NICEST: Noisy Label Correction and Training for Robust Scene Graph Generation

Lin Li, Jun Xiao, Hanrong Shi, Hanwang Zhang, Yi Yang, Senior Member, IEEE, Wei Liu, Fellow, IEEE, and Long Chen, Member, IEEE

Abstract—Nearly all existing scene graph generation (SGG) models have overlooked the ground-truth annotation qualities of mainstream SGG datasets, i.e., they assume: 1) all the manually annotated positive samples are equally correct; 2) all the un-annotated negative samples are absolutely background. In this article, we argue that neither of the assumptions applies to SGG: there are numerous “noisy” ground-truth predicate labels that break these two assumptions and harm the training of unbiased SGG models. To this end, we propose a novel Noisy label CorrEction and Sample Training strategy for SGG: NICEST, which rules out these noisy label issues by generating high-quality samples and designing an effective training strategy. Specifically, it consists of: 1) NICE: it detects noisy samples and then reassigns higher-quality soft predicate labels to them. To achieve this goal, NICE contains three main steps: negative Noisy Sample Detection (Neg-NSD), positive NSD (Pos-NSD), and Noisy Sample Correction (NSC). First, in Neg-NSD, it is treated as an out-of-distribution detection problem, and the pseudo labels are assigned to all detected noisy negative samples. Then, in Pos-NSD, we use a density-based clustering algorithm to detect noisy positive samples. Lastly, in NSC, we use weighted KNN to reassign more robust soft predicate labels rather than hard labels to all noisy positive samples. 2) NIST: it is a multi-teacher knowledge distillation based training strategy, which enables the model to learn unbiased fusion knowledge. A dynamic trade-off weighting strategy in NIST is designed to penalize the bias of different teachers. Due to the model-agnostic nature of both NICE and NIST, NICEST can be seamlessly incorporated into any SGG architecture to boost its performance on different predicate categories. In addition, to better assess the generalization ability of SGG models, we propose a new benchmark, VG-OOD, by reorganizing the prevalent VG dataset. This reorganization deliberately makes the predicate distributions between the training and test sets as different as possible for each subject-object category pair. This new benchmark helps disentangle the influence of subject-object category biases. Extensive ablations and results on different backbones and tasks have attested to the effectiveness and generalization ability of each component of NICEST.

Index Terms—Multi-Teacher knowledge distillation, noisy label learning, out-of-distribution, scene graph generation.

I. INTRODUCTION

Scene Graph Generation (SGG) [1], [2] is a visual task that involves detecting object instances and classifying their pairwise visual relations in an image, which plays a vital role in comprehensive scene understanding [3], [4]. SGG typically represents scene graphs as visually-grounded graphs, where nodes represent objects and edges represent visual relations. Alternatively, it can be formulated as a set of $<$subject-predicate-object$>$ triplets. With its structured representation, SGG has contributed to various downstream tasks, such as image retrieval [5], [6], [7], visual question answering [8], [9], [10], and image captioning [11], [12], [13]. In recent years, SGG has gained considerable attention [14], [15], [16], [17], [18] due to the release of large-scale SGG benchmarks (e.g., Visual Genome (VG) [19] and GQA [20]) and advancements in object detection techniques [21], [22], [23].

Nevertheless, existing SGG benchmarks [19], such as VG, suffer from highly skewed long-tailed predicate-annotated labels. For example in VG, the number of annotated ground-truth visual relations for the head category “on” is 38 times larger than the tail category “sitting on”. To address this issue, current approaches in SGG [24], [25], [26], [27], [28] can be broadly categorized into two strategies. 1) Re-balancing strategy, which involves class-aware re-sampling or loss re-weighting to balance the proportions of different predicate categories during training. 2) Biased-model-based strategy, which separates unbiased predictions from pretrained biased SGG models [29], [30], [31].

Although these models have dominated on the debiasing SGG metrics (e.g., mean Recall@K), it is worth noting that nearly all existing SGG work has taken two plausible assumptions about the ground-truth annotations for granted:

Assumption 1. All the manually annotated positive samples are equally correct.

Assumption 2. All the un-annotated negative samples are absolutely background.
For the first assumption, by “equally”, we mean that the confidence (or quality) of an annotated ground-truth predicate label for each positive sample\(^1\) is exactly the same, i.e., all the manually annotated positive predicate labels are of high quality. Different from other closed-set classification tasks, where each sample corresponds to a unique ground-truth, some specific subject-object pairs in SGG may have multiple reasonable predicates, i.e., the semantics of different predicate categories are interdependent to some extent. Inevitably, this phenomenon has resulted in two annotation characteristics in SGG datasets: 1) Common-prone: When the semantic granularities of these reasonable visual relations are different, the annotators tend to choose the commonest (or coarse-grained) predicate as the ground-truth label. As shown in Fig. 1(a), both riding and on are “reasonable” for the relation between man and bike, but the annotated ground-truth predicate is the less informative on instead of more convincing riding. And this characteristic occurs frequently in the SGG datasets (cf., more examples in Fig. 1(a)). 2) Synonym-random: When these reasonable visual relations are synonymous for the subject-object pair, the annotators usually randomly select one predicate from the several synonyms (e.g., has and with are synonyms for (man/woman shirt)).

\(^1\)We use “sample” to represent the triplet instance interchangeably, and we use “instance” to denote an instance of visual relation triplet.

datasets are always sparsely identified and annotated [32] (cf., Fig. 1(c)), they still train their models by treating all the un-annotated pairs as background, i.e., there is no visual relation between the subject and object pair. By contrast, we contend that all negative samples are NOT absolutely background, i.e., a part of negative samples are not high-quality, that is, they are actually the foreground with missing annotations.

In this paper, we try to get rid of these two questionable assumptions and propose to reformulate SGG as a noisy label learning problem. Specifically, we propose a novel model-agnostic Noisy label CorRection and Sample Training strategy for SGG, dubbed NICEST. NICEST mitigates the noisy label learning problem from two aspects: generating more high-quality training samples and training models with a more effective mechanism. To achieve this goal, NICEST consists of two parts: Noisy label CorRection (NICE) and Noisy Sample Training (NST).

NICE: For high-quality training data, NICE cannot only detect numerous noisy samples, but also reassign more high-quality predicate labels to them. By “noisy”, we mean that these samples break the two assumptions. After the NICE training, we can obtain a cleaner version of the SGG dataset. Specifically, we can: 1) increase the number of fine-grained predicates (common-prone); 2) decrease annotation inconsistency among similar visual patterns (synonym-random); 3) increase the number of positive samples (assumption 2).

To be more specific, NICE consists of three steps: 1) negative noisy sample detection (Neg-NSD): We reformulate the negative NSD as an out-of-distribution (OOD) detection problem, i.e., regarding all the positive samples as in-distribution (ID) training data, and all un-annotated negative samples as OOD test data. In this way, we can detect the missing annotated (ID) samples with pseudo labels. 2) positive noisy sample detection (Pos-NSD): We use a clustering-based algorithm to divide all the positive samples (including the outputs of Neg-NSD) into multiple sets, and regard samples in the noisiest set as noisy positive samples. The clustering results are based on the local density of
each sample. 3) noisy sample correction (NSC): We utilize the weighted KNN (wKNN) to generate a new pseudo label for each sample. To alleviate the impact of the original noisy label with full probabilities, and give a certain fault tolerance space to the pseudo labels, we reassign soft predicate labels to all noisy positive samples.

NIST: Unfortunately, there is no free lunch, sometimes newly assigned pseudo labels from NICE are not right (e.g., ⟨boy-near-pizza⟩ is mistakenly changed to ⟨boy-eating-pizza⟩ in Fig. 2). Like other debiasing methods, NICE may over-weight tail categories by increasing the sample percentage of tail categories. For further effective unbiased SGG training, we propose a new knowledge distillation based Noisy Sample Training strategy: NIST. NIST is a multi-teacher knowledge distillation method that can integrate the unbiased knowledge of multi-teachers to improve the robustness of distillation. Typically, we can use two different teachers, one is trained on the original dataset (i.e., more in favor of head predicates), and the other is trained on the NICE refined dataset (i.e., more in favor of tail predicates).

A novel dynamic weighting strategy is devised to punish both biases to obtain unbiased fusion knowledge. Specifically, we first measure the degree of bias of each teacher model by calculating the cross entropy between the predictions of each teacher and the original ground-truth. Then, we fuse the predictions of the low-biased teacher model with the ground-truth. In this way, the biases on different predicates can be eliminated through dynamic weighting. As original ground-truth predicates are proven to be noisy [33], NIST can make the model more robust by smoothing noisy targets to supervise the training. NIST is also a general trade-off training strategy, which can relieve the bias of different predicates for unbiased SGG. More detailed ablations about its generalization ability will be discussed in Section VI-D.

New Benchmark VG-OOD: In addition, for the widely-used SGG benchmarks (e.g., VG [19]), their predicate distributions of the training set and test set for each subject-object category pair are similar. As displayed in Fig. 8, the predicate distributions of woman-shirt in the original VG dataset are extremely similar (e.g., wearing is the most common predicate in both training and test sets). Due to this property, a naive baseline that simply uses the frequency priors of the predicates of each subject-object category pair (without considering visual appearance) can achieve satisfactory results [24]. For a deeper disentanglement of this prior knowledge and to benchmark the OOD performance of debiasing SGG models, we propose a new dataset for OOD evaluation: VG-OOD. Specifically, we re-split the VG dataset and try to make the predicate distribution of each subject-object category pair in the training and test sets as inconsistent as possible. In this way, the performance of the test set on VG-OOD can better measure the OOD generalization ability of SGG models.

NICEST was evaluated on the widely recognized scene graph generation (SGG) benchmarks, including VG [19], GQA [20], and the newly introduced VG-OOD dataset. Within NICEST, NICE plays a crucial role in enhancing the quality of annotations in original datasets, while NIST focuses on absorbing unbiased knowledge from two teachers. Notably, both NICE and NIST are model-agnostic, allowing for seamless integration into various SGG architectures to enhance their performance. The effectiveness and generalization capabilities of each component within NICEST have been extensively validated through meticulous ablation studies and comprehensive experimental results.

This paper is a substantial and systematic extension of previous CVPR oral work [33] with several significant improvements:

1) In previous NICE-v1,2 we merely assign a hard label to noisy samples in NSC, which may introduce new noise. Thus, in this paper, we propose a more robust method to generate soft labels based on the weights in wKNN.

2) Beyond NICE, we propose a noisy label training strategy NIST. It comprehensively considers the unbiased fusion knowledge from multi-teachers and ground-truth labels, which helps achieve robust model training.

3) We propose a new SGG benchmark for OOD evaluation: VG-OOD. It is designed to supplement the evaluation of the debiasing and generalization ability of SGG models.

4) We conduct more qualitative and quantitative analyses to demonstrate the effectiveness of NICEST.

II. RELATED WORK

Scene Graph Generation: SGG helps machines understand visual relationships between objects, which aims to transform the raw visual input into semantic graph structures. Early SGG work [32], [34] usually regards each object as an individual and directly predicts their pairwise relations without considering the visual context. Subsequent SGG methods [1], [25], [35] begin to exploit this overlooked visual context by regarding each image as a whole and adopting the fully connected graph [36], [37], or the tree-structured graph [26] to model the contexts among objects. Due to the long-tailed distribution and other language bias issues [38], unbiased SGG has recently drawn remarkable attention. In general, existing unbiased SGG models can be roughly divided into: re-balancing strategy [27], [39], [40], [41], [42], [43] and biased-model-based strategy [29], [30], [31], [44]. Different from existing work [45], [46], we are the first to explicitly refine the original noisy ground-truth annotations on SGG datasets, and address the problem in terms of both label quality and training strategy. Although some previous work also has discussed the issue of sparse annotations [31], [47] or

2For the sake of distinction, we use NICE-v1 to denote NICE in [33].
 semantic imbalance [44], they still heavily rely on these original noisy annotations in the training.

**Learning with Noisy Labels:** Existing noisy label learning methods can be roughly grouped into two categories: 1) Utilizing an explicit or implicit noisy model to estimate the distributions of noisy and clean labels, and then deleting or correcting these noisy samples. These models can be in different formats, such as neural networks [48], [49], [50], [51], conditional random field [52], or knowledge graphs [53]. However, they always need abundant clean training samples, which is inapplicable for many noisy label learning datasets. 2) Constructing a more balanced loss function to reduce the influence of noisy training samples [54], [55], [56], [57], [58]. In this paper, we are the first to formulate SGG as a noisy label learning problem, and propose a novel noisy sample correction strategy and an effective training strategy.

**Knowledge Distillation (KD):** Early KD was proposed to effectively learn lightweight student models from pretrained cumbersome teacher models [59], [60], [61]. Benefiting from the soft targets generated by teacher models, KD has shown potentials in numerous vision tasks, i.e., object detection [62], [63], [64], long-tailed learning [65], [66], [67], scene graph generation [68], and visual question answering (VQA) [69]. However, most traditional KD methods only consider one teacher model. To overcome the limitation of the data diversity and single-teacher knowledge, multi-teacher knowledge distillation [70], [71], [72], [73] integrates comprehensive knowledge of multiple teachers. In this paper, we propose a new multi-teacher KD-based training strategy, which can incorporate the unbiased fusion knowledge of multi-teachers for robust SGG training under noisy labels.

**OOD Benchmarks for Visual Scene Understanding:** Recently, several scene understanding tasks, such as VQA [10], [69], [74], [75] and video grounding [76], [77], try to deliberately make the ground-truth distributions different in the training and test splits. For example, Agrawal et al. [78] reorganized splits of previous VQA datasets by designing different answer distributions for each question type in the training and test sets. Similarly, another video grounding dataset for OOD evaluation was proposed in a similar manner [76]. Inspired by them, we introduce VG-OOD to better measure the OOD generalization abilities of SGG models.

### III. NICE: NOISY LABEL CORRECTION

Given an image dataset $\mathcal{I}$, the SGG task aims to convert each image $I_i \in \mathcal{I}$ into a graph $G_i = \{N_i, \mathcal{E}_i\}$, where $N_i$ and $\mathcal{E}_i$ denote the node set (i.e., objects) and edge set (i.e., visual relations) of image $I_i$, respectively. Generally, each graph $G_i$ can also be viewed as a set of visual relation triplets (i.e., $\langle$subject-predicate-object$\rangle$), denoted as $T_i$. For each triplet set $T_i$, we can further divide it into two subsets: $T_i^+$ and $T_i^-$, where $T_i^+$ denotes all the annotated positive triplets (or samples) in image $I_i$, and $T_i^-$ denotes all the un-annotated negative triplets in image $I_i$. Similarly, we use $\mathcal{T}^+ = \{T_i^+\}$ and $\mathcal{T}^- = \{T_i^-\}$ to represent all positive and negative triplets in the whole dataset $\mathcal{I}$.

Fig. 3 illustrates the overall pipeline of NICE. In this section, we successively introduce each part of NICE, including negative noisy sample detection (Neg-NSD), positive NSD (Pos-NSD), and noisy sample correction (NSC). To be specific, given an image and its corresponding ground-truth triplet annotations (i.e., $\mathcal{T}^+ \cup \mathcal{T}^-$), we first use the Neg-NSD to detect all possible noisy negative samples, i.e., missing annotated foreground triplets. $\mathcal{T}^-$ can be further divided into $\mathcal{T}^\text{clean}$ and $\mathcal{T}^\text{noisy}$. Meanwhile, Neg-NSD assigns pseudo positive predicate labels for all samples in $\mathcal{T}^\text{noisy}$ (e.g., painted on for $\langle$letter-window$\rangle$ in Fig. 3). $\mathcal{T}^\text{noisy}$ with pseudo positive labels and original $\mathcal{T}^+$ compose a new positive set $\mathcal{T}^\text{clean}$. Next, we use Pos-NSD to detect all possible noisy positive triplets.

3In Fig. 3, we use a single image as input for a clear illustration. In our experiments, we directly process the whole dataset in each module.

4For brevity, we omit the subscripts $i$ for image $I_i$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
samples in $\mathcal{T}^+$, i.e., the positive samples which suffer from either common-prone or synonym-random characteristics (e.g., at for $\text{women-laptop}$ in Fig. 3). Similarly, $\mathcal{T}^+$ can be divided into $\mathcal{T}_{\text{clean}}^+$ and $\mathcal{T}_{\text{noisy}}^+$. Then, we use NSC to reassign more high-quality soft predicate labels to all samples in $\mathcal{T}_{\text{noisy}}^+$, denoted as $\mathcal{T}_{\text{noisy-clean}}^+$. Lastly, after processing the ground-truth triplet annotations of all images, we can obtain a cleaner version of the dataset ($\mathcal{T}_{\text{clean}}^+ \cup \mathcal{T}_{\text{noisy-clean}}^+ \cup \mathcal{T}_{\text{clean}}$) for SGG training.

A. Negative Noisy Sample Detection (Neg-NSD)

In this module, our goal is to discover all possible noisy negative samples, i.e., missing annotated visual relation triplets. Meanwhile, Neg-NSD assigns a pseudo positive predicate label for each detected noisy negative sample. It is difficult to directly train and evaluate a binary classifier based on these noisy samples, due to the nature of missing annotations in existing negative samples. To this end, we propose formulating the negative noisy sample detection as an out-of-distribution (OOD) detection problem [79]. Specifically, we regard all annotated positive samples as in-distribution (ID) training data, and all unannotated negative samples as OOD test data. Neg-NSD is constructed on top of a plain SGG model (denoted as $\mathbb{F}_{\text{sgg}}^+$), but it is trained exclusively with the annotated positive samples $\mathcal{T}^+$. During the inference stage, Neg-NSD predicts a score of being foreground and assigns a pseudo positive predicate category to each triplet $t_i^-$ in $\mathcal{T}^-$. Following existing OOD detection methods [80], we also utilize a confidence-based model, i.e., Neg-NSD consists of two network output branches: 1) a classification branch to predict a probability distribution $p$ over all positive predicate categories, and 2) a confidence branch to predict a confidence score $c \in [0, 1]$, which indicates the confidence of being an ID category (foreground). During the inference phase, for each sample $t_i^-$, if its confidence score $c_i$ surpasses a threshold $\theta$, we then regard this negative sample as a noisy negative sample, i.e., the detection function $g(\cdot)$ is defined as follows:

$$g(t_i^-) = \begin{cases} 1, & c_i \geq \theta \\ 0, & c_i < \theta \end{cases}$$  \hspace{1cm} (1)

When $g(t_i^-) = 1$, the pseudo label of $t_i^-$ is directly derived from the classification branch, i.e., $\arg\max(p_i)$. Due to significant variations in the predicted average confidence scores for different predicate categories, we set different thresholds for head, body, and tail categories. (More details are in Section VI.)

Training of Neg-NSD: To train the classification branch and confidence branch, we combine predicted probabilities $p_i$ and corresponding target probability distribution $y_i$, i.e.,

$$p'_i = c_i \cdot p_i + (1 - c_i) \cdot y_i$$ \hspace{1cm} (2)

where $p'_i$ is the adjusted probabilities by confidence $c_i$. The motivation of (2) is that, given the opportunity to request a hint of the ground-truth probability with some penalty, the model will definitely choose to seek the hint if it lacks confidence in its output (i.e., $c_i$ is small). And the training objective for Neg-NSD consists of a weighted cross-entropy loss and a regularization penalty loss:

$$\mathcal{L}_{\text{Neg-NSD}} = -\sum_{j=1} w_j \log(p'_i) y_{ij} - \lambda \log(c_i),$$  \hspace{1cm} (3)

where $p'_ij$ and $y_{ij}$ are the $j$-th elements of $p'_i$ and $y_i$, respectively. $w_j$ is the reciprocal of the frequency of the $j$-th predicate category, which mitigates the impact of the long-tailed issues on confidence. The penalty loss is used to prevent the network from always choosing $c = 0$ and using a ground-truth probability distribution to minimize the task loss.

B. Positive Noisy Sample Detection (Pos-NSD)

As shown in Fig. 3, the original positive set $\mathcal{T}^+$ and outputs of the Neg-NSD (i.e., $\mathcal{T}_{\text{noisy}}^+$) compose a new positive sample set $\mathcal{T}^+$. The Pos-NSD module aims to detect all noisy samples in $\mathcal{T}^+$. In general, we utilize a clustering-based solution to divide all these positive samples into multiple subsets with different degrees of noise, and treat all samples in the noisiest subset as noisy positive samples. Intuitively, if a predicate label is consistent with other visually-similar samples of the same predicate category (i.e., visual features of these samples are close to each other). In that case, this predicate is more likely to be a clean sample since these annotations are consistent with each other. Otherwise, it is likely to be a noisy sample. As shown in Fig. 4, the two clean triplets (window-in-room) have more visually-similar neighbors than the noisy triplets (e.g., plant-in-window).

Based on these observations, we propose a local density based solution for positive noisy sample detection. To be specific, we utilize an off-the-shelf pretrained SGG model (denoted as $\mathbb{F}_{\text{sgg}}^+$) to extract all visual relation triplet features, and utilize $h^k$ to represent the visual feature of the $i$-th sample of predicate category $k$ (this sample is denoted as $t_i^k$). Then, we use a distance matrix $D^k = (d^k_{ij})_{N \times N} \in \mathbb{R}^{N \times N}$ to measure the similarity between all positive samples of the same predicate $k$, and $d^k_{ij}$ is calculated by:

$$d^k_{ij} = \| h^k_i - h^k_j \|^2,$$ \hspace{1cm} (4)

where $\| \cdot \|$ is the euclidean distance. Thus, a smaller distance $d^k_{ij}$ means a relatively higher similarity between sample $t_i^k$ and sample $t_j^k$. Then following [81], we define the local density $\rho^k_i$ of each sample $t_i^k$ as the number of samples (within the same predicate category) whose distances to sample $t_i^k$ are closer than
a threshold \( d^*_k \), i.e.,

\[
\rho_i^k = \sum_j 1((d^*_k - d_{ij}) > 0),
\]

where \( 1(\cdot) \) is the indicator function and \( d^*_k \) is the cutoff distance for predicate \( k \), which is ranked at \( \alpha \% \) of sorted \( N \times N \) distances in \( D_N \) from small to large. Thus, a sample with higher local density \( \rho \) means that this sample is more similar to the samples of the same predicate category. Analogously, samples with low local density \( \rho \) are considered as noisy samples. Finally, we use an unsupervised K-means algorithm [82] to divide all data samples into multiple subsets with respect to different \( \rho \) values, i.e., different degrees of noise [83]. All samples in the subset with the lowest \( \rho \) are regarded as noisy positive samples (i.e., \( \mathcal{F}_{\text{noisy}} \)), and passed into the subsequent NSC module for label correction.

**Influence of the Cutoff Distance \( d^*_k \):** From (5), we can observe that the distribution of local density \( \rho \) is directly decided by the selection of the cutoff distance \( d_i \), or the hyperparameter \( \alpha \%). As shown in Fig. 5, when the cutoff distance is ranked at 50% and 1%, local densities of samples diffuse outwards from large to small with one and two centers, respectively, i.e., a smaller cutoff distance (e.g., 1% for \( \alpha \% \)) may divide the whole feature space into more clusters. Meanwhile, different predicate categories may contain various types of semantic meanings. For example, in Fig. 5, the predicate \text{in} of samples inside the red circle represents “inside” (e.g., \{plant-in-vase\}), while predicate \text{in} of the samples inside the green circle represents “wearing” (e.g., \{man-in-shirt\}). Therefore, we set different cutoff distances for different categories.

In addition, more detailed discussions about the influence of \( d^*_k \) on the clustering results are left in the appendix, available online.

**C. Noisy Sample Correction (NSC)**

Given all the detected noisy positive samples from Pos-NSD, the NSC module aims to correct and assign more robust soft labels to these noisy positive predicate labels. There are two non-zero (positive) categories in the newly assigned soft label \( r^{\text{ori}} \): the original predicate category and a raw predicate category. The raw predicates serve as relatively clean labels assigned by wKNN, following the approach in NICE-v1 [33]. The subsequent sections elaborate on the raw predicate categories and the calculation of ground-truth values for the two predicates in the soft labels.

**Selection of Raw Predicates:** The motivation of our NSC is that the predicate label of a sample should be consistent with other visually-similar training samples, especially for those samples with the same \text{subject} and \text{object} categories. For example, in Fig. 6, consider the noisy sample \{girl-in-chair\}. By retrieving all other samples with the same \{girl, chair\}, we observe that most visually-similar samples are annotated as \{girl-sitting on-chair\}. Thus, we employ the simple yet effective weighted K-Nearest Neighbor (wKNN) algorithm to derive the most probable labels for noisy positive samples. wKNN assigns larger weights to the closest samples and smaller weights to those that are far away. Specifically, let \( N(i) \) be the set of \( K \) neighbors of sample \( t_i \), then the raw predicate label \( r^{\text{raw}}_i \) for \( t_i \) is:

\[
r^{\text{raw}}_i = \arg\max_v w_{ij} \cdot 1(v = r_j),
\]

where \( v \) is a predicate category, \( r_j \) is predicate label of \( t_j \), and \( 1(\cdot) \) is an indicator function. The weight \( w_{ij} \) is assigned to each neighbor, defined as \( a \cdot \exp(\frac{-(d_{ij} - b)^2}{2c^2}) \), \( d_{ij} \) is the euclidean distance between \( h_i \) and \( h_j \) (4)), and \( a, b, c \) are hyperparameters.

**Calculation of Predicate Score:** However, the raw predicate labels are not always “perfect”. For example, in Fig. 2, \{boy-near-pizza\} is mistakenly changed to \{boy-eating-pizza\}. To ensure the robustness of training, we propose a scoring function that considers the distances of triplets in weighted KNN as the confidence of the labels and fuses the raw predicate label \( r^{\text{raw}} \) with the original predicate label \( r^{\text{ori}} \) to obtain a new soft label. The scores of the raw predicate label \( s^{\text{raw}} \) and the original predicate label \( s^{\text{ori}} \) in soft labels are calculated as follows:

\[
s^{\text{raw}} = \frac{\sum w^{\text{raw}}(v = r^{\text{raw}})}{\sum w^{\text{raw}}(v = r^{\text{raw}}) + \sum w^{\text{ori}}(v = r^{\text{ori}})},
\]

\[
s^{\text{ori}} = \frac{\sum w^{\text{ori}}(v = r^{\text{ori}})}{\sum w^{\text{raw}}(v = r^{\text{raw}}) + \sum w^{\text{ori}}(v = r^{\text{ori}})},
\]
where $w^{raw}$ and $w^{ori}$ represent the weights of the raw predicate and the original predicate in KNN, respectively.

**Training Objectives:** We apply the binary cross entropy objective for training with soft labels, which is written as:

$$\mathcal{L}_{SL} = -\frac{1}{N} \sum_i N_i \log(p_i) + (1 - N_i) \log(1 - p_i),$$

where $N_i$ is the value of the $i$-th category in the soft label, $p_i$ is the prediction probability of the $i$-th category, and $N$ is the total number of predicate categories.

**Highlights:** In the original NSC of NICE-v1 [33], we directly assign hard labels (i.e., raw predicate labels) with full probability to noisy samples. If the newly assigned labels were unreasonable, new noise might be introduced. In the new NSC of this paper, we assign soft labels to noisy samples, which takes into account both the raw predicate labels by wKNN and the original ground-truth labels. It not only increases the error tolerance rate of noisy sample correction, but also is more conducive to robust training.

**IV. NIST: Noisy Sample Training**

As mentioned above, the labels corrected by NICE [33] are not always accurate. Since most of them are corrected from the head to tail predicates, it may lead to a somewhat over-weighting of tail predicates. We propose a novel multi-teacher knowledge distillation strategy for effective model training that enables the model to learn from the unbiased trade-off fusion of two teachers. The structures of both the two teacher models and the student model are the same as the backbone (i.e., Motifs [24] and VCTree [26]). In this way, we generate a relatively smooth target label that mitigates the performance degradation caused by absolute confidence in the noisy ground-truth with full probability by assigning values to other categories.

The pipeline of the multi-teacher knowledge distillation training paradigm is illustrated in Fig. 7. Specifically, we initially select two models, where one excels in head predicates (i.e., can achieve good performance in Recall) and the other excels in tail predicates (i.e., can achieve good performance in mean Recall). For the sake of simplification, these models are denoted as $T_{Head}$ and $T_{Tail}$, respectively. We aim to obtain an unbiased trade-off teacher that can strike a balance between Recall and mean Recall (i.e., penalize excessive bias against both head and tail predicates). Therefore, we measure the bias of the model towards head and tail predicates, and define the degrees of biases for head and tail predicates as $s_{Head}$ and $s_{Tail}$, respectively. The calculation process is as follows:

$$s_{Head} = \frac{1}{\mathcal{XE}(p_{GT}^{T}, p_{Head}^{S})} = \frac{1}{\sum_{j=1} \log p_{GT}^{T,j} - \log p_{Head}^{S,j}},$$

$$s_{Tail} = \frac{1}{\mathcal{XE}(p_{GT}^{T}, p_{Tail}^{S})} = \frac{1}{\sum_{j=1} \log p_{GT}^{T,j} - \log p_{Tail}^{S,j}},$$

where $\mathcal{XE}$ denotes the cross-entropy function, and $p_{GT}^{T}$, $p_{Head}^{S}$, $p_{Tail}^{S}$ denote the ground-truth labels, the output of $T_{Head}$, and the output of $T_{Tail}$, respectively. We then adopt a dynamic weighting strategy to fuse the biased knowledge of the two teachers fairly, using weights $w_{Head}$ and $w_{Tail}$. The final trade-off knowledge $p^T$ to supervise the training is determined by:

$$p^T = w_{Head} p_{Head}^{S} + w_{Tail} p_{Tail}^{S},$$

where $w_{Head} = \frac{s_{Tail}}{s_{Head} + s_{Tail}}$, $w_{Tail} = \frac{s_{Head}}{s_{Head} + s_{Tail}}$.

**Group Dynamic Weighting:** Acknowledging that the ground-truth (GT) predicate label is more accurate than $T_{Head}$ for head predicates. Similarly, the GT predicate label is more accurate than $T_{Tail}$ for tail predicates. Therefore, we employ a dynamic weighting mechanism for the head, body, and tail groups to obtain more accurate knowledge for distillation:

- For head predicates, $p_{Head}^{S}$ is the GT label, and $p_{Head}^{T}$ is the output probability distribution of $p_{Tail}^{S}$.
- For body predicates, since the GT labels are more accurate than both two models, we directly adopt the GT label as $p_{Head}^{S}$ and $p_{Tail}^{S}$.
- For tail predicates, $p_{Head}^{S}$ is the output probability distribution of $p_{Head}^{T}$, and $p_{Tail}^{T}$ is the GT label.

**Training Objectives:** Since the outputs of the student model and the unbiased teacher model are probability distributions, we adopt KL divergence [84] as the loss function for multi-teacher KD, denoted as:

$$\mathcal{L}_{KD} = KL(p^{S}, p^{T}) = \sum_i N_i \log \left(\frac{p_i^{S}}{p_i^{T}}\right),$$

where $p^{S}$ is the output probability distribution of student model.

**Highlights:** In NICE-v1 [33], the original dataset is corrected correctly. This correction is not always correct and will exaggerate the proportion of tail predicates. In this paper, we propose a training strategy based on multi-teacher knowledge distillation, which integrates different biased knowledge of teacher models to supervise student model training, allowing the student model to account for the performance of both head and tail predicates.

**V. Benchmark: VG-OOD**

Visual Genome [19], the most widely utilized SGG dataset at present, has a consistent predicate distribution for each subject-object category pair in both the training and test sets. Some studies [24] have found that good performance can be obtained just through the frequency prior bias of the commonest predicate category for each subject-object category pair (i.e., 62.2% on R@100 in Predcls). To reduce the influence of frequency bias and supplement the evaluation of the generalization ability of the model, we propose a new benchmark called VG-OOD by
re-splitting the VG dataset. The statistical distributions of all predicates per subject-object category pair (e.g., man-shirt) in VG-OOD are different in the test set compared to the training set.

Dataset Construction: We first count the number of all possible predicates for each subject-object category pair, and arrange the predicates based on their frequency in ascending order. Second, we add triplets whose total number is less than 20% of the overall subject-object category pairs in the test triplet list. For example, in Fig. 8, for the woman-shirt pair, we sort the number of predicates and include those with a cumulative number of 20%, along with their subject-object category pairs, into the list of test triplets (e.g., woman-holding-shirt and woman-holding-shirt). Finally, we split the image into the test set when 70% of its triplets are in the test triplet list.

Dataset Statistics: The statistics of the number of images, the number of triplets, and the difference in distribution (i.e., KL and KL-mean) between the VG and VG-OOD training and test sets are shown in Table I. Among them, KL divergence computes the disparity in predicate distributions between the two splits across all possible subject-object category pairs. The greater the KL divergence, the more different the two distributions are. Therefore, we can see that the re-splitting process in VG-OOD results in significant difference in predicate distributions for each subject-object category pair between the training and test sets. Furthermore, we visualize the distribution of predicates under woman-shirt and tree-snow in Fig. 8. Notably, there is a marked contrast in the number of predicates with a large proportion between the training and test sets (e.g., wearing vs. wears, and in vs. near). Consequently, relying solely on the frequency statistical prior of triplets does not yield satisfactory performance in the test set. Moreover, for some triplets only appearing in the test set, the generalization ability of models can be better verified (e.g., woman-holding-shirt).

VG-OOD vs. VG: As mentioned earlier, since the predicate distributions under each subject-object category pair in the training and test sets are consistent in the original VG dataset, a simple frequency bias baseline can achieve decent results. To verify the generalization ability of SGG models to be independent of the frequency bias, we report the performance of models with and without frequency priors in VG and VG-OOD datasets in Table II. It can be observed from Table II that the model can achieve significant gains on VG (e.g., 41.8% vs. 40.9% on Mean) by using the frequency prior. However, for the VG-OOD dataset, the model cannot obtain desired results with the frequency prior (e.g., 34.9% vs. 36.5% on Mean). Thus, training the model on the VG-OOD dataset can better measure the ability to classify predicates using visual information rather than the frequency bias.

VI. EXPERIMENTS

A. Experimental Settings and Implementation Details

Datasets: To ensure a comprehensive evaluation of our proposed method, we conducted all experiments on three datasets: the challenging VG [19], our newly split VG-OOD and GQA [20]. 1) VG: VG is the most widely utilized benchmark for SGG with over 108 k images. We selected VG to thoroughly evaluate NICEST and ensure a fair comparison with SOTA methods. We followed widely-used splits [1], dividing images into training (70%), testing (30%), and a validation set (5,000 images sampled from the training set [24]). Besides, we followed [89] to divide all predicate categories into three parts based on the number of samples in the training set: head sets for all samples. KL-mean is calculated by averaging the KL divergences of the predicate distributions between the two splits across all possible subject-object category pairs. The greater the KL divergence, the more different the two distributions are. Therefore, we can see that the re-splitting process in VG-OOD results in significant difference in predicate distributions for each subject-object category pair between the training and test sets. Furthermore, we visualize the distribution of predicates under woman-shirt and tree-snow in Fig. 8. Notably, there is a marked contrast in the number of predicates with a large proportion between the training and test sets (e.g., wearing vs. wears, and in vs. near). Consequently, relying solely on the frequency statistical prior of triplets does not yield satisfactory performance in the test set. Moreover, for some triplets only appearing in the test set, the generalization ability of models can be better verified (e.g., woman-holding-shirt).

VG-OOD vs. VG: As mentioned earlier, since the predicate distributions under each subject-object category pair in the training and test sets are consistent in the original VG dataset, a simple frequency bias baseline can achieve decent results. To verify the generalization ability of SGG models to be independent of the frequency bias, we report the performance of models with and without frequency priors in VG and VG-OOD datasets in Table II. It can be observed from Table II that the model can achieve significant gains on VG (e.g., 41.8% vs. 40.9% on Mean) by using the frequency prior. However, for the VG-OOD dataset, the model cannot obtain desired results with the frequency prior (e.g., 34.9% vs. 36.5% on Mean). Thus, training the model on the VG-OOD dataset can better measure the ability to classify predicates using visual information rather than the frequency bias.
We evaluated NICEST on three SGG tasks [1]: 1) Predicate Classification (PredCls): Given the ground-truth objects with labels, we need to only predict pairwise predicate categories. 2) Scene Graph Classification (SGCls): Given the ground-truth object bounding boxes, we need to predict both the object categories and predicate categories. 3) Scene Graph Generation (SGGen): Given an image, we need to detect all object bounding boxes, and predict both the object categories and predicate categories.

Evaluation Metrics: We evaluated all results on three metrics: 1) \( \text{Recall}@K \) (\( R@K \)): It calculates the proportion of top-K confident predicted relation triplets that are in the ground-truth. Following prior work, we used \( K = \{50, 100\} \). 2) mean \( \text{Recall}@K \) (\( mR@K \)): It calculates the recall for each predicate category separately, and then averages \( R@K \) over all predicates, i.e., it puts relatively more emphasis on the tail categories. 3) \( \text{Mean} \): It is the mean of all \( mR@K \) and \( R@K \) scores [33]. \( R@K \) favors head predicates, while \( mR@K \) favors tail ones. Therefore, it is a comprehensive metric that can better reflect model performance on different predicates.

NICE Training Details: In Neg-NSD, we used the Motifs [24] as OOD detection model \( F_{\text{SG}}^{m} \). The training settings (e.g., learning rate and batch size) followed the configurations of [29] under the PredCls task, except that it was trained with only foreground samples. In Pos-NSD, we used a pretrained Motifs [24] provided by [29] as \( F_{\text{SG}}^{m} \) to extract triplet features (cf., \( h_{2}^{\text{f}} \) in (4)) under PredCls task. The number of divided subsets was set to 4. In NSC, the \( a, b, \) and \( c \) were set to 1, 0, and 10, respectively. Note that though we used two Motifs models (one is an off-the-shelf model [24], [29]) in NICE, we only needed to train NICE once, and we could use the obtained cleaner annotations for any model.

NIST Training Details: In the experiment, the models with the bias of head predicates were the widely used baseline models (i.e., Motifs [24] and VCTree [26]). The models with the bias of tail predicates were trained after NICE cleaning (i.e., Motifs+NICE and VCTree+NICE). The models obtained by NIST strategy training were denoted as Motifs+NICE and VCTree+NICE, respectively. With the implementation of multi-teacher knowledge distillation, our proposed method incurs twice the time and GPU space during the training stage compared to the baseline. However, there is no additional overhead during the inference stage.

SGG Training Details: Our NICEST is a model-agnostic strategy, and thus, for different baselines (e.g., Motifs and VCTree), we followed their respective configurations. We adopted the Faster R-CNN [85] with the ResNeXt-101-FPN [87] backbone to detect objects in the image. The parameters of the detector were frozen during training. Besides, we utilized the SGG benchmark provided by [29] for all baselines.

B. Comparisons With State-of-The-Arts

1) Performance on VG: Settings: Since NICE and NIST are model-agnostic strategies, they can be seamlessly incorporated into any advanced SGG model. In this section, we equipped NICE and NIST into two baselines: Motifs [24] and VCTree [26], and compared them with the state-of-the-art SGG methods. According to the generalization of these methods, we group them into two categories: 1) TDE [29], PCPL [41], CogTree [30], DLFE [31], BPL-SA [44] and our previous work NICE-v1 [33]. These methods are all model-agnostic SGG debiasing strategies. For fair comparisons, we also reported their performance on the Motifs and VCTree baselines. 2) KERN [37], G-RCNN [36], MSDN [88], BGNV [39], and DT2-ACBS [43]. These methods are specifically designed for SGG models. All results are reported in Table III. Besides, we reported the \( R@100 \) metric of each predicate category group under the PredCls setting in Table VII and presented the \( R@100 \) across all predicates in appendix, available online. Additionally, a comparison of parameters and computational costs is provided in the appendix, available online.

Results: From the results in Tables III and VII, we have the following observations: 1) Compared to the two strong baselines (i.e., Motifs and VCTree), our NICE can consistently improve model performance on metric \( mR@K \) over all three tasks (e.g., 5.9% \( \sim \) 14.3% and 3.7% \( \sim \) 14.7% absolute gains on metric \( mR@100 \) over Motifs and VCTree, respectively). 2) Compared to the state-of-the-art model-agnostic debiasing strategies (i.e., NICE-v1 [33]), NICE can achieve comparable performance with NICE-v1 on \( m@K \) and better performance on Mean (e.g., 44.3% vs. 43.6% on metric Mean under PredCls over Motifs). This proves that our improvements to the NSC module do make the model more robust to all predicates compared to the original NICE-v1. 3) NICEST achieves the highest performance on the Mean metric across the three tasks (e.g., 45.3% on Mean under PredCls over Motifs). It proves that NIST can realize a better trade-off between accuracy among different predicate categories. 4) NICEST is capable of achieving a better trade-off between head and tail predicates (e.g., 44.7% vs. 41.7% on Avg).

2) Performance on VG-OOD: Settings: Similarly, for the VG-OOD dataset, we also conducted NICE and NIST on the widely utilized baselines: Motifs [24] and VCTree [26]. At the same time, we compared the most classic method (i.e., TDE [29]) and current the state-of-the-art method (i.e., NICE-v1 [33]) of model-agnostic unbiased SGG to this dataset in Table IV. In addition, we visualized the performance comparison of \( R@100 \) across all predicates and the distribution of \( R@100 \) and \( mR@100 \) in the appendix, available online.

Results: From the results in Table IV, we can observe that: 1) Compared with the two common baselines (i.e., Motifs and VCTree), the \( mR@K \) of our NICE has been significantly improved in all three tasks (e.g., 4.4% \( \sim \) 17.0% and 4.7% \( \sim \) 17.0%...
## TABLE III

| B          | Models                              | PredCls        | SGCls        | SGGen        |
|------------|-------------------------------------|----------------|--------------|--------------|
| **VGG-16** |                                     |                |              |              |
|            | Motifs [24]                         | mR@50/100: 17.7 | R@50/100: 35.8 | mR@50/100: 25.7 |
|            | VC Tree [26]                        | R@50/100: 38.1 | 35.8/36.5    | 27.3/30.3 |
|            | KERN [37]                           | 10.1/10.8      | 38.1/38.8    | 17.5/19.4   |
|            | PCPL [41]                           | 9.4/10.0       | 36.7/37.4    | 9.6/11.7    |
|            | MSDN [88]                           | 18.6/19.6      | 27.6/28.4    | 4.6/7.3     |
|            | R-GCN [36]                          | 5.6/8.0        | 27.9/31.3    | 18.5/21.7   |
|            | BGGN [39]                           | 9.0/9.5        | 38.5/37.0    | 28.8/31.3 |
|            | DT2-ACR [43]                        | 9.5/11.7       | 14.6/16.8    | 13.6/15.4   |

## TABLE IV

| B          | Models                              | PredCls        | SGCls        | SGGen        |
|------------|-------------------------------------|----------------|--------------|--------------|
| **VG-OOD** |                                     |                |              |              |
|            | Motifs [24]                         | mR@50/100: 17.7 | R@50/100: 35.8 | mR@50/100: 25.7 |
|            | VC Tree [26]                        | R@50/100: 38.1 | 35.8/36.5    | 27.3/30.3 |
|            | KERN [37]                           | 10.1/10.8      | 38.1/38.8    | 17.5/19.4   |
|            | PCPL [41]                           | 9.4/10.0       | 36.7/37.4    | 9.6/11.7    |
|            | MSDN [88]                           | 18.6/19.6      | 27.6/28.4    | 4.6/7.3     |
|            | R-GCN [36]                          | 5.6/8.0        | 27.9/31.3    | 18.5/21.7   |
|            | BGGN [39]                           | 9.0/9.5        | 38.5/37.0    | 28.8/31.3 |
|            | DT2-ACR [43]                        | 9.5/11.7       | 14.6/16.8    | 13.6/15.4   |

### Results

1) The proposed NICEST can significantly improve the mR@K scores of the two strong baselines (e.g., 12.6% and 12.6% absolute gains on metric mR@100 over Motifs and VCTree, respectively).

2) The NICEST module outperforms the original NICEST-v1 [33] in the comprehensive performance of the Mean metric over three tasks, which proves that the improved NICEST module is indeed more robust to all predicates.

3) NICEST is designed to be independent of the frequency bias in the dataset. In terms of the VG-OOD dataset, NICEST can still achieve the best performance in the Mean metric, showcasing its robust generalization capabilities and its capacity to maintain balanced performance across various predicates.

4) Performance on GQA: Settings: For the GQA dataset, we assessed the performance of the following models:

- Motifs [24], VCTree [26], and their combinations with NICEST-v1 [33], NICE, and NICEST.

The evaluation was conducted on the PredCls task. Moreover, we conducted statistical tests for comparisons on three datasets in the appendix, available online.

**Results:** The results on the GQA dataset are summarized in Table V. From the table, we have the following observations:

1) The incorporation of NICEST significantly improves the mR@K scores of the two strong baselines (e.g., 12.6% absolute gains on metric mR@100 over Motifs and VCTree, respectively).

2) Our NICE and NICEST are able to maintain a high and comparable R@K on the large GQA dataset.

3) NICEST achieves the highest performance on the metric.
The hyperparameter in Neg-NSD is the threshold \( \theta \) for the confidence score (cf., (1)). Particularly, when the threshold \( \theta \) for one category is set to 100\%, which means we never assign this category as pseudo labels. Without loss of generality, we choose three representative hyperparameter settings, i.e., we mine missing annotated triplets on 1) all predicate categories, or 2) only body and tail categories, or 3) only tail categories. The thresholds \( \theta \) for the corresponding head, body, and tail categories were set as 95\%, 90\%, and 60\%, respectively. To disentangle the influence of the other two modules (i.e., Pos-NSD and NSC), we directly use the outputs of Neg-NSD and pristine positive samples for SGG training.

**Results:** From the results in Table VIII(c), we can observe:

1) Different threshold settings have a slight influence on the mR@K metrics, but a relatively more pronounced effect on the R@K metrics. 2) The model gains the best performance when only mining the missing tail predicates in Neg-NSD.

- **Effectiveness of Neg-NSD:** We evaluated the effectiveness of the Neg-NSD by using the same refined samples by Neg-NSD and pristine positive samples for SGG training.

**Results:** All results are reported in Table VIII(a). Compared to the baseline model (#1), the Neg-PSD module (#2) can significantly improve model performance on mR@K metrics (e.g., 25.2\% vs. 17.8\% in mR@100), which proves that these harvested “positive” samples (noisy negative samples with pseudo labels) are indeed beneficial for SGG training.

**2) Ablation Studies on Pos-NSD: Hyperparameter Settings of Pos-NSD:** The hyperparameter in Pos-NSD is the cutoff distance \( d_c \) ranked at \( \alpha \% \) for different categories (cf., (5)). As mentioned in Section III-B, different \( d_c \) directly impacts the clustering results of each predicate category, and smaller \( d_c \) is more suitable for predicates with multiple semantic meanings. Therefore, without loss of generality, we choose five typical settings for different categories, including \((L, L, L), (M, M, M), \) and so on. In our experiments, \( L, M, S \) denote that \( d_c \) is large, medium, and small, respectively. Their corresponding \( \alpha \% \) values were set to 50.0\%, 25.0\% and 12.5\%. Similarly, we disentangle the influence of Neg-NSD and use pristine negative and refined positive samples (outputs of NSC) for SGG training. The results are shown in Table VIII(b).

**Results:** When the cutoff distance \( d_c \) for all predicate categories is set to large or small or from large to small, the model achieves relatively worse results. These results are also consistent with our expectation, i.e., for the predicate categories with multiple semantic meanings (head categories), small \( d_c \) is better for noisy sample detection. Instead, for predicate categories with a unique semantic meaning (tail categories), a larger \( d_c \) is better. Thus, we utilize the \((S, M, L)\) setting for all experiments.

- **Effectiveness of Pos-NSD:** We evaluated the effectiveness of Pos-NSD by using either clean positive samples detected from \( T^+ \) with pristine negative samples or from the new positive samples set \( \tilde{T}^+ \) for SGG training.

**Results:** As shown in Table VIII(a) (#3), the single Pos-NSD component can still improve mR@K metrics using much fewer positive samples. Besides, the baseline can also be exceeded on mR@K (#4) after Neg-NSD and Pos-NSD alone with fewer training samples. It proves that the clean subset divided by Pos-NSD is better for SGG training, and numerous noisy positive samples actually hurt performance.

### Table V

| Models | PredCls | mR@50/100 R@50/100 Mean |
|--------|---------|-------------------------|
| Motifs [24] | 13.9 / 14.7 | 65.1 / 66.8 | 40.1 |
| Motifs+NICE-v1 [35] | 24.8 / 27.4 | 55.8 / 58.5 | 41.6 |
| Motifs+NICE (Ours) | 24.5 / 27.3 | 57.1 / 59.1 | 42.0 |
| Motifs+NSC (Ours) | 24.0 / 26.4 | 57.1 / 60.6 | 42.2 |

The baseline model is Motifs [24].

### Table VI

| Models | PredCls mR@50/100 R@50/100 Mean |
|--------|-------------------------|
| Motifs [2] | 16.5 / 17.6 | 63.5 / 67.2 | 41.8 |
| Motifs+Label Correction [9] | 15.6 / 16.9 | 65.7 / 67.5 | 41.4 |
| Motifs+Label Smoothing [92] | 17.3 / 19.2 | 61.7 / 64.1 | 40.6 |
| Motifs+NICE-v1 [35] | 29.9 / 32.3 | 55.1 / 57.2 | 43.6 |
| Motifs+NICE (Ours) | 30.0 / 32.1 | 56.6 / 58.6 | 44.3 |
| Motifs+NSC (Ours) | 29.5 / 31.6 | 59.1 / 61.0 | 45.3 |

### Table VII

| Models | PredCls | Head | Body | Tail | Avg |
|--------|---------|------|------|------|-----|
| Motifs [2] | 80.5 | 29.3 | 5.7 | 38.5 |
| Motifs+NICE-v1 [35] | 67.9 | 33.1 | 24.2 | 41.7 |
| Motifs+NICE (Ours) | 69.6 | 36.9 | 26.0 | 44.2 |
| Motifs+NSC (Ours) | 72.9 | 35.7 | 25.4 | 44.7 |

AVG is the average of the three groups.
3) Ablation Studies on NSC: Hyperparameter of NSC: The hyperparameter of NSC is the $K$ in $K$NN. We investigated $K = \{1, 3, 5\}$. All results are reported in Table VIII(d). From the results, we can observe that the SGG performance is robust to different $K$. To better make a trade-off of different metrics, we set $K$ to 3.

Effectiveness of NSC: Based on Pos-NSD, NSC replaces the original labels of the noisy positive samples with cleaner and more robust soft labels. As shown in Table VIII(a), NSC markedly improves SGG performance on both mR@K and R@K (# 5 vs. # 3), i.e., NSC reassigns better labels to noisy samples.

Results: Compared with the NSC in NICE-v1 [33], both mR@K and R@K of PredCls are improved by our refined NSC method (33.1% vs. 33.0% in mR@100 and 58.3% vs. 56.9% in R@100, based on VCTree). It proves that our approach is relatively improved across all predicate categories.

T-SNE Visualizations: We visualized the t-SNE distributions of features of \{man-on/riding-skateboard\} on the VG and GQA datasets, comparing the feature space before and after the NSC in Fig. 9. The former is a pair of predicates of different granularity (common-prone), while the latter is a pair of predicates of the same granularity (synonym-random). As shown in Fig. 9, NSC can help to alleviate the inconsistency of ground-truths, i.e., similar visual patterns always have more consistent ground-truth predicate annotations, which is beneficial for SGG training.

D. Ablation Studies on NIST

Hyperparameter of NIST: The hyperparameter of NIST is the weight of two teachers. We evaluated the performance of the three weighted methods in NIST. Among them, “fixed” means that the weights of two teacher models are constant 0.5, “adapt” means that two teacher models adopt adaptive trade-off strategies (cf. (12)), and “group” means that two teacher models...
Fig. 10. Examples of the triplets detected by Negative Noisy Sample Detection that do not appear in the training set.

Fig. 11. Scene graphs generated by Motifs (left) and Motifs+NICE (Middle), Motifs+NICEST (right) on PredCls. Red predicates are error (i.e., not GT and unreasonable), Green predicates are correct (i.e., GT), and Brown predicates are reasonable (not in GT but reasonable). Only detected boxes overlapping with GT are shown.

TABLE X
ABLATION STUDIES (%) ON DIFFERENT WEIGHTING METHODS OF DIFFERENT TEACHER MODELS IN NIST

| Weighting Type | Fixed | Adapt | Group | PredCls | mR@50/100 | R@50/100 | Mean |
|----------------|-------|-------|-------|---------|-----------|----------|------|
|                | ✓     | ✓     | ✓     |         | 28.9 / 31.1 | 57.9 / 59.8 | 44.4 |
|                |       |       |       |         | 30.4 / 32.6 | 57.4 / 59.4 | 45.0 |
|                |       |       |       |         | 29.5 / 31.6 | 59.1 / 61.0 | 45.3 |

Results: In Tables III, IV, and IX, we can find that the model trained by NIST can effectively improve the Mean performance across all the baseline models and datasets (e.g., 36.9% vs. 38.1% and 37.7% vs. 39.9% based on Motifs for TDE on VG and VG-OOD, respectively). Consistent improvement across all models and datasets demonstrates the universality of NIST.

E. Visualization

1) Qualitative Results of Neg-NSD: In Fig. 10, we visualized some “unseen” visual relation triplet categories mined by Neg-NSD that never appear in the original VG dataset. Some of these triplets are easily overlooked by the annotators, such as the relation against between bike and bike, or the relation along between rock and street. These harvested new visual relation triplet categories increase both the number and diversity of samples in tail categories.

2) Qualitative Results of NICE: Fig. 11 demonstrates some qualitative results generated by Motifs, Motifs+NICE, and Motifs+NICEST. From Fig. 11, we can observe that Motifs tends to predict coarse-grained (i.e., head) predicates, such as near, while Motifs+NICE tends to predict fine-grained (i.e., tail) predicates, such as sitting on and covering. The Motifs+NICEST can obtain trade-off predicates that bring predictions closer to the ground-truth.

adopt different adaptive weighting strategies for the head, body, and tail category groups respectively. All results are reported in Table X.

Results: As displayed in Table X, our proposed group weighting strategy achieves the best performance on the Mean metric, proving the effectiveness of the grouping weighting.

Effectiveness of NIST: To prove the generality of NIST, we used the two most common baseline models (i.e., Motifs [24] and VCTree [26]) as teachers biased towards head predicates (i.e., high Recall), and two models trained using unbiased methods (i.e., Motifs-TDE [29] and Motifs-NICE) as teachers biased towards the tail (i.e., high mean Recall). The results of the combination of different teacher models in NIST are reported in Table IX.
VII. CONCLUSIONS AND LIMITATIONS

In this article, we argued that two plausible assumptions about the ground-truth annotations are inapplicable to existing SGG datasets. To this end, we reformulated SGG as a noisy label learning problem and proposed a novel model-agnostic noisy label correction and sample training strategy: NICEST. It is composed of NICE and NIST, which solves the problem of noisy label learning by generating high-quality samples and efficient training, respectively. NICE can not only detect the noisy samples, but also reassign robust soft predicate labels to them. NIST compensates for the tail-biased training defect of NICE by adopting a multi-teacher distillation strategy to enable the model to learn unbiased fusion knowledge. We re-organized the VG dataset to create VG-OOD for a better evaluation of the model generalization capability. Extensive experiments on VG, VG-OOD and QGA datasets prove the effectiveness of each component of NICEST.

Limitations: The use of Multi-Teacher Knowledge Distillation in NICEST may introduce minor computational overhead. In the case of online training, it requires twice the GPU space compared to the baseline. However, with offline training, there is no additional GPU consumption overhead. Additionally, since certain hyperparameters have varying impacts on different predicate categories, it becomes challenging to achieve the optimal trade-off between different evaluation metrics.

REFERENCES

[1] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, “Scene graph generation by iterative message passing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 5410–5419.
[2] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. G. Hauptmann, “A comprehensive survey of scene graphs: Generation and application,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 1–26, Jan. 2023.
[3] L. Li, J. Xiao, G. Chen, J. Shao, Y. Zhang, and L. Chen, “Zero-shot visual relation detection via composite visual cues from large language models,” 2023, arXiv:2305.12476.
[4] L. Li, G. Chen, J. Xiao, and L. Chen, “Compositional zero-shot learning via progressive language-based observations,” 2023, arXiv:2311.14749.
[5] J. Johnson et al., “Image retrieval using scene graphs,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3668–3678.
[6] S. Wang, R. Wang, Z. Yao, S. Shan, and X. Chen, “Cross-modal scene graph matching for relationship-aware image-text retrieval,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., 2020, pp. 1508–1517.
[7] L. Feng and B. Bhama, “Semantic concept co-occurrence patterns for image annotation and retrieval,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 4, pp. 785–799, Apr. 2016.
[8] S. Lee, J.-W. Kim, Y. Oh, and J. H. Jeon, “Visual question answering over scene graph,” in Proc. 1st Int. Conf. Graph. Comput., 2019, pp. 45–50.
[9] J. Shi, H. Zhang, and J. Li, “Explainable and explicit visual reasoning over scene graphs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 8376–8384.
[10] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhuang, “Counterfactual samples synthesizing for robust visual question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10800–10809.
[11] D.-J. Kim, T.-H. Oh, J. Choi, and I. S. Kweon, “Dense relational image captioning via multi-task triple-stream networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 11, pp. 7348–7362, Nov. 2022.
[12] Z. Wang, J. Xiao, L. Chen, F. Gao, J. Shao, and L. Chen, “Learning combinatorial prompts for universal controllable image captioning,” 2023, arXiv:2303.06338.
[13] Z. Wang et al., “Explicit image caption editing,” in Proc. 17th Eur. Conf. Comput. Vis., Tel Aviv, Israel, Springer, 2022, pp. 113–129.
[14] H. Li, H. Zhang, G. Zhou, N. Jiang, and X. Yuan, “PPDL: Predicate probability distribution based loss for unbiased scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 19447–19456.
[15] X. Lin, C. Ding, Y. Zhan, Z. Li, and D. Tao, “HL-Net: Heterophily learning network for scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 19476–19485.
[16] X. Li, L. Chen, G. Chen, Y. Feng, Y. Yang, and J. Xiao, “Decomposed prototype learning for few-shot scene graph generation,” 2023, arXiv:2303.10863.
[17] K. Gao, L. Chen, Y. Niu, J. Shao, and X. Jun, “Classification-then-grounding: Reforumating video scene graphs as temporal bipartite graphs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 19475–19484.
[18] K. Gao, L. Chen, H. Zhang, J. Xiao, and Q. Sun, “Compositional prompt tuning with motion cues for open-vocabulary video relation detection,” 2023, arXiv:2302.00268.
[19] R. Krishna et al., “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” Int. J. Comput. Vis., vol. 123, pp. 32–73, 2017.
[20] D. A. Hudson and C. D. Manning, “QGA: A new dataset for real-world visual reasoning and compositional question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6700–6709.
[21] Z.-S. Hung, A. Mallya, and S. Lazebnik, “Contextual translation embedding for visual relationship detection and scene graph generation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 11, pp. 3820–3832, Nov. 2021.
[22] F. Shao et al., “Improving weakly supervised object localization via causal intervention,” in Proc. 29th ACM Int. Conf. Multimedia, 2021, pp. 3321–3329.
[23] F. Shao et al., “Deep learning for weakly-supervised object detection and localization: A survey,” Neurocomputing, vol. 496, pp. 192–207, 2022.
[24] R. Zellers, M. Yateskar, S. Thomson, and Y. Choi, “Neural motifs: Scene graph parsing with global context,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 5831–5840.
[25] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang, “Counterfactual critic multi-agent training for scene graph generation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 4613–4623.
[26] K. Tang, H. Zhang, B. Wu, W. Liu, and W. Liu, “Learning to compose dynamic tree structures for visual contexts,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6619–6628.
[27] X. Lin, C. Ding, J. Zeng, and D. Tao, “GPS-Net: Graph property sensing network for scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3746–3753.
[28] L. Li, G. Chen, J. Xiao, Y. Yang, C. Wang, and L. Chen, “Compositional feature augmentation for unbiased scene graph generation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2023, pp. 21685–21695.
[29] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, “Unbiased scene graph generation from biased training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3716–3725.
[30] J. Yu, Y. Chai, Y. Hu, and Q. Wu, “Cogtree: Cognition tree loss for unbiased scene graph generation,” in Proc. Int. Joint Conf. Artif. Intell., 2021, pp. 1274–1280.
[31] M.-J. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng, “Recovering the unbiased scene graphs from the biased ones,” in Proc. ACM Int. Conf. Multimedia, 2021, pp. 1581–1590.
[32] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, “Visual relationship detection with language priors,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 852–869.
[33] L. Li, L. Chen, Y. Huang, Z. Zhang, S. Zhang, and J. Xiao, “The devil is in the labels: Noisy label correction for robust scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 18869–18878.
[34] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, “Visual translation embedding network for visual relation detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 5532–5540.
[35] Y. Li, W. Ouyang, X. Wang, and X. Tang, “VIP-CNN: Visual phrase guided convolutional neural network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1347–1356.
[36] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, “Graph R-CNN for scene graph generation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 670–685.
[37] T. Chen, W. Yu, R. Chen, and L. Lin, “Knowledge-embedded routing network for scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6163–6171.
[38] L. Misra, C. Lawrence Zitnick, M. Mitchell, and R. Girshick, “Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2930–2939.
[39] R. Li, S. Zhang, B. Wan, and X. He, “Bipartite scene graph network with adaptive message passing for unbiased scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 11109–11119.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.

[41] S. Yan et al., “PCPL: Predicate-correlation perception learning for unbiased scene graph generation,” in Proc. ACM Int. Conf. Multimedia, 2020, pp. 265–273.

[42] B. Knyazev, H. de Vries, C. Cangea, G. W. Taylor, A. Courville, and E. Belilovsky, “Diph density-aware losses for novel composition in scene graph generation,” 2020, arXiv: 2005.08230.

[43] A. Desai, T.-Y. Wu, S. Tripathi, and N. Vasconcelos, “Learning of visual relations: The devil is in the tails,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 15404–15413.

[44] Y. Guo et al., “From general to specific: Informative scene graph generation via graph planner,” in Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 16383–16392.

[45] L. Li et al., “Label semantic knowledge distillation for unbiased scene graph generation,” 2022, arXiv:2208.03763.

[46] G. Chen, L. Li, Y. Luo, and J. Xiao, “Addressing predicate overlap in scene graph generation with semantic granularity controller,” in Proc. IEEE Int. Conf. Multimedia Expo, 2023, pp. 78–83.

[47] T.-J. Wang, S. Pehlivan, and J. Laaksonen, “Tackling the unannotated: Scene graph generation with bias-reduced models,” in Proc. Brit. Mach. Vis., 2020.

[48] J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer,” in Proc. Int. Conf. Learn. Representations, 2017, pp. 1–9.

[49] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, “MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in Int. Conf. Mach. Learn., 2018, pp. 2304–2313.

[50] K.-H. Lee, X. He, L. Zhang, and L. Yang, “CleanNet: Transfer learning for scalable image classifier training with label noise,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 5447–5456.

[51] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 4334–4343.

[52] A. Vahdat, “Toward robustness against label noise in training deep discriminative neural networks,” in Proc. Conf. Workshop Neural Inf. Process. Syst., 2017.

[53] Y. Li, Y. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, “Learning from noisy labels with distillation,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 1910–1918.

[54] 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.

[55] X. Ma et al., “Distributionally-driven learning with noisy labels,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 3355–3364.

[56] Z. Zhang and M. R. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels,” in Proc. Conf. Workshop Neural Inf. Process. Syst., 2018, pp. 8792–8802.

[57] Y. Wang, X. Ma, Z. Chen, Y. Lao, J. Yi, and J. Bailey, “Symmetric cross entropy for robust learning with noisy labels,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 322–330.

[58] Y. Xu, P. Cao, Y. Kong, and Y. Wang, “L_DMI: An information-theoretic noise-robust loss function,” 2019, arXiv:1909.03388.

[59] L. Wang and K.-J. Yoon, “Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 6, pp. 3048–3068, Jun. 2022.

[60] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, “Revisiting knowledge distillation via label smoothing regularization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.

[61] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, “Scene graph generation from objects, phrases and region proposals,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4971–4980.

[62] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” 2016, arXiv:1610.02136.

[63] T. DeVries and G. W. Taylor, “Learning confidence for out-of-distribution detection in neural networks,” in 2018, arXiv:1802.04965.

[64] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” Science, vol. 344, pp. 1492–1496, 2014.

[65] K. Hechenbichler and K. Schliep, “Weighted k-nearest-neighbor tech- niques and ordinal classification,” discussion paper, vol. 399, no. 1, 2004.

[66] S. Guo et al., “CurriculumNet: Weakly supervised learning from large-scale web images,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 135–150.

[67] S. Kullback and R. A. Leibler, “On information and sufficiency,” Ann. Math. Statist., vol. 22, pp. 79–86, 1951.

[68] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Proc. Conf. Workshop Neural Inf. Process. Syst., 2015, pp. 91–99.

[69] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[70] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.

[71] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, “Scene graph generation from objects, phrases and region captions,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 1261–1270.

[72] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, “Large-scale long-tailed recognition in an open world,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2537–2546.

[73] X. Dong, T. Gan, X. Song, J. Wu, Y. Cheng, and L. Nie, “Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 19427–19436.

[74] X. Wang, Y. Hu, E. Kodirov, D. A. Clifton, and N. M. Robertson, “Pro- Seller: Progressive self label correction for training robust deep neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 752–761.

[75] R. Müller, S. Kornblith, and G. E. Hinton, “When does label smoothing help?”, in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 4696–4705.
Lin Li received the BS degree from the College of Software, Qufu Normal University, Qufu, China, in 2019. She is currently working toward the PhD degree with the College of Computer Science at Zhejiang University, Hangzhou, China. Her current research interests include deep learning and computer vision.

Jun Xiao received the PhD degree in computer science and technology from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2007. He is currently a professor with the College of Computer Science, Zhejiang University. His current research interests include computer animation, multimedia retrieval, and machine learning.

Hanrong Shi received the BEng degree in computer science from Zhejiang University, Hangzhou, China, in 2022. He is currently working toward the master degree with the College of Computer Science, Zhejiang University. His current research interests include computer vision and deep learning.

Hanwang Zhang received the BEng degree in computer science from Zhejiang University, Hangzhou, China, in 2009, and the PhD degree in computer science from the National University of Singapore, in 2014. He is currently an assistant professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He was a research scientist with the Department of Computer Science, Columbia University, USA. His research interest includes computer vision, multimedia, and social media. He is the recipient of the Best Demo runner-up award in ACM MM 2012, the Best Student Paper award in ACM MM 2013, and the Best Paper Honorable Mention in ACM SIGIR 2016, and TOMM best paper award 2018. He is the winner of Best PhD Thesis Award of School of Computing, National University of Singapore, 2014.

Yi Yang (Senior Member, IEEE) received the PhD degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently a distinguished professor with Zhejiang University, China. He was a professor and director of the ReLER Lab, Australian Artificial Intelligence Institute (AAIL), University of Technology Sydney, Australia. He was a post-doctoral research with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interest include machine learning and its applications to multimedia content analysis and computer vision, such as multimedia indexing and retrieval, surveillance video analysis and video semantics understanding.

Wei Liu (Fellow, IEEE) received the PhD degree in EECS from Columbia University, USA. He is currently a distinguished scientist with Tencent, China. He takes the director of Ads Multimedia AI with Tencent Data Platform, and has taken the director of computer vision with Tencent AI Lab. He was a research scientist and staff member of IBM T. J. Watson Research Center, USA. He has long been devoted to fundamental research and technology development in core fields of AI. His research work wins numerous awards and honors, e.g., the 2011 Facebook Fellowship, the 2012 IBM Josef Raviv Memorial Postdoctoral Fellowship, the 2013 Jury Award for Best Thesis of Columbia University, the 2016 and 2017 SIGIR Best Paper Award Honorable Mentions, the 2018 "AI’s 10 to Watch" Award, the 2020-2021 World’s Top 2% Scientists honor, etc. He is an associate/action editor of internationally leading AI journals IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), IEEE Intelligent Systems, and Transactions on Machine Learning Research (TMLR), and is an area chair of top-tier computer science and AI conferences NeurIPS, ICML, CVPR, ICCV, IJCAI, and AAAI. He is a fellow of the IAPR and IMA, and is an elected member of the ISI.

Long Chen (Member, IEEE) received the BEng degree in electrical information engineering from the Dalian University of Technology, in 2015, and the PhD degree in computer science from Zhejiang University, in 2020. He is currently an Assistant Professor with The Hong Kong University of Science and Technology (HKUST). He was a postdoctoral research scientist with Columbia University and a senior researcher with Tencent AI Lab. His research interests are computer vision, machine learning, and multimedia.