2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-9346-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/WACV56688.2023.00220

Dissecting Deep Metric Learning Losses for Image-Text Retrieval

Hong Xuan, Xi (Stephen) Chen
Microsoft
{Hong.Xuan|Chen.Stephen}@microsoft.com

Abstract

Visual-Semantic Embedding (VSE) is a prevalent approach in image-text retrieval by learning a joint embedding space between the image and language modalities where semantic similarities would be preserved. The triplet loss with hard-negative mining has become the de-facto objective for most VSE methods. Inspired by recent progress in deep metric learning (DML) in the image domain which gives rise to new loss functions that outperform triplet loss, in this paper we revisit the problem of finding better objectives for VSE in image-text matching. Despite some attempts in designing losses based on gradient movement, most DML losses are defined empirically in the embedding space. Instead of directly applying these loss functions which may lead to sub-optimal gradient updates in model parameters, in this paper we present a novel Gradient-based Objective AnaLysis framework, or GOAL, to systematically analyze the combinations and reweighting of the gradients in existing DML functions. With the help of this analysis framework, we further propose a new family of objectives in the gradient space exploring different gradient combinations. In the event that the gradients are not integrable to a valid loss function, we implement our proposed objectives such that they would directly operate in the gradient space instead of on the losses in the embedding space. Comprehensive experiments have demonstrated that our novel objectives have consistently improved performance over baselines across different visual/text features and model frameworks. We also showed the generalizability of the GOAL framework by extending it to other models using triplet family losses including vision-language model with heavy cross-modal interactions and have achieved state-of-the-art results on the image-text retrieval tasks on COCO and Flick30K.

1. Introduction

Recognizing and describing the visual world with language is a basic human ability but still remains challenging for artificial intelligence. With recent advances in Deep Neural Networks, tremendous progress has been made in bridging the vision-language modalities. Visual-semantic embedding (VSE) [8, 15, 7] is one of the major topics to build a connection between images and natural language. It aims to map images and their descriptive text information into a joint space, such that a relevant pair of image and text should be mapped close to each other while an irrelevant pair of image and text should be mapped far from each other. In this paper, we focus on visual-semantic embedding for the task of image-text matching and retrieval, but our approach is generalizable to other image-text retrieval models using the triplet loss family [17, 4, 20, 40].

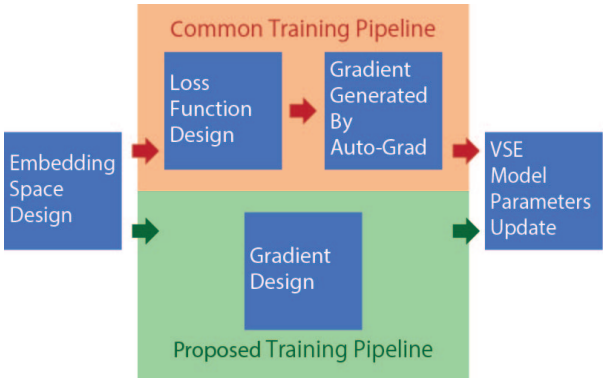

Figure 1. To realize a desired visual semantic embedding space, a common method is to design a loss function which can be calculated on deep learning platforms such as PyTorch or TensorFlow. The auto-grad mechanism on these platforms automatically calculates the gradients to update the model parameters to form a desired embedding space. In practice, the goal of visual semantic embedding is about optimizing the clustering or separation of feature points extracted from image and text, and the loss function is a somewhat indirect approach to reach that goal, while the gradient more directly affects the update of the embedding space. We propose a method to directly design the gradient to train models.

A VSE model usually consists of feature extractors for image and text, a feature aggregator [2], and an objective function during training. Despite significant advances of VSE in feature extractors [31, 6, 1] and feature aggregators [32, 2], there is less attention on the loss function for training the model. A hinge-based triplet ranking loss with hard-negative sampling [26, 7] has become the de-facto

2642-9381/23/$31.00 ©2023 IEEE
DOI 10.1109/WACV56688.2023.00220

training objective for many VSE approaches [17, 20, 41]. Few innovations have been made in designing the loss function for learning joint image-text embeddings since then.

On the other hand, designing deep metric learning (DML) losses has been well-studied for image-to-image retrieval. Many loss functions have been proposed to improve the training performance on image embedding tasks, showing that triplet loss is not optimal for general metric learning [37, 28, 33, 36, 29]. Early losses such as triplet loss and contrastive loss [26, 27] are defined with the intuition that positive pairs should be close while negative pairs should be apart in the embedding space. However, such defined loss functions may not lead to desirable gradients which can explicitly impact the update of model parameters. Some attempts have been made in defining the loss function to achieve desirable gradient updates [37, 29]. However, such approaches lack a systematic view and analysis of the combinations in gradients, and are only limited to integrable gradients so that the resulting losses are differentiable. Therefore, these loss functions may not be optimal and applicable to the image-text retrieval task.

Instead of directly applying established loss functions to VSE for image-text matching, in this paper we present Gradient-based Objective AnaLysis framework, or **GOAL**, a novel gradient-based analysis framework for the VSE problem. We firstly propose a new gradient framework to dissect the losses at the gradient level, and extract their key gradient elements. Then, we explore a new training idea to directly define the gradient to update the model in each training step instead of defining the loss functions, as shown in Figure 1. This new framework allows us to simply combine the key gradient elements in DML losses to form a family of new gradients and avoids the concern of integrating the gradient into a loss function. Finally, the new gradients continue to improve existing VSE performance on image-text retrieval tasks.

In brief, our contributions can be summarized as the following:

- We propose a general framework **GOAL** to comprehensively analyze the update of gradients of existing deep metric learning loss functions and apply this framework to help find better objectives for the VSE problem.
- We propose a new method to deal with image-text retrieval task directly by optimizing the model with a family of gradient objectives instead of using a loss function.
- We show consistent improvement over existing methods, achieving state-of-the-art results in image-text retrieval tasks on COCO datasets.

2. Related Work

Visual Semantic Embedding for Image-text matching There is a rich line of literature focused on mapping visual and text modalities to a joint semantic embedding space for image-text matching [8, 15, 7, 17, 35, 2]. VSE++ is proposed in [7] as a fundamental VSE schema where visual and text embeddings are pretrained separately then aggregated with AvgPool after being projected to a shared space, which later are jointly optimized by a triplet loss with hard-negative mining. Since then consistent advances have been made to improve visual and text feature extractors [11, 6, 12, 31, 5] and feature aggregators [14, 19, 32, 35]. In contrast to dominant use of spatial grids of the feature map as visual features, bottom-up attention [1] has been introduced to learn visual semantic embeddings for image-text matching, which is commonly realized by stacking the region representations from pretrained object detectors [17, 41]. [2] proposed Generalized Pooling Operators (GPO) to learn the best pooling strategy which outperforms approaches with complex feature aggregators. Inspired by the success of large-scale pretraining in language models [5, 21], there is a recent trend of performing task-agnostic vision-language pretraining (VLP) on massive image-text pairs for generic representations, then fine-tune on task-specific data and losses to achieve state-of-the-art results in downstream tasks including image-text retrieval [23, 30, 4, 20, 40]. However, as opposed to our proposed method, prevalent approaches choose to optimize the triplet loss as the de-facto objective for the image-text matching task. In this paper, we will strive to revisit the problem of finding better training objectives for visual semantic embeddings.

Deep Metric Learning is useful in extreme classification settings such as fine-grained recognition [28, 22, 34, 16, 26]. The goal is to train networks to map semantically related images to nearby locations and unrelated images to distant locations in an embedding space. There are many loss functions that have been proposed to solve the deep metric learning problem. Triplet loss function [13, 26] and its variants such as circle loss [29] form a triplet that contains anchor, positive and negative instances, where the anchor and positive instance share the same label, and anchor and negative instance share different labels. Pair-wise loss functions such as contrastive loss [10], binomial deviance loss [37], lifted structure loss [28] and multi-similarity loss [33] penalize when the distance is large between a pair of instances with the same labels and when the distance is small between a pair of instances with different labels. All these loss functions encourage the distance of positive images pairs to be smaller than the distance of the negative images pairs. Due to the fact that the training goal of DML is similar to VSE problem, in this paper, we borrow these loss design ideas of DML to improve the VSE problem.

Gradient Modification Recent works in DML such as Multi-Similarity Loss and Circle Loss [33, 29, 36] start with standard triplet loss formulations and adjust the gradients of loss functions to give clear improvements with very simple code modifications. These works all find explicit loss functions whose gradients are desirable. Other strategies start with a desired gradient weighting function and integrate the desired gradients to derive a loss function that comes with gradients of appropriate properties. This is often limited to simple weighting strategies, such as the simple linear form in [29] and simple gradient removal for positive pairs when triplets contain hard negative in [36], because it may be hard to find the loss function whose gradient is consistent with complex weighting strategies. The most related work is P2Sgrad [42], which analyzes the gradient in the family of margin-based softmax loss and directly modifies the gradient with the cosine similarity for better optimization. Comparing to P2Sgrad, our work focuses on the triplet loss and its variant loss functions.

The framework in this paper directly explores the space of desired gradient updates. By not limiting ourselves to designing a loss function with appropriate gradients, we can be more explicit in experimentally dissecting the effects of different parts of the gradient. Furthermore, we can recombine the gradient terms that are experimentally most useful in a form of gradient surgery [39] that very slightly alters existing algorithms to give improved performance.

3. Gradient-based Objective Design Framework

We define a collection of terms for how a batch of images and texts affect a network. Let $\mathbf{X}$ be a batch of input images, $\mathbf{Y}$ be a batch of input texts, $\mathbf{x}$ be the $L2$ normalized feature vectors of the images extracted with the image extractor, $\mathbf{y}$ be the $L2$ normalized feature vectors of the texts extracted with the text extractor, l be the loss value for the batch, θ be the parameters of the image extractor, ϕ be the parameters of the text extractor, η be the learning rate, $f_\theta(\cdot)$ be the mapping function of the image extractor, $g_\phi(\cdot)$ be the mapping function of the text extractor, and $L(\cdot)$ be loss function. In the forward training step, the expression is:

$$l = L(\mathbf{x}, \mathbf{y}), \text{ where } \mathbf{x} = f_\theta(\mathbf{X}) \text{ and } \mathbf{y} = g_\phi(\mathbf{Y}) \tag{1}$$

.

The image and text extractor weights are updated as:

$$\begin{cases} \theta^{t+1} = \theta^t - \eta \dfrac{\partial l}{\partial \mathbf{x}} \dfrac{\partial \mathbf{x}}{\partial \theta} \\ \phi^{t+1} = \phi^t - \eta \dfrac{\partial l}{\partial \mathbf{y}} \dfrac{\partial \mathbf{y}}{\partial \phi} \end{cases} \tag{2}$$

These two equations highlight that the updates of extractors parameters are combined with two sets of derivative terms. The first set of derivative terms $\frac{\partial l}{\partial \mathbf{x}}$ and $\frac{\partial l}{\partial \mathbf{y}}$ represent how the change of the image and text embedded features affects the loss, and this is the term explored the most in detail in this work. The second set of derivative terms $\frac{\partial \mathbf{x}}{\partial \theta}$ and $\frac{\partial \mathbf{y}}{\partial \phi}$ represent how the change in the model's parameter affects the embedded features. This term can always be expanded with the multiplication of multiple terms for each layer in a modern deep network with multiple layers because of the derivative chain rule, which is not discussed in the work.

The first set of derivative terms are always constrained by the analytic form of the loss function. For example, due to the exponential form of lifted structure loss [28] and binomial deviance loss [37], their derivatives also contain an exponential term. Such a term may cause gradient instability and it is an example of how the design of the loss function can at best, only implicitly control the extractor's learning behavior.

With the latest deep learning platform such as Pytorch [25] which supports forward modules with customized gradient backward calculation, instead of depending on the derivative of the loss, we can explicitly define the gradient update based on the proposed GOAL framework to directly impact the extractors learning behavior. In the following discussion, we focus on the particular forms of the first set of terms in many triplet loss functions from DML literature, and then propose to directly define the first set of terms for model training.

3.1. Gradient Components

Given a pair of image and text feature $\mathbf{x}$ and $\mathbf{y}$, when the image feature $\mathbf{x}$ is treated as an anchor, we denote its text hard negative feature $\mathbf{y}'$ mined in the text batches $\mathbf{Y}$; when the text feature $\mathbf{y}$ is treated as an anchor, we denote its image hard negative feature $\mathbf{x}'$ mined in the image batches $\mathbf{X}$. Then, we can get two triplets $(\mathbf{x}, \mathbf{y}, \mathbf{y}')$ and $(\mathbf{y}, \mathbf{x}, \mathbf{x}')$. In the first triplet, $S_{\mathbf{x},\mathbf{y}} = \mathbf{x}^T\mathbf{y}$ and $S_{\mathbf{x},\mathbf{y}'} = \mathbf{x}^T\mathbf{y}'$ are the cosine similarity computed as the dot-product of the positive and negative pair of normalized image feature and the normalized text feature. Similar cosine similarity is computed for the second triplet, $S_{\mathbf{y},\mathbf{x}} = \mathbf{y}^T\mathbf{x}$ and $S_{\mathbf{y},\mathbf{x}'} = \mathbf{y}^T\mathbf{x}'$. Finally, these cosine similarities are input into a symmetric triplet loss function $l = L(S_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x},\mathbf{y}'}) + L(S_{\mathbf{y},\mathbf{x}}, S_{\mathbf{y},\mathbf{x}'})$.

The gradients w.r.t. the image and text feature are:

$$\begin{cases} \dfrac{\partial l}{\partial \mathbf{x}} = \dfrac{\partial l}{\partial S_{\mathbf{x},\mathbf{y}}}\dfrac{\partial S_{\mathbf{x},\mathbf{y}}}{\partial \mathbf{x}} + \dfrac{\partial l}{\partial S_{\mathbf{x},\mathbf{y}'}}\dfrac{\partial S_{\mathbf{x},\mathbf{y}'}}{\partial \mathbf{x}} + \dfrac{\partial l}{\partial S_{\mathbf{y},\mathbf{x}}}\dfrac{\partial S_{\mathbf{y},\mathbf{x}}}{\partial \mathbf{x}} \\ \quad = \dfrac{\partial L(S_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x},\mathbf{y}'})}{\partial S_{\mathbf{x},\mathbf{y}}}\mathbf{y} + \dfrac{\partial L(S_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x},\mathbf{y}'})}{\partial S_{\mathbf{x},\mathbf{y}'}}\mathbf{y}' + \dfrac{\partial L(S_{\mathbf{y},\mathbf{x}}, S_{\mathbf{y},\mathbf{x}'})}{\partial S_{\mathbf{y},\mathbf{x}}}\mathbf{y} \\ \dfrac{\partial l}{\partial \mathbf{y}} = \dfrac{\partial l}{\partial S_{\mathbf{x},\mathbf{y}}}\dfrac{\partial S_{\mathbf{x},\mathbf{y}}}{\partial \mathbf{y}} + \dfrac{\partial l}{\partial S_{\mathbf{y},\mathbf{x}}}\dfrac{\partial S_{\mathbf{y},\mathbf{x}}}{\partial \mathbf{y}} + \dfrac{\partial l}{\partial S_{\mathbf{y},\mathbf{x}'}}\dfrac{\partial S_{\mathbf{y},\mathbf{x}'}}{\partial \mathbf{y}} \\ \quad = \dfrac{\partial L(S_{\mathbf{x},\mathbf{y}}, S_{\mathbf{x},\mathbf{y}'})}{\partial S_{\mathbf{x},\mathbf{y}}}\mathbf{x} + \dfrac{\partial L(S_{\mathbf{y},\mathbf{x}}, S_{\mathbf{y},\mathbf{x}'})}{\partial S_{\mathbf{y},\mathbf{x}}}\mathbf{x} + \dfrac{\partial L(S_{\mathbf{y},\mathbf{x}}, S_{\mathbf{y},\mathbf{x}'})}{\partial S_{\mathbf{y},\mathbf{x}'}}\mathbf{x}' \end{cases} \tag{3}$$

There are two major elements in the above gradi-

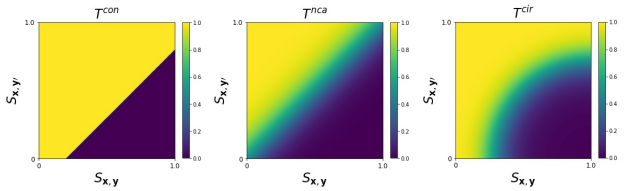

Figure 2. A triplet diagram characterizes the behavior of triplet weights as a function of the the similarity of the positive image-text pair (along the x-axis) and the negative image-text pair (along the y-axis). Triplets where the anchor, positive and negative features are all very similar will be in the top right of the right, and triplets where the positive pairs are similar and the negative pairs are not similar are in the bottom right corner. Using this diagram, (left) shows the constant triplet weight T^{con}, (middle) shows the NCA triplet weight T^{nca}, (right) shows the Circle triplet weight T^{cir}.

ents: scalars $\frac{\partial L(S_{\mathbf{x},\mathbf{y}},S_{\mathbf{x},\mathbf{y}'})}{\partial S_{\mathbf{x},\mathbf{y}}}$, $\frac{\partial L(S_{\mathbf{x},\mathbf{y}},S_{\mathbf{x},\mathbf{y}'})}{\partial S_{\mathbf{x},\mathbf{y}'}}$, $\frac{\partial L(S_{\mathbf{y},\mathbf{x}},S_{\mathbf{y},\mathbf{x}'})}{\partial S_{\mathbf{y},\mathbf{x}}}$, $\frac{\partial L(S_{\mathbf{y},\mathbf{x}},S_{\mathbf{y},\mathbf{x}'})}{\partial S_{\mathbf{y},\mathbf{x}'}}$, and the unit gradient directions $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}'$, $\mathbf{y}'$[1].

The difference in triplet loss and its variants is primarily coming from the scalars. In DML literature, there are majorly two sets of scalar forms: the scalars related to both positive and negative pair similarity of a triplet, which we denote as (**Triplet Weight** T), and the scalars related to either positive and negative pair similarity of a triplet, which we denote as (**Pair Weight** P).

3.2. Triplet Weights

For standard triplet loss function with hard negative mining, the gradient can be derived as:

$$\begin{cases} l^{tri} = \max(m + s_{\mathbf{x},\mathbf{y}'} - s_{\mathbf{x},\mathbf{y}}, 0) + \max(m + s_{\mathbf{y},\mathbf{x}'} - s_{\mathbf{y},\mathbf{x}}, 0) \\ \frac{\partial l^{tri}}{\partial \mathbf{x}} = \delta(m + s_{\mathbf{x},\mathbf{y}'} - s_{\mathbf{x},\mathbf{y}})(\mathbf{y}' - \mathbf{y}) - \delta(m + s_{\mathbf{y},\mathbf{x}'} - s_{\mathbf{y},\mathbf{x}})\mathbf{y} \\ \frac{\partial l^{tri}}{\partial \mathbf{y}} = -\delta(m + s_{\mathbf{x},\mathbf{y}'} - s_{\mathbf{x},\mathbf{y}})\mathbf{x} + \delta(m + s_{\mathbf{y},\mathbf{x}'} - s_{\mathbf{y},\mathbf{x}})(\mathbf{x}' - \mathbf{x}) \end{cases} \tag{4}$$

where m is the margin parameter and $\delta(\cdot)$ is the Heaviside function.

In the gradients of the triplet loss, all scalars are triplet weights because it contains the similarity of both positive and negative pairs of a triplet. The triplet weight is denoted as constant triplet weight T^{con}:

$$T^{con} = \delta(m + s_{\mathbf{x},\mathbf{y}'} - s_{\mathbf{x},\mathbf{y}}) \tag{5}$$

For simplicity, we only show the weights related to triplet ($\mathbf{x}$, $\mathbf{y}$, $\mathbf{y}'$) in the following discussion, and the discussion of weights for triplet ($\mathbf{y}$, $\mathbf{x}$, $\mathbf{x}'$) is similar. When triplets activate the Heaviside function, T^{con} is a constant 1, indicating that these eligible triplets will be treated equally. When triplets don't activate the Heaviside function, T^{con} is 0, indicating that these triplets has no impact on gradient.

[1]They are all unit vector due to the $L2$ normalization

A second common loss function is the NT-Xent loss derived from NCA [9], denote as l^{nca}. Instead of taking all negative candidates into account, in this paper, we adopt the hard-negative mined version as a fair comparison to triplet loss function.

$$\begin{aligned} l^{nca} = -[\log(\frac{\exp(\tau S_{\mathbf{x},\mathbf{y}})}{\exp(\tau S_{\mathbf{x},\mathbf{y}}) + \exp(\tau S_{\mathbf{x},\mathbf{y}'})}) \\ + \log(\frac{\exp(\tau S_{\mathbf{y},\mathbf{x}})}{\exp(\tau S_{\mathbf{y},\mathbf{x}}) + \exp(\tau S_{\mathbf{y},\mathbf{x}'})})] \end{aligned} \tag{6}$$

where τ is the scaling parameter. The scalars in its gradient are also a triplet weight which is denoted as NCA triplet weight T^{nca}(the derivation is shown in Appendix):

$$T^{nca} = \frac{1}{1 + \exp(\tau(S_{\mathbf{x},\mathbf{y}} - S_{\mathbf{x},\mathbf{y}'}))} \tag{7}$$

T^{nca} is rely on the difference of $S_{\mathbf{x},\mathbf{y}}$ and $S_{\mathbf{x},\mathbf{y}'}$. When a triplet in a correct configuration, $S_{\mathbf{x},\mathbf{y}} - S_{\mathbf{x},\mathbf{y}'} > 0$, the triplet weight is small. Otherwise, the triplet weight will be large.

Because T^{nca} only considers the similarity difference $S_{\mathbf{x},\mathbf{y}} - S_{\mathbf{x},\mathbf{y}'}$, some corner cases such as triplet with both large $S_{\mathbf{x},\mathbf{y}}$ and $S_{\mathbf{x},\mathbf{y}'}$ or both small $S_{\mathbf{x},\mathbf{y}}$ and $S_{\mathbf{x},\mathbf{y}'}$ are not well treated. Circle loss [29] proposed a circle triplet weight T^{cir} to deal with the cases:

$$T^{cir} = \frac{1}{1 + \exp(\tau(S_{\mathbf{x},\mathbf{y}}(2 - S_{\mathbf{x},\mathbf{y}}) - S^2_{\mathbf{x},\mathbf{y}'}))} \tag{8}$$

The idea of T^{cir} is to introduce a non-linear mapping for $S_{\mathbf{x},\mathbf{y}}$ and $S_{\mathbf{x},\mathbf{y}'}$ in the exponential term in order to weight more on the corner cases.

Figure 2 shows the triplet weight diagram, a triplet visualization tool from [36], for T^{con} with $m = 0.2$ and T^{nca} and T^{cir} with $\tau = 10$. The equal weight line in T^{nca} is straight lines with form $S_{\mathbf{x},\mathbf{y}} - S_{\mathbf{x},\mathbf{y}'} = \text{const}$. And the equal weight line in T^{cir} is circular lines with form $(S_{\mathbf{x},\mathbf{y}} - 1)^2 + S^2_{\mathbf{x},\mathbf{y}'} = \text{const}$, demonstrating how it increases the weight to the corner cases.

3.3. Pair Weight

In addition to triplet weights, many DML works [37, 28, 33, 36, 29] also proposed pair weights in loss functions. For detailed discussion of pair-weight P, we denote the weight of positive pairs P_+ and the weight of negative pairs P_-. Let a constant scaling parameter to be a baseline for fair comparison. In this case, both pair weights are set with constant 1, as:

$$P^{con}_+ = P^{con}_- = 1; \tag{9}$$

Recent works [33, 36, 29] argued that the weight for negative pairs should be large when they are close to each other. Otherwise, as mentioned in [36], the optimization for DML

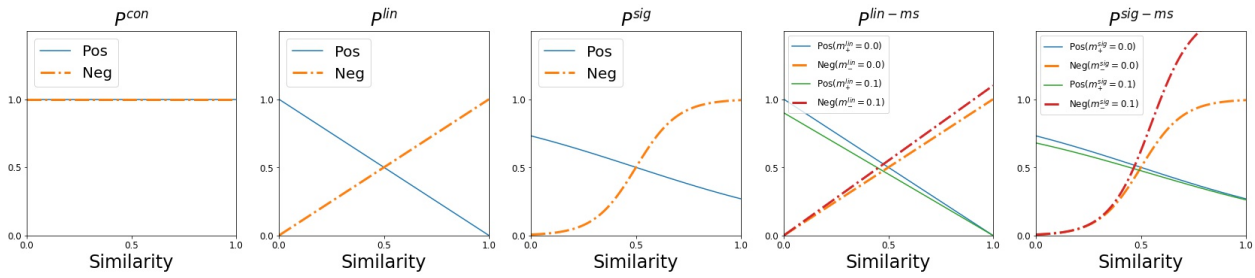

Figure 3. Visualization of constant pair weight P^{con}, linear pair weight P^{lin}, sigmoid pair weight P^{sig} with $\alpha = 2$, $\beta = 10$, $\lambda = 0.5$, linear-MS pair weightP^{lin-ms} with $m_+^{lin} = m_-^{lin} = 0.1$ and sigmoid-MS pair weight P^{sig-ms} with $m_+^{sig} = m_-^{sig} = 0.1$

tasks will quickly converge to a bad local minima. The solution in Circle loss [29] is to apply a linear pair weight P^{lin}: for negative pairs, the weight is large if the similarity is large and small if the similarity is small; for positive pairs, the weight is large if the similarity is small and small if the similarity is large:

$$\begin{cases} P_+^{lin} = 1 - S_{\mathbf{x},\mathbf{y}} \\ P_-^{lin} = S_{\mathbf{x},\mathbf{y}'} \end{cases} \tag{10}$$

Early work binomial deviance loss [37] uses a similar pair weight but with a nonlinear sigmoid form P^{sig}:

$$\begin{cases} P_+^{sig} = \dfrac{1}{1 + \exp\left(\alpha(S_{\mathbf{x},\mathbf{y}} - \lambda)\right)} \\ P_-^{sig} = \dfrac{1}{1 + \exp\left(-\beta(S_{\mathbf{x},\mathbf{y}'} - \lambda)\right)} \end{cases} \tag{11}$$

where α, β and λ are three hyper-parameters.

Multi-similar(MS) loss [33] combines ideas from the lifted structure loss [28] and binomial deviance loss [37], which includes not only the self-similarity of a selected pair but also the relative similarity from other pairs.

We follow [33] to cast their weighting function P^{sig-ms} in our framework. Given a triplet, the self-similarity of the selected positive pair and negative pair are $S_{\mathbf{x},\mathbf{y}}$ and $S_{\mathbf{x},\mathbf{y}'}$. The similarity of other positives in a batch and negatives to the same anchor is considered as relative-similarity, noted as ${R_{\mathbf{x},\mathbf{y}}}^i$ and ${R_{\mathbf{x},\mathbf{y}'}}^j$. In addition, [33] also defines $\mathcal{P}$ and $\mathcal{N}$ be the sets of selected ${R_{\mathbf{x},\mathbf{y}}}^i$ and ${R_{\mathbf{x},\mathbf{y}'}}^j$, where

$$\mathcal{P} = \{{R_{\mathbf{x},\mathbf{y}}}^i \colon {R_{\mathbf{x},\mathbf{y}}}^i < \max\{S_{\mathbf{x},\mathbf{y}'}, {R'_{\mathbf{x},\mathbf{y}}}^j\} + \epsilon\}$$
$$\mathcal{N} = \{{R_{\mathbf{x},\mathbf{y}'}}^j \colon {R'_{\mathbf{x},\mathbf{y}}}^j > \min\{S_{\mathbf{x},\mathbf{y}}, {R_{\mathbf{x},\mathbf{y}}}^i\} - \epsilon\}$$

$$\begin{cases} P_+^{sig-ms} = \dfrac{1}{m_+^{sig} + \exp\left(\alpha(S_{\mathbf{x},\mathbf{y}} - \lambda)\right)} \\ P_-^{sig-ms} = \dfrac{1}{m_-^{sig} + \exp\left(-\beta(S_{\mathbf{x},\mathbf{y}'} - \lambda)\right)} \end{cases} \tag{12}$$

where

$$m_+^{sig} = \frac{1}{|\mathcal{P}|} \sum_{\mathcal{P}} \exp\left(\alpha(S_{\mathbf{x},\mathbf{y}} - {R_{ap}}^i)\right)$$
$$m_-^{sig} = \frac{1}{|\mathcal{N}|} \sum_{\mathcal{N}} \exp\left(-\beta(S_{\mathbf{x},\mathbf{y}'} - {R_{an}}^j)\right)$$

There are two terms in MS loss dynamically changing the pair weight. The self-similarity term has the same effect of sigmoid pair weight P^{sig}. As for the relative-similarity term, the major effect is to increase or decrease the maximum magnitude of the pair weight.

Given a negative pair, when its relative-similarity term $m_-^{sig} > 1$, this indicates the selected negative example is relatively closer to anchor compared to other negative examples. Then, the negative weight increases because the relative term decreases the denominator in P_-^{sig-ms}. When its relative-similarity term $m_-^{sig} < 1$, indicating the selected negative example is relatively far away from anchor comparing to other negative examples, the negative weight decreases because the relative term increases the denominator in P_-^{sig-ms}. The latter situation will not exist under the training with hard negative mining.

Given a positive pair, when its relative-similarity term $m_+^{sig} > 1$, this indicates the selected positive pair has similarity larger than other positive pairs in its batch, the positive weight decreases because the relative term increases the denominator in P_+^{sig-ms}. When its relative-similarity term $m_+^{sig} < 1$, indicating the selected positive pair has similarity less than other positive pairs in its batch, the positive weight increases because the relative term decreases the denominator in P_+^{sig-ms}.

When $m_+^{sig} = m_-^{sig} = 1$ the pair weights simplify back to the sigmoid form in equation 11.

In sum, the main effect caused by the relative-similarity term is to dynamically increase or decrease the maximum penalty for positive and negative pairs as shown in right graph of Figure 3.

In practice, training MS loss needs to tune four hyper-parameters α, β, λ and ϵ to fit different datasets, making the training not convenient and not efficient. With analysis on

	T^{con}	T^{nca}	T^{cir}
P^{con}	Triplet loss	NT-Xent loss	New
P^{lin}	New	New	Circle loss [29]
P^{sig}	Binomial deviance [37]	New	New
P^{lin-ms}	New	New	New
P^{sig-ms}	MS loss [33]	New	New

Table 1. Mapping different gradient combinations of triplet weight and pair weight into existing DML loss functions. Under our GOAL framework, the combinations labeled as "New" are able to be explored.

	Image→ Text	Text → Image
Method	R@1	R@1
VSE++(R152,FT)	41.3	30.3
VSE++(R152,FT) ours	41.0 ± 0.3	30.2 ± 0.1
VSE∞(BUTD)	58.3	42.4
VSE∞(BUTD) ours	58.3 ± 0.7	43.1 ± 0.0
VSE∞(WSL)	66.4	51.6
VSE∞(WSL) ours	66.2 ± 0.2	51.6 ± 0.3

Table 2. Results verification of the model trained with triplet loss function backward vs the model trained with gradient backward on three VSE methods on COCO dataset. Full table is in Appendix

relative-similarity terms m_+^{sig} and m_-^{sig}, we define a clearer and parameter free version of pair weight called linear MS pair weight P^{lin-ms}, which behaves similar to the original MS weight:

$$\begin{cases} P_+^{lin-ms} = (1 - m_+^{lin})(1 - S_{\mathbf{x},\mathbf{y}}) \\ P_-^{lin-ms} = (1 + m_-^{lin}) S_{\mathbf{x},\mathbf{y}'} \end{cases} \tag{13}$$

where

$$m_+^{lin} = \frac{1}{|\mathcal{P}|} \sum_{\mathcal{P}} (S_{\mathbf{x},\mathbf{y}} - R_{\mathbf{x},\mathbf{y}}{}^i)$$

$$m_-^{lin} = \frac{1}{|\mathcal{N}|} \sum_{\mathcal{N}} (S_{\mathbf{x},\mathbf{y}'} - R_{\mathbf{x},\mathbf{y}'}{}^j)$$

3.4. Combinations of Gradient Components

In this section, we have dissected many previous loss functions from DML in terms of their triplet weights and pair weights. Table 1 shows how to map different combinations of gradient components into existing loss functions. In addition to these combinations, the remaining combinations labeled as "New" are all unexplored. These gradient component combinations are hard to be explored if the training needs a loss function and possibly impossible if they are not integrable. However, under our GOAL framework, we are able to train a model with these gradients.

4. Experiments

4.1. Settings

We run a set of experiments on the MS-COCO [3] and Flickr [38] dataset. All experiments are run on the PyTorch platform [25] with Nvidia Tesla V100 32GB GPU. We directly replace the loss module with gradient objective in three open source works: VSE++ [7], VSE∞ [2] and X-VLM [40] and keep all other training settings the same as their original work. We test all possible gradient objectives formed by the combination of the triplet weights and pair weights in Section 3.2 and 3.3 for these three works. Each objective is run for 3 times to remove the effect caused by the randomness coming from the random sampling of the batch and random initialization of the mapping layers to joint space. We report two common retrieval results, image to text retrieval and text to image retrieval, with mean and standard deviation of Recall@1 as metric for both datasets. We show MS-COCO 5K test result in the main paper and Flickr 1k test result in the appendix.

4.2. Validation on Gradient Method

In Tabel 2, we show the results from origin VSE++ and VSE∞ work trained with triplet loss and the results implemented with the equivalent gradient methods with combination of T^{con} and P^{con}. For VSE++ method, we re-implement the experiment "ResNet152, fine-tune" result, denoted as "VSE++(R152,FT) ours". For VSE∞, we re-implement the experiment with pre-extracted object features (BUTD feature) and the Grid features with a pretrained model on Instagram (WSL) [24], denoted as "VSE∞(BUTD) ours" and "VSE∞(WSL) ours". The re-implemented results are almost the same as originally reported numbers, validating our gradient objective with combination of T^{con} and P^{con} has equivalent effect to the triplet loss.

4.3. Results on VSE++ and VSE∞

VSE++ divides the training into two steps. The first step is to freeze the image extractor backbone and train the text extractor and the mapping layers to joint space. In the second step, all parameters of the image and text extractors and the mapping layers are included in the training. We re-implement the original experiments VSE++ (ResNet152) and VSE++ (ResNet152, fine-tuned) for these two steps and replace the triplet loss function with all possible gradient objectives. In addition, we run the same experiments with ViT [6](ViT-base-patch16) which has been popularly used in vision language tasks to compare the performance of gradient objective on different models.

In Table 3 and 4, the pair weights P^{lin}, P^{sig}, P^{lin-ms}, P^{sig-ms} show clear improvement in Recall@1 over the baseline pair weight P^{con}. In addition to pair weight, triplet weights T^{nca}, T^{cir} help pair weight continue to improve the Recall@1 results in the fine-tuning step.

Besides, all DML loss functions mentioned in Table 1 perform better than triplet loss in both steps. In the fine-tuning step, we find the best loss function is MS

VSE++ (ResNet152)						
	Image→ Text			Text → Image		
	T^{con}	T^{nca}	T^{cir}	T^{con}	T^{nca}	T^{cir}
P^{con}	33.9 ± 0.9	34.9 ± 0.4	34.2 ± 0.6	22.8 ± 0.4	22.7 ± 0.2	22.3 ± 0.5
P^{lin}	34.5 ± 0.2	34.5 ± 0.2	34.6 ± 0.3	23.5 ± 0.1	23.2 ± 0.2	23.4 ± 0.4
P^{sig}	34.9 ± 0.1	35.2 ± 0.4	35.0 ± 0.4	23.7 ± 0.1	23.7 ± 0.2	23.6 ± 0.5
P^{lin-ms}	35.0 ± 0.5	35.3 ± 0.5	34.6 ± 0.4	23.8 ± 0.2	23.5 ± 0.2	23.2 ± 0.3
P^{sig-ms}	35.6 ± 0.1	34.9 ± 0.4	35.3 ± 0.4	24.1 ± 0.1	23.7 ± 0.2	23.7 ± 0.1
VSE++ (ResNet152, fine-tuned)						
P^{con}	40.8 ± 0.3	41.0 ± 0.3	41.2 ± 0.1	30.2 ± 0.1	30.5 ± 0.0	30.1 ± 0.2
P^{lin}	41.3 ± 0.2	42.6 ± 0.3	42.3 ± 0.5	30.5 ± 0.1	30.6 ± 0.1	30.7 ± 0.1
P^{sig}	42.2 ± 0.2	43.4 ± 0.1	43.3 ± 0.0	31.1 ± 0.2	31.1 ± 0.2	31.3 ± 0.2
P^{lin-ms}	41.8 ± 0.3	42.6 ± 0.6	42.8 ± 0.2	30.7 ± 0.1	30.9 ± 0.1	31.0 ± 0.2
P^{sig-ms}	43.6 ± 0.3	43.8 ± 0.5	43.8 ± 0.5	30.8 ± 0.3	30.9 ± 0.1	31.1 ± 0.2

Table 3. Result of Image→ Text and Text → Image Recall@1 with different gradient combinations on two steps VSE++ training with ResNet152.

VSE++ (ViT-base-patch16)						
	Image→ Text			Text → Image		
	T^{con}	T^{nca}	T^{cir}	T^{con}	T^{nca}	T^{cir}
P^{con}	37.6 ± 0.2	38.8 ± 0.1	37.9 ± 0.1	26.4 ± 0.2	26.6 ± 0.1	26.4 ± 0.1
P^{lin}	37.7 ± 0.3	38.3 ± 0.3	39.0 ± 0.2	27.0 ± 0.2	27.1 ± 0.0	27.0 ± 0.3
P^{sig}	38.4 ± 0.5	40.0 ± 0.4	39.5 ± 0.8	27.2 ± 0.0	27.5 ± 0.1	27.4 ± 0.3
P^{lin-ms}	38.1 ± 0.3	39.1 ± 0.3	39.0 ± 0.4	27.2 ± 0.1	27.3 ± 0.1	27.0 ± 0.5
P^{sig-ms}	39.9 ± 0.3	40.1 ± 0.3	39.6 ± 0.6	27.7 ± 0.1	27.6 ± 0.1	27.3 ± 0.2
VSE++ (ViT-base-patch16, fine-tuned)						
P^{con}	48.2 ± 0.5	49.6 ± 0.6	48.6 ± 0.9	36.5 ± 0.3	37.0 ± 0.2	36.7 ± 0.3
P^{lin}	48.3 ± 0.3	49.4 ± 0.3	49.3 ± 0.3	36.4 ± 0.1	37.2 ± 0.3	37.4 ± 0.2
P^{sig}	49.2 ± 0.6	50.9 ± 0.2	51.1 ± 0.4	37.2 ± 0.3	37.9 ± 0.3	37.9 ± 0.2
P^{lin-ms}	48.9 ± 0.4	50.2 ± 0.1	49.7 ± 0.2	36.8 ± 0.2	37.4 ± 0.1	37.6 ± 0.4
P^{sig-ms}	50.4 ± 0.8	50.7 ± 0.5	51.7 ± 0.2	37.4 ± 0.4	37.3 ± 0.1	37.9 ± 0.2

Table 4. Result of Image→ Text and Text → Image Recall@1 with different gradient combinations on two steps VSE++ training with ViT.

loss(P^{sig-ms},T^{con}). But it is still sub-optimal when we combine triplet weight T^{nca} or T^{cir} with pair weight P^{sig-ms}, demonstrating the advantage of exploring the gradient space with GOAL.

VSE∞ We re-implement two training setups in VSE∞. Both setups use BERT-base [5] as the text feature extractor. For the image feature, one uses the pre-extracted object features (BUTD feature) and another uses the grid features with a pretrained model on Instagram (WSL feature) [24]. A learned Generalized Pooling Operator(GPO) aggregates and projects the image and text feature vectors independently into the joint embedding space to further compute the loss. Still, we only replace the triplet loss function used in the training with gradient objectives.

Table 5 shows similar improvement pattern as shown in the result of VSE++, verifying our GOAL is general to different of VSE methods.

4.4. State-of-the-Art Results

Finally, we compare two sets of state-of-the-art approaches on MS-COCO 5K test and Flickr 1K test. One set is VSE related and another set is VLP related. In Table 6, We first show our best improved result of VSE++ and VSE∞ method, denoted as "VSE++(R152, FT) ours", "VSE∞(BUTD) ours" and "VSE∞(BUTD) ours", which are trained with combination of (T^{cir}, P^{sig-ms}). In MS-COCO 5K test, the gain of Image→ Text R@1 and Text→ Image R@1 on VSE++(R152, FT) is 3%, 0.7%,

	Image→ Text			Text → Image		
VSE∞(BUTD)						
	T^{con}	T^{nca}	T^{cir}	T^{con}	T^{nca}	T^{cir}
P^{con}	58.9 ± 0.7	61.2 ± 0.7	60.3 ± 0.2	43.1 ± 0.0	43.2 ± 0.3	42.5 ± 0.1
P^{lin}	58.1 ± 0.2	60.7 ± 0.1	60.1 ± 0.2	43.1 ± 0.3	43.4 ± 0.2	43.0 ± 0.1
P^{sig}	59.8 ± 0.6	62.0 ± 0.4	62.0 ± 0.4	43.5 ± 0.2	43.9 ± 0.1	43.8 ± 0.2
P^{lin-ms}	60.0 ± 0.1	61.7 ± 0.2	61.2 ± 0.2	43.6 ± 0.2	43.9 ± 0.2	43.4 ± 0.2
P^{sig-ms}	61.8 ± 0.2	63.1 ± 0.2	63.2 ± 0.3	44.6 ± 0.2	44.8 ± 0.1	44.9 ± 0.1
VSE∞(WSL)						
P^{con}	66.2 ± 0.2	67.6 ± 0.4	67.2 ± 0.4	51.6 ± 0.3	51.4 ± 0.2	49.9 ± 0.1
P^{lin}	66.9 ± 0.7	68.5 ± 0.4	68.4 ± 0.5	52.5 ± 0.1	52.8 ± 0.2	52.5 ± 0.2
P^{sig}	68.2 ± 0.6	69.7 ± 0.2	70.2 ± 0.5	53.0 ± 0.1	53.1 ± 0.2	52.9 ± 0.2
P^{lin-ms}	67.8 ± 0.1	69.7 ± 0.2	69.5 ± 0.1	52.8 ± 0.2	53.3 ± 0.2	52.7 ± 0.1
P^{sig-ms}	70.3 ± 0.2	71.5 ± 0.4	71.4 ± 0.5	53.9 ± 0.4	54.2 ± 0.0	53.6 ± 0.6

Table 5. Result of Image→ Text and Text → Image Recall@1 with different gradient combinations on VSE∞(BUTD) and VSE∞(WSL).

			MS-COCO 5K test						Flickr 1K test					
Tasks			Image→ Text			Text→ Image			Image→ Text			Text→ Image		
Method	Pre-train	Data size	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
VSE++(R152, FT) [7]	✗	-	41.3	-	81.2	30.3	-	72.4	52.9	-	87.2	39.6	-	79.5
VSE++(R152, FT) **ours**	✗	-	44.3	73.3	83.7	31.0	60.2	72.5	57.0	82.6	89.2	42.4	72.4	81.0
SCAN [17]	✗	-	50.4	82.2	90.0	38.6	69.3	80.4	67.4	90.3	95.8	48.6	77.7	85.2
VSRN [19]	✗	-	53.0	81.1	89.4	40.5	70.6	81.1	71.3	90.6	96.0	54.7	81.8	88.2
VSE∞(BUTD) [2]	✗	-	58.3	85.3	-	42.4	72.7	-	81.7	95.4	97.6	61.4	85.9	91.5
VSE∞(BUTD) **ours**	✗	-	63.2	87.2	93.0	44.4	74.2	83.9	82.3	95.8	98.4	64.0	87.5	92.7
VSE∞(WSL) [2]	✗	-	66.4	89.3	-	51.6	79.3	-	88.4	98.3	99.5	74.2	93.7	96.8
VSE∞(WSL) **ours**	✗	-	71.9	92.0	95.9	53.7	80.6	88.4	90.6	99.2	99.6	76.7	94.6	97.3
VinVL [41]	✓	5.6M	75.4	92.9	96.2	58.8	83.5	90.3	-	-	-	-	-	-
ALBEF [18]	✓	14M	77.6	94.3	97.2	60.7	84.3	90.5	95.9	**99.8**	100.0	85.6	**97.5**	98.9
X-VLM [40]	✓	4M	80.4	95.5	**98.2**	63.1	85.7	**91.6**	96.8	99.8	100.0	86.1	97.4	98.7
X-VLM **ours**	✓	4M	**81.4**	**95.6**	97.9	**63.6**	**86.0**	91.5	**97.0**	99.6	**100.0**	**86.3**	97.4	**99.0**

Table 6. Our state-of-the art image-text retrieval results on MS-COCO 5K and Flickr 1K test using the novel loss function designed with the proposed GOAL framework.

on VSE∞(BUTD) is 4.9%, 2.0% and on VSE∞(WSL) is 5.5%, 2.1%. In Flickr 1K test, the gain of Image→ Text R@1 and Text→ Image R@1 on VSE++(R152, FT) is 4.1%, 2.8%, on VSE∞(BUTD) is 0.6%, 2.6% and on VSE∞(WSL) is 2.2%, 2.5%.

In addition, we apply the same gradient objective in the latest state-of-the-art approach X-VLM [40] with replacement of its contrastive loss item in the downstream fine-tuning. The result is denoted as "X-VLM ours". We continue to push the boundary of state-of-the-art result on MS-COCO 5K test and Flickr 1K test.

5. Conclusion

We provide a new framework GOAL to train image-text matching tasks with a combination of gradient components dissected from deep metric learning loss functions. In practice, the proposed gradient objectives can be easily applied as a drop-in replacement to training with loss functions. Extensive experiments on exhaustive combinations of triplet weights and pair weights demonstrate both triplet weights and pair weights have individual impact on the retrieval performance and generally the combination of T^{cir}, P^{sig-ms} achieve the best performance on image-text retrieval. This framework helps find better gradient objectives which have never been explored for this domain and provides consistent retrieval improvement on multiple established methods, including achieving new state-of-the-art results.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

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

[3] 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.

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

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[8] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[9] Jacob Goldberger, Geoffrey E Hinton, Sam T. Roweis, and Ruslan R Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2005.

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE, 2006.

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

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[14] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2018.

[15] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[17] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

[18] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, 2021.

[19] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 4654–4662, 2019.

[20] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[22] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[24] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

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

[27] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.

[28] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[30] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020.

[33] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019.

[34] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[35] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6609–6618, 2019.

[36] Hong Xuan, Abby Stylianou, Xiaotong Liu, and Robert Pless. Hard negative examples are hard, but useful. In *The European Conference on Computer Vision (ECCV)*, September 2020.

[37] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, pages 34–39. IEEE, 2014.

[38] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[39] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[40] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*, 2021.

[41] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021.

[42] Xiao Zhang, Rui Zhao, Junjie Yan, Mengya Gao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. P2sgrad: Refined gradients for optimizing deep face models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9906–9914, 2019.