Cross-Modal Retrieval With Noisy Correspondence via Consistency Refining and Mining

Xinran Ma, Mouxing Yang, Yunfan Li, Peng Hu, Jiancheng Lv, *Senior Member, IEEE*, and Xi Peng, *Senior Member, IEEE*

***Abstract*— The success of existing cross-modal retrieval (CMR) methods heavily rely on the assumption that the annotated cross-modal correspondence is faultless. In practice, however, the correspondence of some pairs would be inevitably contaminated during data collection or annotation, thus leading to the so-called Noisy Correspondence (NC) problem. To alleviate the influence of NC, we propose a novel method termed Consistency REfining And Mining (CREAM) by revealing and exploiting the difference between correspondence and consistency. Specifically, the correspondence and the consistency only be coincident for true positive and true negative pairs, while being distinct for false positive and false negative pairs. Based on the observation, CREAM employs a collaborative learning paradigm to detect and rectify the correspondence of positives, and a negative mining approach to explore and utilize the consistency. Thanks to the consistency refining and mining strategy of CREAM, the overfitting on the false positives could be prevented and the consistency rooted in the false negatives could be exploited, thus leading to a robust CMR method. Extensive experiments verify the effectiveness of our method on three image-text benchmarks including Flickr30K, MS-COCO, and Conceptual Captions. Furthermore, we adopt our method into the graph matching task and the results demonstrate the robustness of our method against fine-grained NC problem. The code is available on https://github.com/XLearning-SCU/2024-TIP-CREAM.**

***Index Terms*— Robust cross-modal retrieval, noisy correspondence, multi-modal learning, graph matching.**

I. Introduction

CROSS-MODAL retrieval (CMR) [2], [3], [4], [5], [6] aims at matching associated samples across different modalities, which has attracted increasing attention from both academic and industry communities. The key of CMR is to bridge the modality gap, hoping similar cross-modal samples would gather together in the feature space. To this end, most existing works [7], [8], [9], [10], [11] aim to learn the cross-modal consistency from the correspondence of associated (*i.e.*, positive) pairs. Although achieving promising performance, these works heavily rely on the assumption that the annotated cross-modal correspondence is faultless. In practice, however, it is daunting and even impossible to precisely annotate all data pairs, and thus the assumption is always violated. In particular, most modern cross-modal data is crawled from the Web [1], [12], [13]. As a result, it is inevitable to wrongly treat some unrelated pairs (*i.e.*, False Positives, FPs) as associated, leading to the so-called Noisy Correspondence (NC, see Fig. 1(a)) problem. Although the importance of combating NC is obvious, there are only a few studies have been conducted [14], [15]. To alleviate the influence of NC, these works try to model the association confidence of positive pairs,

(a) Noisy Correspondence

(b) Diverse Potential Consistency

Fig. 1. Illustrations of the NC problem and our observations. (a) Noisy Correspondence (NC): the dataset consists of both true and false positive pairs, while the ground truth is agnostic. NC will reduce the consistency of positives and mislead the optimization direction, thus degrading the performance of CMR models. (b) Diverse Potential Consistency: give an anchor, we observe that the consistency is different from the correspondence and with various forms in the negative bank. As shown, the pairs with concrete consistency are treated as negative, which however should be positive. The implicit pairs show the consistency beyond words, and the partial pairs have the consistency at the token instead of the instance level. Examples are selected from the Conceptual Captions [1] dataset, and the numerical values in images imply that our model could mine and exploit the potential consistency.

Manuscript received 9 February 2023; revised 9 December 2023; accepted 28 February 2024. Date of publication 25 March 2024; date of current version 1 April 2024. This work was supported in part by NSFC under Grant U21B2040 and Grant 62176171 and in part by the Fundamental Research Funds for Central Universities under Grant CJ202303. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Siwei Ma. *(Xinran Ma and Mouxing Yang contributed equally to this work.) (Corresponding author: Xi Peng.)*

The authors are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: xinranma.gm@gmail.com; yangmouxing@gmail.com; yunfanli.gm@gmail.com; penghu.ml@gmail.com; lvjiancheng@scu.edu.cn; pengx.gm@gmail.com).

Digital Object Identifier 10.1109/TIP.2024.3374221

and accordingly down-weight and even discard the unconfident pairs. Namely, these unconfident pairs are probably FPs which will mislead the model optimization.

Although these works have achieved promising performance, we observe that they ignore the complexity of real-world data to some extent. Specifically, the consistency could be identical to the correspondence only for true positives and true negatives, while being remarkably different for false positives and false negatives. As shown in Fig. 1(b), even though two given data points are unassociated as annotated (*i.e.*, negative), they could be with concrete, implicit, or partial consistency. Clearly, such a diverse potential consistency could be utilized to boost the performance of NC-contaminated models, which however has not been explored so far to the best of our knowledge.

Based on the above observation, we propose Consistency REfining And Mining (CREAM) for robust CMR by rectifying possible noisy correspondence in the positive bank and exploiting the diverse potential consistency in the negative bank. Our CREAM embraces the following two merits. On the one hand, it is able to prevent the model from fitting FPs, thus enjoying robustness against NC. On the other hand, it complements the consistency in negatives so that the CMR performance could be further boosted. In detail, CREAM first computes the association confidence for each data pair resorting to the memorization effect [16] of Deep Neural Networks (DNNs). Based on the estimated confidence, CREAM partitions the pairs into clean, vague, and noisy groups for consistency refining and mining. After that, CREAM will recalibrate the correspondence of positives while mining and exploiting the consistency of negatives. With the recast consistency, CREAM employs a novel contrastive loss to achieve robust CMR against noisy correspondence.

The contributions and novelties of this work could be summarized as follows:

- For the first time, we reveal that the correspondence and consistency cannot be simply treated as identical for the cross-modal pairs. Such a new observation is largely ignored by existing works, which however could boost the CMR performance.
- With our observation, we propose a novel CMR method (dubbed CREAM) that achieves robustness against the NC through consistency refining and mining. On the one hand, CREAM could rectify the correspondence of positive pairs, thus preventing overfitting to NC. On the other hand, CREAM could mine and exploit the diverse potential consistency rooted in negative pairs.
- Extensive experiments on three widely-used CMR benchmarks (Flickr30K, MS-COCO, and Conceptual Captions) verify the effectiveness of our method compared with six state-of-the-art methods.
- Beyond robustness against the instance-level image-text NC, the experiments on the graph matching task across three benchmarks (Willow Object, Pascal VOC, SPair-71k) further validate the effectiveness and generality of our method in handling the fine-grained patch-level NC.

II. Related Work

In this section, we briefly review some recent developments in four related areas, *i.e.*, cross-modal retrieval, learning with noisy labels, learning with noisy correspondence and contrastive learning.

A. Cross-Modal Retrieval

CMR [2], [5], [8], [17], [18], [19] aims to search semantic-relevant samples from different modalities, wherein the key is to alleviate the modality gap. For this purpose, existing CMR methods mainly focus on exploiting the cross-modal consistency hidden in the correspondence of associated pairs, so that different modalities could be bridged. According to the strategy of exploiting consistency, the existing CMR works could be roughly categorized into the following two groups: i) Coarse-grained CMR methods [20], [21], [22], which adopt different backbones to exact modality-specific features and align those features from a global perspective. ii) Fine-grained CMR methods [2], [8], [17], [23], which narrow the modality gap through designing different fine-grained consistency measurements such as multi-level attention [2] and similarity graph [8], [17]. Although promising performance has been achieved, most existing methods would suffer from performance degradation when encountering noisy correspondence as verified in the empirical results.

B. Learning With Noisy Labels

The most relevant paradigm to noisy correspondence might be learning with noisy labels (LNL) [24], [25], [26], [27], [28] which has attracted a lot of attention from both academic and industrial community. Most of the existing LNL works [29], [30], [31], [32] mainly focus on combating noisy annotations in the classification task, thus learning a robust classifier. Different from the instance-level annotation error of standard noisy labels, noisy correspondence refers to the pairwise association error in the cross-modal pairs, *i.e.*, some mis-associated pairs are wrongly regarded as positive pairs. Clearly, the significant paradigm difference prohibits the existing label noise methods from handling noisy correspondence in cross-modal retrieval. Therefore, it is desirable to develop customized methods for learning with noisy correspondence.

C. Learning With Noisy Correspondence

Noisy correspondence refers to the mismatched pairs while being wrongly treated as associated, drawing considerable attention from the community. Given that many tasks and applications require data pairs as input, customizing task-specific methods against noisy correspondence has emerged as a promising direction across numerous applications including but not limited to cross-modal retrieval [14], [15], [33], [34], person re-identification [35], [36], graph matching [37], multi-view clustering [38], [39], image-text pre-training [40], audio-visual action recognition [41], image captioning [42].

Among the aforementioned works, the most related works could be the NC-robust cross-modal retrieval ones.

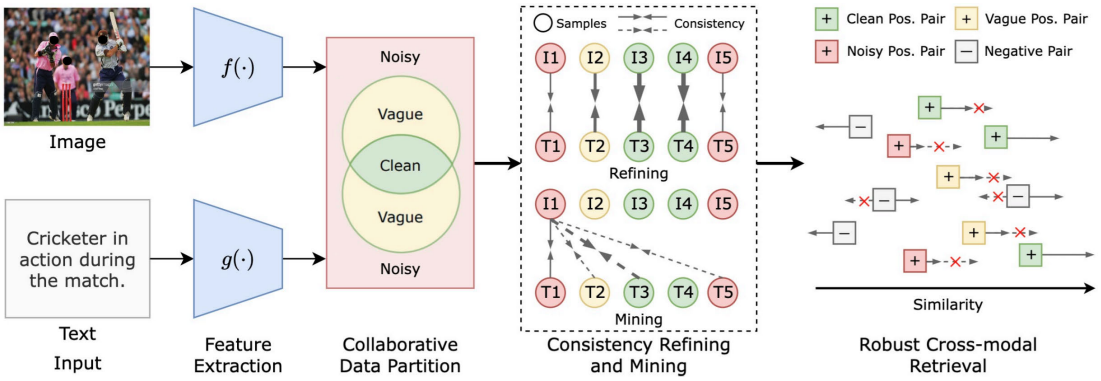


Fig. 2. Overview of the proposed CREAM (best viewed in color). Given paired image-text data, based on the memory effect of DNNs, CREAM first partitions the pairs into three types, namely, clean (green), vague (yellow), and noisy (red). After that, CREAM refines consistency for positive pairs (the strength is indicated by thickness) to alleviate the influence of NC, while mining consistency between some correlated negative pairs (denoted by dash line) to boost the CMR performance. Finally, with the recast consistency, CREAM adopts a novel contrastive loss, which modulates the gradient for different types of positive pairs and reverses the gradient for some negative pairs, leading to robust CMR against NC.

Different from them, we reveal that the correspondence is not always in accord with the consistency, especially for the false positive and false negative pairs. Based on this observation, CREAM achieves robustness against NC through consistency refining and mining. On the one hand, CREAM employs a collaborative learning paradigm to rectify the correspondence of positives, so that the overfitting on FPs is eliminated. On the other hand, CREAM explores and exploits the diverse potential consistency, thus boosting the performance.

D. Contrastive Learning

The contrastive learning paradigm has achieved state-of-the-art performance recently in representation learning [43], [44], [45], [46], [47]. The basic idea of contrastive learning is to maximize the similarities of positive pairs while minimizing those of negative ones. In single-modal contrastive learning, pairs are constructed through data augmentations, such as random crop and color distortion [43]. Samples augmented from the same instance are treated as positive, while other samples in the mini-batch [43] or memory bank [44], [48] are considered as negative. In multi-modal contrastive learning, pairs are constructed based on the correspondence between cross-modal samples [12], [13], [49], [50]. With the pairing information crawled from the Internet or annotated by humans, only paired cross-modal data is defined as positive and others are treated as negative.

The instance-level discrimination nature of contrastive learning is favored for CMR. In this work, we propose a novel contrastive loss that significantly differs from existing cross-model contrastive learning works in the following two aspects. On the one hand, considering FPs in CMR, we rectify the weight for positive pairs to alleviate the influence of noisy correspondence. On the other hand, instead of treating all unpaired samples as absolutely negative, we propose to mine the hidden associations in the unpaired data. Such an operation helps the model to capture more cross-modal consistency information, which further boosts the performance.

III. Method

In this section, we elaborate on the proposed CREAM which consists of two modules, together with a novel objective function to achieve robust cross-modal retrieval against noisy correspondence. Section III-A introduces the Collaborative Data Partition (CDP) module which divides data pairs into three subsets based on the memory effect of DNNs. Section III-B introduces the Consistency Refining and Mining (CRM) module which recast the consistency from two aspects. Section III-C details the proposed objective function to achieve Robust Cross-modal Retrieval (RCR). The framework of CREAM is shown in Fig. 2.

A. Collaborative Data Partition

We first formulate the cross-modal retrieval task as follows by taking image-text matching as an example. Given N cross-modal image-text pairs $\{(I_i, T_i), y_i\}_{i=1}^N$, cross-modal retrieval aims to build correlations between image and text samples in the unlabeled test set, where $y_i \in \{0, 1\}$ is the annotated correspondence indicating whether the i-th image I_i and i-th text T_i belong to the same instance. Cross-modal retrieval with noisy correspondence considers a more challenging setting where an unknown portion of data pairs is mismatched. Namely, some pairs (I_i, T_i) are intrinsically negative (*i.e.*, $y_i = 0$) but are wrongly labeled as positive (*i.e.*, $y_i = 1$).

To tackle such a problem, CREAM first identifies those mislabeled pairs by observing their patterns in pair similarities. Specifically, let $f(\cdot)$ and $g(\cdot)$ be the feature extractors for images and text respectively, the pair similarity is measured by the cosine distance $s(f(I_i), g(T_i))$ in the feature space, which we abbreviate as $s(i, i)$ for simplicity in the following. Some pioneer works [16] have shown that DNNs are apt to learn clean patterns first, and then gradually fit noisy ones, which is the so-called memorization effect. Motivated by this empirical finding, it is possible to distinguish clean and noisy pairs by their different patterns in losses. In this work, the loss for each image-text pair is defined by the vanilla cross-modal InfoNCE [12] as follows,

$$l_i = -\log \frac{\exp(s_{i,i}/\tau)}{\sum_{j=1}^{N} \exp(s_{i,j}/\tau)} - \log \frac{\exp(s_{i,i}/\tau)}{\sum_{j=1}^{N} \exp(s_{j,i}/\tau)}, \tag{1}$$

where $\tau = 0.07$ is the temperature parameter fixed in all our experiments.

To distinguish clean and noisy pairs, we fit the losses of all pairs by a two-component Gaussian Mixture Model [24], [51], namely,

$$P(l\,|\theta) = \alpha_1 \phi(l\,|\theta_1) + \alpha_2 \phi(l\,|\theta_2)\,, \tag{2}$$

where α_k and $\phi(l\,|\theta_k)$ denote the mixture coefficient and the probability density of the k-th component, respectively. According to the DNNs' memorization effect, the component with a smaller mean value corresponds to clean pairs, and the other component corresponds to noisy ones. The probability of pair i belonging to the clean component is calculated by the posterior probability as

$$p_i = P(\theta_k|l_i) = P(\theta_k)P(l_i|\theta_k)/P(l_i). \tag{3}$$

To improve the accuracy of pair partition, we design a collaborative partition scheme. To be specific, we construct two identical models $A = \{f_A, g_A\}$ and $B = \{f_B, g_B\}$ with different random initialization. After warming up networks A and B by minimizing Eq. 1 for all pairs, we could obtain the probability of the i-th pair being clean from each network, denoted as p_i^A, p_i^B. Further, pair i is considered clean by network A(B) if $p_i^{A(B)} > \gamma$ and vice versa, where the threshold γ is set to be 0.5 in our experiments for simplicity. Based on the agreement of two DNNs, we partition all pairs into three groups as follows:

$$(I_i, T_i) \in \begin{cases} \mathcal{D}_c, \hat{p}_i^A + \hat{p}_i^B = 2, \\ \mathcal{D}_v, \hat{p}_i^A + \hat{p}_i^B = 1, \\ \mathcal{D}_n, \hat{p}_i^A + \hat{p}_i^B = 0, \end{cases} \quad \hat{p}_i^{A(B)} = \begin{cases} 1, p_i^{A(B)} > \gamma, \\ 0, p_i^{A(B)} \le \gamma, \end{cases} \tag{4}$$

where $\mathcal{D}_c$, $\mathcal{D}_v$, and $\mathcal{D}_n$ refer to the clean, vague, and noisy set, respectively. In other words, the pair would be regarded as clean/noisy *i.f.f.* both/neither of the two DNNs agree that the pair is highly-confident. For the rest pairs, they would be treated as vague due to the disagreed judgment between the two DNNs.

B. Consistency Refining and Mining

After partitioning data pairs into three groups, we apply different strategies to recast the consistency based on the characteristics of each group. The consistency recast is conducted dually and composed of consistency rectification for positive pairs as well as consistency mining for some negative pairs.

1) Consistency Refining: Consistency refining aims to recalibrate the correspondence of false positive pairs so that their consistency could be properly embedded. For the clean split $\mathcal{D}_c$, the original correspondence is likely to be correct, and thus we slightly reduce the correspondence intensity with the clean confidence p_i. To prevent error accumulation, we swap the probability score between networks A and B, leading to the following formulation, *i.e.*,

$$\begin{aligned} y'^A_{c_i} &= p_i^B y_i + (1 - p_i^B)\hat{y}_i^A, \\ y'^B_{c_i} &= p_i^A y_i + (1 - p_i^A)\hat{y}_i^B, \end{aligned} \tag{5}$$

where $\hat{y}^A$ and $\hat{y}^B$ denote the current predictions from networks A and B respectively, which are computed by the bi-directional retrieval results as follows (similar for $\hat{y}^B$),

$$\begin{aligned} \hat{y}_i^A &= \frac{1}{2}\left[\hat{y}_{i2t}^A + \hat{y}_{t2i}^A\right] \\ &= \frac{1}{2}\left[\frac{\exp\left(s_{i,i}^A/\tau\right)}{\sum_{j=1}^N \exp\left(s_{i,j}^A/\tau\right)} + \frac{\exp\left(s_{i,i}^A/\tau\right)}{\sum_{j=1}^N \exp\left(s_{j,i}^A/\tau\right)}\right]. \end{aligned} \tag{6}$$

For the vague split $\mathcal{D}_v$, the original annotated correspondence is not as reliable as those in the clean split. Accordingly, we lower the intensity of the original correspondence by averaging of clean confidences from two DNNs, and the rectified correspondence is defined as

$$y'^{A(B)}_{v_i} = \frac{p_i^A + p_i^B}{2} y_i + \left(1 - \frac{p_i^A + p_i^B}{2}\right)\hat{y}_i^{A(B)}. \tag{7}$$

For the noisy split $\mathcal{D}_N$, since the annotated correspondence is no longer reliable, we average the current prediction from two DNNs as the rectified correspondence, namely,

$$y'^{A(B)}_{n_i} = \frac{\hat{y}_i^A + \hat{y}_i^B}{2}. \tag{8}$$

2) Consistency Mining: Besides refining consistency of positive pairs, we mine the potential consistency rooted in negative pairs and establish correspondence for them accordingly. Specifically, the consistency mining is performed within each branch of networks A and B independently. For conciseness, we omit the mark $A(B)$ in the following. The intensity of correspondence established on negative pairs is determined by the rectified value of the anchor and pair-wise similarities in the feature space. Specifically,

$$\begin{aligned} w_{i,j}^{i2t} &= (1 - y'_i)\frac{s_{i,j}}{\sum_{k=1,k\neq i}^N s_{i,k}}, \\ w_{i,j}^{t2i} &= (1 - y'_j)\frac{s_{i,j}}{\sum_{k=1,k\neq j}^N s_{k,j}}, \end{aligned} \tag{9}$$

where $y'_{i(j)}$ is the refined correspondence, $w_{i,j}^{i2t}$ and $w_{i,j}^{t2i}$ denote the consistency between cross-modal sample i and j in image-to-text and text-to-image retrieval, respectively.

Next, instead of building correspondence between all negative pairs, we sieve out those pairs with relatively low consistency. Such a filtering operation encourages the network to focus on reliable consistency, which is more likely to benefit the CMR model optimization as verified in our experiments. The threshold for filtering is designed in a data-driven manner, namely,

$$\beta = \frac{1}{N}\left(\bar{y}'_c N_c + \bar{y}'_v N_v + \bar{y}'_n N_n\right), \tag{10}$$

where N_c, N_v, and N_n denote the number of pairs in the clean, vague, and noisy split ($s.t.\ N_c + N_v + N_n = N$), respectively. In practice, we filter those negative pairs with similarity lower than the threshold β, namely,

$$\hat{w}_{i,j}^{i2t} = \begin{cases} 0, & \text{if } w_{i,j}^{i2t} < \beta, \\ w_{i,j}^{i2t}, & \text{else}, \end{cases} \tag{11}$$

and $\hat{w}_{i,j}^{t2i}$ is filtered similarly, namely, the finally established correspondence for negative pairs. Such a design could adaptively adapt to different ratios of noisy correspondence. Concretely, when the noise ratio is low, most of the cross-modal sample pairs would be regarded as clean, and thus the computation of Eq. 10 would be dominated by pairs in the clean split. Accordingly, the filtering threshold would be high, which could tighten the correspondence establishment. On the contrary, under a high noise rate, the filtering threshold would be reduced by noisy pairs. As a result, a low threshold would allow the model to mine more potential consistency in the negative bank to complement the lost consistency caused by noisy pairs. For a comprehensive understanding of our adaptive filtering threshold, we present some analytical studies on the experiments.

C. Robust Cross-Modal Retrieval

Given the recast consistency, *i.e.*, rectified correspondence for positive pairs and newly established correspondence for negative pairs, we propose the following objective function for robust cross-modal retrieval, *i.e.*,

$$\begin{aligned} L &= L_p + \frac{1}{2}\left[L_n^{i2t} + L_n^{t2i}\right] \\ &= \frac{1}{N}\sum_{i=1}^{N} y_i' l_i + \frac{1}{2N}\left[\sum_{\substack{i=1 \\ i\neq j}}^{N} -\hat{w}_{i,j}^{i2t} \log \frac{\exp\left(s_{i,j}/\tau\right)}{\sum_{k=1}^{N}\exp\left(s_{i,k}/\tau\right)}\right. \\ &\quad \left. + \sum_{\substack{j=1 \\ j\neq i}}^{N} -\hat{w}_{i,j}^{t2i} \log \frac{\exp\left(s_{i,j}/\tau\right)}{\sum_{k=1}^{N}\exp\left(s_{k,j}/\tau\right)}\right], \end{aligned} \tag{12}$$

where L_p, L_n^{i2t}, and L_n^{t2i} denote the loss for refined positive pairs and bi-directional negative pairs in cross-modal retrieval, respectively. With the above loss design, the model could alleviate the influence of false positive pairs and exploit the potential consistency rooted in some negative pairs, leading to robust cross-modal retrieval against noisy correspondence.

IV. Experiments

In this section, we verify the robustness of CREAM against the instance-level image-text NC and the more fine-grained patch-level NC. To this end, we conduct extensive experiments on both the image-text retrieval and graph matching tasks across six benchmarks. The organization of this section is as follows. In Section IV-A, we elaborate on the experiment settings including datasets and implementation details. In Section IV-B, we carry out extensive experiments on three benchmarks to evaluate the effectiveness of CREAM. In Section IV-C, we conduct detailed ablation studies to investigate the effects of each module. In Section IV-D, we perform a series of analytical experiments to give a comprehensive understanding of CREAM. In Section IV-E, we extend CREAM to the graph matching task and verify its effectiveness on handling the fine-grained patch-level NC. Due to the space limitation, we present more results in the Appedix.

A. Experiment Settings

In this section, we elaborate on the experiment settings including the used datasets and implementation details.

1) Datasets: The detailed descriptions of the used datasets are presented as follows.

- **Conceptual Captions 3M** [1]: This is a large-scale web-harvested dataset consisting of approximately 3.3M image-caption pairs [1]. In the experiment, following [14], we use a randomly-selected subset of Conceptual Captions 3M for evaluation, named CC152K. CC152K contains 150K pairs for training, 1K pairs for validation, and 1K pairs for testing.
- **Flickr30K** [52]: The dataset contains 31K images collected from the Flickr website. Each image has 5 manually annotated captions. As a result, there are 155K image-text pairs in the datasets. Following [2], we use 5K pairs for validation, 5K pairs for testing, and 145K pairs for training.
- **MS-COCO** [53]: The dataset consists of 123,287 images, and each image is annotated with 5 text descriptions. Therefore, there are 616,435 image-text pairs in the dataset, which is split into 566,435 pairs for training, 25K pairs for validation (as it is slow to validate on 25K pairs, only 5K pairs are used in all experiments), and the rest 25K for testing. Following [2], we use two kinds of evaluation protocols, namely, 5 fold of 1K test images and full 5K test images. The results are reported by either averaging over 5 folds of 1K test images (denoted by MS-COCO 1K) or testing on the full 5K test images (denoted by MS-COCO 5K).
- **SPair-71k** [54]: The dataset comprises 70,958 image pairs covering a total of 18 classes. These image pairs exhibit diverse variations in viewpoint thus suffering from noisy correspondence between keypoints.
- **Pascal VOC** [55]: The dataset comprises 7,020 images allocated for training and 1,682 for testing, covering a total of 20 classes. Each image contains a varying number of keypoints, ranging between 6 and 23.
- **Willow Object** [56]: The dataset includes 256 images across 5 categories, each annotated with 10 distinctive landmarks. Follow [37], we train our CREAM on the initial 20 images and test on the remaining set.

Our primary focus lies in validating CREAM's effectiveness within the image-text retrieval task, leaving the exploration of its extension to the graph matching task in the last. For extensive evaluations, we conduct experiments on both simulated and real-world NC-contaminated datasets. Specifically, for the well-annotated Flickr30K [52] and MS-COCO [53] datasets, following NCR [14], we simulate the NC by randomly shuffling the text of training images in a specific percentage, which is denoted as noise ratio. As for CC152K [14], it is reported to have 3% – 20% mismatched pairs (*i.e.*, NC off-the-shelf) since the data pairs are harvested from the Web [1]. Following [15], we use the widely-used CMR metrics, *i.e.*, Recall@1 (R@1), Recall@5 (R@5), Recall@10 (R@10), and their sum (denoted as R-sum) for the performance measurement.

2) Implementation Details: The proposed CREAM is a generalized NC-robust framework that could be adapted to most

existing CMR models. In the main experiments, we endow the recently-proposed CMR baseline, SGR [8], with robustness against NC. In brief, we maintain the backbone of SGR and train it using the proposed framework. Specifically, following [2] and [8], the off-the-shelf Faster-RCNN [57] is used to extract the features of each image and obtain 36 feature vectors of regions of interest (ROI) for backbone training, and each vector is with 2048 dimensions. Each caption is first processed by word embedding with the size of 300, then fed into Bi-GRU [58] whose hidden state number is 1024.

In our implementation, we first randomly initialize two SGR models and warm up the models using Eq. 1 for better network initialization. Notably, the existing NC-oriented methods [14], [15] also adopt the warm-up strategy while the warm-up epoch varies from different datasets or noise ratios. In this work, the warm-up stage would continue as long as all the metric values on the validation set are increasing, and the maximum warm-up epoch is set as 5. Clearly, our strategy could avoid the labor-intensive tuning on the warm-up epochs. After the warm-up stage, the ROIs features of images and features of captions are fed into both two models for training. The models are trained under the proposed framework, *i.e.*, CREAM with a batch size of 128, and both models share the same data within a single batch. For better data partition, we first use the divided clean data only, and then gradually add the divided vague and noisy data as the training proceeds. For network parameter updating, we use the Adam optimizer [59] with default parameters.

To ensure the practicability of our CREAM, we use the final checkpoints for evaluation instead of using the best checkpoints in the validation set. In the inference stage, we average the predictions of model A and model B as the final prediction for evaluation. All the experiments and evaluations are performed on Ubuntu OS with GeForce RTX 3090 GPUs.

B. Comparisons With State of the Arts

To verify the effectiveness of CREAM, we compare CREAM with six image-text retrieval baselines including SCAN [2], IMRAM [23], SAF [8], SGR [8], NCR [14], and DECL [15]. Among them, the former four baselines are the standard CMR baselines, while NCR and DECL are the existing NC-robust CMR methods. For comprehensive comparisons, besides the results on the CC152K dataset, we vary the noise ratio of Flickr30K and MS-COCO datasets from 20% to 80% with an interval of 20% to simulate more NC scenarios and report the results. The results on the CC152K, Flickr30K, MS-COCO 1K are summarized in Tables I, II and III, respectively. According to the results, one could have the following observations and conclusions. First, with the increasing noise ratio, our CREAM performs relatively stable, whereas the standard CMR baselines encounter remarkable performance degradation, verifying the necessity of developing the NC-robust CMR method. Second, compared to existing NC-robust methods (DECL [15], NCR [14]), our CREAM still achieves promising performance improvement. For example, on the real-world NC-contaminated dataset (CC152K), CREAM improves the "R-Sum" by 4.8% and 7.3% compared to NCR and DECL, respectively. On the simulated NC-contaminated MS-COCO 1K dataset, CREAM achieves absolute improvements of $+5.7, +2.9, +6.6, +10.2$ on "R-Sum" when the noise ratio varies in the range of $[20\%, 40\%, 60\%, 80\%]$, comparing to the best baseline.

TABLE I

EXPERIMENT RESULTS ON CC152K. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN **BOLD** AND UNDERLINE, RESPECTIVELY

Method	CC152K						
	Image to text			Text to image			R-Sum
	R@1	R@5	R@10	R@1	R@5	R@10	
SCAN (ECCV'18)	30.5	55.3	65.3	26.9	53.0	64.7	295.7
IMRAM (CVPR'20)	33.1	57.6	68.1	29.0	56.8	67.4	312.0
SAF (AAAI'21)	31.7	59.3	68.2	31.9	59.0	67.9	318.0
SGR (AAAI'21)	35.0	63.4	73.3	34.9	63.0	72.8	342.4
NCR (NeurIPS'21)	39.5	64.5	73.5	**40.3**	64.6	73.2	355.6
DECL (ACMMM'22)	36.2	63.6	73.2	37.1	63.6	73.7	347.4
Ours	**40.3**	**68.5**	**77.1**	40.2	**68.2**	**78.3**	**372.6**

TABLE II

EXPERIMENT RESULTS ON FLICKR30K

Noise	Method	Flickr30K						
		Image to text			Text to image			R-Sum
		R@1	R@5	R@10	R@1	R@5	R@10	
20%	SCAN (ECCV'18)	56.4	81.7	89.3	34.2	65.1	75.6	402.3
	IMRAM (CVPR'20)	36.0	67.8	78.8	23.1	51.7	63.7	321.1
	SAF (AAAI'21)	51.8	79.5	88.3	38.1	66.8	76.6	401.1
	SGR (AAAI'21)	61.2	84.3	91.5	44.5	72.1	80.2	433.8
	NCR (NeurIPS'21)	76.7	93.9	96.9	57.5	82.8	89.2	497.0
	DECL (ACMMM'22)	75.1	93.6	96.7	56.2	82.4	88.5	492.5
	Ours	**77.4**	**95.0**	**97.3**	**58.7**	**84.1**	**89.8**	**502.3**
40%	SCAN (ECCV'18)	29.9	60.5	72.5	16.4	38.5	48.6	266.4
	IMRAM (CVPR'20)	23.5	53.9	65.8	16.9	41.0	53.2	254.2
	SAF (AAAI'21)	34.3	65.6	78.4	30.1	58.0	68.5	335.0
	SGR (AAAI'21)	47.2	76.4	83.2	34.5	60.3	70.5	372.1
	NCR (NeurIPS'21)	75.3	92.1	95.2	56.2	80.6	87.4	486.8
	DECL (ACMMM'22)	72.2	91.9	96.0	53.8	80.1	87.0	481.0
	Ours	**76.3**	**93.4**	**97.1**	**57.0**	**82.6**	**88.7**	**495.1**
60%	SCAN (ECCV'18)	16.9	39.3	53.9	2.8	7.4	11.4	131.7
	IMRAM (CVPR'20)	14.9	38.1	52.9	11.5	31.4	44.3	193.1
	SAF (AAAI'21)	28.3	54.5	67.5	22.1	47.3	59.0	278.7
	SGR (AAAI'21)	28.7	58.0	71.0	23.8	49.5	60.7	291.7
	NCR (NeurIPS'21)	68.7	89.9	95.5	52.0	77.6	84.9	468.6
	DECL (ACMMM'22)	69.1	90.3	94.9	50.4	76.7	84.6	466.0
	Ours	**70.6**	**91.2**	**96.1**	**53.3**	**79.2**	**87.0**	**477.4**
80%	SCAN (ECCV'18)	5.1	18.1	27.3	3.9	13.1	19.1	86.6
	IMRAM (CVPR'20)	8.5	26.7	38.8	7.1	20.7	30.8	132.5
	SAF (AAAI'21)	12.2	32.8	48.4	11.8	30.5	41.5	177.2
	SGR (AAAI'21)	13.7	35.1	47.6	12.1	30.9	41.9	181.3
	NCR (NeurIPS'21)	1.4	7.1	11.7	1.5	5.4	9.3	36.4
	DECL (ACMMM'22)	55.4	80.4	88.0	37.6	64.9	74.8	401.1
	Ours	**56.1**	**81.2**	**88.4**	**39.2**	**66.7**	**76.2**	**407.8**

TABLE III

EXPERIMENT RESULTS ON MS-COCO 1K

Noise	Method	MS-COCO 1K						
		Image to text			Text to image			R-Sum
		R@1	R@5	R@10	R@1	R@5	R@10	
20%	SCAN (ECCV'18)	28.9	64.5	79.5	20.6	55.6	73.5	322.6
	IMRAM (CVPR'20)	39.1	76.9	88.9	33.1	66.9	79.8	384.7
	SAF (AAAI'21)	41.0	78.4	89.4	38.2	74.0	85.5	406.5
	SGR (AAAI'21)	49.1	83.8	92.7	42.5	77.7	88.2	434.0
	NCR (NeurIPS'21)	77.0	95.6	98.1	61.5	89.3	95.1	516.6
	DECL (ACMMM'22)	77.1	95.9	98.4	61.6	89.1	95.2	517.3
	Ours	**78.9**	**96.3**	**98.6**	**63.3**	**90.1**	**95.8**	**523.0**
40%	SCAN (ECCV'18)	30.1	65.2	79.2	18.9	51.1	69.9	314.4
	IMRAM (CVPR'20)	32.4	68.5	82.2	30.2	64.9	79.9	358.1
	SAF (AAAI'21)	36.0	74.4	87.0	33.7	69.4	82.5	383.0
	SGR (AAAI'21)	43.9	78.3	89.3	37.0	72.8	85.1	406.4
	NCR (NeurIPS'21)	**76.5**	95.0	98.2	60.7	88.5	95.0	513.9
	DECL (ACMMM'22)	75.6	95.0	98.2	59.8	88.2	94.7	511.4
	Ours	**76.5**	**95.6**	**98.3**	**61.7**	**89.4**	**95.3**	**516.8**
60%	SCAN (ECCV'18)	27.8	59.8	74.8	16.8	47.8	66.4	293.4
	IMRAM (CVPR'20)	28.1	62.7	78.5	26.6	59.8	75.1	330.8
	SAF (AAAI'21)	28.2	63.9	79.4	31.1	65.6	80.5	348.7
	SGR (AAAI'21)	37.6	73.3	86.3	33.8	68.6	81.7	381.3
	NCR (NeurIPS'21)	72.7	94.0	97.6	57.9	87.0	94.1	503.3
	DECL (ACMMM'22)	63.4	90.0	95.7	49.6	81.8	91.0	471.4
	Ours	**74.7**	**94.8**	**98.0**	**59.7**	**88.0**	**94.6**	**509.9**
80%	SCAN (ECCV'18)	22.2	51.9	67.5	13.8	41.1	58.6	255.1
	IMRAM (CVPR'20)	21.5	53.0	68.9	20.4	51.0	67.4	282.2
	SAF (AAAI'21)	24.2	57.5	74.1	24.7	57.1	73.0	310.6
	SGR (AAAI'21)	26.7	60.7	75.6	25.3	58.2	72.6	319.1
	NCR (NeurIPS'21)	21.6	52.6	67.6	15.1	38.1	49.8	244.8
	DECL (ACMMM'22)	65.7	91.2	96.1	51.8	82.7	91.1	478.5
	Ours	**68.6**	**92.0**	**96.4**	**54.3**	**84.8**	**92.5**	**488.7**

TABLE IV
ABLATION STUDIES FOR CREAM ON FLICKR30K WITH 40% NOISE. THE DEFAULT SETTINGS ARE MARKED IN GRAY

Method Variants				Image to text		Text to image		R-Sum
CDP	CRM	RCR	WarmUp	R@1	R@10	R@1	R@10	
✓	✓	✓	✓	**76.3**	**97.1**	**57.0**	**88.7**	**495.1**
	✓	✓	✓	75.7	96.4	56.4	88.2	490.4
✓		✓	✓	73.7	96.2	56.5	88.5	490.0
✓	✓		✓	75.0	96.8	55.5	87.5	488.7
✓	✓	✓		73.6	96.2	55.5	88.1	487.9
			✓	68.2	94.4	50.1	77.8	453.8

TABLE V
FINE-GRAINED ABLATION STUDIES ON THE CDP MODULE

CDP Variants	Image to text		Text to image		R-Sum
	R@1	R@10	R@1	R@10	
Clean+Vague+Noisy	**76.3**	97.1	**57.0**	88.7	**495.1**
Clean Only	74.6	96.8	56.1	88.5	491.1
Clean (with Vague) + Noisy	75.0	96.5	56.4	88.6	491.0
Clean + Noisy (with Vague)	74.8	**97.2**	56.9	**88.9**	493.9
Clean+Vague+Noisy (SelfTraining)	70.1	94.8	51.2	82.9	466.7

TABLE VI
FINE-GRAINED ABLATION STUDIES ON THE CRM MODULE

CRM Variants				Image to text		Text to image		R-Sum
y'_c	y'_v	y'_n	$\hat{w}$	R@1	R@10	R@1	R@10	
Eq. 5	Eq. 6	Eq. 7	Eq. 11	**76.3**	**97.1**	57.0	**88.7**	**495.1**
Eq. 5	Eq. 6	Eq. 7	LS	75.7	96.8	**57.6**	88.4	492.9
Eq. 5	Eq. 6	Eq. 7	0	75.9	96.4	56.2	88.3	492.6
Eq. 5	Eq. 6	Eq. 7	w/o Eq. 11	73.7	96.8	56.0	88.2	489.1
1	1	1	Eq. 11	75.4	96.9	56.1	88.5	492.4
1	1	0	Eq. 11	73.7	96.2	56.5	88.5	490.0
1	0	0	Eq. 11	74.0	96.5	55.7	88.6	490.3

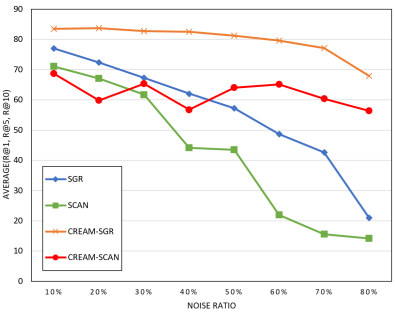


Fig. 3. Retrieval performance of CREAM on the Flickr30K dataset when adapted to SGR and SCAN with varying noise ratio.

C. Ablation Studies

In this section, we first perform a standard ablation study to investigate the importance of each component. Then, we conduct comprehensive fine-grained ablation studies to investigate the effects of the CDP and CRM modules. All the ablation studies are conducted on the Flickr30K dataset with 40% noise.

1) Ablation on Each Module of CREAM: We conduct experiments on the following variants of CREAM by isolating the corresponding module: i) we remove the CDP module, *i.e.*, all the data pairs are regarded as clean ones; ii) we remove the CRM module, *i.e.*, the correspondence of clean and vague pairs are set to be 1 while those of noisy pairs are set to be 0; iii) we replace our loss (Eq. 12) with the loss of NCR [14]; iv) we remove the warm-up procedure; v) we train models with the vanilla contrastive loss (Eq. 1) only while removing the three modules. Table IV summarizes the results and indicates the inseparability of each component.

2) The Effect of the CDP Module: To investigate the effect of the CDP module, we perform the following CDP variants: i) Using the divided clean subsets for training only; ii) Merging the clean and vague subsets; iii) Merging the noisy and vague subsets; iv) Employing a single neural network for partition only, *i.e.*, self-training. The results are summarized in Table V, where one could see that our dedicated partition strategy is more favorable for achieving NC-robust CMR.

3) The Effect of the CRM Module: We first investigate the effect of the consistency mining scheme (Section III-B.2) by conducting the following variants: i) we replace the consistency mining scheme with the vanilla label smooth (LS) strategy; ii) we remove the consistency mining scheme; iii) we perform CRM without the filter operation (denoted as w/o Eq. 11). Moreover, we investigate the effect of the consistency refining scheme (Section III-B.1) by adopting different rectification strategies. From Table VI, one could realize the importance of our CRM module. On the one hand, the performance could be significantly boosted, once the correspondence of pairs with potential consistency is properly established in the negative bank. On the other hand, it is necessary to design the dedicated consistency refining scheme according to the characteristic of each subset.

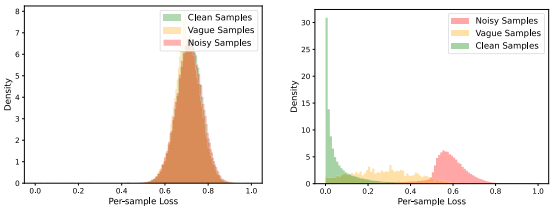
D. Analytical Experiments

In this section, we first show the robustness and generalizability of CREAM. After that, we visually investigate the effectiveness of our CDP and CRM modules. Finally, we present some false positive pairs in the dataset and pairs with diverse potential consistency in the negative bank detected by CREAM.

1) Robustness and Generalizability: To inverstigate the robustness of CREAM, we conduct CREAM and its baseline SGR [8] by varying the noise ratio from 10% to 80% with an interval of 10%. Furthermore, to show the generalizability of CREAM, we adapt it to another CMR baseline SCAN [2] to evaluate the robustness against NC. Fig. 3 depicts that both SGR and SCAN encounter a severe performance drop as the noise ratio increases. In contrast, CREAM could endow the two baselines with robustness against NC, demonstrating the robustness and generalizability of the proposed CREAM.

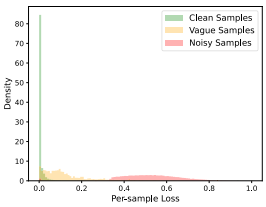
2) Effectiveness of the CDP and CRM Modules: To investigate how our CRM module helps to achieve robust cross-modal retrieval, we visualize the per-sample loss distribution. The results are shown in Fig. 4(a)-(c) One could observe that after warmup, the losses for clean, vague, and noisy samples show different patterns, which proves the effectiveness of our CDP module. After training, the model successfully fits clean and vague samples to different extents, while not being influenced by those noisy samples. Such a result indicates that CRM achieves robustness against NC.

Noticed that in consistency mining, we design an adaptive filter to automatically handle different noisy ratios. To illustrate how the filter works, we visualize the weight distribution for samples in the negative bank and the corresponding filtering thresholds in Fig. 4(d). As shown, the weights are generally larger under a lower noise ratio, since the model

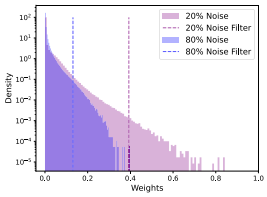


(a) Initialization

(b) After Warmup



(c) After Training



(d) Adaptive Filter

Fig. 4. (a)-(c): The per-sample loss distribution on the Flickr30K dataset with 40% noise ratio across the training process. (d) The weight distribution and the computed thresholds on the Flickr30K dataset under different noise ratios.

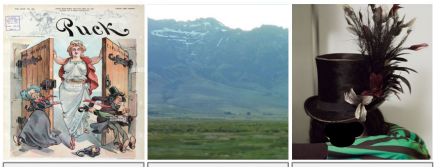
TABLE VII

PERFORMANCE ON THE FLICKR30K DATASET EMPLOYING THE STATE-OF-THE-ART VISION TRANSFORMER BACKBONE

Flickr30K Noise	Method	Image to text R@1	Image to text R@10	Text to image R@1	Text to image R@10	R-Sum
20%	EVA02+SGR	53.5	91.7	43.6	80.5	425.7
	EVA02+NCR	71.7	96.3	57.8	**89.4**	489.4
	EVA02+Ours	**75.2**	**96.9**	**59.9**	89.1	**499.0**
40%	EVA02+SGR	39.8	81.2	31.1	69.3	353.6
	EVA02+NCR	68.8	95.5	55.1	**87.7**	479.6
	EVA02+Ours	**70.8**	**96.1**	**56.5**	87.5	**484.4**
60%	EVA02+SGR	28.4	68.0	19.6	54.7	271.2
	EVA02+NCR	61.6	93.5	49.1	**84.3**	452.7
	EVA02+Ours	**65.1**	**94.9**	**50.6**	83.7	**459.5**
80%	EVA02+SGR	11.6	46.5	9.9	39.5	167.3
	EVA02+NCR	7.0	32.1	1.1	6.7	73.0
	EVA02+Ours	**47.4**	**86.1**	**35.7**	**73.8**	**383.1**

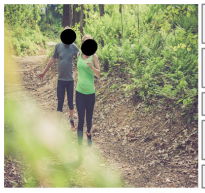
learns better cross-view consistency from more true positive pairs. Accordingly, CREAM computes a stricter threshold following Eq.10. The results indicate that CREAM could always mine the most reliable consistency in the negative bank under different noise ratios.

3) Compatibility of CREAM Across Various Feature Extractors: To investigate the compatibility of CREAM with different extractors, in this section, we use the state-of-the-art vision transformer model, EVA [60], [61], [62], as the visual feature extractor. Specifically, we replace the default Faster-RCNN extractor of SGR with the EVA model, and perform experiments using SGR, NCR and CREAM. From Table VII, one could observe that our CREAM still achieve superior performance than NCR and SGR, implying the compatibility of cream across different feature extractors. Note that, using the EVA model would slightly decrease the performance compared to the Faster-RCNN counterpart although EVA is the SOTA backbone for image classification. The reason could be attributed to the prior information acquired by the Faster-RCNN. More specifically, Faster-RCNN could extract the fine-grained object information which would benefit the cross-modal semantic similarity measurement. In contrast, EVA could only extract the coarse-grained semantic information.



We grow wiser as we grow older. **(0.1246)** | See? That's snow in august ! **(0.0089)** | Visited me in a suit of feathers . **(0.0007)**

(a) Refined Consistency



1. Young woman and elderly man running in the forest. **(GT, 0.9988)**
2. Girl running in forest in the morning. **(0.9959)**
3. Dirt road through the forest. **(0.8794)**
4. Walking path between trees in the park. **(0.7932)**
5. A path to the waterfall. **(0.1497)**

(b) Mined Consistency

Fig. 5. Case study. (a) Consistency refined for FPs. (b) Consistency mined from the negative bank. Values denote the consistency predicted by our model.

TABLE VIII

KEYPOINT MATCHING AVERAGE ACCURACY FOR ALL CLASSES (%)

Method	Willow	Pascal VOC	SPair-71k
GMN [63]	79.3	62.4	65.3
NGM [64]	85.3	64.1	68.9
PCA [65]	87.4	64.8	66.0
CIE [66]	89.0	68.9	73.3
IPCA [67]	90.1	67.7	71.2
ASAR [68]	94.2	81.1	83.1
BBGM [69]	97.2	79.0	82.1
NGM-v2 [64]	97.5	80.1	80.2
COMMON [37]	**99.1**	**82.7**	84.5
Ours	98.8	81.4	**85.1**

4) Case Study: To give a more intuitive understanding of the necessity of consistency refining and mining, we provide some cases from CC152K [14] in Fig. 5. On the one hand, some image-text pairs are wrongly matched (*i.e.*, FPs). CREAM correctly detects those pairs and refines their consistency to prevent them from misleading the optimization. On the other hand, despite the annotated pair, we find that there exists diverse potential consistency in the negative bank (*e.g.*, partial consistency in text 2–4, and implicit consistency in text 5). Mining that hidden consistency could further boost CREAM's performance in cross-modal retrieval.

E. Extension to the Graph Matching Task

Graph matching aims to establish the fine-grained correspondence between the keypoints of given semantic-relevant images/graphs. As pointed by COMMON [37], it is inevitable to wrongly annotate the key point, resulting in noisy correspondence between keypoints. In this section, we investigate the effectiveness of our CREAM in handling such fine-grained noisy correspondence challenge. To this end, we choose the SOTA graph matching method, *i.e.*, COMMON, as our baseline model. More specifically, we keep the backbone of COMMON and train it using our framework. Table VIII summarize the main comparison results between nice SOTA graph matching baselines with our CREAM. More comprehensive results could be accessed in the Appendix. As the results suggested, CREAM performs competitively or even achieves promising performance improvement compared to the existing graph matching baselines, although CREAM is not dedicatedly designed for this task. The performance superiority showcase

TABLE IX
EXPERIMENT RESULTS ON MS-COCO 5K

Noise	Method	MS-COCO 5K						R-Sum
		Image to text			Text to image			
		R@1	R@5	R@10	R@1	R@5	R@10	
20%	SCAN (ECCV'18)	11.6	32.2	44.8	7.3	23.5	35.9	155.4
	IMRAM (CVPR'20)	17.0	44.4	59.4	15.6	38.0	50.8	225.1
	SAF (AAAI'21)	17.7	46.1	61.7	18.7	43.9	58.1	246.2
	SGR (AAAI'21)	23.6	54.6	69.4	22.0	48.8	62.3	280.8
	NCR (NeurIPS'21)	55.0	82.2	90.7	39.6	68.8	79.8	416.1
	DECL (ACMMM'22)	57.3	83.3	90.7	40.0	69.1	79.8	420.1
	Ours	**57.6**	**84.1**	**91.6**	**41.4**	**71.1**	**81.2**	**427.0**
40%	SCAN (ECCV'18)	12.5	33.1	46.0	6.7	21.1	32.5	151.9
	IMRAM (CVPR'20)	13.5	34.9	49.5	13.6	34.6	47.4	193.5
	SAF (AAAI'21)	13.9	40.4	56.4	15.7	39.0	52.4	217.8
	SGR (AAAI'21)	21.1	48.7	63.0	17.8	42.5	56.0	249.0
	NCR (NeurIPS'21)	**55.5**	82.2	89.8	39.5	68.3	79.1	414.4
	DECL (ACMMM'22)	53.4	81.4	89.4	38.6	67.2	78.3	408.3
	Ours	55.3	**82.3**	**90.6**	**39.8**	**69.3**	**80.1**	**417.3**
60%	SCAN (ECCV'18)	10.8	30.0	42.4	5.6	18.7	29.5	136.9
	IMRAM (CVPR'20)	10.7	30.8	44.2	11.6	30.4	42.6	170.3
	SAF (AAAI'21)	10.1	29.7	44.6	13.8	35.3	48.2	181.8
	SGR (AAAI'21)	16.5	40.4	55.5	15.6	38.9	52.2	219.0
	NCR (NeurIPS'21)	49.9	78.5	87.9	36.1	65.4	76.5	394.3
	DECL (ACMMM'22)	39.1	69.1	80.5	28.4	56.4	68.6	342.0
	Ours	**52.1**	**80.4**	**89.0**	**37.8**	**66.9**	**78.0**	**404.3**
80%	SCAN (ECCV'18)	7.3	23.1	34.1	4.5	15.3	24.4	108.7
	IMRAM (CVPR'20)	7.1	22.8	34.6	8.2	22.9	33.9	129.6
	SAF (AAAI'21)	8.8	25.5	38.7	10.2	28.3	39.8	151.3
	SGR (AAAI'21)	9.8	28.5	42.8	10.7	29.5	41.5	162.8
	NCR (NeurIPS'21)	7.4	23.7	34.8	6.0	17.4	25.5	114.8
	DECL (ACMMM'22)	42.5	72.6	82.9	30.5	58.7	70.7	357.8
	Ours	**46.5**	**74.9**	**84.3**	**32.5**	**61.5**	**73.2**	**372.8**

the generality of CREAM from the instance-level image-text NC to the patch-level keypoint NC.

V. CONCLUSION

In this paper, we study a practical but less-touched problem in cross-modal retrieval and graph matching tasks, *i.e.*, noisy correspondence. To learning with noisy correspondence, we propose CREAM that achieves robustness through the consistency rectifying and mining paradigm. Extensive experiments on both the image-text retrieval and graph matching tasks across multiple benchmarks verify the effectiveness of CREAM in handling both the instance-level NC and fine-grained patch-level NC. In the future, we plan to extend our observation and method to other multi-modal applications such as video analysis, image captioning, and so on.

APPENDIX

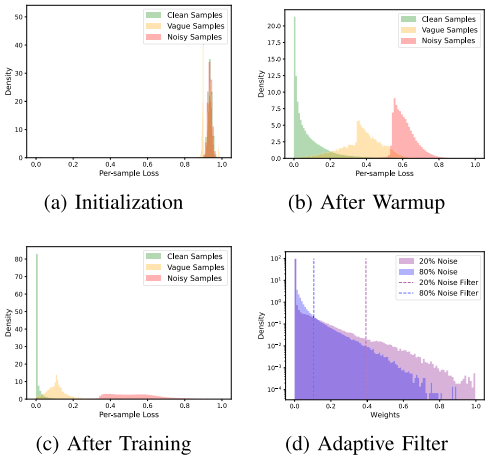
In the appendix, we present more experiments results to provide comprehensive evaluations of our method. The results includes the experiments on the setting of MS-COCO 5K, more analytical experiments, more ablation experiments, more case studies, and the comprehensive comparison results of the graph matching task.

A. Experiment Results on MS-COCO 5K

We have compared CREAM with the state-of-the-art methods on Flickr30K, CC152K and MS-COCO 1K in our manuscript (Section IV-B). Here, we show the experiment results on MS-COCO 5K with noise ratio varying in the range of [20%, 40%, 60%, 80%]. From Table IX, one could see that CREAM has improved R-Sum under different noise ratios.

B. More Analytical Experiments

To verify the effectiveness of the CDP and CRM modules, we have conducted experiments on the Flickr30K dataset in the manuscript (Section IV-D.2). Here, we give more experiment results on MS-COCO under the same experiment settings. The results are visualized in Fig. 6.



(a) Initialization (b) After Warmup

(c) After Training (d) Adaptive Filter

Fig. 6. (a)-(c): The per-sample loss distribution on MS-COCO dataset with 40% noise ratio across the training process. (d) The weight distribution and the computed thresholds on MS-COCO under different noise ratios.

TABLE X
FINE-GRAINED ABLATION STUDIES ON THE FLICKR30K DATASET USING SINGLE GMM FOR DATA PARTITION

Flickr30K Noise	Method	Image to text		Text to image		R-Sum
		R@1	R@10	R@1	R@10	
20%	Ours*	75.8	96.3	55.8	87.5	489.1
	Ours	**77.4**	**97.3**	**58.7**	**89.8**	**502.3**
40%	Ours*	70.1	94.8	51.2	82.9	466.7
	Ours	**76.3**	**97.1**	**57.0**	**88.7**	**495.1**
60%	Ours*	62.8	91.0	43.9	75.7	427.9
	Ours	**70.6**	**96.1**	**53.3**	**87.0**	**477.4**
80%	Ours*	52.6	85.6	35.5	68.8	381.3
	Ours	**56.1**	**88.4**	**39.2**	**76.2**	**407.8**

TABLE XI
KEYPOINT MATCHING ACCURACY (%) ACROSS ALL OBJECTS ON WILLOW OBJECT

Method	Car	Duck	Face	Mbike	Wbottle	Mean
GMN [63]	67.9	76.7	99.8	69.2	83.1	79.3
NGM [64]	84.2	77.6	99.4	76.8	88.3	85.3
PCA [65]	87.6	83.6	**100**	77.6	88.4	87.4
CIE [66]	85.8	82.1	99.9	88.4	88.7	89.0
IPCA [67]	90.4	88.6	**100**	83.0	88.3	90.1
ASAR [68]	92.5	84.0	**100**	95.4	99.0	94.2
BBGM [69]	96.8	89.9	**100**	99.8	99.4	97.2
NGM-v2 [64]	97.4	93.4	**100**	98.6	98.3	97.5
COMMON [37]	97.6	**98.2**	**100**	**100**	99.6	**99.1**
Ours	**97.7**	96.3	**100**	**100**	**99.8**	98.8

From Fig. 6(a)-(c), one could see that after networks initialized, the clean, vague and noisy samples are mixed up. Then, after warmup, the CDP module could divide samples into three components to some degree according to those two neural networks. After training, noisy samples are far away from clean and vague samples and most of clean samples are well learned. Those two networks are not influenced by noisy samples, which shows the effectiveness of CRM module.

Noticed that we have designed an adaptive filter β to help the network select more reliable consistency, which would benefit the optimization of CMR module. As shown in Fig. 6(d), we visualize the distribution of weights and the corresponding filtering thresholds on MS-COCO. The behavior of our method on MS-COCO is consistent with that on Flickr30K. The results indicate that CREAM could always using the most reliable consistency for optimization.

C. More Ablations

To further investigate the effect of our collaborative data partition module, we perform more fine-grained ablation

TABLE XII

KEYPOINT MATCHING ACCURACY (%) ON PASCAL VOC WITH STANDARD INTERSECTION FILTERING

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	Tv	Mean
GMN [63]	41.6	59.6	60.3	48.0	79.2	70.2	67.4	64.9	39.2	61.3	66.9	59.8	61.1	59.8	37.2	78.2	68.0	49.9	84.2	91.4	62.4
PCA [65]	49.8	61.9	65.3	57.2	78.8	75.6	64.7	69.7	41.6	63.4	50.7	67.1	66.7	61.6	44.5	81.2	67.8	59.2	78.5	90.4	64.8
NGM [64]	50.1	63.5	57.9	53.4	79.8	77.1	73.6	68.2	41.1	66.4	40.8	60.3	61.9	63.5	45.6	77.1	69.3	65.5	79.2	88.2	64.1
IPCA [67]	53.8	66.2	67.1	61.2	80.4	75.3	72.6	72.5	44.6	65.2	54.3	67.2	67.9	64.2	47.9	84.4	70.8	64.0	83.8	90.8	67.7
CIE [66]	52.5	68.6	70.2	57.1	82.1	77.0	70.7	73.1	43.8	69.9	62.4	70.2	70.3	66.4	47.6	85.3	71.7	64.0	83.9	91.7	68.9
BBGM [69]	61.9	71.1	79.7	79.0	87.4	**94.0**	89.5	80.2	56.8	79.1	64.6	78.9	76.2	75.1	65.2	98.2	77.3	77.0	94.9	93.9	79.0
NGM-v2 [64]	61.8	71.2	77.6	78.8	87.3	93.6	87.7	79.8	55.4	77.8	89.5	78.8	80.1	79.2	62.6	97.7	77.7	75.7	96.7	93.2	80.1
ASAR [68]	62.9	74.3	79.5	**80.1**	89.2	**94.0**	88.9	78.9	58.8	79.8	88.2	78.9	79.5	77.9	64.9	98.2	77.5	77.1	98.6	93.7	81.1
COMMON [37]	65.6	75.2	80.8	79.5	89.3	92.3	**90.1**	**81.8**	61.6	80.7	**95.0**	**82.0**	**81.6**	**79.5**	66.6	98.9	**78.9**	**80.9**	**99.3**	93.8	**82.7**
Ours	**67.0**	**75.6**	**82.2**	78.1	**89.4**	91.6	89.3	81.6	**62.1**	**82.3**	74.3	81.7	80.9	79.0	**67.7**	**99.3**	**78.9**	73.7	98.3	**94.7**	81.4

TABLE XIII

KEYPOINT MATCHING ACCURACY (%) ON SPAIR-71K FOR ALL CLASSES

Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dog	Horse	Mbike	Person	Plant	Sheep	Train	Tv	Mean
GMN [63]	59.9	51.0	74.3	46.7	63.3	75.5	69.5	64.6	57.5	73.0	58.7	59.1	63.2	51.2	86.9	57.9	70.0	92.4	65.3
PCA [65]	64.7	45.7	78.1	51.3	63.8	72.7	61.2	62.8	62.6	68.2	59.1	61.2	64.9	57.7	87.4	60.4	72.5	92.8	66.0
NGM [64]	66.4	52.6	77.0	49.6	67.7	78.8	67.6	68.3	59.2	73.6	63.9	60.7	70.7	60.9	87.5	63.9	79.8	91.5	68.9
IPCA [67]	69.0	52.9	80.4	54.3	66.5	80.0	68.5	71.4	61.4	74.8	66.3	65.1	69.6	63.9	91.1	65.4	82.9	97.5	71.2
CIE [66]	71.5	57.1	81.7	56.7	67.9	82.5	73.4	74.5	62.6	78.0	68.7	66.3	73.7	66.0	92.5	67.2	82.3	97.5	73.3
NGM-v2 [64]	68.8	63.3	86.8	70.1	69.7	94.7	87.4	77.4	72.1	80.7	74.3	72.5	79.5	73.4	98.9	81.2	94.3	98.7	80.2
BBGM [69]	75.3	65.0	87.6	78.0	69.8	94.0	87.8	78.3	72.8	82.7	76.6	76.3	80.1	75.0	98.7	85.2	96.3	98.0	82.1
ASAR [68]	72.4	61.8	91.8	79.1	71.2	97.4	90.4	78.3	**74.2**	83.1	77.3	**77.0**	83.1	76.4	99.5	85.2	97.8	99.5	83.1
COMMON [37]	77.3	68.2	**92.0**	**79.5**	70.4	97.5	91.6	**82.5**	72.2	**88.0**	80.0	74.1	83.4	82.8	**99.9**	84.4	98.2	99.8	84.5
Ours	**78.4**	**70.3**	90.5	78.6	**72.1**	**98.5**	**91.7**	82.0	71.4	87.1	**82.4**	75.4	**83.5**	**84.4**	99.4	**86.0**	**99.5**	**99.9**	**85.1**

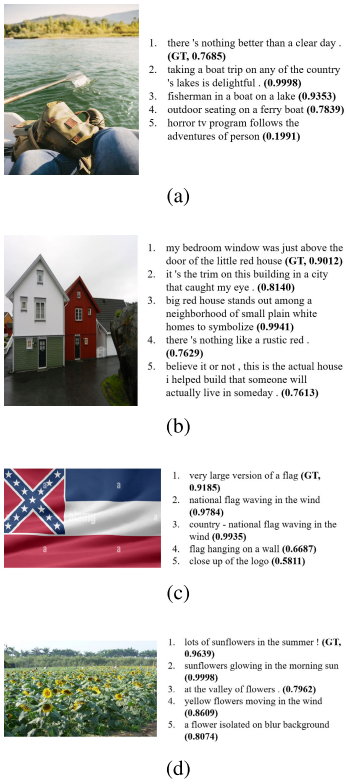


Fig. 7. Case studies on mined consistency using images as queries. The first caption of each query is the ground truth, and the value denotes the consistency predicted by our model.

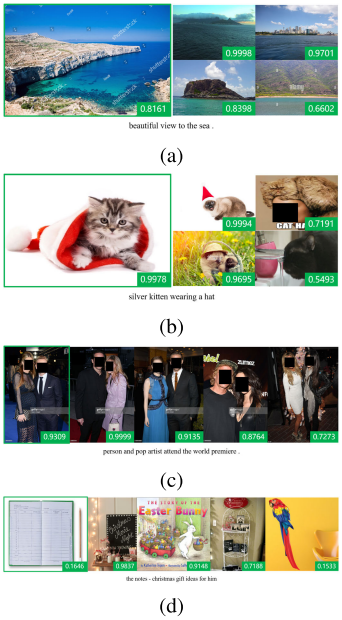


Fig. 8. Case studies on mined consistency using captions as queries. The ground truth image of each query is framed in green, and the value denotes the consistency predicted by our model.

studies by adopting only one neural network to divide the dataset. From the Table X, one could observe that employing a single network with GMM modeling of three components would decrease the performance compared to our default data partition module. It could be attributed to the error accumulation by such self-training manner.

D. More Case Studies

In the manuscript, we have given some case studies for a comprehensive understanding of our method (Section IV-D.4). Here, we give more case studies about the consistency mined in the negative bank by our method. The results are presented in Fig. 7 and Fig. 8, where the former uses images as queries while the latter uses captions as queries. From the results, one could see the powerful ability of mining consistency of our method, which would boost the performance of cross-modal retrieval. Note that, as shown in Fig. 8 (d), our method explores some diverse potential consistency while refining the incorrect correspondence for the ground truth, which could benefit achieving robust cross-modal retrieval.

E. Comprehensive Results of Graph Matching Experiments

In the manuscript, we summary the average keypoint matching accuracy on Willow Object, Pascal VOC and SPair-71k (Section IV-E). Here, in Table XI, Table XII and Table XIII, we show the complete results for all classes in those three datasets, respectively. One could see that CREAM

achieves competitive results among SOTAs, which proves the generalizability of CREAM to the graph matching with noisy correspondence problem.

REFERENCES

[1] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2556–2565.

[2] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. ECCV*, 2018, pp. 201–216.

[3] S. Chun, W. Kim, S. Park, M. Chang, and S. J. Oh, "ECCV caption: Correcting false negatives by collecting machine-and-human-verified image-caption associations for MS-COCO," in *Proc. ECCV*, 2022, pp. 1–19.

[4] J. Wei, X. Xu, Y. Yang, Y. Ji, Z. Wang, and H. T. Shen, "Universal weighting metric learning for cross-modal matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13002–13011.

[5] M. Cheng et al., "ViSTA: Vision and scene text aggregation for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5174–5183.

[6] H. Lu, N. Fei, Y. Huo, Y. Gao, Z. Lu, and J.-R. Wen, "COTS: Collaborative two-stream vision-language pre-training model for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15671–15680.

[7] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2494–2507, May 2017.

[8] H. Diao, Y. Zhang, L. Ma, and H. Lu, "Similarity reasoning and filtration for image-text matching," in *Proc. AAAI*, 2021, pp. 1218–1226.

[9] Y. Peng, X. Huang, and Y. Zhao, "An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2372–2385, Sep. 2018.

[10] P. Hu, X. Peng, H. Zhu, L. Zhen, and J. Lin, "Learning cross-modal retrieval with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5399–5409.

[11] J. Chen, H. Hu, H. Wu, Y. Jiang, and C. Wang, "Learning the best pooling strategy for visual semantic embedding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15784–15793.

[12] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. ICML*, 2021, pp. 8748–8763.

[13] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. ICML*, 2021, pp. 4904–4916.

[14] Z. Huang, G. Niu, X. Liu, W. Ding, X. Xiao, and X. Peng, "Learning with noisy correspondence for cross-modal matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 29406–29419.

[15] Y. Qin, D. Peng, X. Peng, X. Wang, and P. Hu, "Deep evidential learning with noisy correspondence for cross-modal retrieval," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 4948–4956.

[16] D. Arpit et al., "A closer look at memorization in deep networks," in *Proc. ICML*, 2017, pp. 233–242.

[17] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4653–4661.

[18] E. Amrani, R. Ben-Ari, D. Rotman, and A. Bronstein, "Noise estimation using density estimation for self-supervised multimodal learning," in *Proc. AAAI*, 2021, pp. 6644–6652.

[19] X. Xu, T. Wang, Y. Yang, L. Zuo, F. Shen, and H. T. Shen, "Cross-modal attention with semantic consistence for image-text matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5412–5425, Dec. 2020.

[20] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "VSE++: Improving visual-semantic embeddings with hard negatives," 2017, *arXiv:1707.05612*.

[21] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014, *arXiv:1411.2539*.

[22] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5005–5013.

[23] H. Chen, G. Ding, X. Liu, Z. Lin, J. Liu, and J. Han, "IMRAM: Iterative matching with recurrent attention memory for cross-modal image-text retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12652–12660.

[24] J. Li, R. Socher, and S. C. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," 2020, *arXiv:2002.07394*.

[25] X. Xia et al., "Robust early-learning: Hindering the memorization of noisy labels," in *Proc. ICLR*, 2021.

[26] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[27] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[28] E. Yang, D. Yao, T. Liu, and C. Deng, "Mutual quantization for cross-modal search with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7541–7550.

[29] S. Wu et al., "Multi-class classification from noisy-similarity-labeled data," 2020, *arXiv:2002.06508*.

[30] S. Wu et al., "Class2Simi: A noise reduction perspective on learning with noisy labels," in *Proc. ICML*, 2021, pp. 11285–11295.

[31] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in *Proc. ICML*, 2019, pp. 7164–7173.

[32] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized loss functions for deep learning with noisy labels," in *Proc. ICML*, 2020, pp. 6543–6553.

[33] H. Han, K. Miao, Q. Zheng, and M. Luo, "Noisy correspondence learning with meta similarity correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7517–7526.

[34] P. Hu, Z. Huang, D. Peng, X. Wang, and X. Peng, "Cross-modal retrieval with partially mismatched pairs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9595–9610, Aug. 2023.

[35] M. Yang, Z. Huang, P. Hu, T. Li, J. Lv, and X. Peng, "Learning with twin noisy labels for visible-infrared person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14288–14297.

[36] Y. Qin, Y. Chen, D. Peng, X. Peng, J. T. Zhou, and P. Hu, "Noisy-correspondence learning for text-to-image person re-identification," 2023, *arXiv:2308.09911*.

[37] Y. Lin, M. Yang, J. Yu, P. Hu, C. Zhang, and X. Peng, "Graph matching with bi-level noisy correspondence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 23305–23314.

[38] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, "Partially view-aligned representation learning with noise-robust contrastive loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1134–1143.

[39] M. Yang, Y. Li, P. Hu, J. Bai, J. Lv, and X. Peng, "Robust multi-view clustering with incomplete information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1055–1069, Jan. 2023.

[40] R. Huang et al., "NLIP: Noise-robust language-image pre-training," in *Proc. AAAI*, 2023, pp. 926–934.

[41] H. Han, Q. Zheng, M. Luo, K. Miao, F. Tian, and Y. Chen, "Noise-tolerant learning for audio-visual action recognition," 2022, *arXiv:2205.07611*.

[42] W. Kang, J. Mun, S. Lee, and B. Roh, "Noise-aware learning from web-crawled image-text data for image captioning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 2930–2940.

[43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. ICML*, 2020, pp. 1597–1607.

[44] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9726–9735.

[45] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[46] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.

[47] J.-B. Grill et al., "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[48] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[49] Y. Li et al., "Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm," 2021, *arXiv:2110.05208*.
[50] H. Luo et al., "CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning," *Neurocomputing*, vol. 508, pp. 293–304, Oct. 2022.
[51] H. Permuter, J. Francos, and I. Jermyn, "A study of Gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognit.*, vol. 39, no. 4, pp. 695–706, Apr. 2006.
[52] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Trans. Assoc. Comput. Linguistics*, vol. 2, pp. 67–78, Dec. 2014.
[53] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
[54] J. Min, J. Lee, J. Ponce, and M. Cho, "SPair-71k: A large-scale benchmark for semantic correspondence," 2019, *arXiv:1908.10543*.
[55] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Mali, Sep. 2009, pp. 1365–1372.
[56] M. Cho, K. Alahari, and J. Ponce, "Learning graphs to match," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 25–32.
[57] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
[58] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[60] Y. Fang, Q. Sun, X. Wang, T. Huang, X. Wang, and Y. Cao, "EVA-02: A visual representation for neon genesis," 2023, *arXiv:2303.11331*.
[61] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "EVA-CLIP: Improved training techniques for CLIP at scale," 2023, *arXiv:2303.15389*.
[62] R. Wightman. (2019). *Pytorch Image Models*. [Online]. Available: https://github.com/huggingface/pytorch-image-models
[63] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2684–2693.
[64] R. Wang, J. Yan, and X. Yang, "Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5261–5279, Sep. 2022.
[65] R. Wang, J. Yan, and X. Yang, "Learning combinatorial embedding networks for deep graph matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3056–3065.
[66] T. Yu, R. Wang, J. Yan, and B. Li, "Learning deep graph matching with channel-independent embedding and Hungarian attention," in *Proc. ICLR*, 2019.
[67] R. Wang, J. Yan, and X. Yang, "Combinatorial learning of robust deep graph matching: An embedding based approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6984–7000, Jun. 2023.
[68] Q. Ren, Q. Bao, R. Wang, and J. Yan, "Appearance and structure aware robust deep visual graph matching: Attack, defense and beyond," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15242–15251.
[69] M. Rolínek, P. Swoboda, D. Zietlow, A. Paulus, V. Musil, and G. Martius, "Deep graph matching via blackbox differentiation of combinatorial solvers," in *Proc. ECCV*, 2020, pp. 407–424.

Xinran Ma received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2022, where he is currently pursuing the master's degree in computer science with the College of Computer Science. His research interests include multi-modal learning and learning with noisy correspondence.

Mouxing Yang received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science with the College of Computer Science. His research interests include multi-modal representation learning and learning with noisy correspondence.

Yunfan Li received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2020, where he is currently pursuing the Ph.D. degree in computer science with the College of Computer Science. His research interests include unsupervised learning.

Peng Hu received the Ph.D. degree in computer science and technology from Sichuan University, China, in 2019. He is currently an Associate Research Professor with the College of Computer Science, Sichuan University. His main research interests include multi-view learning, cross-modal retrieval, and network compression. In these areas, he has authored more than 40 papers in top-tier conferences and journals.

Jiancheng Lv (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Professor with the Data Intelligence and Computing Art Laboratory, College of Computer Science, Sichuan University, Chengdu. His current research interests include neural networks, machine learning, and big data.

Xi Peng (Senior Member, IEEE) is currently the Cheung Kong Distinguished Professor of the College of Computer Science, Sichuan University. His current research interests include machine learning, multi-media analysis, and AI4Science. In these areas, he has coauthored around 100 articles in *Nature Communications*, *JMLR*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, ICML, and NeurIPS.