

# Improved Probabilistic Image-Text Representations

Sanghyuk Chun  
NAVER AI Lab

## Abstract

Image-Text Matching (ITM) task, a fundamental vision-language (VL) task, suffers from the inherent ambiguity arising from multiplicity and imperfect annotations. Deterministic functions are not sufficiently powerful to capture ambiguity, prompting the exploration of probabilistic embeddings to tackle the challenge. However, the existing probabilistic ITM approach encounters two key shortcomings; the burden of heavy computations due to the Monte Carlo approximation, and the loss saturation issue in the face of abundant false negatives. To overcome the issues, this paper presents an improved Probabilistic Cross-Modal Embeddings (named PCME++) by introducing a new probabilistic distance with a closed-form solution. In addition, two optimization techniques are proposed to enhance PCME++ further; first, the incorporation of pseudo-positives to prevent the loss saturation problem under massive false negatives; second, mixed sample data augmentation for probabilistic matching. Experimental results on MS-COCO Caption and two extended benchmarks, CxC and ECCV Caption, demonstrate the effectiveness of PCME++ compared to state-of-the-art ITM methods. The robustness of PCME++ is also evaluated under noisy image-text correspondences. In addition, the potential applicability of PCME++ in automatic prompt tuning for zero-shot classification is shown. The code is available at <https://naver-ai.github.io/pcmepp/>.

## 1 Introduction

Given images and captions, Image-Text Matching (ITM) is the task of retrieving the most relevant images/captions for the given query caption/image [1–19]. The applications of ITM include cross-modal retrieval [4] from paired image-caption datasets, such as MS-COCO Caption [20], and zero-shot classification [19], by treating class labels as a text (e.g., “a photo of {·}”). Owing to its significant role in image understanding and language comprehension, ITM has emerged as a fundamental Vision Language (VL) downstream task. However, this problem inherently suffers from the ambiguity caused by *many-to-many correspondences* and *sparse annotations* of the ITM datasets.

The nature of image-text matching is *many-to-many*; an image can be described in numerous text explanations, and there are a plentiful number of visual scenes to visualize a text description. However, simultaneously, our datasets are *sparse* annotated. The existing ITM datasets are built by collecting paired image-caption, and treating the collected image-caption pairs are the only positives without considering other potential positives in “negative” pairs [20–24]. For example, Chun et al. [25] showed that the MS-COCO Caption dataset has massive missing positives; 88.2% of caption-to-image positives and 72.1% of image-to-caption positives are labeled as “negative”. Figure 1 shows an example. While humans judge all images and texts are plausibly matched, the dataset only treats a pair  $(x_v^i, x_t^j)$  as positive when  $i = j$ . In this paper, we argue that the inherent multiplicity and the sparse annotations lead to the ambiguity of ITM datasets and make ITM problem challenging (§2.1).

This paper aims to design a proper joint embedding space that represents the inherent ambiguity by probabilistic embeddings [14, 26–35], i.e., encoding an input to a random variable rather than a deterministic vector. Probabilistic embeddings have been introduced for many applications with inherent ambiguity, such as word embeddings [27], face understanding [28, 29], 2D-to-3D pose

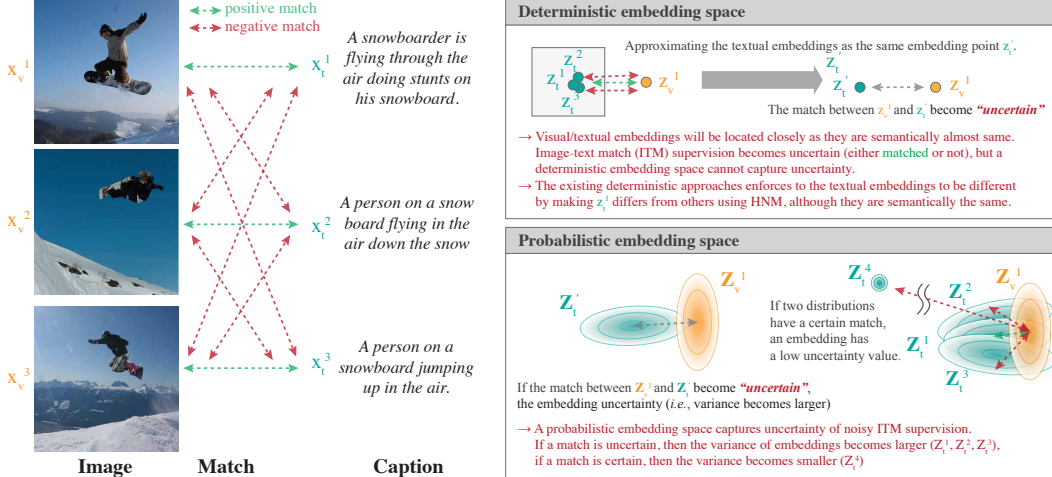


Figure 1: **Inherent ambiguity of ITM.** We assume that the deterministic textual embeddings are mapped to the same point  $z_t^1$ , i.e.,  $z_t^1 \approx z_t^2 \approx z_t^3 \approx z_t^4$ , as well as the probabilistic textual embeddings  $z_t^1 \approx \dots \approx z_t^4$ .

estimation [30], speaker diarization [31], video understanding [32], and composed image retrieval [33]. Especially, Chun et al. [14] investigated the primitive probabilistic approach for ITM, Probabilistic Cross-Modal Embedding (PCME), based on the approach by Oh et al. [26]. Although PCME shows reasonable retrieval performances and interesting observations through uncertainty measures, PCME suffers from expensive computations due to Monte Carlo approximation and fast loss saturation.

Firstly, PCME needs expensive sampling operations for both training and inference. For example, if we randomly draw 7 samples for each input, computing the distance between two samples costs  $O(7 \times 7)$ . Furthermore, due to the sampling operation, PCME retrieval operation cannot be extended to large-scale efficient retrieval systems, such as FAISS [36]. This issue is solved by introducing a new probability distance with a closed-form solution and a new objective function based on the distance (§2.2). In addition, as the proposed closed-form distance consists of Euclidean distance and the relationship between variance embeddings, we can easily adapt approximated KNN to ours (§3.5). Experimental results show that the closed-form distance not only makes the operation efficient but also convergences to a better solution by computing an exact solution instead of an approximation.

Moreover, this paper demonstrates that PCME suffers from fast loss saturation under abundant false negatives (FNs). The PCME training strategy suppresses the gradient step size if the model prediction and ground truth differ significantly (e.g., if a model predicts an image-caption pair is positive with high confidence, but the ground truth is negative). However, as Chun et al. [25] showed, our datasets have abundant FNs; the gradient of FN pairs converges to zero, and FN samples will not contribute to the model update eventually. The issue is mitigated by introducing two techniques: pseudo-positives (§2.3) and mixed sample data augmentation for probabilistic matching (§2.4). This paper conceptually and empirically shows that the proposed techniques can alleviate the zero gradient issue of FNs.

PCME++ is evaluated on MS-COCO Caption [20] and its extended benchmarks CxC [37] and ECCV Caption [25] with state-of-the-art ITM methods (§3.2). In the experiments, PCME++ consistently outperforms the comparison methods on the COCO benchmark. PCME++ is also evaluated on the noisy correspondence benchmark [16], indicating that our method is not only effective for the original task but also holds the potential to address the noisy correspondence problem. Furthermore, this paper shows that the textual uncertainty of PCME++ can be applied to a prompt-tuning for a zero-shot classification with a pre-trained model on large-scale VL datasets, demonstrating the versatility and scalability of our method for a wide range of applications (§3.5). Finally, the qualitative advantages of the learned uncertainty of PCME++ by capturing dataset uncertainty are shown in §3.4.

**Contributions.** This paper introduces PCME++, an improved probabilistic image-text representation, by introducing: a new closed-form probability distance, named CSD, and a new matching objective function based on CSD for substituting expensive sampling-based approximation of PCME [14]; a pseudo-positive strategy and a mixed sample data augmentation strategy for addressing the loss saturation issue of abundant false negatives. PCME++ shows not only good retrieval performances but also the extensibility to various applications, e.g., mitigating noisy correspondences, prompt tuning for zero-shot classification, and understanding the inherent ambiguity of a dataset.

## 2 Improved Probabilistic Cross-Modal Embeddings (PCME++)

### 2.1 Problem definition: Ambiguity of ITM datasets

Let  $x_v$  and  $x_t$  be the input image and caption, respectively. For each image text pair, a binary matching indicator  $m_{vt} \in \{0, 1\}$  denotes whether  $x_t$  describes  $x_v$  well. This paper argues that the inherent multiplicity and the sparse annotations make  $m_{vt}$  ambiguous. For example, as shown in Figure 1,  $x_t^1$  (“A person on a snowboard flying in the air down the snow”) and  $x_t^2$  (“A person on a snowboard jumping up in the air.”) are semantically almost the same, hence we may assume that  $x_t^1$  and  $x_t^2$  are mapped to almost the same embedding point  $z'_t$ , i.e.,  $f(x_t^1) \approx f(x_t^2) = z'_t$  if we have a proper mapping  $f(\cdot)$  between the input space and the embedding space. In this case, if  $x_t^1$  and  $x_v^1$  are a positive match,  $x_t^2$  and  $x_v^1$  should be a positive match in the embedding space. However, because our dataset contains only sparse matching relationships [25, 37],  $x_t^2$  and  $x_v^1$  are a negative match. In other words, in the embedding space, the matching between  $z'_v$  and  $z'_t$  ( $\approx f(x_t^1) \approx f(x_t^2)$ ) becomes ambiguous (i.e., it can be either positive or negative). As shown in Figure 1, a deterministic embedding space cannot capture the inherent uncertainty originated by the multiplicity and the sparse annotations. The existing deterministic approaches, therefore, rely on Hardest Negative Mining (HNM) strategy [4], selecting the closest pair as the only negative for computing a triplet loss. The HNM strategy enforces sparse positive pairs to be closer than other false negative (FN) pairs, resulting in a twisted embedding space that cannot capture the inherent uncertainty of VL datasets. We empirically show that the HNM strategy eventually converges to a suboptimal embedding space when the ambiguity intensifies, i.e., under strong noisy correspondences (§3.2). In contrast, probabilistic embeddings can naturally mitigate the issue by capturing the ambiguity of  $m_{vt}$  with a probability distribution.

### 2.2 Probabilistic contrastive learning

We first define a visual embedding and a text embedding of the given image  $x_v$  and  $x_t$  as normally distributed random variables,  $\mathbf{Z}_v \sim \mathcal{N}(\mu_v, \Sigma_v)$  and  $\mathbf{Z}_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ , respectively. For simplicity, we assume diagonal covariance matrices and simplify the notations as  $\mathcal{N}(\mu_v, \sigma_v^2)$  and  $\mathcal{N}(\mu_t, \sigma_t^2)$ , where  $\mu$  and  $\sigma$  are  $D$ -dimensional vectors. As shown in Figure 1, our purpose is to learn probabilistic embeddings  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$  satisfying the following properties: (a) there exists a proper probabilistic distance between  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$ . (b) if the match  $m_{vt}$  is certain, then  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$  have small variances. (c) if the match between  $x_v$  and  $x_t$  ( $m_{vt}$ ) is ambiguous, then  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$  have large variances.

The probabilistic distance  $d(\cdot)$  between two probabilistic embeddings  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$ , named closed-form sampled distance (CSD), is defined as follows:

$$d(\mathbf{Z}_v, \mathbf{Z}_t) = \mathbb{E}_{\mathbf{Z}_v, \mathbf{Z}_t} \|\mathbf{Z}_v - \mathbf{Z}_t\|_2^2 = \|\mu_v - \mu_t\|_2^2 + \|\sigma_v^2 + \sigma_t^2\|_1, \quad (1)$$

where  $\|\cdot\|_p$  is a p-norm operation. To be self-contained, the full derivation of Equation (1) is provided in Appendix A.1. Equation (1) satisfies most of the properties of a metric function (i.e., positivity, symmetry, and triangular inequality) except zero self-distance;  $d(\mathbf{Z}, \mathbf{Z})$  is  $2\|\sigma^2\|_1$ , not zero. I.e., Equation (1) satisfies the condition (a). There are two ways to make  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$  closer/further; making  $\mu_v$  and  $\mu_t$  closer/further, or making  $\sigma_v$  and  $\sigma_t$  smaller/larger. Hence, if we assume fixed  $\mu_v$  and  $\mu_t$ , we have to decrease  $\sigma_v$  and  $\sigma_t$  to minimize  $d(\mathbf{Z}_v, \mathbf{Z}_t)$ ; if  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$  are a certain positive match (i.e.,  $m_{vt} = 1$ ), then  $\sigma_v$  and  $\sigma_t$  will be collapsed to zero (i.e., satisfying the condition (b)), and  $d(\mathbf{Z}_v, \mathbf{Z}_t)$  will become Euclidean distance. On the other hand, if the match between  $\mathbf{Z}_v$  and  $\mathbf{Z}_t$  is ambiguous (i.e.,  $m_{vt}$  can be either positive or negative), then  $\sigma_v$  and  $\sigma_t$  will not be collapsed to zero, for increasing  $d(\mathbf{Z}_v, \mathbf{Z}_t)$  for the negative match case;  $d(\mathbf{Z}_v, \mathbf{Z}_t)$  also satisfies the condition (c).

CSD has a similar form to Wasserstein 2-distance (WD),  $\inf_{\mathbf{Z}_v, \mathbf{Z}_t} \mathbb{E}_{\mathbf{Z}_v, \mathbf{Z}_t} \|\mathbf{Z}_v - \mathbf{Z}_t\|_2^2 = \|\mu_v - \mu_t\|_2^2 + \|\sigma_v - \sigma_t\|_2^2$ , where WD includes the infimum operation. However, WD is not a proper probabilistic distance in the matching problem, especially WD cannot satisfy the condition (b). Assume the scenario when  $\mu$  values are fixed again. In this case,  $\sigma_v$  and  $\sigma_t$  have no motivation to be decreased, but they are just enforced to have the same values. Hence, the learned  $\sigma$  by WD cannot represent the sample certainty. Figure 2 shows a 2-D toy scenario where CSD satisfies the proper uncertainty conditions while WD cannot. In the figure, red, yellow, and green dots are certain samples, and

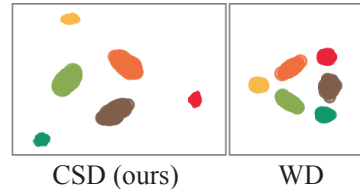


Figure 2: **Learned 2-D embedding spaces by CSD and WD.** The full animations can be found in <https://naver-ai.github.io/pcmepp/>.

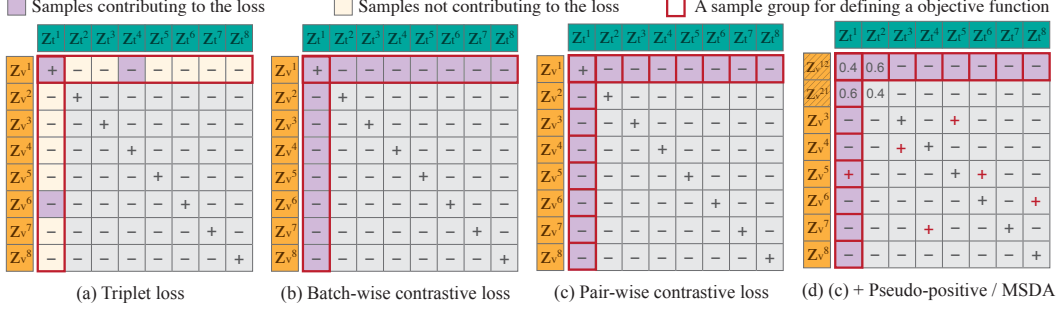


Figure 3: **Comparisons of different objective functions.** For given  $i$ -th visual embeddings  $\mathbf{z}_v^i$  and  $j$ -th textual embedding  $\mathbf{z}_t^j$ , we illustrate how each sample contributes to different loss functions. (a) Only two image-caption pairs contributed to the loss in each row/column for triplet loss. (b) Batch-wise contrastive loss, such as InfoNCE, is defined for each row/column. (c) Pair-wise contrastive loss, such as PCME++, is defined for each image-caption pair. Hence, our loss is computed multiple times for each row/column. (d) As our loss is computed pair-wise, it is straightforward to apply pseudo-positives or mixed sample data augmentation (MSDA).

others are uncertain samples. The size of each dot denotes the intensity of the learned  $\sigma$  values. Here, we observe that  $\frac{\bar{\sigma}_{\text{uncertain}}^2}{\bar{\sigma}_{\text{certain}}^2}$ , the average  $\sigma^2$  value for uncertain/certain samples by CSD are 1.82, while we have 1.04 for WD. More details of the toy experiment are described in Appendix A.2. CSD is also related to the matching probability [26] used by PCME [14], ( $\mathbb{E}_{\mathbf{Z}_v, \mathbf{Z}_t} \text{sigmoid}(-a \|\mathbf{Z}_v - \mathbf{Z}_t\|_2 + b)$ ) where the matching probability cannot be computed in a closed-form due to sigmoid but should be computed by an expensive Monte-Carlo approximation.

Now, based on Equation (1), the probabilistic matching objective function is defined as the follows:

$$\mathcal{L}_{\text{match}} = m_{vt} \log \text{sigmoid}(-a \cdot d(\mathbf{Z}_v, \mathbf{Z}_t) + b) + (1 - m_{vt}) \log \text{sigmoid}(a \cdot d(\mathbf{Z}_v, \mathbf{Z}_t) - b), \quad (2)$$

where  $m_{vt} \in \{0, 1\}$  is the matching indicator between  $v$  and  $t$ .  $a$  and  $b$  are learnable scalar values, following Oh et al. [26] and Chun et al. [14]. In practice, Equation (2) can be easily implemented by binary cross entropy (BCE) loss. We compute  $\mathcal{L}_{\text{match}}$  for all pairs in the mini-batch as contrastive learning objectives, such as InfoNCE [19]. The overview of the comparisons between our objective function, a standard triplet loss, and batch-wise contrastive loss are shown in Figure 3.

To prevent the collapse of  $\sigma$  (i.e.,  $\sigma \rightarrow 0$ ), PCME++ employs Variational Information Bottleneck (VIB) loss [38],  $\mathcal{L}_{\text{VIB}}$ , following Oh et al. [26] and Chun et al. [14]. As derived by Oh et al. [26],  $\mathcal{L}_{\text{VIB}}$  can be computed by the KL divergence between the learned distribution and  $\mathcal{N}(0, I)$ .

### 2.3 Pseudo-positives (PP) for handling numerous false negatives

Let  $-a \cdot d(\mathbf{Z}_v, \mathbf{Z}_t) + b = l_{vt}$ , then we can derive  $\frac{\partial \mathcal{L}_{\text{match}}}{\partial l_{vt}} = 1 - \text{sigmoid}(l_{vt})$  and  $1 - \text{sigmoid}(-l_{vt})$  when  $m_{vt} = 0$  and 1, respectively. Therefore, if there exists a negative pair ( $m = 0$ ) with a large  $l_{vt}$  (i.e., a small probabilistic distance  $d(\mathbf{Z}_v, \mathbf{Z}_t)$ ), the gradient will be converged to zero. Unfortunately, as observed by Chun et al. [25], image-text paired datasets have numerous false negatives (e.g., captions in the COCO Caption dataset have  $\times 8.47$  positive images than the “ground-truth” positive images.), i.e., if we have a plausible matching model, then the false negative pairs will not contribute to the objective function Equation (2). We can also observe the same phenomenon for false positives. Note that PCME [14] also suffers from the same issue, as discussed in Appendix A.3.

To tackle the issue, PCME++ employs a simple pseudo-labeling strategy: for a positive match ( $v, t$ ),  $t'$  is a pseudo-positive (PP) match with  $t$  if  $d(\mathbf{Z}_v, \mathbf{Z}_{t'}) \leq d(\mathbf{Z}_v, \mathbf{Z}_t)$ . Using the pseudo-positives, we compute the pseudo-positive matching loss  $\mathcal{L}_{\text{pseudo-match}}$  using (2). The objective function becomes:

$$\mathcal{L}_{\text{match}} + \alpha \mathcal{L}_{\text{pseudo-match}} + \beta \mathcal{L}_{\text{VIB}}, \quad (3)$$

where  $\alpha$  and  $\beta$  are control parameters of PP matching loss and VIB loss. In the experiments,  $\alpha = 0.1$  and  $\beta = 0.0001$  are chosen (Appendix C.2). Pseudo-code for Equation (3) is shown in Appendix A.4.

### 2.4 Mixed Sample Data Augmentation (MSDA) for probabilistic matching

MSDA, such as Mixup [39] or CutMix [40], shows not only great improvements in empirical performances but also shows good theoretical properties, such as generalization [41, 42] or calibration

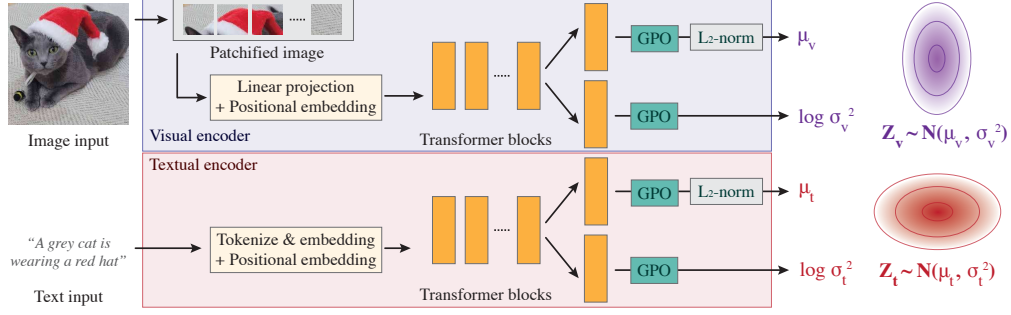


Figure 4: **Architecture overview.** We use the same visual and textual backbones as CLIP [19]. Each modality encoder encodes  $D$ -dimensional  $\ell_2$ -normalized mean vector  $\mu$  and the variance vector  $\log \sigma^2$ , followed by Generalized Pooling Operator (GPO) [15], to represent a normally distributed random variable  $\mathbf{Z} \sim \mathcal{N}(\mu, \sigma^2)$ .

[43]. MSDA consists of two parts; input mixing (*i.e.*, a generative process to generate a new mixed sample) and label mixing (*i.e.*, modifying the supervision of the mixed sample). The intensity of the augmentation is controlled by  $\lambda$ , usually sampled from a pre-defined Beta distribution. For example, a mixed sample by Mixup is  $x_{\text{mix}} = \lambda x_1 + (1 - \lambda)x_2$ . Usually, it is not straightforward to apply MSDA to metric learning or contrastive learning because their losses are computed in a batch-dependent way (See Figure 3 (a) and (b)). On the other hand, as our objective function is computed in a pair-wise manner (See Figure 3 (c)), it is easier to apply MSDA to our objective function.

There are two issues with designing MSDA for probabilistic matching. First, MSDA for the textual modality is not straightforward. Hence, PCME++ only mixes visual inputs using Mixup [39] and CutMix [40]. Second, we cannot directly mix labels because our scenario has no class label. Instead, we let  $m_{vt}$  smooth in Equation (2), *i.e.*,  $m_{vt} \in [0, 1]$ . This approach controls the gradient step size by mixing intensity  $\lambda$ : Assuming  $m_{vt} = \lambda$ , then we have  $\frac{\partial \mathcal{L}_{\text{match}}}{\partial l} = (1 - \lambda) - (1 - 2\lambda) \text{sigmoid}(l_{vt})$ . Here, an extreme case  $\text{sigmoid}(l_{vt}) \approx 1$  (*i.e.*, model predicts a positive match with high confidence) has a gradient value  $\lambda$ . Therefore, MSDA also makes highly confident false negative samples ( $m_{vt} = 0$ , but the model predicts  $\text{sigmoid}(l_{vt}) \approx 1$ ) contribute to the parameter updates as pseudo-positives.

The overview of the optimization procedure with pseudo-positives and MSDA is illustrated in Figure 3 (d). In the experimental results, 25% of mini-batch images are mixed by sampling the mixing intensity  $\lambda$  from Beta(2, 2). For every mini-batch, Mixup or CutMix is randomly chosen for the mixing strategy. The empirical study shows that this strategy is slightly better than the widely-used batch-wise mixing strategy, *i.e.*, randomly mixing the whole mini-batch or using the original mini-batch (Appendix C.2).

## 2.5 Architecture

PCME++ trains visual and textual encoders separately, such as visual semantic embeddings [4, 15] or CLIP [19]. Each encoder has two heads,  $\mu$  and  $\log \sigma^2$  heads whose output vectors are  $D$ -dimensional. An input is mapped to a normal distribution parameterized by the output of  $\mu$  and  $\log \sigma^2$  heads.

PCME++ employs a Vision Transformer (ViT) [44] as the visual backbone and a 12-layer 512-wide Transformer [45] as the textual backbone, following Radford et al. [19]. PCME++ duplicates the last transformer layer for  $\mu$  and  $\log \sigma^2$  heads, *e.g.*, a textual backbone has a shared feature extractor with a 11-layer Transformer and  $\mu$  and  $\log \sigma^2$  are 1-layer Transformer blocks. The  $\log \sigma^2$  head is randomly initialized, while the  $\mu$  head is initialized as the same as the backbone initialization (*e.g.*, from a pre-trained model). We empirically observe that using more layers for  $\log \sigma^2$  marginally improves the performances, but we set the number of layers for  $\log \sigma^2$  head to 1 for computational efficiency. Finally, we employ Generalized Pooling Operator (GPO) [15] for the feature aggregation with the same parameter setting of Chen et al. [15]. We observe that GPO brings both training stability and performance improvements. The model architecture overview is illustrated in Figure 4.

## 3 Experiments

### 3.1 Experimental protocol

**Datasets and evaluation metrics.** PCME++ is evaluated on MS-COCO Caption [20], a widely used ITM benchmark, containing 123,287 images from MS-COCO [46] and five human-annotated captions



per image. 113,287/5,000/5,000 images are used for training/validation/testing [47]. Although Recall@ $k$  (R@ $k$ ) is a common evaluation metric in COCO Caption, as Musgrave et al. [48] showed, R@ $k$  is often insufficient to measure retrieval performances. Furthermore, recent studies [37, 25] observed that many COCO Caption negatives are actually positives; *e.g.*, Chun et al. [25] showed that 88.2% and 72.1% positive images and captions are annotated as negative in COCO. In other words, COCO R@ $k$ , relying on the noisy COCO annotation  $m_{vt}$ , is not fully reliable.

To mitigate the problem of R@ $k$  evaluation, two extended benchmarks, ECCV Caption (EC) [25] and CxC [37], are employed for the test split. Both datasets are validated by human annotators; EC contains more plentiful positives than CxC but its queries are the subset of the original COCO Caption; CxC has fewer positives than EC, but its annotations cover the whole COCO test split, and the annotations are less noisy. Note that the original COCO Caption, EC, and CxC have the same images and captions ( $x_v, x_t$ ) but with different match annotations  $m_{vt}$ . The overview of each benchmark can be found in Appendix B.1. In the experiments, following Chun et al. [25], R@ $k$  for all benchmarks and mAP@R and R-Precision for EC are reported. The conventional 5-fold 1K COCO R@1 and “rsum”, the summation of R@1, R@5, R@10 for image-to-text and text-to-image retrieval are also reported. For the main paper, the averaged scores on each modality is reported, while the full results for each modality and R@5, R@10 results are in Appendix C.6.

**Comparison methods.** VSE $_{\infty}$  [15] is based on a conventional triplet loss and hardest negative mining. InfoNCE is the CLIP [19] pre-training objective. PCME [14] is a primitive probabilistic ITM model with sampling-based matching probability. As we initialize all models by CLIP pre-trained models, CLIP zero-shot (ZS) is also reported as a baseline. All models have the same visual and textual backbones, except probabilistic models; they have an additional  $\log \sigma^2$  head (See Figure 4). All models are trained three times for each setting and the average evaluation metric are reported.

**Training details and model selection.** PCME++ is initialized with the official pre-trained CLIP models [19], while newly introduced modules, such as  $\log \sigma^2$  head and GPO are randomly initialized. All models are trained for 25 epochs using AdamP optimizer [49] by setting the initial learning rate as 0.0005 and weight decay as 0.0001. The learning rate is decayed by a factor of 0.1 for the last 10 epochs. Following Chen et al. [15], different learning rate multipliers are applied for the visual backbone ( $\times 0.01$ ) and the textual backbone ( $\times 0.1$ ). The visual backbone is frozen for 2 epochs, and a linear learning rate warmup is applied for the first epoch after the freezing. Also, layer-wise learning rate decay (LLRD) for each transformer block is applied by 0.7. The batch size is set to 128. Lastly, for the generalizability of GPO, SizeAugment is employed as Chen et al. [15].

The hyperparameters of PCME++ are set as follows; the affine transform is initialized by  $a = b = 5$  in Equation (2);  $\alpha$  for pseudo-positives as 0.1; VIB  $\beta$  as 0.0001. PCME++ mixes 25% of images in the mini-batch by Mixup or CutMix with a mixing ratio drawn from Beta(2, 2). Finally, we adopt stochastic weight average (SWA) [50] on PCME++ for the last 10 epochs to obtain a more generalizable and robust solution [51], except the L/14 backbone due to the GPU memory issue. For comparison methods, The triplet loss margin is set to 0.2 (for VSE $_{\infty}$  [15]) and the initial softmax temperature for InfoNCE [19] is set to 1.0. PCME [14] uses the same initialization of PCME++ for affine transform and VIB, while 8 samples are drawn per input for computing matching probability.

For the evaluation, the best model based on the validation rsum is selected. When SWA is applied, models are not selected based on validation scores but the last averaged model is used. More detailed training settings and resource information for the experiments are described in Appendix B.2.

### 3.2 COCO ITM results

**Main results.** Table 1 shows the main comparison results of PCME++ and other ITM methods. We first observe that PCME++ consistently outperforms other methods in all evaluation metrics on different backbones. Second, we observe that the scale-up of PCME++ leads to consistent performance increases without hyperparameter tuning, while deterministic counterparts (*e.g.*, VSE $_{\infty}$  and InfoNCE) suffer from performance drops when scaling up from ViT-B/16 to ViT-L/14. The full image-to-text and text-to-image retrieval results, R@5 and R@10 are separately reported in Appendix C.6. Appendix C.1 shows more comparisons with other methods using different backbones.

**Noisy correspondence.** Table 2 shows the additional comparisons under noisy correspondence, *i.e.*, by assuming that the training annotations are noisy. Following Huang et al. [16], the image-text

Table 1: **COCO cross-modal retrieval performances.** Comparisons of ITM methods with various backbones in ECCV Caption, CxC and COCO Caption. “Prob?” denotes whether a method is a probabilistic method or not. Each number is the average between the image-to-text retrieval and text-to-image retrieval results, and is the average of three different runs. The full numbers and standard errors are in Appendix C.6. † denotes the re-evaluated results by the official checkpoints, otherwise, numbers are produced by our trained models.

Backbone	Method	Prob?	ECCV Caption [25]			CxC [37]		COCO [20]	
			mAP@R	R-P	R@1	R@1	1K R@1	5K R@1	RSUM
ViT-B/32 (151M)	CLIP ZS <sup>†</sup> [19]	✗	26.8	36.9	67.1	42.0	59.5	40.3	471.9
	VSE <sub>∞</sub> [15]	✗	40.0	49.5	83.1	57.1	75.5	55.2	536.5
	InfoNCE [19]	✗	39.0	48.7	81.7	54.9	74.0	53.0	532.6
	PCME [14]	✓	39.1	48.9	81.4	54.7	73.8	53.0	532.0
	Ours	✓	40.0	49.6	83.3	57.0	75.5	55.3	537.1
	Ours + SWA	✓	<b>40.2</b>	<b>49.8</b>	<b>83.6</b>	<b>57.2</b>	<b>75.6</b>	<b>55.5</b>	<b>537.3</b>
ViT-B/16 (150M)	CLIP ZS <sup>†</sup>	✗	29.3	39.0	71.1	44.3	62.0	42.7	481.0
	VSE <sub>∞</sub>	✗	41.7	50.6	86.3	62.3	79.1	60.7	547.2
	InfoNCE	✗	41.1	50.4	84.8	60.9	78.3	59.3	545.5
	PCME	✓	41.0	50.3	84.3	59.9	77.8	58.2	544.2
	Ours	✓	41.9	51.0	86.3	62.8	79.6	61.3	548.7
	Ours w/ SWA	✓	<b>42.0</b>	<b>51.1</b>	<b>86.6</b>	<b>63.1</b>	<b>79.7</b>	<b>61.6</b>	<b>548.9</b>
ViT-L/14 (428M)	CLIP ZS <sup>†</sup>	✗	28.0	37.8	72.2	48.1	64.8	46.4	491.6
	VSE <sub>∞</sub>	✗	20.2	31.5	46.2	24.3	44.5	22.7	424.3
	InfoNCE	✗	35.6	45.8	75.6	48.0	69.5	45.9	520.6
	PCME	✓	41.2	50.3	86.0	63.4	80.3	61.9	550.4
	Ours	✓	<b>42.1</b>	<b>50.8</b>	<b>88.8</b>	<b>65.9</b>	<b>81.8</b>	<b>64.3</b>	<b>554.7</b>

Table 2: **COCO noisy correspondence.** Noisy correspondence results using the ViT-B/32 backbone, except NCR [16] are shown. NCR scores are re-evaluated by the official weights. Noise ratio 0% is the same as Table 1.

Noise ratio	Method	ECCV Caption			CxC		COCO	
		mAP@R	R-P	R@1	R@1	1K R@1	5K R@1	RSUM
20%	VSE <sub>∞</sub>	37.1	46.6	<b>80.1</b>	<b>53.3</b>	<b>72.0</b>	<b>51.4</b>	<u>520.2</u>
	InfoNCE	35.5	46.0	75.5	47.2	67.8	45.2	513.3
	PCME	37.3	47.2	79.2	49.9	69.9	48.1	519.3
	PCME++ (ours)	<b>37.9</b>	<b>47.7</b>	<u>79.7</u>	<u>51.3</u>	<u>70.8</u>	<u>49.5</u>	<b>522.4</b>
	NCR <sup>†</sup> [16]	35.9	46.0	78.0	50.6	70.1	48.8	518.6
50%	VSE <sub>∞</sub>	17.6	28.2	43.6	20.0	38.5	18.4	390.5
	InfoNCE	33.2	43.8	72.1	43.3	64.0	41.3	498.2
	PCME	<b>34.7</b>	<b>45.2</b>	73.3	45.0	<u>65.8</u>	43.0	<u>505.7</u>
	PCME++ (ours)	<u>34.4</u>	<u>44.6</u>	<u>75.0</u>	<u>46.0</u>	65.7	<u>44.0</u>	503.9
	NCR <sup>†</sup>	34.0	44.3	<b>75.1</b>	<b>47.3</b>	<b>66.8</b>	<b>45.5</b>	<b>508.5</b>

relationships are randomly shuffled with probability of 20% and 50%. A specifically designed method for solving the noisy correspondence problem, NCR [16], is also compared with the comparison methods. Following Huang et al. [16], the model selection criterion is also based on the clean validation rsum as the clean dataset scenario. There are three findings in the table. First, the hardest negative mining-based triplet loss (VSE<sub>∞</sub>) shows vulnerability on strong noisy correspondence, *e.g.*, 50%. Second, although the probabilistic methods, such as PCME and PCME++, are not designed for tackling noisy correspondence, they successfully handle the noisy correspondence scenario, especially showing outperforming precision-based metrics than NCR. Lastly, we observe that under a strong noisy annotation scenario with a 50% noise ratio, PCME shows better scores than PCME++ in some metrics. I presume that it is because the effect of the proposed techniques, such as pseudo-positives and MSDA, can be weakened under an extremely noisy scenario. It will be an interesting topic to combine noisy correspondence and probabilistic embedding, and I leave this for future work.

### 3.3 Ablation study

**Optimization.** Table 3 shows that all the proposed techniques effectively improve probabilistic ITM. More detailed hyperparameter studies for each optimization are in Appendix C.2. Table 4 shows

Table 3: **Effect of optimization methods.** Ablation study on VIB [38], Pseudo-Positives (PP), Mixed Sample Data Augmentation (MSDA), and SWA [50] with a ViT-B/32 backbone are shown.

VIB	PP	MSDA	SWA	ECCV Caption			CxC	COCO		
				mAP@R	R-P	R@1	R@1	1K R@1	5K R@1	RSUM
✗	✗	✗	✗	38.9	48.6	82.2	56.7	75.2	54.9	535.9
✓	✗	✗	✗	39.2	49.0	82.2	56.1	74.9	54.3	535.1
✗	✓	✗	✗	39.0	48.6	82.7	56.8	75.2	55.0	536.0
✓	✓	✗	✗	39.6	49.2	82.6	56.3	74.8	54.5	534.8
✓	✓	✓	✗	40.0	49.6	83.3	57.0	75.5	55.3	537.1
✓	✓	✓	✓	<b>40.2</b>	<b>49.8</b>	<b>83.6</b>	<b>57.2</b>	<b>75.6</b>	<b>55.5</b>	<b>537.3</b>

Table 4: **Effect of probability distance on training objective.** Results on ViT-B/32 backbone with VIB loss.

Probability distance	ECCV Caption			CxC	COCO		
	mAP@R	R-P	R@1	R@1	1K R@1	5K R@1	RSUM
Wasserstein 2-distance	26.7	35.5	69.0	46.3	64.5	44.9	484.6
Match probability (PCME [14])	39.1	48.9	81.4	54.7	73.8	53.0	532.0
Proposed (Equation (1))	<b>39.2</b>	<b>49.0</b>	<b>82.2</b>	<b>56.1</b>	<b>74.9</b>	<b>54.3</b>	<b>535.1</b>

Table 5: **Impact of architecture design choice.** Details are the same as the previous tables.

# layers for $\log \sigma^2$	GPO	ECCV Caption			CxC	COCO		
		mAP@R	R-P	R@1	R@1	1K R@1	5K R@1	RSUM
1	✗	37.4	47.4	79.2	51.0	70.4	49.2	521.8
2	✓	<b>40.2</b>	<b>49.7</b>	83.2	56.6	75.3	54.8	536.5
1	✓	40.0	49.6	<b>83.3</b>	<b>57.0</b>	<b>75.5</b>	<b>55.3</b>	<b>537.1</b>

the impact of the probability distance on training objective (Equation (2)) by replacing  $d(\mathbf{Z}_v, \mathbf{Z}_t)$ . For a fair comparison, all newly proposed optimization techniques except VIB for experiments are omitted. As we already observed in Figure 2, we confirm that Wasserstein distance is not a proper uncertainty estimate as a training objective. Also, the table shows that PCME++ outperforms PCME in all metrics. I presume it is because the matching probability is an approximated value by Monte Carlo approximation, therefore, the distance value will have an approximation gap.

**Architecture.** Table 5 shows the architecture ablation study: (1) GPO improves overall performances; (2) if we use a more complex  $\log \sigma^2$  head, ECCV Caption metrics are slightly improved by capturing ambiguity caused by FNs well. However, the performance improvements are marginal, and it shows inferior  $R@k$  scores than a shallower  $\log \sigma^2$  head. Therefore, PCME++ uses the number of layers for the  $\log \sigma^2$  head as 1.

### 3.4 Uncertainty analysis

From Equation (1), we can define the data uncertainty as  $\|\sigma^2\|_1$ , i.e., the summation of the variance. Based on the data uncertainty, Figure 5 shows how the uncertainty captures the ambiguity of datasets. The average COCO 1K  $R@1$ s for each modality in each of the 10 uncertainty bins are reported in the figure. We observe that by the uncertainty increased, COCO  $R@1$  (the same distribution as the training dataset) is decreased. The results support that the learned uncertainty by PCME++ can capture the inherent ambiguity of the matching annotations.

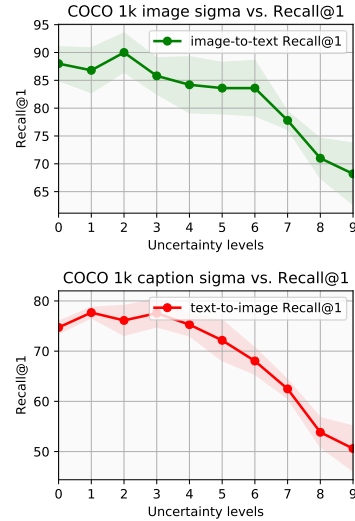


Figure 5:  $\|\sigma^2\|_1$  vs.  $R@1$ .

Figure 6 shows examples of uncertain images and captions, and their retrieved items (more examples are in Appendix C.5). The figure shows that data that can be matched with more samples have higher uncertainty values. As shown in the figure, the retrieved items for uncertain inputs are highly plausible even though the retrieved items are not in the COCO ground truth. In Section 3.5 and Section 4, more benefits of the uncertainty-aware learning and the learned uncertainty are discussed.



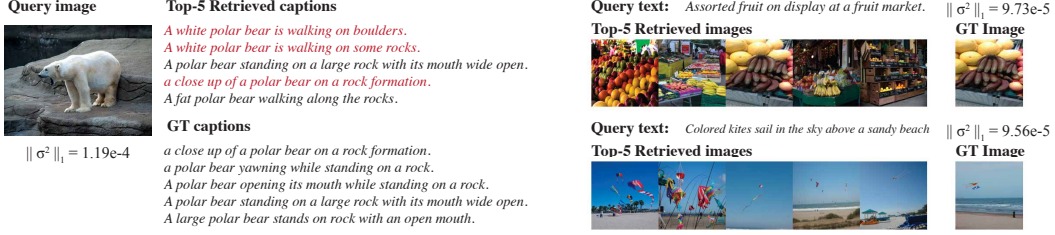


Figure 6: Example of images and captions with high uncertainty. More examples are shown in Appendix C.5.

### 3.5 More applications

**Large-scale retrieval system.** Lack of scalability is a common drawback of probabilistic retrieval systems, *i.e.*, it is difficult to apply probabilistic embeddings on a large-scale retrieval system with a billion-scale index. As the proposed probability distance, CSD (Equation (1)), is the summation of Euclidean distance of  $\mu$  and the intensity of  $\sigma^2$  of each input, we can easily and efficiently combine PCME++ and approximated KNN (ANN). First, a Euclidean distance-based index system for  $\mu$  is built as usual, while  $\sigma^2$  are saved into key-value storage. Then,  $K$  items are retrieved by performing ANN on the  $\mu$  index. Lastly, the retrieved items are re-ranked by computing the summation of the  $\mu$  distance and  $\sigma^2$  value of the retrieved items. In Appendix C.3, the comparisons of diverse retrieval strategies are shown, including ANN based on FAISS [36] and the modified ANN for PCME++. CSD is not only stronger than other probability distances but also more practical and scalable.

**Uncertainty-based prompt-tuning.** Zero-shot (ZS) classification is the task of predicting an unseen class during training. Usually, ZS classification is done by converting class information as a text sentence (*e.g.*, “a photo of a cat”) and mapping into a shared embedding space with the input. For image ZS classification tasks, large-scale ITM pre-training, such as CLIP [19], has become a standard approach. Despite their usability and generalizability, ZS needs hand-crafted prompt engineering for converting class information into proper text sentences. For example, Radford et al. [19] showed that taking the average of 80 different context prompts improves ImageNet [52] top-1 ZS accuracy by 3.5% over a single prompt (“a photo of { · }”). However, designing the best-performing prompts for every novel task is time-consuming.

This paper investigates the potential of PCME++ for automatic prompt engineering using the learned text uncertainty: The uncertainties of prompts for each class are computed, (*e.g.*, “a photo of a cat”, “a photo of many cat”, ...), and the most uncertain text prompts are discarded. Table 6 shows a study on the proposed simple automatic prompt tuning. For the experiment, ViT-S/16 models using InfoNCE loss and PCME++ are trained on the RedCaps dataset [24] for 100K iterations with 1K batch size. Here, “Top-K certain prompts” denotes that every class uses the same top-K for the filtering, and “Best top-K for each class” denotes the best top-K for each class are chosen, *e.g.*, “coral fungus” needs all 80 prompts, while “ringlet butterfly” only needs Top-1 certain prompt while other uncertain 79 prompts are discarded. With this simple strategy, the ZS performance is increased with a significant gap (8.58  $\rightarrow$  14.75). The full description of our ZS experiments are provided in Appendix C.4.

Table 6: ImageNet (IN) Zero-shot (ZS).

Model	Prompts	Top-1 Acc
InfoNCE	“A photo of { · }”	13.05
	All 80 prompts	13.41
PCME++	“A photo of { · }”	8.58
	All 80 prompts	9.33
	Top-K certain prompts	9.37
	Best top-K for each class	<b>14.75</b>

## 4 Limitations and Discussions

**Normal distribution with diagonal covariance would be insufficient?** One can argue that the uncertainty modeling power of PCME++ can be improved by relaxing the diagonal covariance condition. However, Oh et al. [26] showed that if the dimensionality of the embedding space and the number of “hidden features” are the same (*e.g.*, if an image is the combination of two digits, then the number of potential latent features for each input is two), then the diagonal covariance condition can sufficiently capture the inherent uncertainty of the dataset. In practice, we use a very high dimensional embedding space (*e.g.*, 1024) that can sufficiently capture complex relationships between features. Also, in practice, if we relax the condition, the dimensionality of the  $\log \sigma^2$  head output should be about 1M (= 1024  $\times$  1024), which will require expensive computational budgets and large memory.

**Additional sampling is still required if we use other density functions.** The proposed probabilistic distance is defined in distribution-free:  $\mathbb{E}_{\mathbf{Z}_v, \mathbf{Z}_t} \|\mathbf{Z}_v - \mathbf{Z}_t\|_2^2$ . However, the closed-form solution (CSD) is specifically designed for normally distributed embeddings. If one needs probabilistic embeddings with different distributions, such as von Mises–Fisher distribution [35] or Laplacian distribution [34], CSD is no longer applicable. Instead, we can adapt any distribution to PCME++ by using a Monte Carlo approximation, *i.e.*, by computing  $\frac{1}{n \times m} \sum_{z_v^i = z_v^1}^{z_v^n} \sum_{z_t^j = z_t^0}^{z_t^m} \|z_v^i - z_t^j\|_2^2$ , where  $z_v^i \sim \mathbf{Z}_v$  and  $z_t^j \sim \mathbf{Z}_t$ . This change will share the expensive computation issue of previous approaches [26, 14], but the additionally introduced techniques in PCME++ for mitigating the loss saturation issue (*i.e.*, pseudo-positives and MSDA) will still be effective. Applying other probabilistic densities to PCME++ and discovering the effect of different distribution choices will be interesting future work.

**How does uncertainty help learning image-text representations?** As shown in the main experiments, the probabilistic approach is helpful for improving the retrieval performances, but the gaps are not significant (*e.g.*, Table 1 shows that in ViT-B/32, the gap between VSE $\infty$  and PCME++ with SWA is not significant). However, as shown in larger backbone experiments (ViT-B/16 and ViT-L/14) and noisy correspondence experiments (Table 2), PCME++ shows more generalizable performances compared to the existing state-of-the-art ITM methods with the same backbone. Furthermore, as shown in Section 3.4 and Section 3.5, the learned uncertainty by PCME++ shows high interpretability of the datasets as well as the controllability by the users when the rejection of the retrieved items is required. Thus, I believe that the uncertainty-aware learning paradigm and the learned uncertainty will be helpful for image-text matching problems and downstream tasks, such as zero-shot classification.

## 5 Conclusion

This paper addresses the inherent ambiguity of ITM tasks by PCME++. A novel closed-form probability distance and a new matching objective function for efficiency and effectiveness is presented. PCME++ is further enhanced by incorporating a pseudo-positive strategy and a mixed sample data augmentation strategy, successfully addressing the loss saturation issue associated with abundant false negatives. Experimental results demonstrate the extensibility of PCME++ to various applications, such as image-caption cross-modal retrieval, mitigating noisy correspondences, automatic prompt tuning for zero-shot classification, and understanding the inherent ambiguity of a dataset.

## Acknowledgement

I would like to thank my NAVER AI Lab colleagues for valuable discussions, including Sangdoo Yun, Wonjae Kim, Jiyoung Lee, Dongyoon Han, Byeongho Heo, Taekyung Kim, Song Park and Jung-Woo Ha. NAVER Smart Machine Learning (NSML) platform [53] is used for the experiments.

## Societal Impact

This work aims to learn better image-text representations based on a probabilistic approach. As shown in the experiments, PCME++ has the potential to improve the interpretability and the controllability of learned representations by providing an additional degree of freedom to the users. Accordingly, PCME++ shares the potential impact of developing general image-text representations with better interpretability and controllability. For example, as shown by Radford et al. [19], visual-textual representations trained on a large-scale web dataset often suffers from biases in the web; PCME++ can both mitigate or enhance the biases using its interpretability and controllability.

## Appendix

More additional materials are included. More details of our method are described in §A, including the full derivation of the closed-form probabilistic distance (§A.1), the toy experiments (§A.2), comparisons between PCME and PCME++ (§A.3), and pseudo-code of PCME++ (§A.4). In experimental protocol details section (§B), the benchmark dataset details (§B.1), hyperparameter and resource details (§B.2) are shown. Finally, additional experimental results (§C), including comparisons with state-of-the-art (§C.1), the full ablation studies (§C.2), the comparisons of different retrieval strategies (§C.3), automatic prompt-tuning experiments (§C.4), more qualitative examples (§C.5), and the full experimental results and error bars (§C.6) are presented.

## A Method Details

### A.1 Derivation of the closed-form probability distance

In this subsection, the full derivation of Equation (1) is shown. We first show two simple well-known lemmas and conclude the full proof using them.

**Lemma 1.** *Let  $X$  and  $Y$  be independent normally distributed random variables where  $X \sim \mathcal{N}(\mu_X, \Sigma_X)$  and  $Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$ . Then, the subtraction between  $X$  and  $Y$  is another normal distribution, i.e.,  $(X - Y) \sim \mathcal{N}(\mu_X - \mu_Y, \Sigma_X + \Sigma_Y)$ .*

*Proof.* Let  $\phi_X(u) = \exp(it^\top \mu_X - \frac{1}{2}t^\top \Sigma_X t)$  be a characteristic function of normally distributed random variable  $X$ . Using the fact that  $-Y \sim \mathcal{N}(-\mu_Y, \Sigma_Y)$ , we can compute the summation of  $\phi_X(u)$  and  $\phi_{-Y}(u)$  as follows:

$$\phi_{X-Y}(u) = \exp(it^\top \mu_X - \frac{1}{2}t^\top \Sigma_X t) \exp(-it^\top \mu_Y - \frac{1}{2}t^\top \Sigma_Y t) = \exp(it^\top (\mu_X - \mu_Y) - t^\top (\Sigma_X + \Sigma_Y)t). \quad (\text{A.1})$$

Hence,  $X - Y$  is another normal distribution,  $\mathcal{N}(\mu_X - \mu_Y, \Sigma_X + \Sigma_Y)$ .  $\square$

**Lemma 2.** *Let  $X \sim \mathcal{N}(\mu, \Sigma)$ . Then  $\mathbb{E}\|X\|_2^2 = \|\mu\|_2^2 + \text{tr}(\Sigma)$ .*

*Proof.* We first re-parameterize a random variable  $X$  as  $X = \mu + SZ$ , where  $S$  is the square root matrix of  $\Sigma$ , i.e.,  $SS^\top = \Sigma$ , and  $Z$  is a standard normal distribution. Note that  $S$  always exists because  $\Sigma$  is a positive semi-definite by definition. Using  $\mathbb{E}[Z] = 0$ , the property of Frobenius norm  $\|A\|_F^2 = \text{tr}(A)$  and the property of trace  $\text{tr}(AB) = \text{tr}(BA)$ , we have:

$$\begin{aligned} \mathbb{E}\|X\|_2^2 &= \mathbb{E}_Z[\|\mu\|_2^2 + 2\mu^\top SZ + \|Z^\top S^\top SZ\|_2^2] = \|\mu\|_2^2 + \mathbb{E}_Z\|Z^\top S^\top SZ\|_2^2 \\ &= \|\mu\|_2^2 + \mathbb{E}_Z\text{tr}(Z^\top S^\top SZ) = \|\mu\|_2^2 + \text{tr}(S^\top S \mathbb{E}_Z[Z Z^\top]) = \|\mu\|_2^2 + \text{tr}(\Sigma). \end{aligned} \quad (\text{A.2})$$

$\square$

**Proposition 1.** *Let  $X$  and  $Y$  be independent normally distributed random variables where  $X \sim \mathcal{N}(\mu_X, \Sigma_X)$  and  $Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$ . Then we have  $\mathbb{E}\|X - Y\| = \|\mu_X - \mu_Y\|_2^2 + \text{tr}(\Sigma_X + \Sigma_Y)$ .*

*Proof.* By combining Lemma 1 and Lemma 2, the proof is completed.  $\square$

### A.2 Toy experiments

In Section 2.2, a 2-D toy dataset is introduced for comparing various objective functions under inherent uncertainty. The toy dataset has three classes with ‘‘confusing samples’’ between classes, i.e., a confusing sample randomly can be either class A or class B. The number of confusing samples are 30% of the total data points. To synthesize the samples, a centroid is randomly chosen for each class. Using the centroid, each sample is randomly drawn from  $\mu + 0.1 \times \mathcal{N}(0, I)$ . 500 samples are drawn for each class and 150 samples of them are chosen as ‘‘confusing samples’’, i.e., there are 1500 samples with 1050 certain samples and 450 confusing samples. Then,  $\log \sigma$  of each sample is randomly drawn from  $\mathcal{U}(-1.5, 1.5)$  where  $\mathcal{U}$  is a uniform distribution. In summary, the dataset has 350 confident samples for class 1, 2 and 3; 150 confusing samples for class (1, 2), (2, 3) and (3, 1).

To show the effects of different probabilistic distances, the samples are directly updated by the objective function Equation (2) with different metrics, i.e., a sample  $(\mu, \sigma)$  is directly updated by Equation (2). The dataset is directly optimized using Adam optimizer [54] with learning 0.02 during 500 epochs. The mini-batch size is set to 128. We employ the same loss function with PCME++ while the probabilistic distance is chosen from either our distance or Wasserstein distance. The animated learning progress of each method can be found in <https://naver-ai.github.io/pcmepp/>.

As described in Section 2.2, we desire the learned uncertainty can capture the data uncertainty, i.e., we expect that certain samples have small  $\sigma^2$ , while uncertain samples have large  $\sigma^2$ . After training, we observe that the average  $\sigma^2$  for certain samples and uncertain samples by PCME++ are 1.68 and 3.05, respectively. On the other hand, Wasserstein distance shows 2.69 and 2.80, respectively. The result and other experiments on large-scale datasets (Section 3.3) support that PCME++ is a proper probability distribution to capture uncertainty, while Wasserstein is not.

```

1 def compute_loss(v_mu, v_sig, t_mu, t_sig, matched):
2     """v_mu, v_sig: mean and variance for (mixed) images (N by D)
3     t_mu, t_sig: mean and variance for captions (M by D)
4     matched: denoting (i, j) image, caption pair is matched.
5             values are between 0 and 1 (N by M)"""
6     # compute a closed-form distance
7     mu_dist = ((v_mu.unsqueeze(1) - t_mu.unsqueeze(0)) ** 2).sum(-1)
8     sigma_dist = ((v_sig.unsqueeze(1) + t_sig.unsqueeze(0))).sum(-1)
9
10    # a, b: a learnable affine transform
11    logits = -a * (mu_dist + sigma_dist) + b
12
13    # match loss can be easily computed by BCE loss
14    match_loss = BCE(logits, matched)
15
16    # compute pseudo-positive (pp) match loss
17    gt_labels, gt_indices = torch.max(matched, dim=1)
18    gt_vals = logits[:, gt_indices].diag()
19    pseudo_gt_indices = (logits >= gt_vals)
20    pp_matched = (gt_labels.unsqueeze(1) * (pseudo_gt_indices))
21    matched[pseudo_gt_indices] = pp_matched[pseudo_gt_indices]
22    pp_match_loss = BCE(logits, matched)
23
24    # compute VIB loss
25    v_vib = -0.5 * (1 + torch.log(v_sig) - v_mu ** 2 - v_sig).mean()
26    t_vib = -0.5 * (1 + torch.log(t_sig) - t_mu ** 2 - t_sig).mean()
27    vib_loss = v_vib + t_vib
28
29    # final loss, alpha and beta are hyperparameters
30    return match_loss + alpha * pp_match_loss + beta * vib_loss

```

Figure A.1: PyTorch pseudo-code of PCME++. Here,  $v\_sig$  and  $t\_sig$  are computed by taking an exponential to the output of  $\log \sigma^2$  heads. BCE denotes a binary cross-entropy function.

### A.3 Comparisons with PCME and PCME++ objective functions

We first recall the definition of matching probability:

$$\mathbb{E}_{Z_v, Z_t} \text{sigmoid}(-a\|Z_v - Z_t\|_2 + b) \approx \frac{1}{J^2} \sum_{z_v, z_t} \text{sigmoid}(-a\|z_v - z_t\|_2 + b), \quad (\text{A.3})$$

where  $J$  is the number of samples  $z_v$  and  $z_t$ . PCME directly optimized the negative log-likelihood:

$$m_{vt} \log \sum_{z_v, z_t} \text{sigmoid}(-a\|z_v - z_t\|_2 + b) + (1 - m_{vt}) \log \sum_{z_v, z_t} \text{sigmoid}(a\|z_v + z_t\|_2 + b) \quad (\text{A.4})$$

Equation (A.4) and Equation (2) share a similar formulation, but the position of the expectation is different. As the expectation is located at the outside of sigmoid, Equation (A.3) cannot be computed in a closed-form solution, but our distance can. Note that the analysis in Section 2.3 also holds for Equation (A.4), hence the PCME loss suffer from the loss saturation under abundant false negatives.

### A.4 PCME++ Pseudo-code

Figure A.1 shows the PyTorch style pseudo-code of PCME++. Note that  $\mu$  and  $\sigma$  are extracted from the augmented inputs, such as MSDA (Section 2.4) and SizeAugment [15].

## B Experimental Protocol Details

### B.1 More details of benchmark datasets

Figure B.1 illustrates the differences between evaluation datasets. Note that all evaluation benchmarks use the same training dataset described in §3.1. The COCO Caption evaluation split consists of 5,000

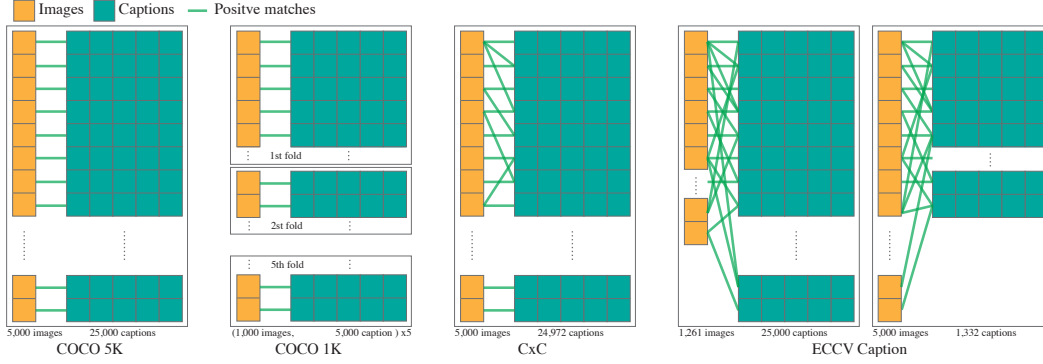


Figure B.1: **Difference between COCO 5K, 1K, CxC [37] and ECCV Caption [25].** All matches not illustrated in the image are negative. ECCV Caption has separated query sets for each modality, while other datasets use the same images and captions for both query and gallery.

Table B.1: **Hyperparameter details**

Method	CLIP ViT B/32, B/16, L/14 COCO	CLIP S/16 RedCaps
Epochs	25	25
Batch size	128	1,536
Optimizer	AdamP	AdamP
Initial learning rate	0.0005	0.0005
LR scheduling	Step	linear warmup and cosine
Layer-wise LR decay	0.7	-
Visual backbone LR decay	0.01	-
Textual backbone LR decay	0.1	-
$\beta_1, \beta_2, \epsilon$	0.9, 0.999, $10^{-8}$	0.9, 0.98, $10^{-6}$
Weight decay	0.0001	0.2
VIB $\beta$	0.0001	$10^{-6}$
PP $\alpha$	0.1	0
MSDA CutMix/Mixup $\lambda$ , mix ratio	2/2/25%	-/-/0%
Size Augment	✓	✗
Embedding dimension	1024	128
Initial $a$ and $b$	5/5	5/5
Resources	ViT B/32 1 V100 (38 hours) ViT B/16 1 V100 (75 hours) ViT L/14 8 V100 (62 hours)	8 V100 (84 hours)

images and 25,000 captions. **COCO 5K** uses the full 5,000 images and 25,000 captions where each image has five positive captions and each caption only has one positive image. For evaluation, COCO 5K measures image-to-text retrieval performances by setting 5,000 images as queries and 25,000 captions as galleries, while text-to-image retrieval performances are measured in the opposite way. **COCO 1K** uses the same positive relationships as COCO 5K, but COCO 1K uses the subset of COCO 5K, *i.e.*, there are 1,000 images and their corresponding 5,000 captions for COCO 1K split. COCO 1K measures the performances by taking an average of five different splits.

CxC [37] and ECCV Caption [25] use the same images and captions of COCO 1K/5K, but with more positive annotations. CxC uses the entire images and valid 24,972 captions among 25,000 captions (by omitting “I cannot see any image” captions). CxC has more positive annotations than COCO, but there are still many missing positives in CxC because their approach is mostly focused on text similarity, not image-text similarity. On the other hand, ECCV Caption is designed for handling false negatives of image-text pairs. ECCV Caption uses the subset of images and captions for the queries, but their retrieval database is the full dataset, *i.e.*, when performing image-to-text retrieval, the number of query images is 1,261 and the number of gallery captions are 25,000; for text-to-image retrieval, the number of query texts is 1,332 and the number of gallery images is 5,000.



Table C.1: **Comparisons with state-of-the-art models.** All numbers are reproduced by the official weights. We highlight **the best scores** except expensive retrieval methods, such as BLIP.

Method	Efficient retrieval?	ECCV Caption			CxC R@1	1K R@1	COCO	
		mAP@R	R-P	R@1			5K R@1	RSUM
CVSE [12]	✓	37.4	47.5	76.7	45.8	67.0	43.8	511.1
VSRN [8]	✓	42.3	<b>51.8</b>	81.5	48.9	69.5	46.7	515.9
NCR [16]	✓	36.4	46.3	79.9	51.8	71.0	50.0	522.6
VSE $\infty$ (BUTD region) [15]	✓	40.5	50.0	82.5	52.4	72.2	50.4	527.5
VSE $\infty$ (WSL)	✓	<b>42.4</b>	51.4	86.4	60.8	78.3	59.0	545.1
VSE $\infty$ (B/16, our implementation)	✓	41.7	50.6	86.3	62.3	79.1	60.7	547.2
ViLT [55]	✗	34.6	44.3	77.8	53.7	72.8	52.2	528.6
VinVL [56]	✗	40.8	49.6	87.8	67.8	82.4	66.4	555.5
BLIP [57]	✗	40.5	48.4	91.0	74.3	86.1	73.1	564.4
CLIP Zero-shot (L/14) [19]	✓	28.0	37.8	72.2	48.1	64.8	46.4	491.6
PCME++ (B/16)	✓	42.0	51.1	86.6	63.1	79.7	61.6	548.9
PCME++ (L/14)	✓	42.1	50.8	<b>88.8</b>	<b>65.9</b>	<b>81.8</b>	<b>64.3</b>	<b>554.7</b>

Table C.2: **Pseudo-positive  $\alpha$  ablation study.**

$\alpha$	ECCV Caption			CxC		COCO	
	mAP@R	R-P	R@1	R@1	1K R@1	5K R@1	RSUM
0.1	40.2	49.8	83.1	56.5	75.1	54.8	536.0
0.5	40.0	49.5	83.1	56.7	75.4	55.0	536.8
2	40.1	49.7	83.0	56.5	75.1	54.8	535.8
5	40.3	49.9	83.1	55.7	74.7	53.9	534.9
10	40.2	49.9	82.5	54.5	73.7	52.6	531.9

As discussed by Musgrave et al. [48] and Chun et al. [25], Recall@K is not an informative metric for measuring retrieval performances in terms of precision. Due to this reason, this paper reports mAP@R and R-Precision of ECCV Caption as the main comparison metrics.

## B.2 Hyperparameter and resource details

Table B.1 shows the detailed hyperparameter settings and the detailed GPU resource information.

## C Additional Experimental Results

### C.1 Comparisons with state-of-the-arts

Table C.1 shows the comparisons of PCME++ and state-of-the-arts with different backbones. Note that ViLT [55], VinVL [56] and BLIP [57] need heavy computations to perform retrieval because they have to compute pair-wise similarity for all pairs. For example, they need  $O(5,000 \times 25,000)$  computation budgets for measuring retrieval performances. On the other hand, methods with separated encoders just need  $O(5,000 + 25,000)$  computation budgets, 4,166 times smaller computation budgets compared to expensive retrieval methods. Therefore, the table only highlights the best retrieval performances among efficient retrieval methods for a fair comparison. PCME++ achieves the best recall scores for all evaluation benchmarks while showing second and third best ECCV mAP@R and R-Precision. I presume that it is because of the capability of the backbone architecture. For example, VSE $\infty$  with CLIP B/16 backbone shows much better recall scores than VSE $\infty$  with WSL backbone, but VSE $\infty$  (WSL) shows better mAP@R and R-Precision than the CLIP backbone. We expect that PCME++ can outperform the previous retrieval methods in precision metrics if we train PCME++ using different backbones, such as large-scale weakly supervised learning (WSL) backbone [58].

### C.2 More ablation studies

Table C.2 shows the ablation study for pseudo positive  $\alpha$ . The table shows that our method is not very sensitive to the choice of  $\alpha$ . We choose  $\alpha = 0.1$ , which shows the second best ECCV Caption

Table C.3: MSDA ablation study.

Mixup $\lambda$	CutMix $\lambda$	Mix ratio	in-batch?	ECCV Caption mAP@R	R-P	R@1	CxC R@1	1K R@1	COCO 5K R@1	RSUM
2	0	25%	✓	37.3	47.3	79.6	51.9	71.7	50.0	525.5
0	2	25%	✓	37.5	47.6	79.2	50.5	70.9	48.8	523.4
1	1	50%	✗	39.7	49.5	81.8	55.2	74.5	53.5	534.0
1	1	25%	✗	39.9	49.6	82.3	55.5	74.6	53.8	534.4
2	2	25%	✗	40.0	49.6	82.8	55.8	74.5	54.0	534.4
1	1	50%	✓	39.9	49.6	82.7	55.4	74.4	53.7	534.1
2	2	25%	✓	<b>40.1</b>	<b>49.7</b>	<b>82.9</b>	<b>56.5</b>	<b>75.0</b>	<b>54.7</b>	<b>535.9</b>

Table C.4: Effect of inference methods. We compare the mean-only inference and probability distance-based inferences using our ViT-B/32 SWA model. Each number is the average of different three runs.

Method	Prob?	ECCV Caption mAP@R	R-P	R@1	CxC R@1	1K R@1	COCO 5K R@1	RSUM
Mean only	✗	<b>40.2</b>	49.8	83.5	56.9	75.2	55.2	536.3
2-Wasserstein	✓	<b>40.2</b>	<b>49.9</b>	83.0	56.6	75.2	54.8	535.9
CSD (ours)	✓	<b>40.2</b>	49.8	<b>83.6</b>	<b>57.2</b>	<b>75.6</b>	<b>55.5</b>	<b>537.3</b>
FAISS [36] (Meany only)	✗	<b>40.1</b>	<b>49.7</b>	<b>83.5</b>	56.4	74.7	54.6	531.2
FAISS + $\sigma$ re-ranking	✓	<b>40.1</b>	<b>49.7</b>	83.2	<b>56.6</b>	<b>74.8</b>	<b>54.8</b>	<b>531.7</b>

mAP@R and COCO recall measures. Table C.3 shows the ablation study for the mixed sample data augmentation design choice. The design choice for PCME++ shows the best performance.

### C.3 Comparisons of different retrieval strategies

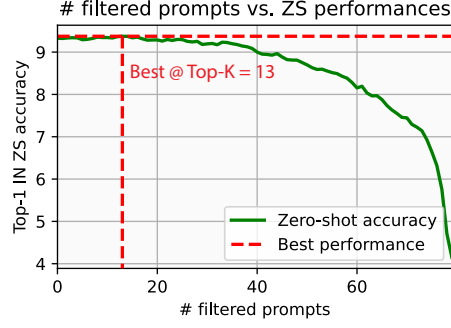
Table C.4 shows the comparisons of different retrieval strategies using PCME++ B/32 model. “Mean only” denotes the retrieval strategy only using  $\mu$  vectors, without  $\sigma$ . “2-Wasserstein” and “CSD” denote that each probabilistic distance is used for the retrieval. In the table, we observe that mean-only retrieval shows sufficiently good performances but using CSD improves the overall performances.

This paper additionally shows the approximated KNN (ANN) results using FAISS [36]. First, a FAISS search index using  $\mu$  vectors is built. Then, ANN is performed on the FAISS index to get the ranked list. Finally, the ranked list is re-ranked by CSD. Here, CSD can be efficiently computed by storing gallery  $\sigma$  into a fast key-value storage, such as Redis. As shown in the table, ANN can be efficiently and effectively applied to PCME++ with a reasonable computation-performance trade-off.

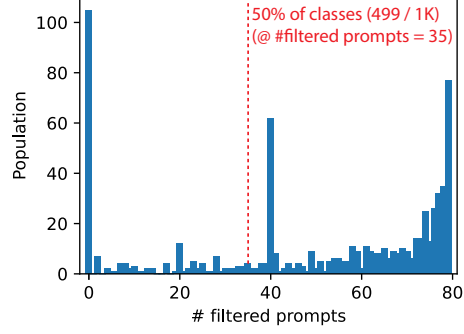
### C.4 Details of automatic prompt-tuning by PCME++

For the experiments, a randomly initialized ViT-S/16 backbone is trained by InfoNCE loss and PCME++ loss on RedCaps [24] using hyperparameters in Table B.1. The pre-trained models are evaluated on the ImageNet [52] zero-shot (ZS) classification task. Specifically, 80 prompts provided by CLIP [19] (shown in the next paragraph) are used for the ZS classification. In Table 6, “A photo of a .” denotes that only “A photo of a .” prompt is used for the zero-shot classification, while “All 80 prompts” denotes that all 80 prompts are used for computing text embeddings and the average text embedding is used for the zero-shot classification.

**80 based prompts.** a photo of a {}, a bad photo of a {}, a photo of many {}, a sculpture of a {}, a photo of the hard to see {}, a low resolution photo of the {}, a rendering of a {}, graffiti of a {}, a bad photo of the {}, a cropped photo of the {}, a tattoo of a {}, the embroidered {}, a photo of a hard to see {}, a bright photo of a {}, a photo of a clean {}, a photo of a dirty {}, a dark photo of the {}, a drawing of a {}, a photo of my {}, the plastic {}, a photo of the cool {}, a close-up photo of a {}, a black and white photo of the {}, a painting of the {}, a painting of a {}, a pixelated photo of the {}, a sculpture of the {}, a bright photo of the {}, a cropped photo of a {}, a plastic {}, a photo of the dirty {}, a jpeg corrupted photo of a {}, a blurry photo of the {}, a photo of the {}, a good photo of the {}, a rendering of the {}, a {} in a video game., a photo of one



(a) The same Top-K filtering results.



(b) The best Top-K for each class.

Figure C.1: **Automatic prompt tuning results.** (a) shows the ImageNet (IN) zero-shot (ZS) results when prompts are filtered by the same top-K for every class. The ZS performance shows the best at the number of filtered prompts is 13, but the performance improvement is marginal. (b) shows the population of best top-K filtering for all classes. Here, 105 classes among 1,000 classes show the best performance when there is no filtering, while 77 classes show the best ZS score when filtering out except the most 1 certain prompt.

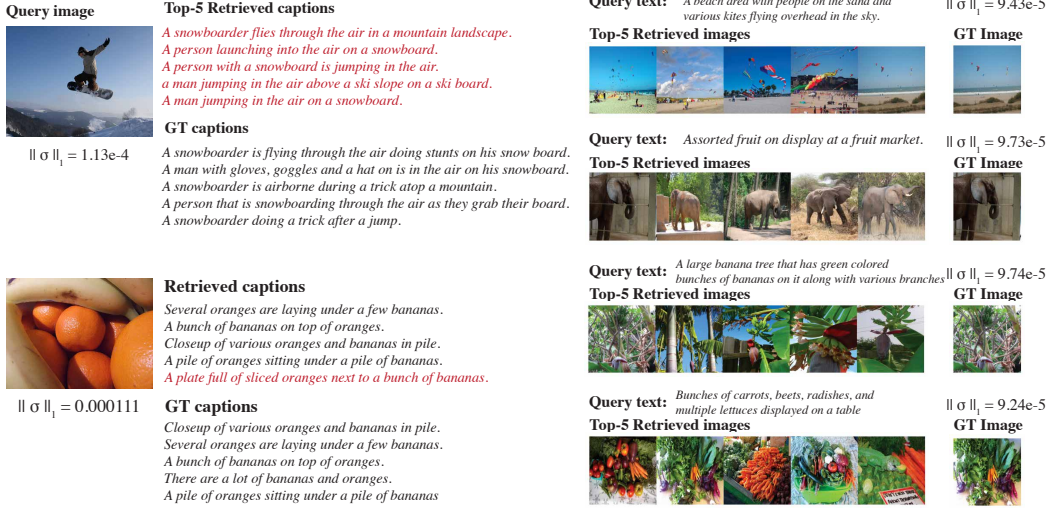


Figure C.2: **More examples of images and captions with high uncertainty**

{}, a doodle of a {}, a close-up photo of the {}, the origami {}, the {} in a video game, a sketch of a {}, a doodle of the {}, a origami {}, a low resolution photo of a {}, the toy {}, a rendition of the {}, a photo of the clean {}, a photo of a large {}, a rendition of a {}, a photo of a nice {}, a photo of a weird {}, a blurry photo of a {}, a cartoon {}, art of a {}, a sketch of the {}, a embroidered {}, a pixelated photo of a {}, itap of the {}, a jpeg corrupted photo of the {}, a good photo of a {}, a plushie {}, a photo of the nice {}, a photo of the small {}, a photo of the weird {}, the cartoon {}, art of the {}, a drawing of the {}, a photo of the large {}, a black and white photo of a {}, the plushie {}, a dark photo of a {}, itap of a {}, graffiti of the {}, a toy {}, itap of my {}, a photo of a cool {}, a photo of a small {}, a tattoo of the {}.

This paper explores the potential of PCME++ for automatic prompt-tuning with a simple uncertainty-based filtering. First, the prompts for every class are sorted by their uncertainty, *i.e.*,  $\|\sigma\|_1$ . Then, uncertain prompts are filtered out, and the remained prompts are used for ZS classification. Here, two strategies are tested. First, the same top-K uncertain prompts for all classes are filtered. As shown in Figure C.1a, this strategy slightly improves the overall performances, but it only shows a marginal improvement against the “all” baseline (+0.04%). To further improve the uncertainty-based filtering, the strategy with different top-K for different prompts is also explored. As shown in Table 6, this strategy shows very effective performance improvement against the baseline (+5.42%). Figure C.1b

Table C.5: Image-to-text retrieval R@5 and R@10 results.

Backbone	Method	CxC		COCO			
		R@5	R@10	1K R@5	1K R@10	5K R@5	5K R@10
ViT-B/32 (151M)	VSE $\infty$	87.1	93.4	96.7	98.8	85.4	92.2
	InfoNCE	87.3	93.4	96.5	98.9	85.9	92.3
	PCME	87.5	93.5	96.6	98.7	85.8	92.3
	PCME++	88.4	94.0	97.0	99.0	87.0	93.0
	PCME++ + SWA	88.5	94.0	97.0	99.0	87.1	92.9
ViT-B/16 (150M)	VSE $\infty$	91.1	95.6	97.8	99.4	89.9	94.8
	InfoNCE	90.9	95.8	97.8	99.3	89.7	94.9
	PCME	90.5	95.4	97.7	99.3	89.2	94.5
	PCME++	91.4	95.7	97.9	99.3	90.3	94.9
	PCME++ + SWA	91.5	95.9	97.9	99.3	90.4	95.1
ViT-L/14 (428M)	VSE $\infty$	58.8	72.6	82.2	91.4	55.7	69.4
	InfoNCE	82.8	91.7	95.3	98.6	80.2	90.0
	PCME	91.8	95.9	98.1	99.4	90.7	95.2
	PCME++	93.4	96.8	98.5	99.6	92.2	96.2

Table C.6: Text-to-image retrieval R@5 and R@10 results.

Backbone	Method	CxC		COCO			
		R@5	R@10	1K R@5	1K R@10	5K R@5	5K R@10
ViT-B/32 (151M)	VSE $\infty$	77.7	86.5	92.2	96.7	75.5	84.8
	InfoNCE	77.3	86.5	92.3	96.9	75.1	84.7
	PCME	77.3	86.4	92.1	96.9	75.0	84.6
	PCME++	78.5	87.1	92.8	97.1	76.5	85.4
	PCME++ + SWA	78.6	87.3	92.8	97.1	76.5	85.5
ViT-B/16 (150M)	VSE $\infty$	82.0	89.5	94.2	97.5	80.3	88.2
	InfoNCE	81.3	89.1	94.0	97.7	79.5	87.7
	PCME	80.9	88.9	93.9	97.7	79.1	87.5
	PCME++	82.0	89.7	94.4	97.8	80.3	88.3
	PCME++ + SWA	82.1	89.7	94.4	97.8	80.4	88.4
ViT-L/14 (428M)	VSE $\infty$	46.4	61.1	74.2	87.4	42.9	57.1
	InfoNCE	73.6	84.2	91.3	96.4	71.0	82.3
	PCME	82.7	90.2	94.5	97.8	81.1	88.8
	PCME++	84.0	90.8	95.1	98.1	82.6	89.7

shows the detailed population of the best top-K filtering per class. Here, the classes whose accuracy is 0% are omitted. Interestingly, we observe that 10% of classes (105) show the best ZS performances when all 80 prompts are used. On the other hand, about half of the classes (499) show the best performance when more than 35 prompts are filtered out.

This primitive study on uncertainty-based prompt tuning has two limitations. First, the baseline pre-trained model is too weak (13.41% top-1 IN ZS performance with InfoNCE) compared to well-known baselines, such as CLIP [19]. Second, this study has no validation split, *i.e.*, the best top-K prompt for each class is directly searched from the ImageNet validation split. Searching for the best top-K for each class without direct tuning on test split using strong probabilistic pre-trained image-text representations will be an interesting future research direction.

### C.5 More examples of uncertain samples

More examples of uncertain images and captions are shown in Figure C.2.

### C.6 Full experimental results

The image-to-text and text-to-image R@5 and R@10 results are shown in Table C.5 and Table C.6. The full experimental results, including separated image-to-text and text-to-image retrieval results for the main table, and standard errors, are included in Table C.7, Table C.8 and Table C.9. The

Table C.7: Image-to-text retrieval full results.

Backbone	Method	ECCV Caption			CxC	COCO	
		mAP@R	R-P	R@1	R@1	1K R@1	5K R@1
ViT-B/32 (151M)	VSE $\infty$	31.7 ( $\pm 1.2$ )	42.8 ( $\pm 0.9$ )	75.6 ( $\pm 3.2$ )	61.8 ( $\pm 4.1$ )	80.4 ( $\pm 3.2$ )	60.2 ( $\pm 4.2$ )
	InfoNCE	31.2 ( $\pm 0.1$ )	42.3 ( $\pm 0.1$ )	75.4 ( $\pm 1.1$ )	61.8 ( $\pm 0.1$ )	80.3 ( $\pm 0.6$ )	60.1 ( $\pm 0.2$ )
	PCME	31.2 ( $\pm 0.0$ )	42.3 ( $\pm 0.0$ )	74.9 ( $\pm 0.3$ )	61.5 ( $\pm 0.6$ )	80.1 ( $\pm 0.2$ )	59.9 ( $\pm 0.6$ )
	PCME++	32.2 ( $\pm 0.1$ )	43.4 ( $\pm 0.1$ )	77.1 ( $\pm 1.0$ )	63.3 ( $\pm 0.2$ )	81.3 ( $\pm 0.3$ )	61.8 ( $\pm 0.2$ )
	PCME++ + SWA	32.4 ( $\pm 0.2$ )	43.5 ( $\pm 0.2$ )	77.8 ( $\pm 0.7$ )	63.7 ( $\pm 0.4$ )	81.4 ( $\pm 0.2$ )	62.3 ( $\pm 0.4$ )
ViT-B/16 (150M)	VSE $\infty$	34.4 ( $\pm 0.1$ )	44.8 ( $\pm 0.2$ )	81.2 ( $\pm 0.7$ )	69.4 ( $\pm 0.2$ )	84.9 ( $\pm 0.4$ )	68.0 ( $\pm 0.1$ )
	InfoNCE	33.7 ( $\pm 0.1$ )	44.4 ( $\pm 0.1$ )	79.7 ( $\pm 0.4$ )	68.2 ( $\pm 0.6$ )	84.3 ( $\pm 0.7$ )	66.8 ( $\pm 0.5$ )
	PCME	33.2 ( $\pm 0.3$ )	44.0 ( $\pm 0.4$ )	79.1 ( $\pm 0.4$ )	66.8 ( $\pm 0.6$ )	83.6 ( $\pm 0.3$ )	65.3 ( $\pm 0.6$ )
	PCME++	34.5 ( $\pm 0.1$ )	45.1 ( $\pm 0.1$ )	81.5 ( $\pm 0.2$ )	69.9 ( $\pm 0.3$ )	85.4 ( $\pm 0.2$ )	68.7 ( $\pm 0.4$ )
	PCME++ + SWA	34.6 ( $\pm 0.1$ )	45.2 ( $\pm 0.1$ )	81.8 ( $\pm 0.8$ )	70.3 ( $\pm 0.1$ )	85.6 ( $\pm 0.1$ )	69.0 ( $\pm 0.1$ )
ViT-L/14 (428M)	VSE $\infty$	15.7	27.2	39.7	28.9	51.2	27.4
	InfoNCE L/14	27.8	39.6	69.0	53.9	75.9	51.9
	PCME	34.1	44.5	81.5	70.7	86.5	69.5
	PCME++	35.4	45.3	84.0	73.3	87.9	71.8

Table C.8: Text-to-image retrieval full results.

Backbone	Method	ECCV Caption			CxC	COCO	
		mAP@R	R-P	R@1	R@1	1K R@1	5K R@1
ViT-B/32 (151M)	VSE $\infty$	47.7 ( $\pm 0.2$ )	55.9 ( $\pm 0.3$ )	88.6 ( $\pm 0.9$ )	49.0 ( $\pm 2.6$ )	67.9 ( $\pm 2.2$ )	46.9 ( $\pm 2.6$ )
	InfoNCE	46.8 ( $\pm 0.5$ )	55.1 ( $\pm 0.5$ )	88.0 ( $\pm 0.8$ )	48.0 ( $\pm 0.3$ )	67.7 ( $\pm 0.2$ )	46.0 ( $\pm 0.3$ )
	PCME	47.1 ( $\pm 0.2$ )	55.5 ( $\pm 0.2$ )	88.0 ( $\pm 0.5$ )	48.0 ( $\pm 0.1$ )	67.6 ( $\pm 0.1$ )	46.1 ( $\pm 0.1$ )
	PCME++	48.0 ( $\pm 0.1$ )	56.1 ( $\pm 0.2$ )	88.8 ( $\pm 0.3$ )	49.9 ( $\pm 0.1$ )	68.9 ( $\pm 0.2$ )	47.9 ( $\pm 0.0$ )
	PCME++ + SWA	48.1 ( $\pm 0.2$ )	56.2 ( $\pm 0.3$ )	89.2 ( $\pm 0.3$ )	50.0 ( $\pm 0.1$ )	69.0 ( $\pm 0.1$ )	48.0 ( $\pm 0.1$ )
ViT-B/16 (150M)	VSE $\infty$	49.1 ( $\pm 0.3$ )	56.5 ( $\pm 0.2$ )	91.3 ( $\pm 0.4$ )	55.3 ( $\pm 0.3$ )	73.3 ( $\pm 0.3$ )	53.4 ( $\pm 0.3$ )
	InfoNCE	48.5 ( $\pm 0.2$ )	56.3 ( $\pm 0.1$ )	89.9 ( $\pm 0.2$ )	53.6 ( $\pm 0.3$ )	72.3 ( $\pm 0.1$ )	51.7 ( $\pm 0.3$ )
	PCME	48.7 ( $\pm 0.2$ )	56.5 ( $\pm 0.2$ )	89.5 ( $\pm 0.1$ )	53.1 ( $\pm 0.9$ )	72.0 ( $\pm 0.6$ )	51.2 ( $\pm 0.9$ )
	PCME++	49.7 ( $\pm 0.2$ )	57.2 ( $\pm 0.2$ )	91.4 ( $\pm 0.6$ )	55.2 ( $\pm 0.2$ )	73.4 ( $\pm 0.1$ )	53.4 ( $\pm 0.2$ )
	PCME++ + SWA	49.8 ( $\pm 0.1$ )	57.2 ( $\pm 0.2$ )	91.4 ( $\pm 0.7$ )	55.5 ( $\pm 0.2$ )	73.5 ( $\pm 0.1$ )	53.6 ( $\pm 0.2$ )
ViT-L/14 (428M)	VSE $\infty$	24.7	35.8	52.7	19.7	37.9	18.0
	InfoNCE L/14	43.4	52.1	82.1	42.1	63.1	39.9
	PCME	48.2	56.0	90.5	56.1	74.1	54.3
	PCME++	48.6	56.3	92.5	58.9	75.8	57.1

full experimental numbers for all experiments, including ablation studies, can be found in <https://naver-ai.github.io/pcmepp/>.

## References

- [1] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Adv. Neural Inform. Process. Syst.*, pages 2121–2129, 2013. 1
- [2] 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.
- [3] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.
- [4] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving visual-semantic embeddings with hard negatives. In *Brit. Mach. Vis. Conf.*, 2018. 1, 3, 5
- [5] Jiuxiang Gu, Jianfei Cai, Shafiq R Joty, Li Niu, and Gang Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7181–7189, 2018.
- [6] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Eur. Conf. Comput. Vis.*, 2018.



Table C.9: Average retrieval full results.

Backbone	Method	ECCV Caption			CxC	COCO	
		mAP@R	R-P	R@1	R@1	1K R@1	5K R@1
ViT-B/32 (151M)	VSE $\infty$	39.7 ( $\pm 0.5$ )	49.3 ( $\pm 0.3$ )	82.1 ( $\pm 2.0$ )	55.4 ( $\pm 3.3$ )	74.2 ( $\pm 2.7$ )	53.6 ( $\pm 3.4$ )
	InfoNCE	41.1 ( $\pm 0.1$ )	50.4 ( $\pm 0.1$ )	84.8 ( $\pm 0.3$ )	60.9 ( $\pm 0.4$ )	78.3 ( $\pm 0.4$ )	59.3 ( $\pm 0.3$ )
	PCME	39.1 ( $\pm 0.1$ )	48.9 ( $\pm 0.1$ )	81.4 ( $\pm 0.4$ )	54.7 ( $\pm 0.2$ )	73.8 ( $\pm 0.1$ )	53.0 ( $\pm 0.3$ )
	PCME++	40.1 ( $\pm 0.1$ )	49.8 ( $\pm 0.1$ )	83.0 ( $\pm 0.6$ )	56.6 ( $\pm 0.1$ )	75.1 ( $\pm 0.2$ )	54.8 ( $\pm 0.1$ )
	PCME++ + SWA	40.2 ( $\pm 0.1$ )	49.8 ( $\pm 0.2$ )	83.5 ( $\pm 0.4$ )	56.9 ( $\pm 0.2$ )	75.2 ( $\pm 0.1$ )	55.2 ( $\pm 0.2$ )
ViT-B/16 (150M)	VSE $\infty$	41.7 ( $\pm 0.2$ )	50.6 ( $\pm 0.2$ )	86.3 ( $\pm 0.5$ )	62.3 ( $\pm 0.1$ )	79.1 ( $\pm 0.3$ )	60.7 ( $\pm 0.1$ )
	InfoNCE	39.0 ( $\pm 0.3$ )	48.7 ( $\pm 0.2$ )	81.7 ( $\pm 1.0$ )	54.9 ( $\pm 0.2$ )	74.0 ( $\pm 0.3$ )	53.0 ( $\pm 0.1$ )
	PCME	41.0 ( $\pm 0.3$ )	50.3 ( $\pm 0.3$ )	84.3 ( $\pm 0.2$ )	59.9 ( $\pm 0.8$ )	77.8 ( $\pm 0.4$ )	58.2 ( $\pm 0.8$ )
	PCME++	42.1 ( $\pm 0.1$ )	51.1 ( $\pm 0.1$ )	86.5 ( $\pm 0.4$ )	62.6 ( $\pm 0.1$ )	79.4 ( $\pm 0.1$ )	61.1 ( $\pm 0.3$ )
	PCME++ + SWA	42.2 ( $\pm 0.0$ )	51.2 ( $\pm 0.1$ )	86.6 ( $\pm 0.5$ )	62.9 ( $\pm 0.1$ )	79.6 ( $\pm 0.1$ )	61.3 ( $\pm 0.1$ )
ViT-L/14 (428M)	VSE $\infty$	20.2	31.5	46.2	24.3	44.5	22.7
	InfoNCE L/14	35.6	45.8	75.6	48.0	69.5	45.9
	PCME	41.2	50.3	86.0	63.4	80.3	61.9
	PCME++	42.0	50.8	88.2	66.1	81.8	64.4

- [7] Yan Huang, Qi Wu, Chunfeng Song, and Liang Wang. Learning semantic concepts and order for image and sentence matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6163–6171, 2018.
- [8] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Int. Conf. Comput. Vis.*, pages 4654–4662, 2019. 14
- [9] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1979–1988, 2019.
- [10] Jonatas Wehrmann, Douglas M Souza, Mauricio A Lopes, and Rodrigo C Barros. Language-agnostic visual-semantic embeddings. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5804–5813, 2019.
- [11] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6609–6618, 2019.
- [12] Haoran Wang, Ying Zhang, Zhong Ji, Yanwei Pang, and Lin Ma. Consensus-aware visual-semantic embedding for image-text matching. In *Eur. Conf. Comput. Vis.*, 2020. 14
- [13] Haiwen Diao, Ying Zhang, Lin Ma, and Huchuan Lu. Similarity reasoning and filtration for image-text matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- [14] Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio De Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 2, 4, 6, 7, 8, 10
- [15] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15789–15798, 2021. 5, 6, 7, 12, 14
- [16] Zhenyu Huang, Guocheng Niu, Xiao Liu, Wenbiao Ding, Xinyan Xiao, hua wu, and Xi Peng. Learning with noisy correspondence for cross-modal matching. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Adv. Neural Inform. Process. Syst.*, 2021. URL <https://openreview.net/forum?id=S9ZyhWC17wJ>. 2, 6, 7, 14
- [17] Ali Furkan Biten, Andres Mafla, Lluís Gómez, and Dimosthenis Karatzas. Is an image worth five sentences? a new look into semantics for image-text matching. In *IEEE/CVF Winter Conf. App. Comput. Vis.*, pages 1391–1400, 2022.
- [18] Dongwon Kim, Namyup Kim, and Suha Kwak. Improving cross-modal retrieval with set of diverse embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23422–23431, 2023.
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pages 8748–8763. PMLR, 2021. 1, 4, 5, 6, 7, 9, 10, 14, 15, 17

- [20] 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. 1, 2, 5, 7
- [21] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.
- [22] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. pages 2556–2565, 2018.
- [23] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3558–3568, 2021.
- [24] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021. 1, 9, 15
- [25] Sanghyuk Chun, Wonjae Kim, Song Park, Minsuk Chang Chang, and Seong Joon Oh. Eccv caption: Correcting false negatives by collecting machine-and-human-verified image-caption associations for ms-coco. In *Eur. Conf. Comput. Vis.*, 2022. 1, 2, 3, 4, 6, 7, 13, 14
- [26] Seong Joon Oh, Kevin Murphy, Jiyan Pan, Joseph Roth, Florian Schroff, and Andrew Gallagher. Modeling uncertainty with hedged instance embedding. In *Int. Conf. Learn. Represent.*, 2019. 1, 2, 4, 9, 10
- [27] Dat Quoc Nguyen, Ashutosh Modi, Stefan Thater, Manfred Pinkal, et al. A mixture model for learning multi-sense word embeddings. In *Proc. of the 6th Joint Conference on Lexical and Computational Semantics (\*SEM 2017)*, pages 121–127, 2017. 1
- [28] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6902–6911, 2019. 1
- [29] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5710–5719, 2020. 1
- [30] Jennifer J Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. View-invariant probabilistic embedding for human pose. In *Eur. Conf. Comput. Vis.*, 2020. 2
- [31] Anna Silnova, Niko Brummer, Johan Rohdin, Themis Stafylakis, and Lukas Burget. Probabilistic embeddings for speaker diarization. In *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 24–31, 2020. 2
- [32] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Probabilistic representations for video contrastive learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14711–14721, 2022. 2
- [33] Andrei Neculai, Yanbei Chen, and Zeynep Akata. Probabilistic compositional embeddings for multimodal image retrieval. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4547–4557, 2022. 2
- [34] Frederik Warburg, Marco Miani, Silas Brack, and Soren Hauberg. Bayesian metric learning for uncertainty quantification in image retrieval. *arXiv preprint arXiv:2302.01332*, 2023. 10
- [35] Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. In *International Conference on Machine Learning*, 2023. 1, 10
- [36] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 2, 9, 15
- [37] Zarana Parekh, Jason Baldridge, Daniel Cer, Austin Waters, and Yinfei Yang. Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for ms-coco. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2021. 2, 3, 6, 7, 13
- [38] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Int. Conf. Learn. Represent.*, 2017. URL <https://openreview.net/forum?id=HyxQzBceg>. 4, 8
- [39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Int. Conf. Learn. Represent.*, 2018. 4, 5

- [40] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Int. Conf. Comput. Vis.*, 2019. 4, 5
- [41] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *Int. Conf. Learn. Represent.*, 2021. 4
- [42] Chanwoo Park, Sangdoo Yun, and Sanghyuk Chun. A unified analysis of mixed sample data augmentation: A loss function perspective. In *Neural Information Processing Systems (NeurIPS)*, 2022. 4
- [43] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, and James Zou. When and how mixup improves calibration. *arXiv preprint arXiv:2102.06289*, 2021. 5
- [44] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>. 5
- [45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, pages 5998–6008, 2017. 5
- [46] 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 *Eur. Conf. Comput. Vis.*, 2014. 5
- [47] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3128–3137, 2015. 6
- [48] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Eur. Conf. Comput. Vis.*, 2020. 6, 14
- [49] Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. Adamp: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *Int. Conf. Learn. Represent.*, 2021. 6
- [50] Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *Conference on Uncertainty in Artificial Intelligence*, 2018. 6, 8
- [51] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *Adv. Neural Inform. Process. Syst.*, 2021. 6
- [52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 9, 15
- [53] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. Nsm1: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 10
- [54] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. 11
- [55] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *Int. Conf. Mach. Learn.*, 2021. 14
- [56] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 14
- [57] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. 14
- [58] 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)*, pages 181–196, 2018. 14