# VSE++: IMPROVING VISUAL-SEMANTIC EMBEDDINGS WITH HARD NEGATIVES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We present a new technique for learning visual-semantic embeddings for cross-modal retrieval. Inspired by the use of hard negatives in structured prediction, and ranking loss functions used in retrieval, we introduce a simple change to common loss functions used to learn multi-modal embeddings. That, combined with fine-tuning and the use of augmented data, yields significant gains in retrieval performance. We showcase our approach, dubbed *VSE++*, on the MS-COCO and Flickr30K datasets, using ablation studies and comparisons with existing methods. On MS-COCO our approach outperforms state-of-the-art methods by $8.8\%$ in caption retrieval, and $11.3\%$ in image retrieval (based on R@1).

## 1 INTRODUCTION

Joint embeddings enable a wide range of tasks in image, video and language understanding. Examples include shape-image embeddings (Li et al. (2015b)) for shape inference, bilingual word embeddings (Zou et al. (2013)), human pose-image embeddings for 3D pose inference (Li et al. (2015a)), fine-grained recognition (Reed et al. (2016a)), zero-shot learning (Frome et al. (2013)), and modality conversion via synthesis (Reed et al. (2016b;a)). Such embeddings entail mappings from two (or more) domains into a common vector space in which semantically associated inputs (e.g., text and images) are mapped to similar locations. The embedding space thus represents the underlying structure of the domains, where locations and often direction are semantically meaningful.

In this paper we focus on learning *visual-semantic embeddings*, central to tasks such as image-caption retrieval and generation Kiros et al. (2014); Karpathy & Fei-Fei (2015), and visual question-answering Malinowski et al. (2015). One approach to visual question-answering, for example, is to first describe an image by a set of captions, and then to find the nearest caption in response to a question (Agrawal et al. (2017); Zitnick et al. (2016)). In the case of image synthesis from text, one approach is to invert the mapping from a joint visual-semantic embedding to the image space (Reed et al. (2016b;a)).

Here we focus on visual-semantic embeddings for the generic task of cross-modal retrieval; i.e. the retrieval of images given captions, or of captions from a query image. As is common in information retrieval, we measure performance by R@$K$, i.e., recall at $K$ – the fraction of queries for which the correct item is retrieved in the closest $K$ points to the query in the embedding space ($K$ is usually a small integer, often $1$). More generally, retrieval is a natural way to assess the quality of joint embeddings for image and language data for use in subsequent tasks (Hodosh et al. (2013)).

To this end, the problem is one of ranking, for which the correct target(s) should be closer to the query than other items in the corpus, not unlike *learning to rank* problems (e.g., Li (2014)), and max-margin structured prediction Chapelle et al. (2007); Le & Smola (2007). The formulation and model architecture in this paper are most closely related to those of Kiros et al. (2014), learned with a triplet ranking loss. In contrast to that work, we advocate a novel loss, the use of augmented data, and fine-tuning, that together produce a significant increase in caption retrieval performance over the baseline ranking loss on well-known benchmark datasets. We outperform the best reported result on MS-COCO by almost $9\%$. We also demonstrate that the benefit from a more powerful image encoder, and fine-tuning the image encoder, is amplified with the use of our stronger loss function. To ensure reproducibility, our code will be made publicly available. We refer to our model as *VSE++*.

Finally, we note that our formulation complements other recent articles that propose new model architectures or similarity functions for this problem. Wang et al. (2017) propose an embedding network to fully replace the similarity function used for the ranking loss. An attention mechanism on both image and caption is used by Nam et al. (2016), where the authors sequentially and selectively focus on a subset of words and image regions to compute the similarity. In Huang et al. (2016), the authors use a multi-modal context-modulated attention mechanism to compute the similarity between an image and a caption. Our proposed loss function and triplet sampling could be extended and applied to other such approaches.

## 2 Learning Visual-Semantic Embeddings

### 2.1 Image-caption Retrieval

For image-caption retrieval the query is a caption and the task is to retrieve the most relevant image(s) from a database. Or the query may be an image and one retrieves relevant captions. The goal is to maximize recall at $K$ (R@$K$), the fraction of queries for which the most relevant item is ranked among the top $K$ items returned.

Let $S = \{(i_n, c_n)\}_{n=1}^N$ be a training set of image-caption pairs. We refer to $(i_n, c_n)$ as *positive pairs* and $(i_n, c_{m\neq n})$ as *negative pairs*; i.e., the most relevant caption to the image $i_n$ is $c_n$ and for caption $c_n$, it is the image $i_n$. We define a similarity function $s(i, c) \in \mathbb{R}$ that should, ideally, give higher similarity scores to positive pairs than negatives. In caption retrieval, the query is an image and we rank a database of captions based on the similarity function; i.e., R@$K$ is the percentage of queries for which the positive caption is ranked among the top $K$ captions using $s(i, c)$. And likewise for image retrieval. In what follows the similarity function is defined on the joint embedding space. This approach differs from others, such as Wang et al. (2017), which use a similarity network to directly classify an image-caption pair as matching or non-matching.

### 2.2 Visual-Semantic Embedding

Let $\phi(i; \theta_\phi) \in \mathbb{R}^{D_\phi}$ be a feature-based representation computed from the image (e.g. the representation before logits in VGG19 (Simonyan & Zisserman (2014)) or ResNet152 (He et al. (2016))). Similarly, let $\psi(c; \theta_\psi) \in \mathbb{R}^{D_\psi}$ be a representation of a caption $c$ in a caption embedding space (e.g. a GRU-based text encoder). Here, $\theta_\phi$ and $\theta_\psi$ denote the model parameters used for the respective mappings to obtain the initial image and caption representations.

The mappings into the *joint embedding space* are then defined in terms of linear projections; i.e.,

$$f(i; W_f, \theta_\phi) = W_f^T \phi(i; \theta_\phi) \quad (1)$$

$$g(c; W_g, \theta_\psi) = W_g^T \psi(c; \theta_\psi) \quad (2)$$

where $W_f \in \mathbb{R}^{D_\phi \times D}$ and $W_g \in \mathbb{R}^{D_\psi \times D}$. We further normalize $f(i; W_f, \theta_\phi)$, and $g(c; W_g, \theta_\psi)$, to lie on the unit hypersphere. The similarity function in the embedding space is then defined as an inner product:

$$s(i, c) = f(i; W_f, \theta_\phi) \cdot g(c; W_g, \theta_\psi) \,. \quad (3)$$

Let $\theta = \{W_f, W_g, \theta_\psi\}$ be the model parameters. If we also fine-tune the image encoder, then we would also include $\theta_\phi$ in $\theta$.

Training entails the minimization of empirical loss with respect to $\theta$, i.e., the cumulative loss over training data $S = \{(i_n, c_n)\}_{n=1}^N$:

$$e(\theta, S) = \tfrac{1}{N} \sum_{n=1}^{N} \ell(i_n, c_n) \quad (4)$$

where $\ell(i_n, c_n)$ is a suitable loss function for a single training exemplar.

Recent approaches to joint visual-semantic embeddings have used a form of triplet ranking loss (Kiros et al. (2014); Karpathy & Fei-Fei (2015); Zhu et al. (2015); Socher et al. (2014)), inspired

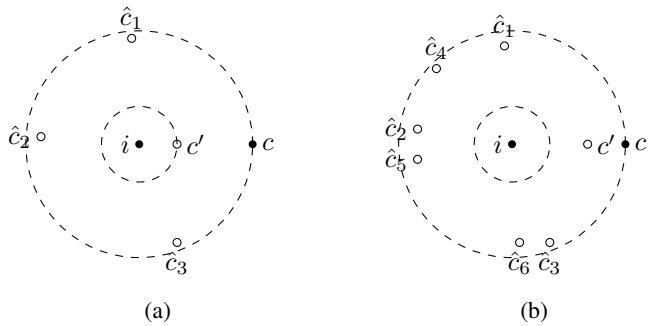



Figure 1: An illustration of typical positive pairs and the nearest negative samples. Here assume similarity score is the negative distance. Filled circles show a positive pair $(i, c)$, while empty circles are negative samples for the query $i$. The dashed circles on the two sides are drawn at the same radii. Notice that the hardest negative sample $c'$ is closer to $i$ in (a). Assuming a zero margin, (b) has a higher loss with the *SH* loss compared to (a). The *MH* loss assigns a higher loss to (a).

its use in image retrieval (Frome et al. (2007); Chechik et al. (2010)). Prior work has employed a hinge-based, triplet ranking loss with margin $\alpha$:

$$\ell_{SH}(i, c) = \sum_{\hat{c}} [\alpha - s(i, c) + s(i, \hat{c})]_+ + \sum_{\hat{i}} [\alpha - s(i, c) + s(\hat{i}, c)]_+ \,, \tag{5}$$

where $[x]_+ \equiv \max(x, 0)$. This hinge loss comprises two symmetric terms, with $i$ and $c$ being queries. The first sum is taken over all negative captions $\hat{c}$ given query $i$. The second negative images $\hat{i}$ given caption $c$. Each term is proportional to the expected loss (or *violation*) over sets of negative samples. If $i$ and $c$ are closer to one another in the joint embedding space than to any negatives pairs, by the margin $\alpha$, the hinge loss is zero. In practice, for computational efficiency, rather than summing over all possible negatives in the training set, it is common to only sum over (or randomly sample) the negatives within a mini-batch of stochastic gradient descent (e.g., see Kiros et al. (2014); Socher et al. (2014); Karpathy & Fei-Fei (2015)).

Of course there are other loss functions that one might consider. One approach is a pairwise hinge loss in which elements of positive pairs are encouraged to be within a radius $\rho_1$ in the joint embedding space, while negative pairs should be no closer than $\rho_2 > \rho_1$. This is problematic as it constrains the structure of the latent space more than does the ranking loss, and it entails the use of two hyper-parameters which can be very difficult to set. Another possible approach is to use Canonical Correlation Analysis to learn $W_f$ and $W_g$, thereby trying to preserve correlation between the text and images in the joint embedding (e.g., Klein et al. (2015); Eisenschtat & Wolf (2016)). By comparison, when measuring performance as R@$K$, for small $K$, a correlation-based loss will not give sufficient influence to the embedding of negative items in the local vicinity of positive pairs, which is critical for R@$K$.

### 2.3 Emphasis on Hard Negatives

Inspired by common loss functions used in structured prediction (Tsochantaridis et al. (2005); Yu & Joachims (2009); Felzenszwalb et al. (2010)), we focus on hard negatives for training, i.e., the negatives closest to each training query. This is particularly relevant for retrieval since it is the hardest negative that determines success or failure as measured by R@1.

Given a positive pair $(i, c)$, the hardest negatives are given by $i' = \arg\max_{j \neq i} s(j, c)$ and $c' = \arg\max_{d \neq c} s(i, d)$. To emphasize hard negatives we therefore define our loss as

$$\ell_{MH}(i, c) = \max_{c'} [\alpha + s(i, c') - s(i, c)]_+ + \max_{i'} [\alpha + s(i', c) - s(i, c)]_+ \,. \tag{6}$$

Like Eq. 5, the loss comprises two terms, one with $i$ and one with $c$ as queries. Unlike Eq. 5, this loss is specified in terms of the hardest negatives, $c'$ and $i'$. Hereafter, we refer to the loss in Eq. 6 as *Max of Hinges (*MH*)* loss, and the loss function in Eq. 5 as *Sum of Hinges (*SH*)* loss.

An example of where the *MH* loss is superior to *SH* is when multiple negatives with relatively small violations combine to dominate the *SH* loss. For example, in Fig. 1, a positive pair is depicted together with two sets of negative samples. In Fig. 1(a), there exists a single negative sample that is

too close to the query. Essentially, moving such a hard negative, might require a significant change to the mapping. However, any training step that pushes the hard negative away, can bring back many small violating negative samples, as in Fig. 1(b). Using the *SH* loss, these 'new' negative samples may dominate the loss, so the model is pushed back to the first example in Fig. 1(a). This may create local minima in the *SH* loss that may not be as problematic for the *MH* loss as it focuses solely on the hardest negative.

For computational efficiency, instead of finding the hardest negatives in the whole training set, we find them in a mini-batch. With random sampling of the mini-batches, this approximate yields other advantages. One is that there is a high probability of getting hard negatives that are harder than at least $90\%$ of the entire training set. Moreover, the loss is potentially robust to label errors in the training data because the probability of sampling the hardest negative over the entire training set is somewhat low. In Appendix A, we analyze the probability of sampling hard negatives further.

## 3 EXPERIMENTS

We first perform experiments with our approach, *VSE++*, and compare it to a baseline formulation with *SH* loss, referred to as *VSE0*, and other state-of-the-art approaches. Essentially, the baseline formulation, *VSE0*, is the same used by Kiros et al. (2014), here referred to as *UVS*.

We experiment with two image encoders: VGG19 by Simonyan & Zisserman (2014) and ResNet152 by He et al. (2016). In what follows below we use VGG19 unless specified otherwise. As in previous work we extract image features directly from FC7, the penultimate fully connected layer. The dimensionality of the image embedding, $D_\phi$, is $4096$ for VGG19 and $2048$ for ResNet152.

In somewhat more detail, we first resize the image to $256 \times 256$, and then use either a single center crop of size $224 \times 224$ or the mean of feature vectors for $10$ crops of similar size, as done by Klein et al. (2015) and Vendrov et al. (2015). We refer to training with one center crop as *1C* and training with $10$ crops as *10C*. We also consider using random crops, denoted by *RC*. For *RC*, we have the full VGG19 model and extract features over a single randomly chosen cropped patch on the fly as opposed to pre-computing the image features once and reusing them.

For the caption encoder, we use a GRU similar to the one used in Kiros et al. (2014). We set the dimensionality of the GRU, $D_\psi$, and the joint embedding space, $D$, to $1024$. The dimensionality of the word embeddings that are input to the GRU is set to $300$.

We further note that in Kiros et al. (2014), the caption embedding is normalized, while the image embedding is not. Normalization of both vectors means that the similarity function is cosine similarity. In *VSE++* we normalize both vectors. Not normalizing the image embedding changes the importance of samples. In our experiments, not normalizing the image embedding helped the baseline, *VSE0*, to find a better solution. However, *VSE++* is not significantly affected by this normalization.

### 3.1 DATASETS

We evaluate our method on the Microsoft COCO dataset (Lin et al. (2014)) and the Flickr30K dataset (Young et al. (2014)). Flickr30K has a standard $30,000$ images for training. Following Karpathy & Fei-Fei (2015), we use $1000$ images for validation and $1000$ images for testing. We also use the splits of Karpathy & Fei-Fei (2015) for MS-COCO. In this split, the training set contains $82,783$ images, $5000$ validation and $5000$ test images. However, there are also $30,504$ images that were originally in the validation set of MS-COCO but have been left out in this split. We refer to this set as *rV*. Some papers use *rV* for training ($113,287$ training images in total) to further improve accuracy. We report results using both training sets. Each image comes with $5$ captions. The results are reported by either averaging over $5$ folds of $1K$ test images or testing on the full $5K$ test images.

### 3.2 DETAILS OF TRAINING

We use the Adam optimizer Kingma & Ba (2014) to train the models. We train models for at most $30$ epochs. Except for fine-tuned models, we start training with learning rate $0.0002$ for $15$ epochs and then lower the learning rate to $0.00002$ for another $15$ epochs. The fine-tuned models are trained by taking a model that is trained for $30$ epochs with a fixed image encoder and then training it for $15$

| # | Model | Trainset | Caption Retrieval | | | Image Retrieval | | |
|---|---|---|---|---|---|---|---|---|
| | | | **R@1** | **R@10** | **Med r** | **R@1** | **R@10** | **Med r** |
| | | **1K Test Images** | | | | | | |
| 1.1 | *UVS* (Kiros et al. (2014), GitHub) | *1C* (1 fold) | 43.4 | 85.8 | 2 | 31.0 | 79.9 | 3 |
| 1.2 | *Order* (Vendrov et al. (2015)) | *10C+rV* | 46.7 | 88.9 | 2.0 | 37.9 | 85.9 | 2.0 |
| 1.3 | Embedding Network (Wang et al. (2017)) | ? | 50.4 | 69.4 | - | 39.8 | 86.6 | - |
| 1.4 | sm-LSTM (Huang et al. (2016)) | ? | 53.2 | 91.5 | **1** | 40.7 | 87.4 | 2 |
| 1.5 | 2WayNet (Eisenschtat & Wolf (2016)) | ? | 55.8 | - | - | 39.7 | - | - |
| 1.6 | *VSE++* | *1C* (1 fold) | 43.6 | 84.6 | 2.0 | 33.7 | 81.0 | 3.0 |
| 1.7 | *VSE++* | *RC* | 49.0 | 88.4 | 1.8 | 37.1 | 83.8 | 2.0 |
| 1.8 | *VSE++* | *RC+rV* | 51.9 | 90.4 | **1.0** | 39.5 | 85.6 | 2.0 |
| 1.9 | *VSE++ (fine-tuned)* | *RC+rV* | 57.2 | 93.3 | **1.0** | 45.9 | 89.1 | 2.0 |
| 1.10 | *VSE++ (ResNet152)* | *RC+rV* | 58.3 | 93.3 | **1.0** | 43.6 | 87.8 | 2.0 |
| 1.11 | *VSE++ (ResNet152, fine-tuned)* | *RC+rV* | **64.6** | **95.7** | **1.0** | **52.0** | **92.0** | **1.0** |
| | | **5K Test Images** | | | | | | |
| 1.12 | *Order* (Vendrov et al. (2015)) | *10C+rV* | 23.3 | 65.0 | 5.0 | 18.0 | 57.6 | 7.0 |
| 1.13 | *VSE++ (fine-tuned)* | *RC+rV* | 32.9 | 74.7 | 3.0 | 24.1 | 66.2 | 5.0 |
| 1.14 | *VSE++ (ResNet152, fine-tuned)* | *RC+rV* | **41.3** | **81.2** | **2.0** | **30.3** | **72.4** | **4.0** |

Table 1: Results of experiments on MS-COCO.

| # | Model | Trainset | Caption Retrieval | | | Image Retrieval | | |
|---|---|---|---|---|---|---|---|---|
| | | | **R@1** | **R@10** | **Med r** | **R@1** | **R@10** | **Med r** |
| 2.1 | *VSE0* | *1C* (1 fold) | 43.2 | 85.0 | 2.0 | 33.0 | 80.7 | 3.0 |
| 1.6 | *VSE++* | *1C* (1 fold) | 43.6 | 84.6 | 2.0 | 33.7 | 81.0 | 3.0 |
| 2.2 | *VSE0* | *RC* | 43.1 | 87.1 | 2.0 | 32.5 | 82.1 | 3.0 |
| 1.7 | *VSE++* | *RC* | 49.0 | 88.4 | 1.8 | 37.1 | 83.8 | 2.0 |
| 2.3 | *VSE0* | *RC+rV* | 46.8 | 89.0 | 1.8 | 34.2 | 83.6 | 2.6 |
| 1.8 | *VSE++* | *RC+rV* | 51.9 | 90.4 | **1.0** | 39.5 | 85.6 | 2.0 |
| 2.4 | *VSE0 (fine-tuned)* | *RC+rV* | 50.1 | 90.5 | 1.6 | 39.7 | 87.2 | 2.0 |
| 1.9 | *VSE++ (fine-tuned)* | *RC+rV* | 57.2 | 93.3 | **1.0** | 45.9 | 89.1 | 2.0 |
| 2.5 | *VSE0 (ResNet152)* | *RC+rV* | 52.7 | 91.8 | 1.0 | 36.0 | 85.5 | 2.2 |
| 1.10 | *VSE++ (ResNet152)* | *RC+rV* | 58.3 | 93.3 | **1.0** | 43.6 | 87.8 | 2.0 |
| 2.6 | *VSE0 (ResNet152, fine-tuned)* | *RC+rV* | 56.0 | 93.5 | 1.0 | 43.7 | 89.7 | 2.0 |
| 1.11 | *VSE++ (ResNet152, fine-tuned)* | *RC+rV* | **64.6** | **95.7** | **1.0** | **52.0** | **92.0** | **1.0** |

Table 2: The effect of data augmentation and fine-tuning. We copy the relevant results for *VSE++* from Table 1 to enable an easier comparison. Notice that applying all the modifications with the exception the *VSE0* model reaches $56.0\%$ for $R@1$, while *VSE++* achieves $64.6\%$.

epochs with a learning rate of $0.00002$. We set the margin to $0.2$ for most of the experiments. We use a mini-batch size of $128$ in all our experiments. Notice that since the size of the training set for different models is different, the actual number of iterations in each epoch can vary. For evaluation on the test set, we tackle over-fitting by choosing the snapshot of the model that performs best on the validation set. The best snapshot is selected based on the sum of the recalls on the validation set.

### 3.3 Results on MS-COCO

The results on the MS-COCO dataset are presented in Table 1. To understand the effect of training and algorithmic variations we report ablation studies for the baseline *VSE0* (see Table 2). Our best result with *VSE++* is achieved by using ResNet152 and fine-tuning the image encoder (row 1.11), where we see $21.2\%$ improvement in R@1 for caption retrieval and $21\%$ improvement in R@1 for image retrieval compared to *UVS* (rows 1.1 and 1.11). Notice that using ResNet152 and fine-tuning can only lead to $12.6\%$ improvement using the *VSE0* formulation (rows 2.6 and 1.1), while our *MH* loss function brings a significant gain of $8.6\%$ (rows 1.11 and 2.6).

Comparing *VSE++ (ResNet152, fine-tuned)* to the current state-of-the-art on MS-COCO, *2WayNet* (row 1.11 and row 1.5), we see $8.8\%$ improvement in R@1 for caption retrieval and compared to *sm-LSTM* (row 1.11 and row 1.4), $11.3\%$ improvement in image retrieval. We also report results on the full $5K$ test set of MS-COCO in rows 1.13 and 1.14.

*Effect of the training set.* We compare *VSE0* and *VSE++* by incrementally improving the training data. Comparing the models trained on *1C* (rows 1.1 and 1.6), we only see $2.7\%$ improvement in R@1 for image retrieval but no improvement in caption retrieval performance. However, when we train using *RC* (rows 1.7 and 2.2) or *RC+rV* (rows 1.8 and 2.3), we see that *VSE++* gains an

| # | Model | Trainset | Caption Retrieval | | | Image Retrieval | | |
|---|---|---|---|---|---|---|---|---|
| | | | **R@1** | **R@10** | **Med r** | **R@1** | **R@10** | **Med r** |
| 3.1 | *UVS* (Kiros et al. (2014)) | *1C* | 23.0 | 62.9 | 5 | 16.8 | 56.5 | 8 |
| 3.2 | *UVS* (GitHub) | *1C* | 29.8 | 70.5 | 4 | 22.0 | 59.3 | 6 |
| 3.3 | Embedding Network (Wang et al. (2017)) | ? | 40.7 | 79.2 | - | 29.2 | 71.7 | - |
| 3.4 | DAN (Nam et al. (2016)) | ? | 41.4 | 82.5 | 2 | 31.8 | 72.5 | 3 |
| 3.5 | sm-LSTM (Huang et al. (2016)) | ? | 42.5 | 81.5 | 2 | 30.2 | 72.3 | 3 |
| 3.6 | 2WayNet (Eisenschtat & Wolf (2016)) | ? | 49.8 | - | - | 36.0 | - | - |
| 3.7 | DAN (ResNet152) (Nam et al. (2016)) | ? | **55.0** | **89.0** | **1** | **39.4** | **79.1** | **2** |
| 3.8 | *VSE0* | *1C* | 29.8 | 71.9 | 3.0 | 23.0 | 61.0 | 6.0 |
| 3.9 | *VSE0* | *RC* | 31.6 | 71.7 | 4.0 | 21.6 | 63.8 | 5.0 |
| 3.10 | *VSE++* | *1C* | 31.9 | 68.0 | 4.0 | 23.1 | 60.7 | 6.0 |
| 3.11 | *VSE++* | *RC* | 38.6 | 74.6 | 2.0 | 26.8 | 66.8 | 4.0 |
| 3.12 | *VSE++ (fine-tuned)* | *RC* | 41.3 | 77.9 | 2.0 | 31.4 | 71.2 | 3.0 |
| 3.13 | *VSE++ (ResNet152)* | *RC* | 43.7 | 82.1 | 2.0 | 32.3 | 72.1 | 3.0 |
| 3.14 | *VSE++ (ResNet152, fine-tuned)* | *RC* | 52.9 | 87.2 | 1.0 | **39.6** | **79.5** | **2.0** |

Table 3: Results on the Flickr30K dataset.

improvement of 5.9% and 5.1%, respectively, in R@1 for caption retrieval compared to *VSE0*. This shows that *VSE++* can better exploit the additional data.

*Effect of a better image encoding*. We also investigate the effect of a better image encoder on the models. Row 1.9 and row 2.4 show the effect of fine-tuning the VGG19 image encoder. We see that the gap between *VSE0* and *VSE++* increases to 6.1%. If we use ResNet152 instead of VGG19 (row 1.10 and row 2.5), the gap is 5.6%. As for our best result, if we use ResNet152 and also fine-tune the image encoder (row 1.11 and row 2.6) the gap becomes 8.6%. The increase in the performance gap shows that the improved loss of *VSE++* can better guide the optimization when a more powerful image encoder is used.

### 3.4 Results on Flickr30K

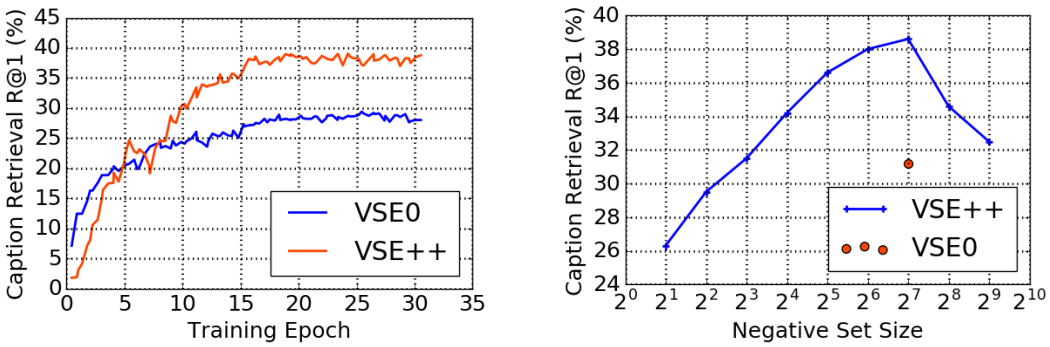
Tables 3 summarizes the performance on Flickr30K. We obtain 23.1% improvement in R@1 for caption retrieval and 17.6% improvement in R@1 for image retrieval (rows 3.1 and 3.14). We observed that *VSE++* over-fits when trained with the pre-computed features of *1C*. The reason is potentially the limited size of the Flickr30K training set. As explained in Sec. 3.2, we select a snapshot of the model before over-fitting occurs, based on performance with the validation set. Over-fitting does not occur when the model is trained using the *RC* training data. Our results show the improvements incurred by our *MH* loss persist across datasets, as well as across models.

### 3.5 Behavior of Loss Functions

We have observed that the *MH* loss can take a few epochs to 'warm-up' during training. Fig. 2(a) depicts such behavior on the Flickr30K dataset using *RC*. One can see that the *SH* loss starts off faster, but after approximately 5 epochs *MH* loss surpasses *SH* loss. To explain this, the *MH* loss depends on a smaller set of triplets compared to the *SH* loss. At the beginning of the training, there is so much that the model has to learn. However, the gradient of the *MH* loss, may only be influenced by a small set of triples. As such, it can take longer to train a model with the *MH* loss. We explored a simple form of curriculum learning (Bengio et al. (2009)) to speed-up the training. We start training with the *SH* loss for a few epochs, then switch to the *MH* loss for the rest of the training. However, it did not perform better than training solely with the *MH* loss.

### 3.6 Effect of Negative Set Size on *MH* Loss

In practice, our *MH* loss searches for the hardest negative only within each mini-batch at each iteration. To explore the impact of this approximation we examined how performance depends on the effective sample size over which we searched for negatives (while keeping the mini-batch size fixed at 128). In the extreme case, when the negative set is the training set, we get the hardest negatives in the entire training set. As discussed in Sec. 2.3, sampling a negative set smaller than the training set can potentially be more robust to label errors.




(a) (b)

Figure 2: Analysis of the behavior of the *MH* loss on the Flickr30K dataset training with *RC*. Fig. (a) compares the *SH* loss to the *MH* loss (Table 3, row 3.9 and row 3.11). Notice that, in the first 5 epochs the *SH* loss achieves a better performance, however, from there-on the *MH* loss leads to much higher recall rates. Fig. (b) shows the effect of the negative set size on the R@1 performance.

| # | **Model** | Caption Retrieval | | | Image Retrieval | | |
|---|---|---|---|---|---|---|---|
| | | **R@1** | **R@10** | **Med r** | **R@1** | **R@10** | **Med r** |
| | | **1K Test Images** | | | | | |
| 4.1 | *Order* (Vendrov et al. (2015)) | 46.7 | 88.9 | 2.0 | 37.9 | 85.9 | 2.0 |
| 4.2 | *VSE0* | 49.5 | 90.0 | 1.8 | 38.1 | 85.1 | 2.0 |
| 4.3 | *Order0* | 48.5 | 90.3 | 1.8 | 39.6 | 86.7 | 2.0 |
| 4.4 | *VSE++* | 51.3 | 91.0 | 1.2 | 40.1 | 86.1 | 2.0 |
| 4.5 | *Order++* | **53.0** | **91.9** | **1.0** | **42.3** | **88.1** | 2.0 |

Table 4: Comparison on MS-COCO. Training set for all the rows is *10C+rV*.

Fig. 2(b) shows the effect of the negative sample size on the *MH* Loss function. We compare the caption retrieval performance for different negative set sizes varied from 2 to 512. In practice, for negative set sizes smaller than the mini-batch size, 128, we randomly sample the negative set from the mini-batch. In other cases where the mini-batch size is smaller than the negative set, we randomly sample the mini-batch from the negative set. We observe that on this dataset, the optimal negative set size is around 128. Interestingly, for negative sets as small as 2, R@1 is slightly below *VSE0*. To understand this, note that the *SH* loss is still over a large sample size which has a relatively high probability of containing hard negatives. For large negative sets, the model takes longer to train for the first epochs. Using the negative set size 512, the performance dropped. This can be due to the small size of the dataset and the increase in the probability of sampling the hardest negative and outliers. Even though the performance drops with larger mini-batch sizes, it still performs better than the *SH* loss.

### 3.7 Improving Order Embeddings

Given the simplicity of our approach, our proposed loss function can complement the recent approaches that use more sophisticated model architectures or similarity functions. Here we demonstrate the benefits of the *MH* loss by applying it to another approach to joint embeddings called order-embeddings Vendrov et al. (2015). The main difference with the formulation above is the use of an asymmetric similarity function, i.e., $s(i, c) = -\|\max(0, g(c; W_g, \theta_\psi) - f(i; W_f, \theta_\phi))\|^2$. Again, we simply replace their use of the *SH* loss by our *MH* loss.

Like their experimental setting, we use the training set *10C+rV*. For our *Order++*, we use the same learning schedule and margin as our other experiments. However, we use their training settings to train *Order0*. We start training with a learning rate of 0.001 for 15 epochs and lower the learning rate to 0.0001 for another 15 epochs. Like Vendrov et al. (2015) we use a margin of 0.05. Additionally, Vendrov et al. (2015) takes the absolute value of embeddings before computing the similarity function which we replicate only for *Order0*.

**GT**: Two elephants are standing by the trees in the wild.

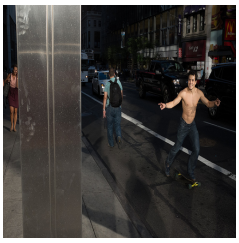
***VSE0***: [9] Three elephants kick up dust as they walk through the flat by the bushes.

***VSE++***: [1] A couple elephants walking by a tree after sunset.

**GT**: A large multi layered cake with candles sticking out of it.

***VSE0***: [1] A party decoration containing flowers, flags, and candles.

***VSE++***: [1] A party decoration containing flowers, flags, and candles.



**GT**: The man is walking down the street with no shirt on.

***VSE0***: [24] A person standing on a skate board in an alley.

***VSE++***: [10] Two young men are skateboarding on the street.

**GT**: A row of motorcycles parked in front of a building.

***VSE0***: [2] a parking area for motorcycles and bicycles along a street

***VSE++***: [1] A number of motorbikes parked on an alley

**GT**: some skateboarders doing tricks and people watching them

***VSE0***: [39] Young skateboarder displaying skills on sidewalk near field.

***VSE++***: [3] Two young men are outside skateboarding together.

**GT**: a brown cake with white icing and some walnut toppings

***VSE0***: [6] A large slice of angel food cake sitting on top of a plate.

***VSE++***: [16] A baked loaf of bread is shown still in the pan.

**GT**: A woman holding a child and standing near a bull.

***VSE0***: [1] A woman holding a child and standing near a bull.

***VSE++***: [1] A woman holding a child looking at a cow.

**GT**: A woman in a short pink skirt holding a tennis racquet.

***VSE0***: [6] A man playing tennis and holding back his racket to hit the ball.

***VSE++***: [1] A woman is standing while holding a tennis racket.

Figure 3: Examples of test images and the top 1 retrieved captions for *VSE0* and *VSE++* (ResNet)-finetune. The value in brackets is the rank of the highest ranked ground-truth caption. GT is a sample from the ground-truth captions.

Table 4 reports the results when the *SH* loss is replaced by the *MH* loss. We replicate their results using our *Order0* formulation and get slightly better results (row 4.1 and row 4.3). We observe $4.5\%$ improvement from *Order0* to *Order++* in R@1 for caption retrieval (row 4.3 and row 4.5). Compared to the improvement from *VSE0* to *VSE++*, where the improvement on the *10C+rV* training set is $1.8\%$, we gain an even higher improvement here. This shows that the *MH* loss can potentially improve numerous similar loss functions used in retrieval and ranking tasks.

## 4 Conclusion

This paper focused on learning visual-semantic embeddings for cross-modal, image-caption retrieval. Inspired by structured prediction, we proposed a new loss based on violations incurred by relatively hard negatives compared to current methods that used expected errors (Kiros et al. (2014); Vendrov et al. (2015)). We performed experiments on the MS-COCO and Flickr30K datasets and showed that our proposed loss significntly improves performance on these datasets. We observed that the improved loss can better guide a more powerful image encoder, ResNet152, and also guide better when fine-tuning an image encoder. With all modifications, our *VSE++* model achieves state-of-the-art performance on the MS-COCO dataset, and is slightly below the best recent model on the Flickr30K dataset. Our proposed loss function can be used to train more sophisticated models that have been using a similar ranking loss for training.

## REFERENCES


Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra. Vqa: Visual question answering. *International Journal of Computer Vision*, 123(1):4–31, 2017. 1

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM, 2009. 6

Olivier Chapelle, Quoc Le, and Alex Smola. Large margin optimization of ranking measures. In *NIPS workshop: Machine learning for Web search*, 2007. 1

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010. 3

Aviv Eisenschtat and Lior Wolf. Linking image and text with 2-way nets. *arXiv preprint arXiv:1608.07973*, 2016. 3, 5, 6

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 3

Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8. IEEE, 2007. 3

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pp. 2121–2129, 2013. 1

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pp. 770–778, 2016. 2, 4

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013. 1

Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal lstm. *arXiv preprint arXiv:1611.05588*, 2016. 2, 5, 6

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE CVPR*, pp. 3128–3137, 2015. 1, 2, 3, 4

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv preprint arXiv:1412.6980*, 2014. 4

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 1, 2, 3, 4, 5, 6, 8

Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *IEEE CVPR*, pp. 4437–4446, 2015. 3, 4

Quoc V. Le and Alexander J. Smola. Direct optimization of ranking measures. *CoRR*, abs/0704.3359, 2007. URL http://arxiv.org/abs/0704.3359. 1

Hang Li. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 7(3):1–121, 2014. 1

Sijin Li, Weichen Zhang, and Antoni B Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2848–2856, 2015a. 1

Yangyan Li, Hao Su, Charles Ruizhongtai Qi, Noa Fish, Daniel Cohen-Or, and Leonidas J Guibas. Joint embeddings of shapes and images via cnn image purification. *ACM Trans. Graph.*, 34(6): 234–1, 2015b. 1

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*, pp. 740–755. Springer, 2014. 4

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 1

Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. *arXiv preprint arXiv:1611.00471*, 2016. 2, 6

Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58, 2016a. 1

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016b. 1

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 4

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Association for Computational Linguistics (ACL)*, 2:207–218, 2014. 2, 3

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484, 2005. 3

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*, 2015. 4, 5, 7, 8

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *arXiv preprint arXiv:1704.03470*, 2017. 2, 5, 6

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Association for Computational Linguistics (ACL)*, 2:67–78, 2014. 4

Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1169–1176. ACM, 2009. 3

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015. 2

C Lawrence Zitnick, Aishwarya Agrawal, Stanislaw Antol, Margaret Mitchell, Dhruv Batra, and Devi Parikh. Measuring machine intelligence through visual question answering. *arXiv preprint arXiv:1608.08716*, 2016. 1

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1393–1398, 2013. 1

# Appendix

## A PROBABILITY OF SAMPLING THE HARDEST NEGATIVE

Let $S = \{(i_n, c_n)\}_{n=1}^{N}$ denote a training set of image-caption pairs, and let $C = \{c_n\}$ denote the set of captions. Suppose we draw $M$ samples in a mini-batch, $Q = \{(i_m, c_m)\}_{m=1}^{M}$, from $S$. Let the permutation, $\pi_m$, on $C$ refer to the rankings of captions according to the similarity function $s(i_m, c_n)$ for $c_n \in S \setminus \{c_m\}$. We can assume permutations, $\pi_m$, are uncorrelated.

Given a query image, $i_m$, we are interested in the probability of getting no captions from the 90th percentile of $\pi_m$ in the mini-batch. Assuming IID samples, this probability is simply $.9^{(M-1)}$, the probability that no sample in the mini-batch is from the 90th percentile. This probability tends to zero exponentially fast and it goes below $1\%$ for $M \geq 44$. Hence, for large enough mini-batch size, with probability close to 1, we sample negative captions in the mini-batch that are harder than $90\%$ of the training set.

The same probability for the 99.9th percentile of $\pi_m$ tends to zero much more slowly. The same probability goes below $1\%$ for $M \geq 6905$ which is a relatively large mini-batch size. This analysis shows that while we can get strong signals just by randomly sampling mini-batches, we are potentially robust to outliers such as negative captions that better describe an image compared to the ground-truth caption.