bound of $\mathcal{L}_{\text{MV-Max-Tri}}$. In order to facilitate understanding, we introduce a rough upper bound of $\mathcal{L}_{\text{MV-Max-Tri}}$:

$$\begin{aligned}\mathcal{L}_{\text{MV-Up-Rough-Tri}} = \frac{1}{K}\sum_{k=1}^{K}\Big\{&[\alpha - s(\mathbf{v}_k,\mathbf{t}) + s^*(V,\hat{T})]_+ \\ &+ [\alpha - s(\mathbf{v}_k,\mathbf{t}) + s^*(\hat{V},T)]_+\Big\}.\end{aligned} \tag{13}$$

Since $s^*(V,T) \geq s(\mathbf{v}_k,\mathbf{t})$, it's easy to get $\mathcal{L}_{\text{MV-Max-Tri}} \leq \mathcal{L}_{\text{MV-Up-Rough-Tri}}$. $\mathcal{L}_{\text{MV-Up-Rough-Tri}}$ can optimize all view branches, but it still brings all visual embeddings $\{\mathbf{v}_k\}_{k=1}^K$ closer to $\mathbf{t}$, suppressing intra-class variations. Therefore, we introduce indicator functions $\mathbb{1}(V,\hat{T})$ and $\mathbb{1}(\hat{V},T)$. When $\mathbb{1}(V,\hat{T}) = 1$, all visual embeddings do not meet the constraint of triplet loss, and all view branches will be optimized. $\mathbb{1}(V,\hat{T}) = 0$ means that at least one visual embedding satisfies the constraint of triplet loss, and the other branches no longer bring $\mathbf{v}_k$ and $\mathbf{t}$ closer, keeping the intra-class variations.

When $[\alpha - s^*(V,T) + s^*(V,\hat{T})] > 0$, $\mathbb{1}(V,\hat{T}) = 1$. Similarly, when $[\alpha - s^*(V,T) + s^*(\hat{V},T)] > 0$, $\mathbb{1}(\hat{V},T) = 1$. Therefore, we can get:

$$\mathcal{L}_{\text{MV-Max-Tri}} \leq \mathcal{L}_{\text{MV-Up-Tri}} \leq \mathcal{L}_{\text{MV-Up-Rough-Tri}}. \tag{14}$$

$\mathcal{L}_{\text{MV-Up-Tri}}$ is the tighter upper bound of $\mathcal{L}_{\text{MV-Max-Tri}}$, which can simultaneously optimize all view branches and retain intra-class variations.

Diversity of views is essential to improve the generalization ability of the model. Optimizing MV-VSE using only $\mathcal{L}_{\text{MV-Up-Tri}}$ will make multiple views too similar, so we optimize $\mathcal{L}_{\text{MV-Max-Tri}}$ and $\mathcal{L}_{\text{MV-Up-Tri}}$ jointly. When optimizing $\mathcal{L}_{\text{MV-Max-Tri}}$, only one view branch is optimized at a time, which naturally introduces the diversity between views. The overall objective of the MV-VSE can be formulated as follows:

$$\mathcal{L}_{\text{MV-VSE-Tri}} = \lambda\mathcal{L}_{\text{MV-Max-Tri}} + (1-\lambda)\mathcal{L}_{\text{MV-Up-Tri}}, \tag{15}$$

where λ is a parameter to balance balance the contributions of the two losses, which can control the diversity of views.

Note that other losses (*e.g.* normalized softmax loss [Zhai and Wu, 2019]) can also be converted to a multi-view version using our method.

4 Experiments

4.1 Datasets and Evaluation Metric

We evaluate our method on two standard benchmarks: Flickr30K [Young *et al.*, 2014] and MS-COCO [Lin *et al.*, 2014]. Flickr30K dataset contains 31,000 images, each image is annotated with 5 sentences. Following the data split of [Faghri *et al.*, 2018], we use 1,000 images for validation, 1,000 images for testing, and the remaining for training. MS-COCO

Data Split	Flickr30K 1K Test							COCO 5-fold 1K Test						
Eval Task	Image-to-Text			Text-to-Image			RSUM	Image-to-Text			Text-to-Image			RSUM
Method	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 + RNN														
VSE++ [Faghri *et al.*, 2018]	52.9	80.5	87.2	39.6	70.1	79.5	409.8	64.6	90.0	95.7	52.0	84.3	92.0	478.6
SCO [Huang *et al.*, 2018]	55.5	82.0	89.3	41.1	70.5	80.1	418.5	69.9	92.9	97.5	56.7	87.5	94.8	499.3
ResNet-101 Faster R-CNN + RNN														
LIWE [Wehrmann *et al.*, 2019]	69.6	90.3	95.6	51.2	80.4	87.2	474.3	73.2	95.5	98.2	57.9	88.3	94.5	507.6
VSRN* [Li *et al.*, 2019]	71.3	90.6	96.0	54.7	81.8	88.2	482.6	76.2	94.8	98.2	62.8	89.7	95.1	516.8
CVSE [Wang *et al.*, 2020]	73.5	92.1	95.8	52.9	80.4	87.8	482.5	74.8	95.1	98.3	59.9	89.4	95.2	512.7
GPO [Chen *et al.*, 2021]	76.5	94.2	**97.7**	56.4	83.4	89.9	498.1	78.5	**96.0**	98.7	61.7	90.3	95.6	520.8
GPO (MV-VSE)	**79.0**	**94.9**	**97.7**	**59.1**	**84.6**	**90.6**	**505.8**	**78.7**	95.7	**98.7**	**62.7**	**90.4**	**95.7**	**521.9**
ResNet-101 Faster R-CNN + BERT														
CAMERA* [Qu *et al.*, 2020]	78.0	95.1	97.9	60.3	85.9	91.7	508.9	77.5	96.3	98.8	63.4	90.9	95.8	522.7
GPO [Chen *et al.*, 2021]	81.7	95.4	97.6	61.4	85.9	91.5	513.5	79.7	96.4	98.9	64.8	**91.4**	**96.3**	527.5
GPO (MV-VSE)	**82.1**	**95.8**	**97.9**	**63.1**	**86.7**	**92.3**	**517.5**	**80.4**	**96.6**	**99.0**	**64.9**	91.2	96.0	**528.1**

Table 1: Experimental results (%) on Flickr30K and MS-COCO 1K. *: Ensemble results of two models.

dataset contains 123,287 images, and each image comes with 5 sentences. We mirror the data split setting of [Faghri *et al.*, 2018]. More specifically, we use 113,287 images for training, 5,000 images for validation, and 5,000 images for testing. We report results on both 1,000 test images (averaged over 5 folds) and the full 5,000 test images.

For the evaluation of image-text retrieval, following the [Faghri *et al.*, 2018], we use the Recall@K (R@K), with $K = \{1, 5, 10\}$ as the evaluation metric for the task. R@K indicates the percentage of queries for which the model returns the correct item in its top K results. We follow [Chen *et al.*, 2021] to use RSUM, which is defined as the sum of recall metrics at $K = \{1, 5, 10\}$ of both text-to-image and image-to-text retrievals, as a summarizing metric to gauge retrieval model's overall performances.

4.2 Implementation Details

We apply the proposed MV-VSE framework to the GPO [Chen *et al.*, 2021], denote as **GPO (MV-VSE)**. We implement multiple generalized pooling operators to aggregate visual features into multi-view visual embeddings. The loss function is the multi-view version of the triplet loss $\mathcal{L}_{\text{MV-VSE-Tri}}$. GPO is the current state-of-the-art VSE model without extra training data. For GPO, image features are bottom-up attention [Anderson *et al.*, 2018] region features extracted by a pretrained Faster R-CNN [Ren *et al.*, 2015] in conjunction with ResNet-101 [He *et al.*, 2016], and we use either BiGRU or BERT-base [Devlin *et al.*, 2019] as the text feature extractor. Parameters are set as $K = 3$, $\lambda = 0.7$, for both Flickr30K and MS-COCO.

4.3 Quantitative Results and Analysis

Table 1 compares MV-VSE with VSE baselines over different feature extractors on Flickr30K and MS-COCO 1K test set. VSE++ [Faghri *et al.*, 2018], SCO [Huang *et al.*, 2018], LIWE [Wehrmann *et al.*, 2019], VSRN [Li *et al.*, 2019], CVSE [Wang *et al.*, 2020] and GPO [Chen *et al.*, 2021] are state-of-the-art VSE methods proposed in recent years. CAMERA [Qu *et al.*, 2020] is the only multi-view VSE method currently. For a fair comparison, we divide the text feature extractor into RNN-based (*e.g.* GRU, LSTM) and BERT-based methods for comparison. Over all feature extractors, MV-VSE outperforms the baselines. The improvement of MV-VSE on Flickr30K is more obvious than that on MS-COCO. Since Flickr30K has larger intra-class variations. This is consistent with our statistics on the similarity within the two datasets as shown in Figure 2. Table 2 shows the experimental results on MS-COCO 5K test set. MV-VSE is also superior to other methods. The improvement of MV-VSE on the MS-COCO 5K test set is more obvious than that on the MS-COCO 1K test set, which reflects the better generalization ability of MV-VSE in large-scale retrieval.

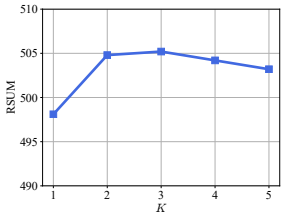
Eval Task	Image-to-Text			Text-to-Image			RSUM
Method	R@1	R@5	R@10	R@1	R@5	R@10	
ResNet-152 + RNN							
VSE++	41.3	71.1	81.2	30.3	59.4	72.4	355.7
SCO	42.8	72.3	83.0	33.1	62.9	75.5	369.6
ResNet-101 Faster R-CNN + RNN							
VSRN*	53.0	81.1	89.4	40.5	70.6	81.1	415.7
GPO	56.6	83.6	**91.4**	39.3	69.9	81.1	421.9
GPO (MV-VSE)	**56.7**	**84.1**	**91.4**	**40.3**	**70.6**	**81.6**	**424.6**
ResNet-101 Faster R-CNN + BERT							
CAMERA*	55.1	82.9	91.2	40.5	71.7	82.5	423.9
GPO	58.3	85.3	92.3	42.4	72.7	**83.2**	434.3
GPO (MV-VSE)	**59.1**	**86.3**	**92.5**	**42.5**	**72.8**	83.1	**436.3**

Table 2: Experimental results (%) on MS-COCO 5K. *: Ensemble results of two models

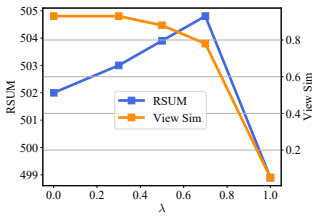
Ablation Study. We conducted an ablation study on the major components of our MV-VSE framework as showed in Table 3. $\mathcal{L}_{\text{Tri}}$ ($K = 1$) represents a common single-view VSE model. It can be seen that the multi-view model is always better than the single-view model. The performance of the proposed $\mathcal{L}_{\text{MV-Up-Tri}}$ is better than that of $\mathcal{L}_{\text{MV-Max-Tri}}$ and

Eval Task	Image-to-Text			Text-to-Image		
Method	R@1	R@5	R@10	R@1	R@5	R@10
$\mathcal{L}_{\text{Tri}}$ $(K=1)$	76.5	94.2	97.7	56.4	83.4	89.9
$\mathcal{L}_{\text{MV-Max-Tri}}$ $(K=3)$	77.2	94.2	96.6	57.6	84.4	90.1
$\mathcal{L}_{\text{MV-Avg-Tri}}$ $(K=3)$	77.2	94.3	97.7	57.9	83.7	90.5
$\mathcal{L}_{\text{MV-Up-Tri}}$ $(K=3)$	77.8	**95.3**	**97.9**	58.0	84.0	90.0
$\mathcal{L}_{\text{MV-VSE-Tri}}$ $(K=3)$	**79.0**	94.9	97.7	**59.1**	**84.6**	**90.6**

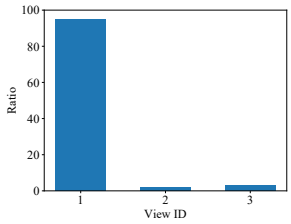
Table 3: Ablation experimental results on Flickr30K.



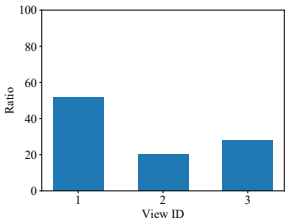
(a) Effects of the Numbers of Views.



(b) Effects of the Diversity of Views.



(c) Ratios of each view selected, when $\lambda = 1$.



(d) Ratios of each view selected, when $\lambda = 0.7$.

Figure 4: Effects of different configurations of hyper-parameters on Flickr30K.

$\mathcal{L}_{\text{MV-Avg-Tri}}$, thanks to the fact that $\mathcal{L}_{\text{MV-Up-Tri}}$ can can simultaneously optimize all view branches and retain intra-class variations. $\mathcal{L}_{\text{MV-VSE-Tri}}$ achieved the best performance, indicating that the introduction of $\mathcal{L}_{\text{MV-Max-Tri}}$ joint optimization increase the diversity of perspectives and improved the generalization ability of the model.

Effects of the Numbers of Views. Figure 4 (a) shows the effect of the number of views K on the performance of MV-VSE. We test the effect of K by fixing $\lambda = 0.7$. As K increases, the retrieval performance increases first and then decreases slightly. When $K = 3$, MV-VSE reaches the best performance. Since there are 5 text descriptions for an image in Flickr30K, there are not many views. If it is applied to data with larger intra-class variations, K can be appropriately increased.

Effects of the Diversity of Views. Figure 4 (b) shows the effects of the hyper-parameter λ on the retrieval performance and the similarities between different views of MV-VSE. We test the effect of λ by fixing $K = 3$. With the increase of λ, the contribution of $\mathcal{L}_{\text{MV-Max-Tri}}$ becomes larger, and the similarities between different views are smaller. The increase of λ increases the diversity of views, and the corresponding retrieval performance is improved. However, when $\lambda = 1$, the optimization goal of the model is $\mathcal{L}_{\text{MV-Max-Tri}}$, only one feature aggregator will be optimized for each backpropagation, which is easy to fall into the local minima and results in bad performance. Figure 4 (c) shows the ratio of each view selected when $\lambda = 1$. It can be seen that the model almost always selects a certain view. Since this view branch is always optimized, and other branches cannot be trained, the entire model falls into local minima. When the proposed multi-view upper bound loss is added, as shown in Figure 4 (d), each view can be optimized, and each view may be selected.

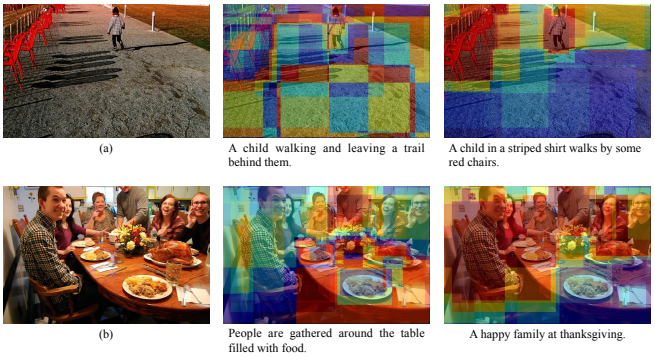


Figure 5: Visualization of MV-VSE on Flickr30K. For each group, we respectively showed the raw image and two attention maps of different views with the matched sentences from left to right.

4.4 Visualization of MV-VSE

To intuitively see the difference between the learned multiple views, we visualize the MV-VSE. We calculate the cosine similarity between the bottom-up attention regional features and the final visual embeddings. The similarity is used as the attention score to visualize the heatmap. Several visualization results are shown in Figure 5. It can be seen that MV-VSE learns embeddings from different views. For example, in Figure 5 (a), the second sentence focuses on the child's clothes and the "red chair" on the left side, and the corresponding visualization results also pay more attention to these regions. In Figure 5 (b), The first sentence pays more attention to "the table filled with food", while the second sentence pays more attention to the "family". The visualization results also give the corresponding attention to the regions. The visualization of MV-VSE demonstrates the successful modeling of intra-class variations.

5 Conclusion

This paper proposes a plug-and-play multi-view visual-semantic embedding (MV-VSE) framework, which can be used for image-text and video-text retrieval without excessively increasing model complexity. To optimize MV-VSE, a multi-view upper bound loss is proposed to jointly optimize the multi-view embedding while retaining intra-class variations between different views. Comprehensive experiments show that MV-VSE can improve the performance of existing VSE methods. In the future, the idea of multi-view will be applied to more retrieval tasks.

Acknowledgments

This research is supported in part by the Fundamental Research Funds for the Central Universities (2021XD-A01-1) and the BUPT Excellent Ph.D. Students Foundation (CX2020104).

References

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

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

[Chen *et al.*, 2021] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *CVPR*, 2021.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[Faghri *et al.*, 2018] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*, 2018.

[Feng *et al.*, 2020] Zerun Feng, Zhimin Zeng, Caili Guo, and Zheng Li. Exploiting visual semantic reasoning for video-text retrieval. In *IJCAI*, 2020.

[Frome *et al.*, 2013] Andrea Frome, Greg S Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: a deep visual-semantic embedding model. In *NeurIPS*, 2013.

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

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

[Kim *et al.*, 2021] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

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

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

[Li *et al.*, 2020] Zheng Li, Caili Guo, Bo Yang, Zerun Feng, and Hao Zhang. A novel convolutional architecture for video-text retrieval. In *ICME*, 2020.

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

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

[Qu *et al.*, 2020] Leigang Qu, Meng Liu, Da Cao, Liqiang Nie, and Qi Tian. Context-aware multi-view summarization network for image-text matching. In *MM*, 2020.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NuerIPS*, 2015.

[Sanakoyeu *et al.*, 2019] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *CVPR*, 2019.

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

[Wang *et al.*, 2020] Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. In *ECCV*, 2020.

[Wehrmann *et al.*, 2019] Jonatas Wehrmann, Douglas M Souza, Mauricio A Lopes, and Rodrigo C Barros. Language-agnostic visual-semantic embeddings. In *ICCV*, 2019.

[Xuan *et al.*, 2020] Hong Xuan, Abby Stylianou, and Robert Pless. Improved embeddings with easy positive triplet mining. In *WACV*, 2020.

[Young *et al.*, 2014] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014.

[Zhai and Wu, 2019] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. In *BMVC*, 2019.

[Zhang *et al.*, 2022] Huatian Zhang, Zhendong Mao, Kun Zhang, and Yongdong Zhang. Show your faith: Cross-modal confidence-aware network for image-text matching. In *AAAI*, 2022.

[Zheng *et al.*, 2021a] Wenzhao Zheng, Chengkun Wang, Jiwen Lu, and Jie Zhou. Deep compositional metric learning. In *CVPR*, 2021.

[Zheng *et al.*, 2021b] Wenzhao Zheng, Borui Zhang, Jiwen Lu, and Jie Zhou. Deep relational metric learning. In *ICCV*, 2021.