

VSE++: Improving Visual-Semantic Embeddings with Hard Negatives

Fartash Faghri¹
faghri@cs.toronto.edu

David J. Fleet¹
fleet@cs.toronto.edu

Jamie Ryan Kiros*²
kiros@google.com

Sanja Fidler¹
fidler@cs.toronto.edu

¹ Department of Computer Science,
University of Toronto
and
Vector Institute

² Google Brain Toronto

Abstract

We present a new technique for learning visual-semantic embeddings for cross-modal retrieval. Inspired by hard negative mining, the use of hard negatives in structured prediction, and ranking loss functions, we introduce a simple change to common loss functions used for multi-modal embeddings. That, combined with fine-tuning and use of augmented data, yields significant gains in retrieval performance. We showcase our approach, VSE++, on 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 (at R@1).

Introduction

Joint embeddings enable a wide range of tasks in image, video and language understanding. Examples include shape-image embeddings ([38]) for shape inference, bilingual word embeddings ([38]), human pose-image embeddings for 3D pose inference ([9]), fine-grained recognition ([25]), zero-shot learning ([9]), and modality conversion via synthesis ([25, 26]). 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 domain structure, where location and often direction are semantically meaningful.

Visual-semantic embeddings have been central to image-caption retrieval and generation [3, 3], and visual question-answering [22]. 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 ([1, 37]). For image synthesis from text, one could map from text to the joint embedding space, and then back to image space ([25, 26]).

Here we focus on visual-semantic embeddings for cross-modal retrieval; i.e. the retrieval of images given captions, or of captions for a query image. As is common in 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 ([10]).

The basic problem is one of ranking; the correct target(s) should be closer to the query than other items in the corpus, not unlike *learning to rank* problems (e.g., [18]), and max-margin structured prediction [3, 14]. The formulation and model architecture in this paper are most closely related to those of [15], learned with a triplet ranking loss. In contrast to that work, we advocate a novel loss, the use of augmented data, and fine-tuning, which, together, produce a significant increase in caption retrieval performance over the baseline ranking loss on well-known benchmark data. We outperform the best reported result on MS-COCO by almost 9%. We also show that the benefit of a more powerful image encoder, with fine-tuning, is amplified with the use of our stronger loss function. We refer to our model as *VSE++*. To ensure reproducibility, our code is publicly available ¹.

Our main contribution is to incorporate hard negatives in the loss function. This was inspired by the use of hard negative mining in classification tasks ([6, 7, 23]), and by the use of hard negatives for improving image embeddings for face recognition ([27, 33]). Minimizing a loss function using hard negative mining is equivalent to minimizing a modified non-transparent loss function with uniform sampling. We extend the idea with the explicit introduction of hard negatives in the loss for multi-modal embeddings, without any additional cost of mining.

We also note that our formulation complements other recent articles that propose new architectures or similarity functions for this problem. To this end, we demonstrate improvements to [6]. Among other methods that could be improved with a modified loss, [32] propose an embedding network to fully replace the similarity function used for the ranking loss. An attention mechanism on both images and captions is used by [24], where the authors sequentially and selectively focus on a subset of words and image regions to compute the similarity. In [12], the authors use a multi-modal context-modulated attention mechanism to compute the similarity between images and captions. Our proposed loss function and triplet sampling could be extended and applied to other such problems.

2 Learning Visual-Semantic Embeddings

For image-caption retrieval the query is a caption and the task is to retrieve the most relevant image(s) from a database. Alternatively, the query may be an image, and the task is to retrieve relevant captions. The goal is to maximize recall at K ($R@K$), i.e., 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 $s(i, c)$. Likewise for image retrieval. In what follows the similarity function is defined on the joint embedding space. This differs from other formulations, such as [32], which use a similarity network to directly classify an image-caption pair as matching or non-matching.

¹<https://github.com/fartashf/vsepp>

2.1 Visual-Semantic Embedding

Let $\phi(i; \theta_\phi) \in \mathbb{R}^{D_\phi}$ be a feature-based representation computed from image i (e.g. the representation before logits in VGG19 ([28]) or ResNet152 ([11])). Similarly, let $\psi(c; \theta_\psi) \in \mathbb{R}^{D_\psi}$ be a representation of caption c in a caption embedding space (e.g. a GRU-based text encoder). Here, θ_ϕ and θ_ψ denote model parameters for the respective mappings to these initial image and caption representations.

Then, let the mappings into the *joint embedding space* be defined by linear projections:

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

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

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. Finally, we define the similarity function in the joint embedding space to be the usual 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 θ_ϕ in θ .

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

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

where $\ell(i_n, c_n)$ is a suitable loss function for a single training exemplar. Inspired by the use of a triplet loss for image retrieval (e.g., [4, 8]), recent approaches to joint visual-semantic embeddings have used a hinge-based triplet ranking loss [13, 15, 29, 36]:

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

where α serves as a margin parameter, and $[x]_+ \equiv \max(x, 0)$. This hinge loss comprises two symmetric terms. The first sum is taken over all negative captions \hat{c} , given query i . The second is taken over all 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 negative, by the margin α , the hinge loss is zero. In practice, for computational efficiency, rather than summing over all negatives in the training set, it is common to only sum over (or randomly sample) the negatives in a mini-batch of stochastic gradient descent [13, 15, 29]. The runtime complexity of computing this loss approximation is quadratic in the number of image-caption pairs in a mini-batch.

Of course there are other loss functions that one might consider. One is a pairwise hinge loss in which elements of positive pairs are encouraged to lie within a hypersphere of radius ρ_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., [6, 17]). 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.

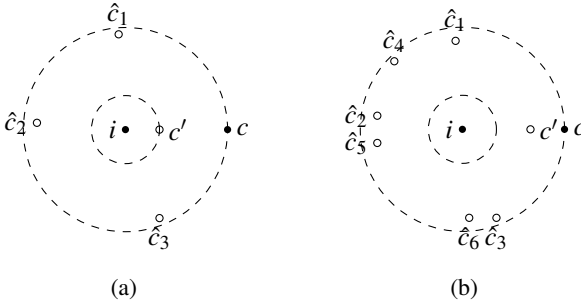


Figure 1: An illustration of typical positive pairs and the nearest negative samples. Here assume similarity score is the negative distance. Filled circles show a positive pair (i, c) , while empty circles are negative samples for the query i . The dashed circles on the two sides are drawn at the same 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).

2.2 Emphasis on Hard Negatives

Inspired by common loss functions used in structured prediction ([10, 30, 35]), 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 define our loss as

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

Like Eq. 5, this 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' . We refer to the loss in Eq. 6 as *Max of Hinges* (MH) loss, and the loss in Eq. 5 as *Sum of Hinges* (SH) loss. There is a spectrum of loss functions from the *SH* loss to the *MH* loss. In the *MH* loss, the winner takes all the gradients, where instead we use re-weighted gradients of all the triplets. We only discuss the *MH* loss as it was empirically found to perform the best.

One case in which the *MH* loss is superior to *SH* is when multiple negatives with small violations combine to dominate the *SH* loss. For example, Fig. 1 depicts a positive pair together with two sets of negatives. In Fig. 1(a), a single negative is too close to the query, which may require a significant change to the mapping. However, any training step that pushes the hard negative away, might cause a number of small violating negatives, as in Fig. 1(b). Using the *SH* loss, these ‘new’ negatives 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, which focuses on the hardest negative.

For computational efficiency, instead of finding the hardest negatives in the entire training set, we find them within each mini-batch. This has the same quadratic complexity as the complexity of the *SH* loss. With random sampling of the mini-batches, this approximation 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.

#	Model	Trainset	Caption Retrieval				Image Retrieval			
			R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
1K Test Images										
1.1	<i>UVS</i> ([1]), GitHub)	<i>IC</i> (1 fold)	43.4	75.7	85.8	2	31.0	66.7	79.9	3
1.2	<i>Order</i> ([2])	<i>10C+rV</i>	46.7	-	88.9	2.0	37.9	-	85.9	2.0
1.3	Embedding Net ([3])	<i>10C+rV</i>	50.4	79.3	69.4	-	39.8	75.3	86.6	-
1.4	sm-LSTM ([4])	?	53.2	83.1	91.5	1	40.7	75.8	87.4	2
1.5	2WayNet ([5])	<i>10C+rV</i>	55.8	75.2	-	-	39.7	63.3	-	-
1.6	<i>VSE++</i>	<i>IC</i> (1 fold)	43.6	74.8	84.6	2.0	33.7	68.8	81.0	3.0
1.7	<i>VSE++</i>	<i>RC</i>	49.0	79.8	88.4	1.8	37.1	72.2	83.8	2.0
1.8	<i>VSE++</i>	<i>RC+rV</i>	51.9	81.5	90.4	1.0	39.5	74.1	85.6	2.0
1.9	<i>VSE++ (FT)</i>	<i>RC+rV</i>	57.2	86.0	93.3	1.0	45.9	79.4	89.1	2.0
1.10	<i>VSE++ (ResNet)</i>	<i>RC+rV</i>	58.3	86.1	93.3	1.0	43.6	77.6	87.8	2.0
1.11	<i>VSE++ (ResNet, FT)</i>	<i>RC+rV</i>	64.6	90.0	95.7	1.0	52.0	84.3	92.0	1.0
5K Test Images										
1.12	<i>Order</i> ([2])	<i>10C+rV</i>	23.3	-	65.0	5.0	18.0	-	57.6	7.0
1.13	<i>VSE++ (FT)</i>	<i>RC+rV</i>	32.9	61.7	74.7	3.0	24.1	52.8	66.2	5.0
1.14	<i>VSE++ (ResNet, FT)</i>	<i>RC+rV</i>	41.3	71.1	81.2	2.0	30.3	59.4	72.4	4.0

Table 1: Results of experiments on MS-COCO.

2.3 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, π_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, π_m , are uncorrelated.

Given a query image, i_m , we are interested in the probability of getting no captions from the 90th percentile of π_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, falling below 1% for $M \geq 44$. Hence, for large enough mini-batches, with high probability we sample negative captions that are harder than 90% of the entire training set. The probability for the 99.9th percentile of π_m tends to zero more slowly; it falls below 1% for $M \geq 6905$, which is a relatively large mini-batch.

While we get strong signals by randomly sampling negatives within mini-batches, such sampling also provides some robustness to outliers, such as negative captions that better describe an image compared to the ground-truth caption. Mini-batches as small as 128 can provide strong enough training signal and robustness to label errors. Of course by increasing the mini-batch size, we get harder negative examples and possibly a stronger training signal. However, by increasing the mini-batch size, we lose the benefit of SGD in finding good optima and exploiting the gradient noise. This can lead to getting stuck in local optima or as observed by [[7](#)], extremely long training time.

3 Experiments

Below we perform experiments with our approach, *VSE++*, comparing it to a baseline formulation with *SH* loss, denoted *VSE0*, and other state-of-the-art approaches. Essentially, the baseline formulation, *VSE0*, is similar to that in [[3](#)], denoted *UVS*.

We experiment with two image encoders: VGG19 by [[28](#)] and ResNet152 by [[10](#)]. In what follows, 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_ϕ , is 4096 for VGG19 and 2048 for ResNet152.

In more detail, we first resize the image to 256×256 , and then use either a single crop of size 224×224 or the mean of feature vectors for multiple crops of similar size. We refer to training with one center crop as *IC*, and training with 10 crops at fixed locations as *IOC*. These image features can be pre-computed once and reused. We also experiment with using a single random crop, denoted by *RC*. For *RC*, image features are computed on the fly. Recent works have mostly used *RC/IOC*. In our preliminary experiments, we did not observe significant differences between *RC/IOC*. As such, we perform most experiments with *RC*.

For the caption encoder, we use a GRU similar to the one used in [15]. We set the dimensionality of the GRU, D_ψ , 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 [15], 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 ([21]) and the Flickr30K dataset ([24]). Flickr30K has a standard 30,000 images for training. Following [13], we use 1000 images for validation and 1000 images for testing. We also use the splits of [13] 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 [14]. Models are trained 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 trained for 30 epochs with a fixed image encoder, and then training it for 15 epochs with a learning rate of 0.00002. We set the margin to 0.2 for most experiments. We use a mini-batch size of 128 in all 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

#	Model	Trainset	Caption Retrieval				Image Retrieval			
			R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
2.1	VSE0	IC (1 fold)	43.2	73.9	85.0	2.0	33.0	67.4	80.7	3.0
1.6	VSE++	IC (1 fold)	43.6	74.8	84.6	2.0	33.7	68.8	81.0	3.0
2.2	VSE0	RC	43.1	77.0	87.1	2.0	32.5	68.3	82.1	3.0
1.7	VSE++	RC	49.0	79.8	88.4	1.8	37.1	72.2	83.8	2.0
2.3	VSE0	RC+rV	46.8	78.8	89.0	1.8	34.2	70.4	83.6	2.6
1.8	VSE++	RC+rV	51.9	81.5	90.4	1.0	39.5	74.1	85.6	2.0
2.4	VSE0 (FT)	RC+rV	50.1	81.5	90.5	1.6	39.7	75.4	87.2	2.0
1.9	VSE++ (FT)	RC+rV	57.2	86.0	93.3	1.0	45.9	79.4	89.1	2.0
2.5	VSE0 (ResNet)	RC+rV	52.7	83.0	91.8	1.0	36.0	72.6	85.5	2.2
1.10	VSE++ (ResNet)	RC+rV	58.3	86.1	93.3	1.0	43.6	77.6	87.8	2.0
2.6	VSE0 (ResNet, FT)	RC+rV	56.0	85.8	93.5	1.0	43.7	79.4	89.7	2.0
1.11	VSE++ (ResNet, FT)	RC+rV	64.6	90.0	95.7	1.0	52.0	84.3	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 after applying all the modifications, VSE0 model reaches 56.0% for R@1, while VSE++ achieves 64.6%.

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++ (ResNet, FT) 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 IC (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 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 3.17). We observed that VSE++ over-fits when trained with the pre-computed features of IC. 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.

#	Model	Trainset	Caption Retrieval				Image Retrieval			
			R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
3.1	UVS ([10])	1C	23.0	50.7	62.9	5	16.8	42.0	56.5	8
3.2	UVS (GitHub)	1C	29.8	58.4	70.5	4	22.0	47.9	59.3	6
3.3	Embedding Net ([10])	10C	40.7	69.7	79.2	-	29.2	59.6	71.7	-
3.4	DAN ([10])	?	41.4	73.5	82.5	2	31.8	61.7	72.5	3
3.5	sm-LSTM ([10])	?	42.5	71.9	81.5	2	30.2	60.4	72.3	3
3.6	2WayNet ([10])	10C	49.8	67.5	-	-	36.0	55.6	-	-
3.7	DAN (ResNet) ([10])	?	55.0	81.8	89.0	1	39.4	69.2	79.1	2
3.8	VSE0	1C	29.8	59.8	71.9	3.0	23.0	48.8	61.0	6.0
3.9	VSE0	RC	31.6	59.3	71.7	4.0	21.6	50.7	63.8	5.0
3.10	VSE++	1C	31.9	58.4	68.0	4.0	23.1	49.2	60.7	6.0
3.11	VSE++	RC	38.6	64.6	74.6	2.0	26.8	54.9	66.8	4.0
3.12	VSE0 (FT)	RC	37.4	65.4	77.2	3.0	26.8	57.6	69.5	4.0
3.13	VSE++ (FT)	RC	41.3	69.1	77.9	2.0	31.4	60.0	71.2	3.0
3.14	VSE0 (ResNet)	RC	36.6	67.3	78.4	3.0	23.3	52.6	66.0	5.0
3.15	VSE++ (ResNet)	RC	43.7	71.9	82.1	2.0	32.3	60.9	72.1	3.0
3.16	VSE0 (ResNet, FT)	RC	42.1	73.2	84.0	2.0	31.8	62.6	74.1	3.0
3.17	VSE++ (ResNet, FT)	RC	52.9	80.5	87.2	1.0	39.6	70.1	79.5	2.0

Table 3: Results on the Flickr30K dataset.

3.5 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 [[51](#)]. 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 [[51](#)] we use a margin of 0.05. Additionally, [[51](#)] takes the absolute value of embeddings before computing the similarity function which we replicate only for *Order0*.

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.

#	Model	Caption Retrieval				Image Retrieval			
		R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
		1K Test Images							
4.1	<i>Order</i> ([51])	46.7	-	88.9	2.0	37.9	-	85.9	2.0
4.2	<i>VSE0</i>	49.5	81.0	90.0	1.8	38.1	73.3	85.1	2.0
4.3	<i>Order0</i>	48.5	80.9	90.3	1.8	39.6	75.3	86.7	2.0
4.4	<i>VSE++</i>	51.3	82.2	91.0	1.2	40.1	75.3	86.1	2.0
4.5	<i>Order++</i>	53.0	83.4	91.9	1.0	42.3	77.4	88.1	2.0

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

3.6 Behavior of Loss Functions

We observe that the *MH* loss can take a few epochs to ‘warm-up’ during training. Fig. 2 depicts such behavior on the Flickr30K dataset using *RC*. Notice 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. Early in training the gradient of the *MH* loss is influenced by a relatively small set of triples. As such, it can take more iterations to train a model with the *MH* loss. We explored a simple form of curriculum learning ([14]) 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 much better than training solely with the *MH* loss.

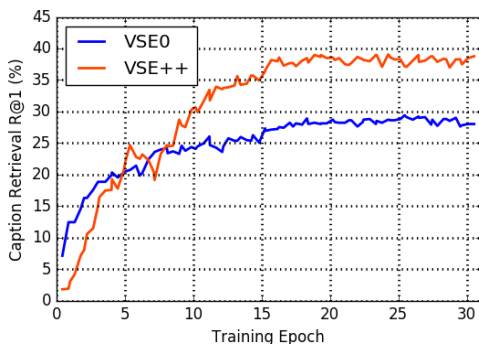


Figure 2: Analysis of the behavior of the *MH* loss on the Flickr30K dataset training with *RC*. This figure compares the *SH* loss to the *MH* loss (Table 3, row 3.9 and row 3.11). Notice that, in the first 5 epochs the *SH* loss achieves a better performance, however, from there-on the *MH* loss leads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

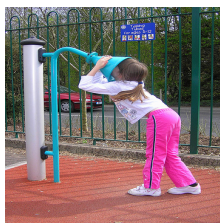
In [27], it is reported that with a mini-batch size of 1800, training is extremely slow. We experienced similar behavior with large mini-batches up to 512. However, mini-batches of size 128 or 256 exceeded the performance of the *SH* loss within the same training time.

3.7 Examples of Hard Negatives

Fig. 3 shows the hard negatives in a random mini-batch. These examples illustrate that hard negatives from a mini-batch can provide useful gradient information.

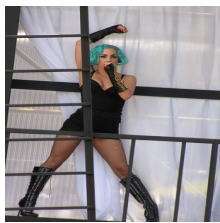
4 Conclusion

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 ([19, 31]). We performed experiments on the MS-COCO and Flickr30K datasets and showed that our proposed loss significantly 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



GT: A little girl wearing pink pants, pink and white tennis shoes and a white shirt with a little girl on it puts her face in a blue Talking Tube.

HN: [0.26] Blond boy jumping onto deck.



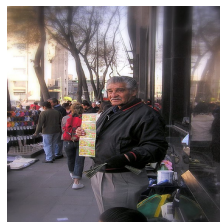
GT: A teal-haired woman in a very short black dress, pantyhose, and boots standing with right arm raised and left hand obstructing her mouth in microphone-singing fashion is standing.

HN: [0.08] Two dancers in azure appear to be performing in an alleyway.



GT: Two men, one in a dark blue button-down and the other in a light blue tee, are chatting as they walk by a small restaurant.

HN: [0.41] Two men with guitars strapped to their back stand on the street corner with two other people behind them.



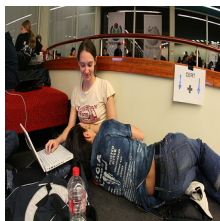
GT: A man wearing a black jacket and gray slacks, stands on the sidewalk holding a sheet with something printed on it in his hand.

HN: [0.26] Two men with guitars strapped to their back stand on the street corner with two other people behind them.



GT: There is a wall of a building with several different colors painted on it and in the distance one person sitting down and another walking.

HN: [0.06] A woman with luggage walks along a street in front of a large advertisement.



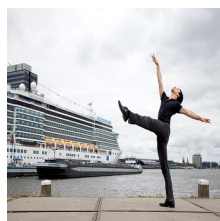
GT: A man is laying on a girl's lap, she is looking at him, she also has her hand on her notebook computer.

HN: [0.18] A woman sits on a carpeted floor with a baby.



GT: A young blond girl in a pink sweater, blue skirt, and brown boots is jumping over a puddle on a cloudy day.

HN: [0.51] An Indian woman is sitting on the ground, amongst drawings, rocks and shrubbery.



GT: One man dressed in black is stretching his leg up in the air, behind him is a massive cruise ship in the water.

HN: [0.24] A topless man straps surfboards on top of his blue car.

Figure 3: Examples from the Flickr30K training set along with their hard negatives in a random mini-batch according to the loss of a trained VSE++ model. The value in brackets is the cost of the hard negative and is in the range $[0, 2]$ in our implementation. HN is the hardest negative in a random sample of size 128. GT is the positive caption used to compute the cost of NG.

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.

Acknowledgements

This work was supported in part by funding to DF from NSERC Canada, the Vector Institute, and the Learning in Brains and Machines Program of the Canadian Institute for Advanced Research.

References

- [1] 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 (IJCV)*, 123(1):4–31, 2017.
- [2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pages 41–48. ACM, 2009.
- [3] Olivier Chapelle, Quoc Le, and Alex Smola. Large margin optimization of ranking measures. In *NIPS workshop: Machine learning for Web search*, 2007.
- [4] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research (JMLR)*, 11(Mar):1109–1135, 2010.
- [5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.
- [6] Aviv Eisenschat and Lior Wolf. Linking image and text with 2-way nets. July 2017.
- [7] 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 (PAMI)*, 32(9):1627–1645, 2010.
- [8] Andrea Frome, Yoram Singer, Fei Sha, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *International Conference in Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.
- [9] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Neural Information Processing Systems (NIPS)*, pages 2121–2129, 2013.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [11] 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.
- [12] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal lstm. 2017.
- [13] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.
- [14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

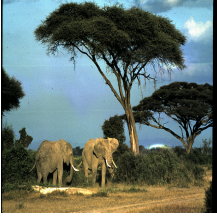
- [15] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. 2014.
- [16] Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4437–4446, 2015.
- [17] Quoc Le and Alexander Smola. Direct optimization of ranking measures. *arXiv preprint arXiv:0704.3359*, 2007.
- [18] Hang Li. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 7(3):1–121, 2014.
- [19] Sijin Li, Weichen Zhang, and Antoni B Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *International Conference in Computer Vision (ICCV)*, pages 2848–2856, 2015.
- [20] 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. 34(6):234–1, 2015.
- [21] 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 *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.
- [22] Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *International Conference in Computer Vision (ICCV)*, 2015.
- [23] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svm for object detection and beyond. In *International Conference in Computer Vision (ICCV)*, pages 89–96. IEEE, 2011.
- [24] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. July 2017.
- [25] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 49–58, 2016.
- [26] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. 2016.
- [27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.
- [28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

- [29] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. 2:207–218, 2014.
- [30] Ioannis Tsochantaris, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6(Sep):1453–1484, 2005.
- [31] Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. 2016.
- [32] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018.
- [33] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. 2017.
- [34] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. 2:67–78, 2014.
- [35] Chun-Nam John Yu and Thorsten Joachims. Learning structural svms with latent variables. In *International Conference on Machine Learning (ICML)*, pages 1169–1176. ACM, 2009.
- [36] 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 *International Conference in Computer Vision (ICCV)*, 2015.
- [37] C Lawrence Zitnick, Aishwarya Agrawal, Stanislaw Antol, Margaret Mitchell, Dhruv Batra, and Devi Parikh. Measuring machine intelligence through visual question answering. *AI Magazine*, 2016.
- [38] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1393–1398, 2013.

Appendix

A Examples of Hard Negatives

Fig. A.1 compares the outputs of *VSE++* and *VSE0* for a few examples.



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 A.1: Examples of MS-COCO 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.