Abstract

We tackle the cross-modal retrieval problem, where learning is only supervised by relevant multi-modal pairs in the data. Although the contrastive learning is the most popular approach for this task, it makes potentially wrong assumption that the instances in different pairs are automatically irrelevant. To address the issue, we propose a novel loss function that is based on self-labeling of the unknown semantic classes. Specifically, we aim to predict class labels of the data instances in each modality, and assign those labels to the corresponding instances in the other modality (i.e., swapping the pseudo labels). With these swapped labels, we learn the data embedding for each modality using the supervised cross-entropy loss. This way, cross-modal instances from different pairs that are semantically related can be aligned to each other by the class predictor. We tested our approach on several real-world cross-modal retrieval problems, including text-based video retrieval, sketch-based image retrieval, and image-text retrieval. For all these tasks our method achieves significant performance improvement over the contrastive learning.

1 Introduction

Cross-modal retrieval, the task of retrieving the most relevant items in the database of one modality (e.g., images) for a given query from another modality (e.g., texts), has received unprecedented attention in computer vision and related areas. It is to learn the underlying relevance or similarity metric between data instances that live in heterogeneous modalities with highly different distributions. Although there are several different learning problem formulations in the literature, in this paper we mainly focus on the paired training data setup, in which training is only supervised by relevant pairs in the training data, and there are no semantic class labels annotated. That is, the training data consist of only pairs of relevant multi-modal data instances, e.g., (image, text), which may require minimal human annotation effort (e.g., web scraping of images and nearby texts).

The contrastive (or triplet loss) learning [Chopra et al., 2005; Hadsell et al., 2006] is recognised as the most popular and successful approach, which aims to learn the cross-modal similarity measure by the intuitive criteria that pull together relevant pairs and push away irrelevant ones. However, it makes potentially wrong assumption that instances in different pairs are automatically irrelevant. The pairs in the training data are usually collected by considering relevant pairs only (e.g., nearby images and texts in a web page), and the relevance of instances in different pairs is usually not checked. However, this is implicitly assumed in the contrastive loss. The issue was also raised in recent work [Kim et al., 2019; Zhou et al., 2020; Patrick et al., 2020; Wray et al., 2021; Chen et al., 2021]. In this paper we propose a novel learning algorithm that addresses the issue via self-labeled clustering approach.

Motivated from the recent clustering-based representation learning in the self-supervised learning literature [Asano et al., 2020; Caron et al., 2020], we propose a novel loss function for cross-modal retrieval that is based on self-labeling of the unknown classes. Specifically, we introduce (latent) semantic class labels to be assigned to data instances, where class labels supposedly decide the relevance of cross-modal data instances (i.e., the same class label means relevant items, and vice versa). We predict class labels of the data instances in each modality, and assign the predicted labels to the corresponding instances in the other modality (i.e., swapping the pseudo labels). With these swapped pseudo labels, we learn the data embedding for each modality using the supervised cross-entropy loss. This way, cross-modal instances from different pairs that
are semantically related can be aligned to each other by the class predictor. The whole process of label prediction and supervised learning with swapped classes is alternated to learn the optimal feature extraction networks. We call this approach Swapped Assignment of Multi-modal Pairs (SwAMP).

The main benefits of the SwAMP are in two folds: i) Unlike the contrastive loss, SwAMP does not make potentially wrong assumption that instances from different pairs are automatically irrelevant. The optimized class assignment finds similar instances from other pairs, and the feature extractor is trained in such a way that the same-class instances, even in different pairs, are well aligned. This feature of aligning instances in different pairs is hardly exploited in the contrastive loss. ii) Since the learning does not fully resort to pair-based losses as in contrastive learning, the sampling complexity can be reduced. This comes from the class-based loss adopted in the SwAMP, where similar ideas were exploited previously in self-supervised representation learning (Caron et al., 2018; Asano et al., 2020; Caron et al., 2020). Our approach is generically applicable to different types of cross-modal retrieval problems. We empirically demonstrate that the SwAMP loss improves retrieval performance significantly over the contrastive learning, on various real-world cross-modal retrieval problems, including text-video, sketch-image, and image-text retrieval.

2 Problem Setup & Background

Let $x^A$ and $x^B$ denote data instances from modality A and modality B, respectively. For instance, $x^A$ is an image from the image modality, while $x^B$ is a text/caption from the text modality. Throughout the paper we deal with modality-wise feature representation, meaning that we have modality-wise feature extractors (neural networks) $\phi^A(\cdot)$ and $\phi^B(\cdot)$ applied to $x^A$ and $x^B$, respectively. Also known as dual encoders, it produces a succinct vector representation for each modality, $\phi^A(x^A) \in \mathbb{R}^d$ and $\phi^B(x^B) \in \mathbb{R}^d$. The shared feature space ($\subset \mathbb{R}^d$) allows us to define the similarity score $s(x^A, x^B)$ as a cosine angle between $\phi^A(x^A)$ and $\phi^B(x^B)$. The goal is to learn the feature extractors so that the relevant pairs $x^A$ and $x^B$ have a high similarity score $s(x^A, x^B)$, while irrelevant pairs have a low similarity score. The main benefit of the modality-wise feature representation is the computational efficiency, scalable to billions of instances at training/test time, thanks to the efficient dot-product. There is an alternative approach that directly computes the similarity score without having modality-wise representation. A typical example is the cross-modal attention models (Lee et al., 2018; Lu et al., 2019; Desai and Johnson, 2020; Huang et al., 2020) (details in Sec. 3). Although they can capture interactions between cross-modal local features, they are computationally demanding, not scalable to large-scale data.

The training data are composed of relevant pairs $D = \{(x^A_i, x^B_i)\}_{i=1}^N$, where $x^A_i$ and $x^B_i$ are the instances in the $i$-th relevant pair. At test time, a query is given from the target modality, say $x^A$, and the goal is to find the most relevant instance, say $x^B$, from the other modality, where the search is performed on the given test set $\{x^B_i\}_{i=N+1}^{N+M}$.

2.1 Contrastive Learning

In contrastive learning (Chopra et al., 2005; Hadsell et al., 2006), it is implicitly assumed that data instances from different pairs are irrelevant, although it may not be true. The loss function is defined to capture the intuition: penalize low (high) similarity scores for relevant (irrelevant, resp.) pairs. By introducing the margin $\alpha$ (e.g., 0.2) and considering the most violating irrelevant pairs (i.e., hard negatives), the loss can be written as (script c stands for contrastive):

$$L_c(\phi^A, \phi^B) = \sum_{i \in \mathcal{D}} \left( s(x^A_i, x^B_i) - \max_{j \in \mathcal{D}\setminus i} s(x^A_i, x^B_j) \right) \geq \alpha$$

$$+ \left( s(x^A_i, x^B_i) - \max_{j \in \mathcal{D}\setminus i} s(x^A_j, x^B_i) \right) \geq \alpha$$ (1)

where $(z) \geq \alpha = \max(0, \alpha - z)$ only incurs positive loss when $z < \alpha$. A main issue of the contrastive learning is that we cannot guarantee that data instances from different pairs in the training data are irrelevant, because the data are usually collected by considering relevant pairs only (e.g., web scraping of images and nearby texts), and the relevance of instances in different pairs is usually not checked. However, this is assumed in the contrastive loss.

3 Our Approach: SwAMP

Our idea is to introduce (latent) semantic class labels for data instances and use them to learn the feature extractors. The class labels supposedly decide the relevance of data instances from different modalities, that is, $x^A$ and $x^B$ are considered relevant if their class labels are the same, and vice versa. Obviously, the paired cross-modal instances in the training data must have the same class labels. But beyond this, instances from different pairs can also be deemed relevant if they belong to the same semantic class labels. The motivation is that if we estimate the class labels accurately, the feature extractor learning can be turned into a supervised classification problem.

More formally, we consider (unknown) class labels to be assigned to the data instances. Let $y^A, y^B \in \{1, \ldots, K\}$ be the class labels for $x^A$ and $x^B$, respectively, where $K$ is chosen by the user. The relevance of $x^A$ and $x^B$ is determined by their class labels: $x^A$ and $x^B$ are deemed relevant if $y^A = y^B$ and irrelevant if $y^A \neq y^B$. If we knew the class labels that bear such semantics in the training data, then training becomes supervised learning that can be done for each modality, which allows us to avoid pairwise terms in the loss function. However, we don’t have class labels,
and we optimize them (i.e., self-supervised learning) together with the feature extractors \( \phi^A(\cdot) \) and \( \phi^B(\cdot) \). To this end, we build linear classifiers \( p(y|x^A) \) and \( p(y|x^B) \) on the extracted features. For each modality \( M \in \{A, B\} \),

\[
p(y = j|M) = \frac{\exp(p_j^M(x)/\tau)}{\sum_i \exp(p_i^M(x)/\tau)},
\]

(2)

where \( P = \{p_1, \ldots, p_K\} \) are trainable parameters that are shared between two modalities, and \( \tau \) is the temperature in the softmax. We can regard each \( p_j \) as the prototype vector for class \( j \) that lies in the shared feature space. Since we have classification models, the (supervised) cross-entropy loss minimization is a natural choice to optimize them. That is, letting \( p_{\text{true}}(y|x^A) \) be the true conditional class distribution for modality \( A \), we minimize \( \mathbb{E}_{p_{\text{true}}(y|x^A)}[-\log p(y|x^A)] \) with respect to \( P \) and the network parameters of \( \phi^A(\cdot) \) (similarly for modality \( B \)). Since we cannot access \( p_{\text{true}}(y|x^A) \), one may be tempted to use the model \( p(y|x^A) \) in (2) instead. However, it can easily lead to a degenerate solution such as the one that puts all the probability mass on a particular single class all the time (thus attaining the optimal cross-entropy loss 0). Moreover, this would make learning \( \phi^A(\cdot) \) and \( \phi^B(\cdot) \) nearly independent and less interacted with each other, merely through the shared prototypes \( P \).

Instead, we form an optimization problem to estimate a surrogate of \( p_{\text{true}}(y|x^A) \), which we denote by \( q(y|x^A) \), using the information from the other modality \( B \), while imposing additional constraints to avoid the degenerate solutions. More specifically, we optimize the surrogate \( q(y|x^A) \) with the following two criteria. First, \( q(y|x^A) \) needs to be well aligned with the current estimate \( p(y|x^B) \) for \( x^B \) that is paired with \( x^A \). This is due to the aforementioned requirements for the class labels, where the class labels (more generally, their distributions) of the paired instances should match. Secondly, the marginal distribution \( q(y) = \mathbb{E}_{x^A \sim D} q(y|x^A) \) is constrained to be a uniform distribution. This constraint naturally arises from the symmetry of class labels, a reasonable assumption about the true class distribution, and successfully leaves out the degenerate solutions discussed above. To summarize, the following is the optimization problem for \( q(y|x^A) \), where \( Q^A \) is the \((N \times K)\) matrix with \( Q^A_{ij} := q(y|x^A) \). Recall that \( D = \{ (x^A_i, x^B_i) \}_{i=1}^N \) is the training data of paired instances.

\[
\min_{Q^A} \mathbb{E}_{i \sim D} \left[ -\log p(y|x^B_i) \right] \quad \text{s.t.} \quad \mathbb{E}_{i \sim D} [q(y|x^A_i)] = 1/K, \forall y.
\]

(3)

We perform similar optimization for \( q(y|x^B) \) \((Q^B_{ij} := q(y|x^B))\) to approximate \( p_{\text{true}}(y|x^B) \) by exchanging the roles of \( A \) and \( B \). The optimal solutions (surrogates) are denoted by \( q^A \) and \( q^B \), where we use the superscript to distinguish the two modalities. Note that during the optimization of (3) for \( q^A \) and \( q^B \), we fix the model parameters, that is, \( P \) and the feature extractor networks. The overall optimization is alternation between: i) surrogate optimization (3) with \( P, \phi^A, \phi^B \) fixed, and ii) supervised (cross-entropy) loss minimization with \( q^A \) and \( q^B \) fixed, where the latter can be written as (subscript \( s \) stands for SwAMP):

\[
\min_{P, \phi^A, \phi^B} \mathcal{L}_s := \mathbb{E}_{i \sim D} \left[ -\log p(y|x^A_i) \right] + \mathbb{E}_{i \sim D} \left[ -\log p(y|x^B_i) \right]
\]

(4)

Now we discuss how to optimize (3). It is essentially the optimal transport (OT) problem \((\text{Villani, 2008;} \text{Cuturi, 2013})\), specifically with the cost matrix \( C_{iy} = -\log p(y|x^B_i) \) and the marginal constraints \( \sum_i Q^A_{iy} = 1/K, \forall y \) (and implicitly \( \sum_y Q^B_{ij} = 1/N, \forall i \in D \)). Although the OT is known to be an instance of the linear program (LP), conventional LP solvers are not suitable for large-scale problems. As is common practice, we relax the problem by augmenting the LP with the entropic regularizer for \( q(y|x^A) \), namely \( \frac{1}{\eta} \sum_i \log Q^A_{iy} \) added to the loss (thus, penalizing small entropy), which can be solved by the efficient Sinkhorn-Knopp (SK) algorithm \((\text{Cuturi, 2013})\). Here \( \eta \) is the regularization trade-off hyperparameter. The SK algorithm finds the optimal solution as \( Q^A = \text{Diag}(u) \cdot A \text{Diag}(v) \), where \( A_{ij} = e^{-\eta C_{ij}} \) and the vectors \( u \in \mathbb{R}^N \) and \( v \in \mathbb{R}^K \) are the fixed points of \( u_i = \frac{1}{N} / (Av)_i \) and \( v_j = \frac{1}{K} / (A^T u)_j \) for \( i = 1, \ldots, N, j = 1, \ldots, K \). The fixed point iteration usually converges quickly after a few iterations. We denote the algorithm as:

\[
Q \leftarrow \text{SK(cost = C, reg = } \eta)\)
\]

(5)

One challenge in optimizing (3) with the SK, however, is that it involves the entire dataset \( D \) in the loss, which means that the model update (4) has to be deferred until \( q \) is optimized for an entire data epoch. Simply replacing \( D \) with a minibatch might be dangerous since the population class marginal distributions are poorly covered by a minibatch. We need an even larger subset of \( D \) to roughly meet the (uniform) class constraint. To this end, we adopt the (FIFO) queues, where we accumulate the embeddings \( \phi^A(x^A) \) and \( \phi^B(x^B) \) from the latest minibatches into the queues. The optimization (3) is then performed on the queue data (\( D \) replaced by the data in the queues). To have the uniform class constraint meaningful, we choose the queue size to be greater than \( K \). Note that (3) is solved by the SK algorithm, and thus no backprop is required, hence enlarging the queue size does not incur computational issue. Similar ideas were used in the self-supervised representation learning literature, e.g., \((\text{He et al., 2019}; \text{Caron et al., 2020})\). To have the queues filled with the latest features, we insert the features
Algorithm 1 SwAMP Training.

Input: Class cardinality $K$, queue size, temp. $\tau$, $\eta$ in SK.

Initialize: $P = \{p_i\}_{i=1}^K$, $\phi^A$, $\phi^B$. Empty queue $Q$.

Output: Trained model $\{P, \phi^A(\cdot), \phi^B(\cdot)\}$.

Repeat until convergence:
1. Sample a minibatch of paired data $B = \{(x_i^A, x_i^B)\}$.
2. Evaluate $\phi^A(x_i^A)$ and $\phi^B(x_i^B)$ for $i \in B$ (forward pass).
3. Insert $\{(\phi^A(x_i^A), \phi^B(x_i^B))\}_{i \in B}$ into the queue $Q$.
4. Solve (5) for modality $A$ and $B$:
   \[
   \begin{align*}
   &\{q^A(y|i)\}_{i \in Q} \leftarrow \text{SK}\left(-\log p(y|x_i^B)\right)_{i \in Q}, \text{reg} = \eta \\
   &\{q^B(y|i)\}_{i \in Q} \leftarrow \text{SK}\left(-\log p(y|x_i^A)\right)_{i \in Q}, \text{reg} = \eta
   \end{align*}
   \]
5. Take the minibatch portions $\{q^A(y|i), q^B(y|i)\}_{i \in B}$: Do SGD update with $L$ in (6).

where $\lambda$ is the trade-off hyperparameter.

As we estimate the surrogate $q^A$ using the current classification model in modality $B$, and vice versa, the class assignment is swapped. The pseudo code of our algorithm is shown in Alg. 1. The idea of optimizing class labels in the representation learning was previously introduced in (Asano et al., 2020) Caron et al. (2020), however, they aimed for self-supervised representation learning as an instance discrimination pretext task with augmented data. In this paper, we deal with the cross-modal retrieval problem, where we estimate the class labels of instances in one modality using the features from the other modality. Unlike the contrastive loss, SwAMP does not make any assumption that instances from different pairs are automatically irrelevant. The OT class assignment finds similar instances from other pairs, and the feature extractor is trained in such a way that the same-class instances, even in different pairs, are well aligned. This feature of aligning instances in different pairs is hardly exploited in the contrastive loss.

4 Related Work

Cross-modal retrieval. It is beyond the scope of the paper to enumerate all previous works on cross-modal retrieval, and we refer the readers to recent survey papers such as (Wang et al., 2020). Recently, the most interesting cross-modal tasks involve, among others, video-text (Liu et al., 2019) Gabeur et al. (2020) Patrick et al. (2020) Miech et al. (2021) (2019, 2020) Wang et al. (2021) Chen et al. (2021), image-text (Chen et al., 2018) Faghri et al. (2018) Lee et al. (2018) Chun et al. (2021) Li et al. (2019) Zhang et al. (2020), and sketch-photo (Dey et al., 2019) Sain et al. (2021). For the training data of relevant pairs, most approaches commonly rely on the idea of contrastive learning (Chopra et al., 2005; Hadsell et al., 2006). Beyond the intuitive triplet forms (Wang et al., 2014) Schroff et al. (2015), more sophisticated losses were introduced in (Sohn et al., 2016) Song et al. (2016) Wang et al. (2019a, b) to deal with a positive and multiple negative pairs as well as hard examples. To reduce the super-linear time computational overhead, several sophisticated sampling strategies were proposed (Wu et al., 2017) Harwood et al. (2017) Yuan et al. (2017). As discussed in Sec. 2 there are broadly two different ways to define the similarity metric between instances of different modalities: modality-wise feature representation and cross-modal attention. The main benefit of the former is the computational efficiency, scalable to billions of instances at training/test time, thanks to the efficient dot-product. The latter directly computes the similarity score without having modality-wise representation (Lee et al., 2018) Lu et al. (2019) Desai and Johnson (2020) Huang et al. (2020) using the transformer-like attentive neural networks which aim to capture interactions between local features in the instances from different modalities. Although they can capture cross-modal interactions between local features of data instances from different modalities, they are computationally demanding and very slow due to the quadratic complexity in the number of local features. In (Miech et al., 2021), a hybrid of the two is introduced, which retains the two models, but performs re-ranking/distillation at test time for speed-up.

Clustering-based approaches. There were previous attempts to cluster (group) data instances, or equivalently self-labeling, to improve saliency in representation learning. Some approaches perform offline K-means clustering for every epoch (Caron et al., 2018) Alwassel et al. (2020), which can make training slow. The idea of optimizing class labels in the representation learning was previously introduced in (Asano et al., 2020) Caron et al. (2020). However, all these previous approaches aimed for self-supervised representation learning as an instance discrimination pretext task with augmented data. On the other hand, we perform simultaneous learning of class labels and the feature extraction networks for the cross-modal retrieval setting. More recently (Chen et al., 2021) proposed a clustering-based cross-modal retrieval method based on the semantic similarity. However, our approach is mainly different from it in that we adopt the OT-based class label assignment forming a joint feature-label optimization, instead of simple fusion of multi-modal features for clustering as in (Chen et al., 2021).

5 Experimental Results

We test the proposed SwAMP loss on several different types of real-world cross-modal retrieval problems. For each problem/dataset, we choose the most popular and successful method in the literature, and replace its loss function (mostly contrastive loss) with the proposed SwAMP loss to demonstrate the performance improvement. To this end, for fair comparison, we faithfully follow the same optimization
We first consider the text-to-video retrieval task where the works, as well as the training/test protocols, we follow the retrieval performance metrics are recall-at-k (R@$k$) with $k = 1, 5, 10$ and the median rank (Med-R). Hence, the random guess attains R@1 = 0.03% Med-R=1, 675. The results are summarized in Table 2. In the bottom four rows, we see the performance improvement achieved by the proposed SwAMP against the contrastive loss \cite{Miech2019}. For both training strategies, No PT (random model initialization) and PT (initialized with the HowTo100M-pretrained model), our SwAMP improves the retrieval performance significantly (e.g., about 12% reduction in Median Rank for the No PT case). SwAMP also outperform the CCA baseline FV-CCA \cite{Klein2015}.

**MSRVTT.** This dataset \cite{Xu2016} collected from YouTube contains videos of specific categories including music, sports, and movie. There are 200K video-caption pairs obtained by human annotation. We follow the retrieval training/test protocol of \cite{Yu2018,Miech2019}. The test set consists of 1K pairs. As reported in Table 2, our SwAMP loss improves the performance over the contrastive learning significantly for both no-pretraining and pretraining cases: about 24% in R@1 in the No PT case, and 27% in the PT case. Furthermore, the SwAMP outperforms with large margin the state-of-the-arts: C+LSTM+SA+FC7 \cite{Torabi2016}, VSE-LSTM \cite{Kiros2014}, Temporal Tessellation \cite{Kauman2017}, CT-SAN \cite{Yu2017}, and JSFusion \cite{Yu2018}.

**LSMDC.** This dataset of movie video clips is comprised of 101K video-caption pairs. The captions are collected either from the movie scripts or the audio descriptions. The test set contains 1K pairs. For this dataset, $\lambda = 0.1$ (impact of the SwAMP loss against contrastive) for the PT case. The results are shown in Table 2. Similar to the other two datasets, our SwAMP is consistently better than the contrastive learning (about 7 ~ 9% in Median Rank).

### 5.1 Text-based Video Retrieval

We first consider the text-to-video retrieval task where the goal is to find the most relevant video clip for a given natural language text query. We consider three datasets for this task: i) **YouCook2** \cite{Zhou2018} of cooking videos and instructions, ii) **MSR-VTT** \cite{Xu2016} of generic videos and captions from YouTube, and iii) **LSMDC** \cite{Rohrbach2017} of movie clips and subtitles. All these datasets provide pairs of video clip and text description, forming a multi-modal paired data format \textit{(text, video)} which conforms to our SwAMP framework.

For the raw text/video features and the feature extractor networks, as well as the training/test protocols, we follow the methods in \cite{Miech2019}, and the details are described in Appendix (Sec. C). Following \cite{Miech2019}, there are two training strategies: i) No-pretraining (No-PT) where the feature extraction networks are randomly initialized, and the training is done on the training split of the dataset, and ii) Pretraining (PT) where the feature extractors are first pretrained on the large-scale HowTo100M dataset \cite{Miech2019}, and finetuned on the target dataset. In \cite{Miech2019}, they adopt the contrastive (triplet) loss for training the feature extractors. Although we also compare our approach with the state-of-the-arts, the main focus in this experiment is to demonstrate the performance improvement achieved by the proposed SwAMP loss against vanilla contrastive learning. The SwAMP hyperparameter $\lambda$, the weight/impact of the SwAMP loss against the contrastive loss in (5) is chosen as $\lambda = 0.25$ for all three datasets, except the LSMDC-PT case for which $\lambda = 0.1$. We also choose temperature in softmax $\tau = 0.25$, entropic regularization trade-off in SK $\eta = 5.0$, the number of classes $K = 500$, and the queue size 2,048 for the SwAMP. The other learning hyperparameters common in SwAMP and contrastive losses are not changed from \cite{Miech2019}, and summarized in Appendix (Sec. C).

**YouCook2.** This cooking video dataset collected from YouTube, contains 89 recipes and 14K video clips annotated with textual descriptions from paid human workers. The test data are formed by taking 3.5K clips from the validation set, and the test set comprises of 3,350 pairs. The retrieval performance metrics are recall-at-k (R@k) with

| Methods               | R@1 ↑ | R@5 ↑ | R@10 ↑ | Med-R ↓ |
|-----------------------|-------|-------|--------|---------|
| Random                | 0.03  | 0.15  | 0.3     | 1675    |
| FV-CCA                | 4.6   | 14.3  | 21.6    | 75      |
| Contrastive (No PT)   | 4.2   | 13.7  | 21.5    | 65      |
| SwAMP (No PT)         | 4.8   | 14.5  | 22.5    | 57      |
| Contrastive (PT)      | 8.2   | 24.5  | 35.3    | 24      |
| SwAMP (PT)            | 9.4   | 24.9  | 35.3    | 22      |
tion consists of three losses: **Triplet loss** is the conventional triplet loss, **Domain loss** uses an adversarial domain classifier to penalize misalignment between embedding distributions of photo images and sketches, and **Semantic loss** urges the embeddings of the photo images and sketches to reconstruct the pretrained word embedding of the corresponding object word. We also use the same attention-based embedding networks for photo and sketch modalities. Then, we add our SwAMP loss to the Doodle2Search’s loss with the impact $\lambda = 0.1$ for all three datasets. We use the queue size $K = 500$, softmax temperature $\tau = 0.25$, entropic regularization impact $\eta = 5.0$. The retrieval performances on the three datasets are summarized in Table 3. The performance metrics are mean average precision (mAP), mAP@200, and the precision-at-200 (P@200). As shown, our SwAMP loss when added to the existing contrastive-based loss, significantly improves the retrieval performance (about 9% in mAP for Sketchy and about 60% for TU-Berlin).

### 5.3 Image-Text Retrieval

For the image-text cross-modal retrieval task, we follow the features and protocols from the well-known stacked cross attention network (SCAN) (Lee et al., 2018). In their framework, each image is represented by a set of local features $V = \{v_1, \ldots, v_k\}$, where $v_i (\in \mathbb{R}^D) = W_v f_i + b_v$ and $f_i$’s are the CNN features extracted from salient image regions detected by the Faster-R-CNN model (Ren et al., 2015). The raw features $f_i$’s are fixed and $\{W_v, b_v\}$ are learnable parameters. The text (sentence) is also treated as a set of word features $E = \{e_1, \ldots, e_n\}$, where $e_i (\in \mathbb{R}^D) = (h_i^r + h_i^l)/2$ and $h_i^{r/l}$ are the outputs of the bi-directional GRU (Bahdanau et al., 2015; Schuster and Paliwal, 1997) with the sequence of word embeddings as input. Both the word embeddings and GRU parameters are learnable. These image/text features contain rich local information, however, one challenge is that both representations are sets, hence the number of elements ($k$ and $n$) can vary from instance to instance.

In (Lee et al., 2018), they proposed a cross-modal attention model, where each local feature from one modality is transformed by the attention (Vaswani et al., 2017) with the set of local features in the other modality; e.g., $v_i$ is transformed to $\text{attn}(v_i; \{e_j\}_{j=1}^n) = \text{the weighted sum of values } \{e_j\}_{j=1}^n$, with $v_i$ as a query and $\{e_j\}_{j=1}^n$ as keys (this denoted by i-t, while the other attention direction t-i can be used alternatively). Then the similarity score between image $V$ and text $E$ is defined as $\text{pool}(\{\cos(v_i, \text{attn}(v_i; \{e_j\}_{j=1}^n))\}_{i=1}^K)$, where $\cos(a, b)$ is the cosine similarity and $\text{pool}$ is the pooling operation, either of $AVG$ (average) or $LSE$ (log-sum-exp). Then the triplet contrastive loss of (10) is employed. Although the cross-attention is useful for capturing interaction between local features, computing the similarity score takes quadratic time in the number of local features in the instances. This is time consuming compared to the simple dot-product of the modality-wise embedding vectors (See Table 12 for wall-clock times).

To have modality-wise succinct representation instead (for SwAMP), we adopt the induced-set attention idea from Set-Transformer (Lee et al., 2019). Specifically, we introduce $p$ learnable prototype (query) vectors $\{q_j\}_{j=1}^p$, $q_j \in \mathbb{R}^D$. Then we compute the attention for each query with $V$ (or $E$), i.e., $z_j = \text{attn}(q_j; \{v_i\}_{i=1}^k)$. We define $\phi_{image}(V) = \text{concat}(z_1, \ldots, z_p)$, similarly for $\phi_{text}(E)$.
When compared with the computationally expensive SCAN, SwAMP mostly outperforms SCAN except for the SCAN’s best attention direction/combination choices. To see the computational advantage of SwAMP-PAR, we compare the actual training/test times for the two approaches in Table 12 measured on the same machine with a single GPU (RTX 2080 Ti) and Core i7 3.50GHz CPU. Our SwAMP-PAR is about 4 times faster than SCAN for training on both datasets, while the difference becomes even more pronounced during test; SwAMP-PAR is about two orders of magnitude faster than the cross-modal attention model.

Table 4: Image-text retrieval results on MS-COCO.

| Methods | Image → Text | Text → Image |
|---------|--------------|--------------|
|         | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 |
| DAN [Nam et al., 2017] | 55.0 | 81.8 | 89.0 | 59.1 | 69.2 | 79.1 | 39.4 | 69.2 | 79.1 |
| DPC [Zhang et al., 2017] | 55.6 | 81.9 | 89.5 | 59.1 | 69.2 | 80.9 | 39.4 | 69.2 | 80.9 |
| VSE++ [Faghri et al., 2018] | 52.9 | - | 87.2 | 39.6 | - | 79.5 | 41.1 | - | 80.1 |
| SCO [Zhang et al., 2018] | 55.5 | 82.0 | 89.3 | 41.1 | 70.5 | 80.1 | 39.4 | 70.5 | 80.1 |
| SCAN t-i AVG | 67.9 | 89.0 | 94.4 | 43.9 | 74.2 | 82.8 | 41.1 | 74.2 | 82.8 |
| SCAN t-i AVG | 61.8 | 87.5 | 93.7 | 45.8 | 74.4 | 83.0 | 41.1 | 74.4 | 83.0 |
| SCAN t-i AVG + i-t LSE | 67.4 | 90.3 | 95.8 | 48.6 | 77.7 | 85.2 | 41.1 | 77.7 | 85.2 |
| Contrastive-PAR | 65.7 | 86.8 | 92.4 | 48.2 | 75.8 | 84.2 | 41.1 | 75.8 | 84.2 |
| SwAMP-PAR | 67.8 | 88.5 | 94.0 | 49.1 | 76.1 | 83.7 | 41.1 | 76.1 | 83.7 |

Table 5: Image-text retrieval results on Flickr30K.

| Methods | 5-fold (1K test images) | Entire (5K test images) |
|---------|-------------------------|-------------------------|
|         | Image → Text | Text → Image | Image → Text | Text → Image |
|         | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 |
| DAN [Nam et al., 2017] | 65.5 | 89.8 | 95.5 | 47.1 | 79.9 | 90.0 | 41.2 | 70.5 | 81.1 | 25.3 | 53.4 | 66.4 |
| DPC [Zhang et al., 2017] | 64.6 | - | 95.7 | 52.0 | - | 92.0 | 41.3 | - | 81.2 | 30.3 | - | 72.4 |
| VSE++ [Faghri et al., 2018] | 68.5 | - | 97.9 | 56.6 | - | 94.5 | 42.0 | - | 84.7 | 31.7 | - | 74.6 |
| SCO [Zhang et al., 2018] | 69.9 | 92.9 | 97.5 | 56.7 | 87.5 | 94.8 | 42.8 | 72.3 | 83.0 | 33.1 | 62.9 | 75.5 |
| PCME [Chun et al., 2021] | 68.8 | - | 94.6 | 54.6 | - | 94.2 | 44.2 | - | 81.9 | 31.9 | - | 74.6 |
| SCAN t-i | 69.2 | 93.2 | 97.5 | 54.4 | 86.0 | 93.6 | 46.4 | 77.4 | 87.2 | 34.4 | 63.7 | 75.7 |
| SCAN t-i + i-t | 72.7 | 94.8 | 98.4 | 58.8 | 88.4 | 94.8 | 50.4 | 82.2 | 90.0 | 38.6 | 69.3 | 80.4 |
| Contrastive-PAR | 71.8 | 94.3 | 97.9 | 56.8 | 86.9 | 93.8 | 48.4 | 78.1 | 88.1 | 34.3 | 64.4 | 76.2 |
| SwAMP-PAR | 72.6 | 94.6 | 98.0 | 57.4 | 87.6 | 94.1 | 49.7 | 79.1 | 88.3 | 35.0 | 65.1 | 76.6 |

Table 6: Running times (seconds) measured on (Core i7 3.50GHz CPU / 128GB RAM / 1 RTX-2080Ti GPU). Per-batch times for training, entire times for test. For MS-COCO, test times for 5K test images (1K test in parentheses).

| Methods | Flickr3K | MS-COCO |
|---------|----------|----------|
|         | Train | Test | Train | Test |
| SCAN t-i AVG | 0.35 | 336.9 | 0.33 | 9352.0 (350.3) |
| SwAMP-PAR | 0.09 | 3.8 | 0.08 | 25.9 (16.3) |

Figure 1: Impact of $K$ (the number of classes) and $\lambda$. 

5.4 Ablation Study

We perform empirical study on the impact of two important hyperparameters in our model: the number of classes $K$ and SwAMP loss trade-off $\lambda$.

**Number of classes ($K$).** Recall that the best $K$ values we chose were: $K = 1000$ for the image-text retrieval datasets.
and $K = 500$ for text-based video retrieval. To see how the retrieval performance is affected by other choices of $K$, we conduct experiments by varying $K$ around the optimal values. The results on MS-COCO ($I \rightarrow T$) and YouCook2 tasks are shown in Fig. 16 (Left). (More results on other datasets can be found in Appendix (Fig. 4–8, Sec. B.).) Clearly, very small $K$ has low retrieval performance (R@1), and increasing $K$ leads to improvement. However, beyond certain points, there is no benefit of increasing $K$ and we even see performance degradation, which agrees with the observations from previous work (Asano et al., 2020; Caron et al., 2020). This is perhaps due to the difficulty of assigning meaningful cluster labels in optimal transport. Overall, with properly chosen $K$, SwAMP outperforms contrastive learning, signifying that SwAMP’s grouping/clustering of similar instances is more effective than vanilla instance discrimination. The fact that the optimal $K$ values are different in two tasks (image-text and video-text) implies that the best cardinality of semantic clusters is highly dependent on the dataset characteristics (e.g., size and semantic diversity).

**SwAMP impact ($\lambda$).** The sensitivity to $\lambda$ is shown in Fig. 16 (Right), and more results and further discussions are in Appendix (Fig. 9–13, Sec. B).

### 5.5 Visualization of Learned Clusters

As qualitative analysis, we visualize the learned clusters to see if they capture meaningful semantic information. On MS-COCO (trained with the number of classes $K = 1000$), we organize images and texts by their assigned cluster labels using the learned prototype classification model (i.e., (7)). We first visually inspect individual clusters, images and texts that belong to each cluster. As we show a few examples in Fig. 2 (more in Appendix (Fig. 2,3, Sec. A)), each cluster contains semantically coherent data samples. Then we inspect texts (captions) in each cluster, and select a few keywords, those words that appear the most frequently in the texts. These keywords for each cluster consist of objects (noun) and/or actions (verb) that faithfully describe the cluster and data samples that belong to it. The full list is shown in Appendix (Fig. 1, Sec. A), but to enumerate a few of them (cluster ID: keywords), for instance, 0014: giraffe/feeding, 0169: soccer/playing, 0283: bus/parked, 0405: pizza/oven, 0597: vase/flowers, 0713: dog/ball, 0818: kite/flying, 0956: parking/meter.

Although the last three clusters in Fig. 2 all have the semantic meaning of *baseball*, they have different details in either activity or focus/scene: *swing*, *base playing*, and *crowd scene*. This means that SwAMP finds clusters based on the whole contents (objects, activities, and scenes), instead of doing merely object-based clustering. Although we have roughly equal numbers of samples per cluster, we found that some clusters are overlapped with others in terms of semantic meaning (redundant clusters in Appendix (Fig. 1),...
increased flexibility, while discovering meaningful latent semantic classes. The efficacy of our approach was demonstrated on several real-world cross-modal retrieval problems in diverse modalities, text-video, sketch-photo, and image-text, where our method achieved significant performance improvement over the contrastive learning for all these tasks.

References

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In Advances in Neural Information Processing Systems (NeurIPS), 2020.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In International Conference on Learning Representations (ICLR), 2020.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2015. International Conference on Learning Representations (ICLR).

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In European Conference on Computer Vision (ECCV), 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In Advances in Neural Information Processing Systems, 2020.

J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Anggie Bogust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, James Glass, Michael Picheny, and Shih-Fu Chang. Multi-modal Clustering Networks for Self-supervised Learning from Unlabeled Videos, 2021. International Conference on Computer Vision.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. arXiv preprint arXiv:1504.00325, 2015.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In IEEE Conference on Computer Vision and Pattern Recognition, 2005.

Sanghyuk Chun, Seong Joon Oh, Rafael Sampaio de Rezende, Yannis Kalantidis, and Diane Larlus. Probabilistic embeddings for cross-modal retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8415–8424, June 2021.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances. In Advances in Neural Information Processing Systems, 2013.

Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. arXiv preprint arXiv:2006.06666, 2020.

Sounak Dey, Pau Riba, Anjan Dutta, Josep Lladós, and Yi-Zhe Song. Doodle to Search: Practical Zero-Shot Sketch-based Image Retrieval. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? ACM Transactions on Graphics, 31(4): 1–10, 2012.

F. Faghri, D.J. Fleet, J.R. Kiros, and S. Fidler. VSE++: Improved visual-semantic embeddings with hard negatives. In Proc. of British Machine Vision Conference, 2018.

V. Gabeur, C. Sun, K. Alahari, and C. Schmid. Multimodal transformer for video retrieval, 2020. European Conference on Learning Representations.

J. Gu, J. Cai, S. Joty, L. Niu, and G. Wang. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In IEEE Conference on Computer Vision and Pattern Recognition, 2006.

K. Haro, H. Kataoka, and Y. Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722, 2019.

Y. Huang, Q. Wu, and L. Wang. Learning semantic concepts and order for image and sentence matching. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-BERT: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

D. Kauman, G. Levi, T. Hassner, and L. Wolf. Temporal tessellation: A unified approach for video analysis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *European Conference on Computer Vision (ECCV)*, 2018.

Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *International Conference on Computer Vision (ICCV)*, 2019.

Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts, 2019. British Machine Vision Conference.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, 2019.

A. Miech, I. Laptev, and J. Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

A. Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *International Conference on Computer Vision (ICCV)*, 2019.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. Thinking fast and slow: Efficient text-to-visual retrieval with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9826–9836, June 2021.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

H. Nam, J.W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

M. Patrick, P. Y. Huang, Y. Asano, F. Metze, A. Hauptmann, J. Henriques, and A. Vedaldi. Support-set bottlenecks for video-text representation learning, 2020. International Conference on Learning Representations.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *International Journal of Computer Vision*, 123:94–120, 2017.

Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards style-agnostic sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8504–8513, June 2021.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics*, 35(4):1–12, 2016.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
Yuming Shen, Li Liu, Fumin Shen, and Ling Shao. Zero-shot sketch-image hashing. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective, 2016. In Advances in Neural Information Processing Systems.

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

A. Torabi, N. Tandon, and L. Sigal. Learning language visual embedding for movie understanding with natural language. arXiv preprint arXiv:1609.08124, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, 2017.

Cédric Villani. Optimal Transport: Old and New. Springer, 2008.

Jiang Wang, Yang Song, T. Leung, C. Rosenberg, Jingbin Wang, J. Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. arXiv preprint arXiv:1607.06215, 2016.

Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: Global-local sequence alignment for text-video retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5079–5088, June 2021.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. Ranked list loss for deep metric learning. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019a.

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, 2019b.

M. Wray, H. Doughty, and D. Damen. On Semantic Similarity in Video Retrieval, 2021. Computer Vision and Pattern Recognition.

Chao-Yuan Wu, R. Mannmatha, Alexander J. Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A large video description dataset for bridging video and language. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Sasikiran Yelamarthi, Shiva Krishna Reddy M, Ashish Mishra, and Anurag Mittal. A zero-shot framework for sketch based image retrieval. In European Conference on Computer Vision (ECCV), 2018.

Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Y. Yu, J. Kim, and G. Kim. A joint sequence fusion model for video question answering and retrieval. In European Conference on Computer Vision (ECCV), 2018.

Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

Qi Zhang, Zhen Lei, Zhaoxiang Zhang, and Stan Z. Li. Context-aware attention network for image-text retrieval. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Z. Zheng, L. Zheng, M. Garrett, Y. Yang, and Y.D. Shen. Dual-path convolutional image-text embedding. arXiv preprint arXiv:1711.05535, 2017.

L. Zhou, C. Xu, and J. J. Corso. Towards automatic learning of procedures from web instructional videos. In The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), 2018.

Mo Zhou, Zhenxing Niu, Le Wang, Zhanming Gao, Qilin Zhang, and Gang Hua. Ladder Loss for Coherent Visual-Semantic Embedding. In AAAI Conference on Artificial Intelligence, 2020.
Appendix

• Visualization of Learned Clusters (Sec. A)
• Ablation Study (Sec. B)
• (Detailed) Text-based Video Retrieval (Sec. C)
• (Detailed) Image-Text Retrieval (Sec. D)
• (Extra Experiments) Synthetic Data (Sec. E)

A Visualization of Learned Clusters

As qualitative analysis, we visualize the learned clusters to see if they capture meaningful semantic information. On MS-COCO (trained with the number of classes $K = 1000$), we organize images and texts by their assigned cluster labels using the learned prototype classification model,

$$p(y = j | x^M) = \frac{\exp(p_j \phi^M(x)/\tau)}{\sum_{i} \exp(p_i \phi^M(x)/\tau)}, \quad M \in \{A, B\}$$  \hspace{1cm} (7)

in our SwAMP. We first visually inspect individual clusters, images and texts that belong to each cluster. Some examples are shown in Fig. 4 and Fig. 5; each cluster contains semantically coherent data samples. Then we inspect texts (captions) in each cluster, and select a few keywords, those words that appear the most frequently in the texts. These keywords for each cluster consist of objects (noun) and/or actions (verb) that faithfully describe the cluster and data samples that belong to it. The full list is provided in Fig. 3.

A.1 Clustering based on Whole Contents

In Fig. 4 and Fig. 5 looking at clusters:

• Group-A = (0100, 0234, 0359, 0405, 0428)
• Group-B = (0195, 0208, 0221, 0253)
• Group-C = (0180, 0683)

the clusters within these groups are all related to the semantic meaning of pizza, baseball, and cat, respectively. However, they have different details in either activity or focus/scene: man eating, people and table, on plate, oven, and with woman for Group-A; swing, on field, base playing, and crowd scene for Group-B; on bed and television for Group-C. This means that SwAMP finds clusters based on the whole contents (objects, activities, and scenes), instead of doing merely object-based clustering.

A.2 Class Imbalance

Although we have roughly equal numbers of samples per cluster, we can see many redundant/repeated clusters in Fig. 3. This means that some clusters are overlapped with others in terms of semantic meaning, constituting larger super-clusters. These clusters are related to dominant data samples (e.g., cat, dog, tennis, baseball). This implies that the SwAMP can effectively deal with imbalance of semantic classes that can reside in data.
B Ablation Study

In our experiments, we chose the hyperparameters by cross validation with grid search. We perform empirical study on the impact of two important hyperparameters in our model: the number of classes \( K \) and SwAMP loss trade-off \( \lambda \).

Number of classes \((K)\). Recall that the best \( K \) values we chose were: \( K = 1000 \) for the image-text retrieval datasets and \( K = 500 \) for text-based video retrieval. To see how the retrieval performance is affected by other choices of \( K \), we conduct experiments by varying \( K \) around the optimal values. The results are shown in Fig. 6, Fig. 7, Fig. 8, Fig. 9, and Fig. 10. Clearly, very small \( K \) has low retrieval performance (R@1), and increasing \( K \) leads to improvement. However, beyond certain points, there is no benefit of increasing \( K \) and we even see performance degradation, which agrees with the observations from previous work \cite{Asano2020, Caron2020}. This is perhaps due to the difficulty of assigning meaningful cluster labels in optimal transport. Overall, with properly chosen \( K \), SwAMP outperforms contrastive learning, signifying that SwAMP’s grouping/clustering of similar instances is more effective than vanilla instance discrimination. The fact that the optimal \( K \) values are different in two tasks (image-text and video-text) implies that the best cardinality of semantic clusters is highly dependent on the dataset characteristics (e.g., size and semantic diversity).

SwAMP loss trade-off \((\lambda)\). We perform sensitivity analysis on \( \lambda \), the strength of the SwAMP loss. For different values of \( \lambda \), the retrieval scores (R@1) are shown in Fig. 11, Fig. 12, Fig. 13, Fig. 14, and Fig. 15. Our model remains better than contrastive learning for large intervals of different \( \lambda \)’s, and the performance is not very sensitive to \( \lambda \).

C Text-based Video Retrieval

We consider the text-to-video retrieval task where the goal is to find the most relevant video clip for a given natural language text query. We consider three datasets for this task: i) YouCook2 \cite{Zhou2018} of cooking videos and instructions, ii) MSR-VTT \cite{Xu2016} of generic videos and captions from YouTube, and iii) LSMDC \cite{Rohrbach2017} of movie clips and subtitles. All these datasets provide pairs of video clip and its text description, forming a multi-modal paired data format \((text, video)\) which conforms to our SwAMP framework.

For the raw text/video features and the feature extractor networks, as well as the training/test protocols, we follow the methods in \cite{Miech2019}. Whereas the details of the datasets and experimental setups are described in the subsequent sections, the features are specifically built by the following procedures. First, the raw features are obtained by the pretrained networks: (a) raw video features (4096D) are concatenation of frame-level and video-level features extracted from the pretrained 2D/3D CNNs (the ImageNet pre-trained Resnet-152 \cite{He2016} for 2D features and the Kinetics \cite{Carreira2017} pre-trained ResNeXt-101 16-frame model \cite{Hara2018} for 3D features), (b) raw text features (4096D) are the GoogleNews pre-trained word2vec embeddings \cite{Mikolov2013} for the pre-processed transcribed video narrations with the common stop words removed. Then the feature extractor networks \( \phi^{video}(\cdot) \) and \( \phi^{text}(\cdot) \) transform these raw features into 4096D features by the sigmoid-gated linear transform where the gating functions are two-layer linear networks \cite{Miech2018}. We fix the raw features and train only the latter sigmoid-gated networks, which comprise about 67M parameters.

Following \cite{Miech2019}, there are two training strategies: i) No-pretraining (No-PT) where the feature extraction networks are randomly initialized, and the training is done on the training split of the dataset, and ii) Pretraining (PT) where the feature extractors are first pretrained on the large-scale HowTo100M dataset \cite{Miech2019}, and finetuned on the target dataset. In \cite{Miech2019}, they adopt the contrastive (triplet) loss for training the feature extractors. Although we also compare our approach with the state-of-the-arts, the main focus in this experiment is to demonstrate the performance improvement achieved by the proposed SwAMP loss against vanilla contrastive learning. The SwAMP hyperparameter \( \lambda \), the weight/impact of the SwAMP loss, is chosen as \( \lambda = 0.25 \) for all three datasets, except the LSMDC-PT case for which \( \lambda = 0.1 \). We also choose temperature in softmax \( \tau = 0.25 \), entropic regularization trade-off in SK \( \eta = 5.0 \), the number of classes \( K = 500 \), and the queue size 2,048 for the SwAMP. The other learning hyperparameters common in SwAMP and contrastive losses are not changed from \cite{Miech2019}.

YouCook2. This cooking video dataset collected from YouTube, contains 89 recipes and 14K video clips annotated with textual descriptions from paid human workers. The test data are formed by taking 3.5K clips from the validation set, and the test set comprises of 3,350 pairs. The retrieval performance metrics are recall-at-\( k \) (R@k) with \( k = 1, 5, 10 \) and the median rank (Med-R). Hence, the random guess attains R@1 = 0.03% Med-R=1,675. The results are summarized in Table 7. In the bottom four rows, we see the performance improvement achieved by the proposed SwAMP against the contrastive loss \cite{Miech2019}. For both training strategies, No PT (random model initialization) and PT (initialized with the HowTo100M-pretrained model), our SwAMP improves the retrieval performance significantly (e.g., about 12% reduction in
Table 7: Text-video retrieval results on YouCook2.

| Methods               | R@1 ↑ | R@5 ↑ | R@10 ↑ | Med-R ↓ |
|-----------------------|-------|-------|--------|---------|
| Random                | 0.03  | 0.15  | 0.3    | 1675    |
| FV-CCA                | 4.6   | 14.3  | 21.6   | 75      |
| Contrastive (No PT)   | 4.2   | 13.7  | 21.5   | 65      |
| SwAMP (No PT)         | 4.8   | 14.5  | 22.5   | 57      |
| Contrastive (PT)      | 8.2   | 24.5  | 35.3   | 24      |
| SwAMP (PT)            | 9.4   | 24.9  | 35.3   | 22      |

Table 8: Text-video retrieval results on MSRVTT.

| Methods               | R@1 ↑ | R@5 ↑ | R@10 ↑ | Med-R ↓ |
|-----------------------|-------|-------|--------|---------|
| Random                | 0.1   | 0.5   | 1.0    | 500     |
| C+LSTM+SA+FC7         | 4.2   | 12.9  | 19.9   | 55      |
| VSE-LSTM              | 3.8   | 12.7  | 17.1   | 66      |
| SNUVL                 | 3.5   | 15.9  | 23.8   | 44      |
| Temporal Tessellation | 4.7   | 16.6  | 24.1   | 41      |
| CT-SAN                | 4.4   | 16.6  | 22.3   | 35      |
| JSFusion              | 10.2  | 31.2  | 43.2   | 13      |
| Contrastive (No PT)   | 12.1  | 35.0  | 48.0   | 12      |
| SwAMP (No PT)         | 15.0  | 38.5  | 50.3   | 10      |
| Contrastive (PT)      | 14.9  | 40.2  | 52.8   | 9       |
| SwAMP (PT)            | 19.0  | 42.4  | 55.2   | 8       |

Table 9: Text-Video retrieval results on LSMDC.

| Methods               | R@1 ↑ | R@5 ↑ | R@10 ↑ | Med-R ↓ |
|-----------------------|-------|-------|--------|---------|
| Random                | 0.1   | 0.5   | 1.0    | 500     |
| C+LSTM+SA+FC7         | 4.3   | 12.6  | 18.9   | 98      |
| VSE-LSTM              | 3.1   | 10.4  | 16.5   | 79      |
| SNUVL                 | 3.6   | 14.7  | 23.9   | 50      |
| Temporal Tessellation | 4.7   | 15.9  | 23.4   | 64      |
| CT-SAN                | 4.5   | 14.1  | 20.9   | 67      |
| JSFusion              | 9.1   | 21.2  | 34.1   | 36      |
| Contrastive (No PT)   | 7.2   | 18.3  | 25.0   | 44      |
| SwAMP (No PT)         | 7.7   | 19.3  | 27.7   | 40      |
| Contrastive (PT)      | 7.1   | 19.6  | 27.9   | 40      |
| SwAMP (PT)            | 8.3   | 20.0  | 28.9   | 37      |

Median Rank for the No PT case). SwAMP also outperform the CCA baseline FV-CCA (Klein et al., 2015).

**MSRVTT.** This generic video-text dataset (Xu et al., 2016) collected from YouTube contains videos of specific categories including music, sports, and movie. There are 200K video-caption pairs obtained by human annotation. We follow the retrieval training/test protocol of (Yu et al., 2018; Miech et al., 2019). The test set consists of 1K pairs. As reported in Table 8, our SwAMP loss improves the performance over the contrastive learning significantly for both no-pretraining and pretraining cases: about 24% in R@1 in the No PT case, and 27% in the PT case. Furthermore, the SwAMP outperforms with large margin the state-of-the-arts: C+LSTM+SA+FC7 (Torabi et al., 2016), VSE-LSTM (Kiros et al., 2014), Temporal Tessellation (Kauman et al., 2017), CT-SAN (Yu et al., 2017), and JSFusion (Yu et al., 2018).

**LSMDC.** The LSMDC (Rohrbach et al., 2017) is a dataset of movie video clips, comprised of 101K video-caption pairs. The captions are collected either from the movie scripts or the audio descriptions. The test set contains 1K pairs. For this dataset, we use the SwAMP hyperparameter (impact of the SwAMP loss against the contrastive loss) $\lambda = 0.1$ for the PT case. The results are shown in Table 9. Similar to the other two datasets, our SwAMP is consistently better than the contrastive learning (about 7 ~ 9% in Median Rank).

https://sites.google.com/site/describingmovies/lsmdc-2016/movieretrieval
D Image-Text Retrieval

For the image-text cross-modal retrieval task, we follow the features and protocols from the well-known stacked cross attention network (SCAN) [Lee et al., 2018]. In their framework, each image is represented by a set of local features $V = \{v_1, \ldots, v_K\}$, where $v_i \in \mathbb{R}^D$ = $W_v f_i + b_v$ and $f_i$’s are the CNN features extracted from salient image regions detected by the Faster-R-CNN model [Ren et al., 2015]. The raw features $f_i$’s are fixed and $\{W_v, b_v\}$ are learnable parameters. The text (sentence) is also treated as a set of word features $E = \{e_1, \ldots, e_n\}$, where $e_i \in \mathbb{R}^D = (h_i^t + h_i^q)/2$ and $h_i^q/t$ are the outputs of the bi-directional GRU [Bahdanau et al., 2015; Schuster and Paliwal, 1997] with the sequence of word embeddings as input. Both the word embeddings and GRU parameters are learnable. These image/text features contain rich local information, however, one challenge is that both representations are sets, hence the number of elements $(k$ and $n$) can vary from instance to instance.

In the original SCAN paper [Lee et al., 2018], they proposed a cross-modal attention model, where each local feature from one modality is transformed by the attention [Vaswani et al., 2017] with the set of local features in the other modality; e.g., $v_i$ is transformed to $\text{attn}(v_i; \{e_j\}_{j=1}^n) = \text{weighted sum of values} \{e_j\}_{j=1}^n$ with $v_i$ as a query and $\{e_j\}_{j=1}^n$ as keys (this denoted by i-t, while the other attention direction i-t can be used alternatively). Then the similarity score between image $V$ and text $E$ is defined as $\text{pool}((\cos(v_i, \text{attn}(v_i; \{e_j\}_{j=1}^n)))_{i=1}^K)$, where $\cos(a, b)$ is the cosine similarity and pool is the pooling operation, either of AVG or LSE (log-sum-exp). Then the triplet contrastive loss is employed. For the details, please refer to [Lee et al., 2018].

Note that in the SCAN, there is no succinct modality-wise embedding vector representation, but the similarity score between instances of two modalities is rather computed by highly complex attention operations. Although this is helpful for capturing the interactions between local features, computing the similarity score takes quadratic time in the number of elements (local features) in the instances. This is time consuming compared to simple dot-product of the modality-wise embedding vectors (See Table 12 for the actual running times compared with the approaches based on modality-wise attention representation). Moreover, it is not applicable to our SwAMP approach since we need to predict the class labels for each modality from modality-wise representation $\phi^{image}(V)$, $\phi^{text}(E)$.

To have modality-wise representation, we adopt the idea of induced-set attention (ISA) from the Set Transformer [Lee et al., 2019]. Specifically, we introduce $p$ learnable prototype (query) vectors $\{q_j\}_{j=1}^p$ where $q_j \in \mathbb{R}^D$. Then we compute the attention for each query with $V$ (or $E$), i.e., $z_j = \text{attn}(q_j; \{v_i\}_{i=1}^K)$. Then we define $\phi^{image}(V) = \text{concat}(z_1, \ldots, z_p)$, similarly for $\phi^{text}(E)$, where concat refers to concatenation. Thus the parameters for $\phi^{image}()$ are $\{W_v, b_v\}$ and $\{q_j\}_{j=1}^p$, and the parameters for $\phi^{text}()$ are the word embeddings, GRU parameters, and $\{q_j\}_{j=1}^p$. We share the same $\{q_j\}_{j=1}^p$ for both modalities. We also have multi-head extension by computing these features multiple times and concatenating them. We call these modality-wise features as prototype attention representation (PAR). Note that computing PAR features has linear complexity in the number of local features (assuming $p$ is constant), and the cross-modal similarity is simply dot-product of PAR features, and can be computed in linear time (See also Table 12).

D.1 Datasets and Results

We test our approach on the popular image-text retrieval datasets, MS-COCO and Flickr30K. There are 31K images and five captions for each image in Flickr30K. MS-COCO contains 123,287 images, where each image is annotated with five text descriptions. Following the widely-used split [Karpathy and Fei-Fei, 2015; Faghri et al., 2018], for the Flickr30K, we have 1K images for validation, 1K images for testing, and the rest for training. For MS-COCO, there are 5K test images (and 25K captions, five captions for each image). We also follow two standard protocols for measuring the test retrieval performance for MS-COCO: 1) using the entire 5K test images or 2) splitting the test set into 5 folds and report the average retrieval performance over the 5 folds.

The results are summarized in Table 10 (Flickr) and Table 11 (MS-COCO). We specifically highlight the comparison between the contrastive loss and our SwAMP loss with the modality-wise feature representation (Contrastive-PAR vs. SwAMP-PAR). For the PAR features, we choose the number of prototypes $p = 20$, attention weight temperature $T = 0.5$, and the number of heads $H = 1$ for Flickr, and $p = 10$, $T = 0.5$, $H = 2$ for MS-COCO. For the SwAMP hyperparameters, we use the impact of SwAMP loss $\lambda = 1.0$, softmax temperature $\tau = 0.025$, the number of classes $K = 1,000$, queue size 1,280 for both datasets. As shown, the SwAMP loss performs consistently better than the contrastive loss. SwAMP also outperforms several state-of-the-arts including the recent sophisticated probabilistic embedding strategy [Chun et al., 2021].

When compared with the computationally expensive SCAN, SwAMP mostly outperforms SCAN except for the SCAN’s best attention direction/combination choices. Note that SwAMP uses the simple feature aggregation strategy (PAR) to have
Table 10: Image-text retrieval results on Flickr30K.

| Methods               | Image → Text | Text → Image |
|-----------------------|--------------|--------------|
|                       | R@1  | R@5  | R@10 | R@1  | R@5  | R@10 |
| DAN [Nam et al., 2017] | 55.0 | 81.8 | 89.0 | 39.4 | 69.2 | 79.1 |
| DPC [Zheng et al., 2017] | 55.6 | 81.9 | 89.5 | 39.1 | 69.2 | 80.9 |
| VSE++ [Faghri et al., 2018] | 52.9 | -    | 87.2 | 39.6 | -    | 79.5 |
| SCO [Huang et al., 2018] | 55.5 | 82.0 | 89.3 | 41.1 | 70.5 | 80.1 |
| SCAN i-t AVG           | 67.9 | 89.0 | 94.4 | 43.9 | 74.2 | 82.8 |
| SCAN t-i AVG           | 61.8 | 87.5 | 93.7 | 45.8 | 74.4 | 83.0 |
| SCAN t-i AVG + i-t LSE | 67.4 | 90.3 | 95.8 | 48.6 | 77.7 | 85.2 |
| Contrastive-PAR        | 65.7 | 86.8 | 92.4 | 48.2 | 75.8 | 84.2 |
| SwAMP-PAR              | 67.8 | 88.5 | 94.0 | 49.1 | 76.1 | 83.7 |

Table 11: Image-text retrieval results on MS-COCO.

| Methods               | 5-fold (1K test images) | Entire (5K test images) |
|-----------------------|--------------------------|--------------------------|
|                       | Image → Text | Text → Image | Image → Text | Text → Image |
|                       | R@1  | R@5  | R@10 | R@1  | R@5  | R@10 | R@1  | R@5  | R@10 |
| DPC [Zheng et al., 2017] | 65.6 | 89.8 | 95.5 | 47.1 | 79.9 | 90.0 | 41.2 | 70.5 | 81.1 |
| VSE++ [Faghri et al., 2018] | 64.6 | -    | 95.7 | 52.0 | -    | 92.0 | 41.3 | -    | 81.2 |
| GXN [Gu et al., 2018]      | 68.5 | -    | 97.9 | 56.6 | -    | 94.5 | 42.0 | -    | 84.7 |
| SCO [Huang et al., 2018]   | 69.9 | 92.9 | 97.5 | 56.7 | 87.5 | 94.8 | 42.8 | 72.3 | 83.0 |
| PCME [Chun et al., 2021]  | 68.8 | -    | 54.6 | -    | -    | -    | 44.2 | -    | 31.9 |
| SCAN i-t                | 69.2 | 93.2 | 97.5 | 54.4 | 86.0 | 93.6 | 46.4 | 77.4 | 87.2 |
| SCAN t-i + i-t          | 72.7 | 94.8 | 98.4 | 58.8 | 88.4 | 94.8 | 50.4 | 82.2 | 90.0 |
| Contrastive-PAR         | 71.8 | 94.3 | 97.9 | 56.8 | 86.9 | 93.8 | 48.4 | 78.1 | 88.1 |
| SwAMP-PAR              | 72.6 | 94.6 | 98.0 | 57.4 | 87.6 | 94.1 | 49.7 | 79.1 | 88.3 |

Table 12: Running time comparison for SCAN (cross-modal attention) and our SwAMP-PAR. Running times (seconds) are measured on the same machine (Core i7 3.50GHz CPU, 128GB RAM, and a single GeForce RTX-2080 Ti GPU). We report per-batch times for training, and entire retrieval times for test. For MS-COCO test, the running times for 5K test images are reported, where times for 1K test images averaged over 5 folds are shown in the parentheses. For SCAN, when we use features in both directions (e.g., t-i AVG + i-t LSE), the running times are roughly doubled.

| Methods               | Flickr30K | MS-COCO |
|-----------------------|-----------|---------|
|                       | Train    | Test    | Train    | Test |
| SCAN i-t AVG           | 0.35     | 336.9   | 0.33     | 9352.0 (350.3) |
| SwAMP-PAR              | 0.09     | 3.8      | 0.08     | 25.9 (16.3) |

E Synthetic Data

In this section we devise a synthetic dataset not only for performing the proof-of-concept test of our SwAMP algorithm, but also analyze the impacts of the various hyperparameters and training options in the proposed algorithm. For the former, we especially focus on the retrieval performance improvement achieved by our SwAMP compared to the contrastive loss or its popular variants (e.g., online hard-example mining loss).

The dataset is constructed by the following procedure: We randomly generate 20 Gaussians in $\mathbb{R}^5$, each of which is considered to represent a semantic class. For each Gaussian (class), we sample a latent vector $z \in \mathbb{R}^5$, and a pair of instances...
(\(x^A \in \mathbb{R}^{100}, x^B \in \mathbb{R}^{100}\)) is then generated by \(x^A = f_A(z)\) and \(x^B = f_B(z)\) where \(f_A\) and \(f_B\) are randomly initialized fully-connected DNNs with two hidden layers of 50 units. We generate 500 pairs for each class that leads to 10,000 data pairs, and split them into 7000/1000/2000 train/validation/test sets. The validation recall-at-1 (R@1) performance is evaluated at every training epoch, and the model at the epoch with the best validation performance is selected as the final model. Note that during training we only use the paired data \((x^A, x^B)\) with the semantic class labels hidden to the training algorithms.

For training, we adopt the embedding networks \(\phi^A(x^A)\) and \(\phi^B(x^B)\) as fully-connected neural nets with two hidden layers of 50 units. The embedding dimension is chosen as 5. We train the model with this same network architecture, using the contrastive loss and our SwAMP loss. For both loss functions, the batch size is 128, and the Adam optimizer (Kingma and Ba, 2015) is used with learning rate \(10^{-3}\), and the maximum epoch is 100.

For the contrastive loss, we adopt the (online) hard-example mining with the margin parameter \(\alpha = 0.1\). For the SwAMP loss, the defaults parameters are as follows: temperature \(\tau = 0.01\) for the softmax classifier, the reciprocal impact of the max-entropy regularizer for the Sinkhorn-Knopp \(\eta = 1/0.05\) (i.e., we add the entropic regularizer with the weight \(\eta^{-1} = 0.05\) to the objective of the OT problem. Also, by default, we choose the number of classes \(K = 1000\) and the queue size 1,280, 10 times the batch size (and greater than \(K\)). For both loss functions, the embedding networks are initialized randomly.

For test, we perform the cross-modal retrieval task \(x^A \rightarrow x^B\), treating each \(x^A\) in the test set as a query, retrieving \(x^B\) from the test set. There are two ways to define the retrieval error: i) pair-based which treats the retrieved \(x^B\) as a correct retrieval only if the query \(x^A\) and the retrieved \(x^B\) are found as a pair in the data, and ii) class-based which compares only the classes of the query \(x^A\) and the retrieved \(x^B\). Hence the pair-based error is more strict than the class-based since it counts only the data item that appears in the data as correct retrieval, without comparing the semantic classes of the retrieved item and the query.

### E.1 Ablation study on hyperparameters

There are several hyperparameters in our SwAMP model, and we have conducted several ablation-type study on the impacts of the hyperparameters. The hyperparameters that are deemed to be the most critical are: i) the number of classes \(K\), ii) the size of the queues, iii) initialization of the feature extraction networks (either random initialization or pretrained one with the contrastive loss), iv) entropic regularization trade-off \(\eta\) in Sinkhorn-Knopp, and v) the soft/hard cluster assignment after OT clustering.

**Number of classes** \((K)\). We vary the number of classes \(K\) for 200, 500, 1000, 2000, 3000, and record the R@1 scores for both pair and class based error types for our SwAMP model. The results are shown in Fig. 16. We see that allowing more clusters improves the performance. However, once \(K\) is around 1000 or greater than 1000, there is no significant benefit of increasing \(K\). This implies that SwAMP does not merely do instance discrimination, but seeks for grouping/clustering of similar instances. Although we did not include it in the figure, having \(K = 20\), i.e., the true number of semantic classes, yielded poor performance (worse than \(K = 200\)). This means that it is very difficult to expect that the model would discover the underlying semantic classes correctly.

**Size of queues.** Another important hyperparameter is the size of the queues, where the OT clustering is performed on the latest features that are stored in the queues. In addition to the default queue size 1280 = 10 \(\times\) 128 (batch size), we try with different queue sizes \(\{0, 1, 2, 5, 20\}\) \(\times\) 128. Note that the OT clustering is performed on the union of the features in the queue and the current batch, hence zero queue size implies that we only use the current batch for OT clustering. The results are reported in Fig. 17. As shown, increasing the queue sizes accordingly improves the performance, where with the queue size of two times the batch size outperforms the contrastive loss. Also, not using the queues (“No queue”) resulted in poor performance, signifying the importance of using the queues. Interestingly, too large queue size \((20 \times)\) deteriorates the performance, which might be explained by the negative effects of the stale features obtained several iterations ago from the
old feature extractor networks. This suggests the trade off of the queue size: too small queue size does not generalize well to the clustering of entire data, while too large queue size can be harmful due to the inconsistent stale features.

**Initialization of feature extractor networks.** In our default setup, the feature extractor networks $\phi^A(\cdot)$ and $\phi^B(\cdot)$ are initialized randomly. Now we test the performance of the SwAMP when the feature extractor networks are initialized from the pretrained ones by the contrastive loss training. We initially expected that this warm-start training may expedite the training with the SwAMP loss, however, as the results in Fig. 18 indicates, it does not outperform the random initialization although the warm-start is still better than contrastive loss training. This may imply that the SwAMP loss defines a very different loss landscape from the contrastive loss, and the contrastive-loss optimized model may lie at the region far from the optima of the SwAMP loss, thus the warm-start even hinders convergence to the SwAMP optima.

**Impact of the entropic regularization ($1/\eta$).** In the Sinkhorn-Knopp (SK) algorithm, we have the reciprocal trade-off $1/\eta$ for the entropy term of the optimization variables $q(y|x)$. Too much emphasizing the entropy term (by increasing $1/\eta$ or decreasing $\eta$) would lead to near uniform $q(y|x)$, which means that it carries little information about the meaningful classes, and cluster assignment can be more or less random. On the other hand, having too small impact of the entropy term would make the SK algorithm converge too slowly, and the output of the SK with only a few iterations would produce non-optimal solutions. To see the impact, we vary $1/\eta$ from 0.01, 0.05 (default), and 0.1, and the results are shown in Fig. 19. We see that there is slight performance degradation for small and large $1/\eta$ values from the optimal choice.

**Soft or hard cluster assignment after OT.** We also check if the hard cluster assignment thresholding after OT optimization would be beneficial or not. Recall that the default is to use the output $q(y|x)$ of the SK algorithm as it is (i.e., soft cluster assignment). In the hard assignment we further threshold $q(y|x)$ to have one-hot encoding, which is then used in the cross-entropy loss optimization. As shown in Fig. 20, the hard assignment is harmful, which implies that retaining uncertainty in cluster estimation is important to have accurate clustering and feature learning.
Figure 3: 1000 clusters trained with MS-COCO. For each cluster, we depict a few most frequent keywords in the captions that belong to the cluster.
Figure 4: Some randomly selected clusters with images and texts that belong to them. Each cluster, titled by ID: keywords, shows randomly chosen 5 images and 4 texts.
Figure 5: More clusters (continued from Fig. 4).
Figure 6: (Flickr30K) Impact of the number of classes ($K$).

Figure 7: (MS-COCO) Impact of the number of classes ($K$).

Figure 8: (YouCook2) Impact of the number of classes ($K$).

Figure 9: (MSRVTT) Impact of the number of classes ($K$).

Figure 10: (LSMDC) Impact of the number of classes ($K$).
Figure 11: (Flickr30K) Impact of the SwAMP loss ($\lambda$).

Figure 12: (MS-COCO) Impact of the SwAMP loss ($\lambda$).

Figure 13: (YouCook2) Impact of the SwAMP loss ($\lambda$).

Figure 14: (MSRVTT) Impact of the SwAMP loss ($\lambda$).

Figure 15: (LSMDC) Impact of the SwAMP loss ($\lambda$).
Figure 16: (Synthetic data) Impact of the number of classes ($K$).

Figure 17: (Synthetic data) Impact of the size of the queues.

Figure 18: (Synthetic data) Impact of the initialization of feature extractor networks.

Figure 19: (Synthetic data) Impact of entropic regularization ($1/\eta$) in Sinkhorn-Knopp.

Figure 20: (Synthetic data) Soft (default) or hard cluster assignment after OT.