Learning the Best Pooling Strategy for Visual Semantic Embedding

Jiacheng Chen[1*] Hexiang Hu[2*] Hao Wu[1] Yuning Jiang[3] Changhu Wang[1]
[1]ByteDance AI Lab [2]University of Southern California [3]Alibaba Inc

Abstract

*Visual Semantic Embedding (VSE) is a dominant approach for vision-language retrieval, which aims at learning a deep embedding space such that visual data are embedded close to their semantic text labels or descriptions. Recent VSE models use complex methods to better contextualize and aggregate multi-modal features into holistic embeddings. However, we discover that surprisingly simple (but carefully selected) global pooling functions (*e.g.*, max pooling) outperform those complex models, across different feature extractors. Despite its simplicity and effectiveness, seeking the best pooling function for different data modality and feature extractor is costly and tedious, especially when the size of features varies (*e.g.*, text, video). Therefore, we propose a Generalized Pooling Operator (GPO), which learns to automatically adapt itself to the best pooling strategy for different features, requiring no manual tuning while staying effective and efficient. We extend the VSE model using this proposed GPO and denote it as VSE∞.*

Without bells and whistles, VSE∞ outperforms previous VSE methods significantly on image-text retrieval benchmarks across popular feature extractors. With a simple adaptation, variants of VSE∞ further demonstrate its strength by achieving the new state of the art on two video-text retrieval datasets. Comprehensive experiments and visualizations confirm that GPO always discovers the best pooling strategy and can be a plug-and-play feature aggregation module for standard VSE models. Code and pre-trained models are available at http://jcchen.me/vse_infty/

1. Introduction

Recognizing and describing the visual world with natural language is an essential capability for artificial intelligence. It motivates the research of image-text matching, which challenges a learning agent to establish accurate and generalizable alignment between visual and textual data, so that one can identify images or videos by text queries or vice versa.

Visual semantic embedding (VSE) [9, 10, 22] tackles this challenge by learning a semantic embedding space, where the distance between paired visual and textual instances in the embedding space is optimized to be small. The core idea of the VSE has three steps:

Step 1. Extract a set (or sequence) of features from data, using *feature extractors* (*e.g.*, ConvNets for visual data).
Step 2. Contextualize and aggregate the extracted features to project them into the joint embedding space as holistic vectors, using *feature aggregators*.
Step 3. Compute the matching score between embeddings with a similarity metric (*e.g.*, cosine distance).

With the feature extractor determined, one might expect that a complex aggregator is required to achieve good results. However, we show (in § 3) that a surprisingly simple and efficient aggregator, a carefully selected pooling function (*e.g.*, max pooling), can surpass prior state-of-the-art VSE methods with complex aggregators [17, 27, 43, 45, 46].

Such pooling functions are both simple and effective. However, searching for the optimal pooling requires extensive manual tuning and repetitive experiments (*e.g.*, grid search) for each data modality and features, which is tedious and costly as it enumerates over a combinatorial number of configurations. This search procedure could be even more complicated when the sets of features have varying sizes.

Can we discover the best pooling strategy automatically? In this paper, we propose a novel parameterized pooling operator, *Generalized Pooling Operator* (GPO), to fully exploit the strengths of pooling-based feature aggregation. GPO generalizes over various pooling functions and learns to adjust itself to the best one for different data modalities and feature extractors. Specifically, GPO learns a generator that predicts the pooling coefficients to weight the elements of sorted feature vectors, and use their weighted sum as the pooling output. The coefficient generator is instantiated as a tiny sequence model to handle variable-sized features. GPO learns to adapt to the optimal pooling strategy, and improve VSE models at a negligible extra computational cost.

With the proposed GPO, we build our multi-modal matching system as VSE∞, which extends a standard VSE framework[22] by using GPO as the feature aggregators for both visual and text features. We train our system optimizing a margin-based triplet ranking objective similar to [9], with

*Authors contributed equally

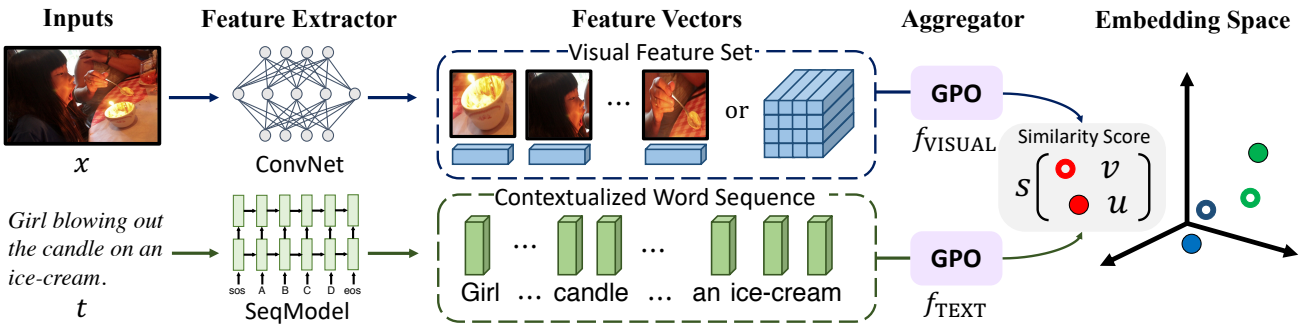

Figure 1. Illustration of the standard Visual Semantic Embedding framework with the proposed pooling-based aggregator, *i.e.*, Generalized Pooling Operator (GPO). It is simple and effective, which automatically adapts to the appropriate pooling strategy given different data modality and feature extractor, and improves VSE models at negligible extra computation cost.

the online hard-negative mining.

Without bells and whistles, VSE∞ surpasses all previous state-of-the-art VSE-based methods on the image-text retrieval tasks, over COCO [21] and Flickr30K [49]. With a straightforward extension, variants of VSE∞ also achieve the best video-text retrieval results on two benchmark datasets, *i.e.*, MSR-VTT [48] and VaTeX [44]. In additional experiments, we show that GPO consistently outperforms other alternative learnable poolings from the literature. To better understanding GPO, we further visualize the pooling strategy found by VSE∞, and compare it with the one from a thorough grid search process.

Our contributions are summarized as the following:

- We empirically find that carefully selecting simple pooling functions can outperform complex visual aggregators in prior VSE methods for image-text matching.
- We propose a novel Generalized Pooling Operator (GPO) that generalizes various pooling functions. It learns to automatically discover the best pooling function for image, text, and video data with various feature extractors.
- We build up VSE∞ with GPO, which achieves the new state-of-the-art performances among VSE methods on image-text and video-text retrieval.
- We visualize the pooling strategies learned by GPO, and verify that GPO learns the best pooling strategies given the data by comparing VSE∞ with a thorough grid search over pooling functions of all modalities.

2. Visual Semantic Embedding for Multi-modal Matching

We begin by revisiting the formal formulation of Visual Semantic Embedding (VSE). A VSE model (illustrated in Figure 1) leverages a visual embedding function $\mathbf{\Phi}(\boldsymbol{x})$ such as convolutional neural networks (*e.g.* CNNs [12, 47]), and a text embedding function $\mathbf{\Psi}(\boldsymbol{t})$ such as sequence models (*e.g.* LSTMs [14], Transformers [42]), to compute the set of visual features and text features, respectively:

$$\texttt{ConvNet}(\boldsymbol{x}) : \boldsymbol{x} \rightarrow \{\boldsymbol{\phi}_n\}_{n=1}^{\mathsf{N}},$$
$$\texttt{SeqModel}(\boldsymbol{t}) : \boldsymbol{t} \rightarrow \{\boldsymbol{\psi}_m\}_{m=1}^{\mathsf{M}}$$

Here the set of visual features $\{\boldsymbol{\phi}_n\}_{n=1}^{\mathsf{N}}$ has N elements of convolutional local representations with $\boldsymbol{\phi}_n \in \mathbb{R}^{d_1}$. As aforementioned, the concrete form of $\boldsymbol{\phi}_n$ can be feature vectors of spatial grids from the feature map, object proposals [1], or spatial-pyramids [13], depending on the feature extractor. Similarly, text features $\{\boldsymbol{\psi}_t\}_{m=1}^{\mathsf{M}}$ 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.

The output visual features $\{\boldsymbol{\phi}_n\}_{n=1}^{\mathsf{N}}$ and textual features $\{\boldsymbol{\psi}_t\}_{m=1}^{\mathsf{M}}$ are then aggregated by visual and textual aggregators $f_{\text{VISUAL}}(\cdot)$ and $f_{\text{TEXT}}(\cdot)$, to further encode the holistic visual and text embedding $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{R}^{d_3}$ as follows:

$$\boldsymbol{v} = f_{\text{VISUAL}}\Big(\{\boldsymbol{\phi}_n\}_{n=1}^{\mathsf{N}}\Big), \quad \boldsymbol{u} = f_{\text{TEXT}}\Big(\{\boldsymbol{\psi}_m\}_{m=1}^{\mathsf{M}}\Big).$$

The compatibility score is then defined as the cosine similarity between $\boldsymbol{v}$ and $\boldsymbol{u}$, formally as:

$$s_{(\boldsymbol{x},\boldsymbol{t})} = \frac{\boldsymbol{v}^\top \boldsymbol{u}}{\|\boldsymbol{v}\| \cdot \|\boldsymbol{u}\|}$$

During the inference, the $s_{(\boldsymbol{x},\boldsymbol{t})}$ scores are used to rank a query text against all candidate images, and the top candidate are returned as the prediction. We note that the inference procedure is efficient as the visual and text embedding $\boldsymbol{v}$ and $\boldsymbol{u}$ can be pre-computed. The pair-wise scores are then computed by matrix multiplication.

Learning Multi-modal Matching To learn a VSE model, existing methods mostly optimize the hinge-based triplet ranking loss with online hard negative mining proposed by VSE++ [9]. The concrete matching objective is defined by:

$$\ell_{\text{MATCH}} = \sum_{(\boldsymbol{x},\boldsymbol{t})\sim\mathcal{D}} [\alpha - s_{(\boldsymbol{x},\boldsymbol{t})} + s_{(\boldsymbol{x},\hat{\boldsymbol{t}})}]^+ + [\alpha - s_{(\boldsymbol{x},\boldsymbol{t})} + s_{(\hat{\boldsymbol{x}},\boldsymbol{t})}]^+ \quad (1)$$

Table 1. **Image-text retrieval results in R@1** of VSE models with different visual *aggregator*, evaluated with MS-COCO 1K. See § 5.1 for details.

		Region [1]		Grid [18]	
Aggregator	#Param	T → I	I → T	T → I	I → T
AvgPool [9]	0	54.0	68.5	58.9	72.4
Seq2Seq [15]	6.3M	58.5	69.9	61.5	73.3
SelfAttn [43, 45]	3.2M	56.2	70.2	60.3	73.0
GCN+AvgPool [27]	4.2M	54.9	69.0	59.5	71.8
GCN+Seq2Seq [27]	23.1M	**60.7**	72.5	59.5	71.1
Best Pooling Function	0	**60.7**	**74.5**	**61.6**	**76.3**

where α is a hyper-parameter. $(\boldsymbol{x}, \boldsymbol{t})$ is a positive image-text pair in the dataset $\mathcal{D}$ and $[x]^+ \equiv \max(0, x)$. We represent $\hat{\boldsymbol{t}} = \texttt{argmax}_{\boldsymbol{t}' \neq \boldsymbol{t}} \boldsymbol{s}_{(\boldsymbol{x}, \boldsymbol{t}')}$ and $\hat{\boldsymbol{x}} = \texttt{argmax}_{\boldsymbol{x}' \neq \boldsymbol{x}} \boldsymbol{s}_{(\boldsymbol{x}', \boldsymbol{t})}$ as the hardest negative text and image examples measured by the learned VSE model within a mini-batch.

3. VSE∞ with Generalized Pooling Operator

In this section, we first present an empirical finding that highlights the effectiveness of well-selected pooling function in VSE model, which motivates our methodological pursuit (§ 3.1). We then propose our method, Generalized Pooling Operator (GPO), with a introduction of its formal definition (§ 3.2), followed by the details of GPO's concrete model architecture (§ 3.3). Finally, we summarize our multi-modal system (VSE∞) that leverages GPO (§ 3.4).

3.1. Simple Pooling Works the Best

As aforementioned in § 1, complex aggregators f have been investigated in the VSE literature [17, 27, 43, 45, 46], such as sequence-to-sequence encoder (Seq2Seq), graph convolution network (GCN), self-attention encoder (SelfAttn), *etc*. However, we surprisingly find that these aggregation models with millions of parameters underperform carefully selected pooling functions.

Table 1 highlights a comparison between different aggregators, across two widely used image feature extractors in the literature [18] – *Grid feature* is the feature maps from ConvNets and *Region feature* is the ROI features from object detectors [1] (details in § 5). The results are reported in recall@1 for text-based image retrieval (T→I) and vice versa. Given the candidates of Average Pooling (AvgPool), Max Pooling (MaxPool) and K-Max Pooling (K-MaxPool [20], details in § 3.2) with different K, it shows that the best among them consistently outperform complex aggregators. Here, the best results for Region and Grid feature are achieved by MaxPool and K-MaxPool (K=20), respectively.

Analyses of the Empirical Findings. Most complex aggregators are designed to contextualize the input features spatially, leveraging the relationship between spatial grids or regions. However, these aggregators introduce a large set of parameters in addition to the vanilla VSE model, which causes a higher risk of over-fitting comparing to simple pooling functions. In this paper, instead of investigating why complex aggregators are suboptimal, we focus on maximizing the advantages of pooling-based aggregation.

While the optimal pooling strategy enjoys simplicity and effectiveness, searching it requires repetitive experiments over numerous configurations (*e.g.*, different K for K-MaxPool), which is both tedious and costly. This process can be more complicated when the feature extractor changes, or when the features have variable lengths (*e.g.*, text).

Motivated by these, we aim for a general and plug-and-play pooling operator that generalizes over different pooling patterns (*e.g.*, Avg, Max and K-MaxPool with arbitrary K) for variable-sized inputs, and learns to automatically adapt itself to the best strategy according to the data (*e.g.*, image, text, video *etc*.) and feature extractors. We denote our proposed module as the Generalized Pooling Operator (GPO).

3.2. Generalizing over Different Pooling Strategies

Suppose that we have a set of N feature vectors $\{\boldsymbol{\phi}_n^i\}_{n=1}^{\mathsf{N}}$ and our goal is to obtain a holistic vectorized embedding $\boldsymbol{v}^i$ out from the N elements, for each dimension $i = 1, \ldots, d_1$. Here we use the superscript i to index the i-th dimension of the feature vector. We further denote $\texttt{max}_k(\cdot)$ as the operator that *takes the k-th maximum value from an ordered list*. Then, we can formally define commonly used pooling strategies as the following:

- **AvgPool** The average pooling computes the mean value among the N elements, as $\boldsymbol{v}^i = \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \boldsymbol{\phi}_n^i, \forall i$.
- **MaxPool** The max pooling computes the maximum value among the N elements, as $\boldsymbol{v}^i = \texttt{max}_1(\{\boldsymbol{\phi}_n^i\}_{n=1}^{\mathsf{N}}), \forall i$.
- **K-MaxPool** The K-max pooling computes the mean value of the top-K maximum values among the N elements, as $\boldsymbol{v}^i = \frac{1}{\mathsf{K}} \sum_{k=1}^{\mathsf{K}} \texttt{max}_k(\{\boldsymbol{\phi}_n^i\}_{n=1}^{\mathsf{N}}), \forall i$.

Main Idea As described above, GPO aims to generalize over various pooling strategies, so that the pooling operator can automatically find the most appropriate strategy for different features. Therefore, GPO learns to generate the pooling coefficients $\boldsymbol{\theta}$, and the pooling is defined as a weighted sum over sorted features:

$$\boldsymbol{v}^i = \sum_{k=1}^{\mathsf{N}} \boldsymbol{\theta}_k \cdot \texttt{max}_k\Big(\{\boldsymbol{\phi}_n^i\}_{n=1}^{\mathsf{N}}\Big), \forall i, \qquad (2)$$

$$\text{where} \sum_{k=1}^{\mathsf{N}} \boldsymbol{\theta}_k = 1.$$

Here, the coefficients $\boldsymbol{\theta}$ are of the size N, with a scalar weight $\boldsymbol{\theta}_k$ for the k-th maximum value among the N elements. The constraint $\sum_{k=1}^{\mathsf{N}} \boldsymbol{\theta}_k = 1$ is enforced via Softmax. The parameterized pooling operator can approximate AvgPool,

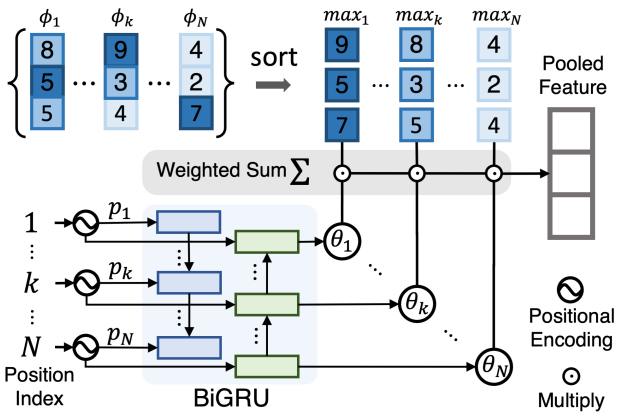

Figure 2. Detailed illustration of the GPO architecture.

MaxPool, K-MaxPool with arbitrary K, and more complex pooling functions. For instance, the learned pooling strategy could weight the top-K elements unevenly, or only set non-zero values for $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_N$. We visualize some learned pooling coefficients in § 5.3.

Learning to Generate the Pooling Coefficients The most straightforward way to parameterize $\boldsymbol{\theta}$ is to define it as a trainable vector, but this can only deal with the scenario where N is a constant integer. When the features are of variable sizes, which is common in video and text sequences, learning a fixed set of coefficients $\boldsymbol{\theta}$ is no longer feasible. To address this issue, we propose to learn a parameterized function $g(\cdot,\cdot)$ as the *coefficient generator*:

$$\boldsymbol{\theta}_k = g(k, \mathsf{N}), \text{where } k = 1, \ldots, \mathsf{N}. \tag{3}$$

As a consequence, for each position k, the coefficient generator $g(\cdot,\cdot)$ outputs a coefficient $\boldsymbol{\theta}_k$ to aggregate $\{\boldsymbol{\phi}_n^i\}_{n=1}^{\mathsf{N}}$.

3.3. Implementing Generalized Pooling Operator

Now we discuss the concrete implementation of the GPO function $g(\cdot,\cdot)$. Figure 2 provides an illustration of the architecture. There are two major components in the GPO design: (1) A positional encoding function based on trigonometric function; (2) A sequence model that takes the positional encoding sequence to generate pooling coefficients, based on bidirectional Gated Recurrent Unit (BiGRU).

Encoding Position Every position index k is uniquely represented by a dense vector, such that the vector can be further transformed to $\boldsymbol{\theta}_k$ by parameterized functions. A common approach here is to learn an embedding matrix in which row k is the embedding for k. However, this presumes the input positions $\{1, \ldots, k, \ldots, \mathsf{N}\}$ orthogonal to each other. To make more efficient use of the prior information between position indices, we adopt the positional encoding strategy used in Transformers [42] to vectorize positional indices:

$$\boldsymbol{p}_k^i = \begin{cases} \sin(w_j, k), & \text{when } i = 2j \\ \cos(w_j, k), & \text{when } i = 2j+1 \end{cases}, \forall i. \tag{4}$$

where $w_j = \frac{1}{10000^{2j/d_3}}$ and d_3 is the number of dimensions for the positional encoding.

Generating Pooling Coefficients with a Sequence Model Using the positional encoding above, we transform every position index k into a dense vector $\boldsymbol{p}_k \in \mathbb{R}^{d_3}$. Next, we learn a sequence model to produce the pooling coefficients. Since the size of feature set N varies, it is necessary for the coefficient generator to be aware of the size of feature set. Therefore, we make use of a sequence-to-sequence decoder function, which takes the sequence of positional encodings $\boldsymbol{p} = \{\boldsymbol{p}_k\}_{k=1}^{\mathsf{N}}$ as input and outputs the sequence of pooling coefficients $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k=1}^{\mathsf{N}}$. The decoder function consists of a small BiGRU and a multi-layer perceptron (MLP):

$$\{\boldsymbol{h}_k\}_{k=1}^{\mathsf{N}} = \texttt{BiGRU}(\{\boldsymbol{p}_k\}_{k=1}^{\mathsf{N}}), \quad \boldsymbol{\theta}_k = \texttt{MLP}(\boldsymbol{h}_k) \tag{5}$$

Here $\boldsymbol{h}_k$ is the output of the BiGRU at the position k.

Learning Generator with Diverse Set Sizes To make GPO's coefficient generator $g(\cdot,\cdot)$ better approximate different pooling patterns for variable-sized inputs, we perform a data augmentation strategy to allow it observing a larger variety of feature set sizes. During the training, we randomly drop 20% inputs vectors to perturb the size of the input feature set, which we call Size Augmentation. We show in Appendix that applying this strategy to both image and text effectively improve the performance of VSE models.

3.4. Building up VSE∞ using GPO

We build up our multi-modal matching model (dubbed VSE∞) by pluging GPO into the standard VSE framework (§ 2). Specifically, we replace the visual and text aggregators in the standard VSE framework (*i.e.*, AvgPool) with two GPOs. The two GPOs project the image feature vectors and text feature vectors independently into two holistic embeddings, to further compute the matching score. VSE∞ is closely related to previous VSE models. We adopt the learning framework of VSE++ [9] (Eq. 1), which improves early VSE models [10, 22] with an additional online hard negative mining procedure. We refer to § 5 for more details.

4. Related Works

Existing image-text matching methods can be categorized differently based on how the *cross-modal interaction* is implemented. As aforementioned, Visual Semantic Embedding (VSE) [9, 10, 22, 27, 46] learns a joint embedding space, such that the compatibility score can be computed as a inner-product between the two holistic image and text vectors.

Table 2. Image-Text Retrieval Results of VSE-based methods on COCO and Flickr30K datasets, using different visual and textual backbones (denoted by **bold section title**). ⋆: Ensemble results of two models; on IN/IN+VG/IG: Models pre-trained on ImageNet [38], ImageNet and VisualGenome [23], or Instagram [32], respectively. The best and second best results (in RSUM) are marked **bold** in **red** and **black**. We refer to the Appendix for extensions of this table with more baselines and COCO 5K results.

Data Split		COCO 5-fold 1K Test [5]							Flickr30K 1K Test [49]						
Eval Task		IMG → TEXT			TEXT → IMG				IMG → TEXT			TEXT → IMG			
Method	Feature Type	R@1	R@5	R@10	R@1	R@5	R@10	RSUM	R@1	R@5	R@10	R@1	R@5	R@10	RSUM
ResNet-101 Faster-RCNN on IN+VG (BUTD) [1] + BiGRU															
LIWE [45]$_{2019}$	Region	73.2	95.5	98.2	57.9	88.3	94.5	507.6	69.6	90.3	95.6	51.2	80.4	87.2	474.3
VSRN⋆[27]$_{2019}$	Region	76.2	94.8	98.2	62.8	89.7	95.1	516.8	71.3	90.6	96.0	54.7	81.8	88.2	482.6
CVSE [43]$_{2020}$	Region	69.2	93.3	97.5	55.7	86.9	93.8	496.4	70.5	88.0	92.7	54.7	82.2	88.6	476.7
Our: VSE++	Region	68.5	92.6	97.1	54.0	85.6	92.7	490.5	62.2	86.6	92.3	45.7	73.6	81.9	442.3
Our: VSE∞	Region	78.5	96.0	98.7	61.7	90.3	95.6	**520.8**	76.5	94.2	97.7	56.4	83.4	89.9	**498.1**
Our: VSE∞	Region+Grid	80.0	97.0	99.0	64.8	91.6	96.5	**528.8**	80.7	96.4	98.3	60.8	86.3	92.3	**514.8**
ResNet-101 Faster-RCNN on IN+VG (BUTD) [1] + BERT [7]															
Our: VSE++	Region	67.9	91.9	97.0	54.0	85.6	92.5	488.9	63.4	87.2	92.7	45.6	76.4	84.4	449.7
Our: VSE∞	Region	79.7	96.4	98.9	64.8	91.4	96.3	**527.5**	81.7	95.4	97.6	61.4	85.9	91.5	**513.5**
Our: VSE∞	Region+Grid	82.2	97.5	99.5	68.1	92.9	97.2	**537.4**	85.3	97.2	98.9	66.7	89.9	94.0	**532.0**
ResNeXT-101 on IG (WSL) [32] + BERT [7]															
Our: VSE++	Grid	79.6	97.1	99.0	66.4	91.1	95.5	528.7	80.9	96.6	98.9	65.2	89.5	93.7	524.8
Our: VSE∞	Grid	84.5	98.1	99.4	72.0	93.9	97.5	**545.4**	88.4	98.3	99.5	74.2	93.7	96.8	**550.9**
Our: VSE∞ ⋆	Grid	85.6	98.0	99.4	73.1	94.3	97.7	**548.1**	88.7	98.9	99.8	76.1	94.5	97.1	**555.1**

Therefore, VSE relies on learning strong image and text embedding functions to obtain high-quality joint embedding space. Frome *et al*. [10] used this approach for zero-shot image recognition [2, 24, 34], via matching visual embeddings with semantic word embeddings. Kiros *et al*. [22] extends the idea by using bi-directional LSTMs to encode sentence as the semantic embedding. Faghri *et al*. proposes VSE++, which learns with online hard-negative mining and further improves the quality of VSE models [9]. VSE++ is one of the most fundamental VSE methods that use AvgPool as the feature aggregator. Beyond the above, more research along this line focused on improving the visual or text embedding function (especially the aggregator), or designing auxiliary training objectives [8, 11, 17, 27, 33, 40, 41, 46].

Recently, methods using BERT models for vision-language data (V+L BERTs) [6, 16, 26, 28, 29, 31] learns to perform rich cross-modal interaction, via tailored mechanisms such as (single/multi-headed) cross-attention [25, 42]. These methods typically use a BERT [7] as the text feature extractor and learn additional cross-modal Transformers for rich cross-modal interactions. At the same time, these methods perform large-scale visual-linguistic pre-training with a collection of datasets with paired images and text (*e.g*., the Conceptual Caption dataset [39]). Comparing to this family of methods, VSE models are inferior in empirical performances as its lack of strong cross-modal interaction. However, VSE models are orders of magnitude more efficient than V+L BERTs in terms of cross-modal retrieval as the latter requires the huge BERT model to forward over all pairs of images and texts. In § 5.1.1, we show that the best VSE∞ can attain a close image-text matching performance to the best V+L BERT method while being much faster in large-scale multi-modal retrieval.

5. Experiments

We conduct experiments to validate VSE∞ on image-text (§ 5.1.1) and video-text matching (§ 5.1.2). We compare GPO with alternative poolings in § 5.2, and analyze the learned GPO in § 5.3. We refer to the Appendix for complete experimental details and more ablation studies.

5.1. Multi-modal Retrieval Experiments

Multi-modal retrieval is typically evaluated using the metric of recall at K (R@K), with $K = \{1, 5, 10\}$. We follow [3, 46] to use RSUM, which is defined as the sum of recall metrics at $K = \{1, 5, 10\}$ of both I→T (I2T) and T→I (T2I) retrievals, as a summarizing metric to gauge retrieval model's overall performances. In all experiments, we set the dimensions of the positional encoding and BiGRU to be 32. Therefore, GPO has $0.1M$ parameter in total, which is less than 1% of the entire model.

5.1.1 Image-text Retrieval

Setup For image-text retrieval, we perform experiments on MS-COCO [5, 30] and Flickr30K [49] over various feature

Table 3. Comparison between variants of VSE∞ and V+L BERTs. All methods uses BERT-base. ⋆: ensemble results of two models. R/G in parenthesis represents Region/Grid features.

Data Split			COCO 5K Test [5]			
Eval Task			IMG → TEXT		TEXT → IMG	
Method	Pretrain	CNN	R@1	R@5	R@1	R@5
ViLBERT[50]	✓	BUTD	53.5	79.7	38.6	68.2
ViLBERT DG[50]	✓	BUTD	57.5	84.0	41.8	71.5
UNICODER VL[26]	✓	BUTD	62.3	87.1	46.7	76.0
UNITER[6]	✓	BUTD	64.4	87.4	50.3	78.5
OSCAR[29]	✓	BUTD	70.0	91.1	54.0	80.8
Our Methods						
VSE∞ (R)	✗	BUTD	58.3	85.3	42.4	72.7
VSE∞ (R+G)	✗	BUTD	62.5	87.8	46.0	75.8
VSE∞ (G)	✗	WSL	66.4	89.3	51.6	79.3
VSE∞ (G) ⋆	✗	WSL	68.1	90.2	52.7	80.2

extractors. Each image of these two datasets is associated with five text descriptions. COCO contains 123,287 images, we use the data split of [9, 21, 25] where there are 113,287 training images, 5000 test images, and 5000 validation images. Flickr30K contains 31,000 images, we also use the same data split as [9], where there are 29,000 training images, 1000 test images, and 1000 validation images. COCO results are reported in 5K and 1K, where the 1K results are averaged over the five 1K data folds. The image feature extractors are categorized into *Region feature* and *Grid feature* following the naming convention in [18], where grid feature represents the feature maps from a CNN, and region feature represents object-level features from a detector.

Implementation Details The dimension of the joint embedding space is 1024. We use pre-extracted object features [1] as the region feature (BUTD feature). For grid feature, the CNN backbone is fine-tuned, and we increase the resolution of input images to 512×512 as suggested by [18]. We experiment with two different CNNs: (1) ResNet-101 of Faster-RCNN [37] pre-trained on ImageNet and Visual Genome (BUTD) [1] and (2) ResNeXT-101(32×8d) [47] pre-trained on Instagram (WSL) [32]. Meanwhile, we use either BiGRU or BERT-base as the text feature extractor. We refer to the Appendix for full training details and more results.

Main Results Table 2 compares VSE∞ with VSE baselines over different feature extractors. VSE++ is the fundamental VSE method as described in § 4, we re-implement it (denoted as *Our: VSE++*) and apply it on latest feature extractors (*e.g.*, BUTD image features, BERT, *etc.*). The major difference to its original implementation is the input image size for grid feature. LIWE [45], VSRN [27], and CVSE [43] are state-of-the-art VSE methods proposed in recent two years (we compare with more baselines in the Appendix). We use numbers directly from original papers except for CVSE, for which we re-run the official code after removing unfair additional label inputs and fixing its 1K evaluation setting (details in Appendix). *Region+Grid* means training two separate models with region and grid feature and averaging their similarity outputs. Over all three combinations of feature extractors, VSE∞ outperforms the baselines *without using complicated aggregator*. Besides, VSE∞ with WSL+BERT as feature extractors achieves the best empirical results, improving over the second best feature extractors by a large margin. VSE∞ is better than the baselines in both *performance* and *simplicity*. We present the COCO 5K Test results, and results with additional feature extractors in the Appendix.

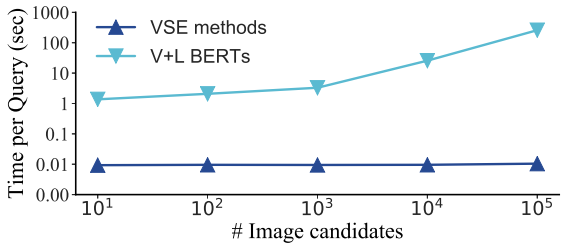

Figure 3. We compare single GPU inference time for text-based image retrieval (lower the better). VSE methods are much faster than V+L BERTs, especially when number of images grows large.

Comparing VSE∞ with V+L BERTs We further compare VSE∞ with state-of-the-art V+L BERTs in Table 3. We report results on COCO 5K as the 1K results reported by V+L BERTs is computed on the first 1K fold, instead of the average result over the five 1K folds. Without large-scale V+L pre-training, our VSE∞ (R+G) is no worse than three out of five V+L BERTs using the same feature extractors. By using the WSL CNN to compensate for the lack of pre-training, VSE∞ further outperforms UNITER and gets very close to OSCAR [29], which is the current best V+L BERT. This is a promising result since VSE models by design do not have *any fine-grained cross-modal interaction* as V+L BERTs (see § 4). Meanwhile, VSE methods are orders of magnitude faster for large-scale multi-modal retrieval as the holistic embeddings can be pre-computed or indexed [19], and matrix multiplication is all we need to compute the compatibility score. To demonstrate this, we perform an additional text-to-image retrieval experiment with increasing size of image candidates, and visualize the model's inference time in Figure 3. When the number of image candidates is small, we observe that VSE is a hundred time faster than V+L BERT. As the number of image candidates grows, the gap of time cost increases almost quadratically. VSE∞ fully exploits existing feature extractors and pushes the performance of VSE-based methods to a new height, which have significant impact in real-world problems such as image search with text query.

Evaluating VSE∞ with Crisscrossed Captions We eval-

Table 4. Results on video-text retrieval benchmarks. ∞: Methods modified to using the GPO to aggregate frame and word features.

Method	VIDEO → TEXT			TEXT → VIDEO			
	R@1	R@5	R@10	R@1	R@5	R@10	RSUM
VSE++[9]	14.4	34.1	45.6	8.3	24.0	34.1	160.5
VSE∞	16.0	38.6	50.2	8.7	25.3	35.9	**174.7**
HGR [4]	15.0	36.7	48.8	9.2	26.2	36.5	172.4
HGR∞	15.0	39.0	51.7	9.1	25.9	36.3	**177.0**

(a) **MSR-VTT Video-Text Retrieval [48]**

Method	VIDEO → TEXT			TEXT → VIDEO			
	R@1	R@5	R@10	R@1	R@5	R@10	RSUM
VSE++[9]	47.8	78.6	86.2	34.7	71.3	81.7	400.3
VSE∞	51.2	78.7	86.3	34.2	71.6	81.9	**403.9**
HGR [4]	48.9	79.1	87.9	35.6	73.5	83.4	408.4
HGR∞	51.0	78.8	87.7	37.3	73.4	82.4	**410.6**

(b) **VATEX Video-Text Retrieval [44]**

Table 5. Evaluations on COCO 5K test set with Crisscrossed Caption (CxC). All models are trained on the COCO dataset.

Method	I → T		T → I		T → T		I → I	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
VSRN [27]	52.4	81.9	40.1	71.1	41.0	64.8	44.2	76.7
DE [35]	55.9	84.2	41.7	72.3	42.6	64.9	38.5	73.6
VSE∞ (BUTD)	60.6	87.4	46.2	76.3	45.9	68.7	44.4	78.3
VSE∞ (WSL)	67.9	90.6	53.6	81.1	46.7	69.2	51.3	83.2

uate our best models (with BERT and Grid features on either BUTD or WSL backbones) on the Crisscrossed Captions(CxC) [35] extension of COCO, which evaluates image-text matching systems more holistically with additional intra-modal and inter-modal semantic similarity annotations. Table 5 shows that our model can significantly outperform the baseline for both inter-modality and intra-modality (on BUTD features). Moreover, VSE∞ with WSL feature can further boost the performances.

5.1.2 Video-text Retrieval

Setup We evaluate our method on two video datasets: MSR-VTT [48] and VATEX [44]. MSR-VTT contains 10,000 videos while each video has 20 text descriptions, and we use the standard split with 6573 videos for training, 2990 for testing and 497 for validation. VATEX contains 25,991 videos for training, 6000 for testing and 3000 for validation, and the 10 English descriptions for each video are used in the experiments. We splits the original validation set into new validation and testing set, each with 1500 videos, as [4].

Implementation Details We use ResNet-152 pre-trained on ImageNet to extract frame features for MSR-VTT and use the official I3D feature for VATEX. All implementations are based on the official code of the video-text matching method HGR [4], and we re-train all models. BiGRU is the text backbone for all experiments and the VSE setting is similar to 5.1.1 except that visual features are frame-level video features. Complete details are in the Appendix.

Main Results Table 4 presents the effectiveness of VSE∞ on video-text matching. VSE++ for video-text matching is an extension of the image-text version. HGR [4] is the current state-of-the-art method, which employs hierarchical matching strategies. By replacing the AvgPool on frames and text with GPO, VSE∞ clearly outperforms VSE++ in terms of RSUM. Additionally, we change the pooling function in the global-matching branch of HGR [4] with GPO (denoted as HGR∞), and get consistent improvements.

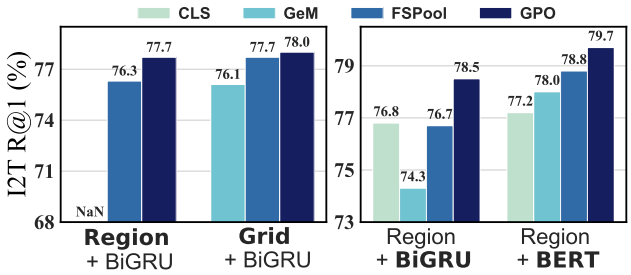

Figure 4. **Left figure** studies different aggregators on two visual features, with AvgPool as the text aggregator for BiGRU. **Right figure** studies different aggregators on two textual features, while using GPO as the visual aggregator for the region features.

5.2. Comparing GPO to Alternative Poolings

We compare GPO with several representative learnable pooling methods across four combos of visual and text feature extractors. GPO's Size Augmentation is used in all cases for fair comparison. The baselines include:

- **Generalized Mean Pooling (GeM) [36]**, an adaptive pooling function with a single trainable parameter, and is popular in image search literature;
- **Feature-sorting Pooling (FSPool) [51]**, a learnable pooling that handles variable-sized inputs by interpolating a fixed-size learnable vector, which was proposed to encode sets in permutation-invariant manner. FSPool generates different pooling coefficients for each feature dimension.
- **CLS** token based aggregation, which is widely used for aggregating text features in the BERT models [7]. For BiGRU, we simply take the feature of the first token for the CLS aggregation.

Figure 4 presents the comparison in R@1 of COCO I2T, we skip T2I results since the conclusions are the same. In the left part of visual pooling, FSPool is close to GPO on

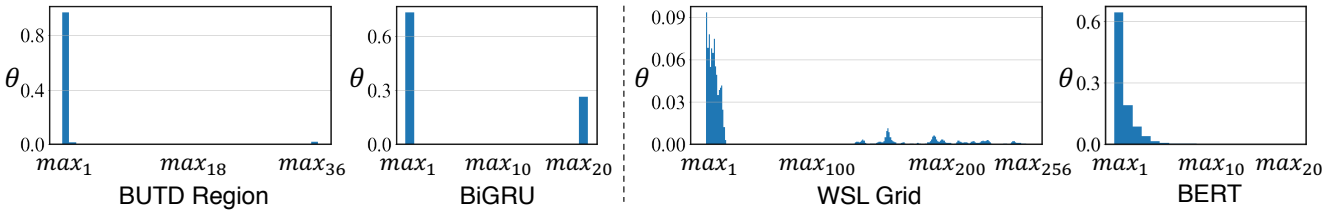

Figure 5. Visualization of pooling coefficients learned by GPO. The left and right figures are the VSE models on "BUTD Region+BiGRU" and "WSL Grid+BERT" features, respectively.

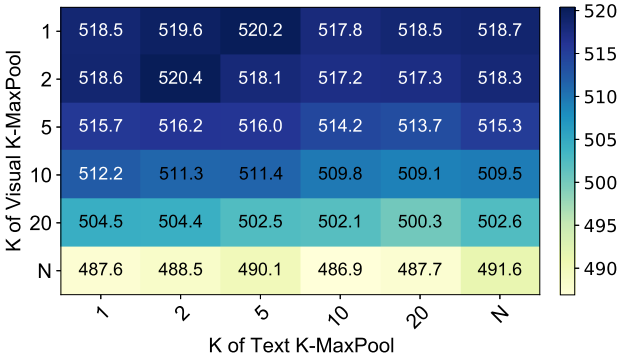

Figure 6. Illustrating the grid search process using K-MaxPool with various K, on BUTD region and BiGRU features. The results are the RSUM values of COCO 5-fold 1K evaluation.

grid feature, but much worse on region feature. We note that the official GeM implementation is not numerical stable and it causes gradient explosion when training with region feature and BiGRU. In the right of Figure 4, we vary over different textual pooling strategies, with BiGRU or BERT being the text feature extractor. Again, GPO outperforms all alternative pooling methods. It is worth noting that BERT's default CLS aggregator is far from being optimal in the context of multi-modal matching. Above all, GPO is the best pooling strategy on various combinations of features, and can serve as a plug-and-play per-modality feature aggregator.

5.3. Visualizing and Understanding GPO

To better understand the pooling patterns learned by GPO, we visualize the learned pooling coefficients of GPO in Figure 5. On BUTD region feature, GPO approximates MaxPool, which is consistent with the observation in § 3. On grid feature, the coefficients are less regular, but large position indices take up most large values. We additionally observe that GPO generates non-zero coefficients for the maximum and minimum values of BiGRU features, which goes beyond the pattern of K-MaxPool. The learned pooling strategy for BERT is close to MaxPool.

Comparing GPO against Grid Search. We recall that the main motivation of GPO is to fully exploit the advantages of simple pooling functions but eliminate the repetitive manual experiments for seeking the best pooling hyperparameter. To verify how GPO address this challenge, we conduct a manual grid search over K-MaxPool with different K values for image-text matching with BUTD region and BiGRU features. GPO's Size Augmentation is used here for fair comparison as it improves performance (see Appendix for details). As shown by Figure 6, the best RSUM given by the grid search is 520.4, which is slightly worse than the 520.8 of the corresponding GPO entry in Table 2, which means that GPO successfully refrains us from the costly repetitive search. Note that GPO generates a pooling strategy beyond K-MaxPool for BiGRU (Figure 5), although it does not make it significantly better than the best-selected K-MaxPool.

Figure 6 shows that the best combination of pooling functions for visual and text modalities are entangled with each other. For instance, the best textual pooling function varies when the visual pooling function is changed. Therefore, a $n \times n$ search is necessary to find the optimal combinations of K, where n is the number of grids for each modality. This could become worse when the visual feature includes multiple feature extractors (*e.g.*, region+grid), as the search complexity can further become $O(n^3)$.

In summary, GPO keeps the effectiveness and efficiency of best-selected pooling functions, and avoids the annoying grid search process. GPO can serve as a plug-and-play aggregation module to improve VSE models.

6. Conclusion

In this paper, we propose the Generalized Pooling Operator (GPO), which learns to automatically adapt itself to the best pooling strategy for different data and feature backbone. As a result, we build up our VSE∞ by extending the standard VSE model with GPO as the feature aggregators. VSE∞ outperforms previous VSE methods significantly on image-text retrieval benchmarks across popular feature extractors. We further demonstrate that our VSE model achieves comparable image-text matching performances to vision+language BERT models, without visual-linguistic pre-training. Comprehensive ablation experiments confirm that GPO discovers proper pooling strategies. With simple adaptations, variants of VSE∞ further demonstrate effectiveness by achieving the new state of the art on two video-text retrieval datasets.

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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 3, 5, 6

[2] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 5

[3] H. Chen, G. Ding, Xudong Liu, Zijia Lin, J. Liu, and J. Han. Imram: Iterative matching with recurrent attention memory for cross-modal image-text retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12652–12660, 2020. 5

[4] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 7

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5, 6

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *Eur. Conf. Comput. Vis.*, 2020. 5, 6

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 5, 7

[8] Aviv Eisenschtat and Lior Wolf. Linking image and text with 2-way nets. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 5

[9] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. In *BMVC*, 2017. 1, 2, 3, 4, 5, 6, 7

[10] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Adv. Neural Inform. Process. Syst.*, 2013. 1, 4, 5

[11] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 2

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. 2

[15] Hexiang Hu, Ishan Misra, and Laurens van der Maaten. Binary image selection (bison): Interpretable evaluation of visual grounding. *arXiv preprint arXiv:1901.06595*, 2019. 3

[16] Yan Huang and Liang Wang. Acmm: Aligned cross-modal memory for few-shot image and sentence matching. In *Int. Conf. Comput. Vis.*, 2019. 5

[17] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 3, 5

[18] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik G. Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 3, 6

[19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 6

[20] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *ACL*, 2014. 3

[21] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 2, 6

[22] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *NeurIPS Workshop Deep Learning*, 2014. 1, 4, 5

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, J. M. Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2016. 5

[24] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2013. 5

[25] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Eur. Conf. Comput. Vis.*, 2018. 5, 6

[26] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *AAAI*, 2019. 5, 6

[27] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Int. Conf. Comput. Vis.*, 2019. 1, 3, 4, 5, 6, 7

[28] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 5

[29] Xiujun Li, Xi Yin, C. Li, X. Hu, Pengchuan Zhang, Lei Zhang, Longguang Wang, H. Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *Eur. Conf. Comput. Vis.*, 2020. 5, 6

[30] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014. 5

[31] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Int. Conf. Comput. Vis.*, 2019. 5

[32] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Eur. Conf. Comput. Vis.*, 2018. 5, 6

[33] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching.

In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 5

[34] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *Adv. Neural Inform. Process. Syst.*, 2014. 5

[35] Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for ms-coco. *arXiv preprint arXiv:2004.15020*, 2020. 7

[36] Filip Radenović, Giorgos Tolias, and O. Chum. Fine-tuning cnn image retrieval with no human annotation. *TPAMI*, 41:1655–1668, 2019. 7

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*, 2015. 6

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2014. 5

[39] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 5

[40] Richard Socher, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014. 5

[41] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 5

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017. 2, 4, 5

[43] Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. *Eur. Conf. Comput. Vis.*, 2020. 1, 3, 5, 6

[44] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Y. Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Int. Conf. Comput. Vis.*, pages 4580–4590, 2019. 2, 7

[45] Jonatas Wehrmann, Mauricio A. Lopes, Douglas M. Souza, and Rodrigo C. Barros. Language-agnostic visual-semantic embeddings. In *Int. Conf. Comput. Vis.*, pages 5803–5812, 2019. 1, 3, 5, 6

[46] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 3, 4, 5

[47] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2, 6

[48] J. Xu, T. Mei, Ting Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5288–5296, 2016. 2, 7

[49] 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*, 2:67–78, 2014. 2, 5

[50] Bowen Zhang, Hexiang Hu, Vihan Jain, Eugene Ie, and Fei Sha. Learning to represent image and text with denotation graph. In *EMNLP*, 2020. 6

[51] Y. Zhang, Jonathon S. Hare, and A. Prügel-Bennett. Fspool: Learning set representations with featurewise sort pooling. *Int. Conf. Learn. Represent.*, 2020. 7