Label Semantic Knowledge Distillation for Unbiased Scene Graph Generation

Lin Li, Jun Xiao, Hanrong Shi, Wenxiao Wang, Jian Shao, An-An Liu, Senior Member, IEEE, Yi Yang, Senior Member, IEEE, and Long Chen

Abstract—The Scene Graph Generation (SGG) task aims to detect all the objects and their pairwise visual relationships in a given image. Although SGG has achieved remarkable progress over the last few years, almost all existing SGG models follow the same training paradigm: they treat both object and predicate classification in SGG as a single-label classification problem, and the ground-truths are one-hot target labels. However, this prevalent training paradigm has overlooked two characteristics of current SGG datasets: 1) For positive samples, some specific subject-object instances may have multiple reasonable predicates. 2) For negative samples, there are numerous missing annotations. Regardless of the two characteristics, SGG models are easy to be confused and make wrong predictions. To this end, we propose a novel model-agnostic Label Semantic Knowledge Distillation (LS-KD) for unbiased SGG. Specifically, LS-KD dynamically generates a “soft” label for each subject-object instance by fusing a predicted Label Semantic Distribution (LSD) with its original one-hot target label. LSD reflects the correlations between this instance and multiple predicate categories. Meanwhile, we propose two different strategies to predict LSD: iterative self-KD and synchronous self-KD. Extensive ablations and results on three SGG tasks have attested to the superiority and generality of our proposed LS-KD, which can consistently achieve decent trade-off performance between different predicate categories.

Index Terms—Unbiased scene graph generation, label confusion, knowledge distillation.

I. INTRODUCTION

Scene Graph Generation (SGG), i.e., detecting all the objects (including both localization and classification) and classifying their pairwise visual relationships in a given image, is a challenging and fundamental scene understanding task [1]. Typically, a scene graph can be formulated as a series of subject-predicate-object triplets (e.g., “man-on-chair” in Fig. 1(a)). Due to its symbolic representation nature, SGG has played an important role in many downstream tasks, e.g., image captioning [2], image retrieval [3], [4], visual question answering [5], and so on [6]. Meanwhile, with the release of several large-scale SGG datasets, e.g., Visual Genome (VG) [7]) and the maturity of advanced object detectors [8], SGG has raised unprecedented attention over the recent few years, and hundreds of SGG models have been proposed [9], [10], [11], [12].

However, one notorious challenge in existing SGG benchmarks is the highly-skewed long-tailed relation annotations, i.e., the head predicate categories have much more annotated ground-truth samples (or relation triplets) than the tail ones. Thus, a recent surge of interest in SGG is developing unbiased scene graph generation models, which can achieve more balanced performance on all predicate categories (i.e., head, body, and tail categories). Currently, existing state-of-the-art unbiased SGG models can be coarsely grouped into four categories: 1) Class-aware Re-sampling: They try to sample more tail category training samples to balance the data distribution [13]. 2) Loss Re-weighting: They aim to reduce the contributions of the head training samples in the loss calculation by designing different weighting strategies [14], [15]. 3) Biased-model-based: They disentangle unbiased representations from some biased trained models [16], [17], [18], [19]. 4) Noisy Label Learning: They first detect noisy annotated samples and then reassign them to more high-quality predicate labels [20], [21]. Although different advanced techniques have been proposed in each category, almost all of them follow the same training paradigm: regarding the object and predicate classification as a single-label classification problem by assigning each triplet sample only one predicate label, and using the cross-entropy loss (between predicted distributions and one-hot target labels) as the training objective.

Although today’s unbiased SGG models have achieved significant progress on some challenging SGG benchmarks (e.g., VG), it is worth noting that this training paradigm has overlooked two characteristics of existing SGG datasets: 1) For brevity, we use “tail”, “body”, and “head” categories to represent the predicate categories that appear in the tail, body, and head parts of the number distributions of different ground-truth predicate annotations in SGG datasets.
For instance, "boy-shirt" in Fig. 1(b) are all reasonable predicates in these "man-chair" pairs. 2) For the negative samples, there are numerous missing annotations [22], and all these negative samples are treated as absolute background category in the training process. For instance, "boy-shirt", "boy-helmet" in Fig. 1(b) are all missing-annotated subject-object instances. Similarly, some "negative" samples may even have multiple reasonable predicates (e.g., "wearing" and "has" for "boy-shirt").

By training the predicate classifier in SGG models with one-hot target labels, these neglected characteristics will respectively result in two adverse impacts: 1) It may lead to confused SGG predictions by simply assigning one-hot target labels without considering multiple reasonable predicates for each subject-object pair. Meanwhile, some over-confident predictions on the head categories also enlarge the performance gap between different categories. 2) It may lead to wrong predictions by directly assigning absolute background (i.e., one-hot target labels) to the missing-annotated subject-object pairs. Since the total probability is given to the wrong background, the model trusts the wrong label excessively.

To alleviate these issues, in this paper, we propose a novel model-agnostic Label Semantic Knowledge Distillation for unbiased SGG, dubbed LS-KD. Specifically, LS-KD integrates label semantic distributions (LSD, i.e., the knowledge that reflects the correlations between subject-object instances and predicate categories) and original one-hot target labels to "soft" labels, and uses these "soft" labels to supervise the model training. On one hand, LS-KD can model the correlation between multiple labels, reducing the impact of confused predictions. On the other hand, LS-KD can assign other possible labels to unannotated samples, avoiding the problem of directly assigning absolute background to missing-annotated subject-object pairs. Particularly, we utilize self-knowledge distillation (self-KD) to predict these LSDs, by transferring label semantic knowledge from a teacher model to the student model (i.e., the model used for final prediction). Meanwhile, we propose two different self-KD strategies: 1) Iterative self-KD: the student model trained at an earlier step is utilized as the teacher model iteratively. 2) Synchronous self-KD: the student and teacher models share the same backbone with two different classifiers, and both models are trained synchronously. Thanks to these designs, the student model can utilize the past and current predictions (i.e., LSD) of the teacher model in iterative and synchronous self-KD as informative supervision, respectively. Besides, we fuse the output of the teacher model with the original one-hot labels to generate soft target labels, which differs from other fusion strategies in the standard self-KD methods [23], [24], [25].

We evaluate our method on the most widely-used SGG benchmark: VG [7]. Since LS-KD is a model-agnostic debiasing method, it can be seamlessly incorporated into any advanced SGG architectures and consistently improve their performance. Extensive ablations and results on multiple SGG tasks and backbones have shown the effectiveness and generalization ability of LS-KD.

In summary, we make three main contributions in this paper:

1) We are the first to analyze the reason why the prevalent SGG training paradigm may not be optimal, i.e., the effect of treating predicate classification in SGG as a single-label classification, going beyond previous work that only discussed the existence of multiple reasonable predicates for each subject-object pair.

---

2 We use "positive" and "negative" samples to denote annotated ground-truth visual relation triplets with foreground and background predicate categories, respectively. Foreground categories indicate the predicate categories annotated in the dataset indicating the existence of interaction between the subject and object. Background categories are those in which the dataset assumes no interaction between the subject and object.

3 The concept of "model-agnostic" means that methods that are general and can be easily applied to existing SGG models to improve their performance.
2) We propose a novel model-agnostic LS-KD, which can capture the correlations between the subject-object instances and different predicate categories.

3) We conduct extensive experiments to prove the superiority of LS-KD, which can achieve a decent trade-off performance between different predicates.

II. RELATED WORK

A. Scene Graph Generation (SGG)

Scene graphs are structural representations that transform raw visual input into semantic categories [26]. After Lu et al. [22] firstly brought a large-scale dataset to attention, plentiful studies have been conducted in SGG. Existing SGG works can be divided into two groups: 1) Two-stage SGG: They first utilize a detector to obtain object proposals, then predict the classes of objects and their pairwise relations [1], [27], [28], [29], [30], [31], [32], [33], [34]. In order to exploit the rich global context, some works regard each image as a whole and adopt the fully connected graph [31], [32], the chained graph [30], and the tree-structured graph [33] to model the contexts among objects. 2) One-stage SGG: They use the fully convolutional network or Transformer to detect the objects and relations from image features directly [35], [36]. In this paper, we propose a model-agnostic debiasing method that can be used in any SGG model. In our experiments, two popular two-stage SGG models (Motifs [30] & VCTree [33]) are used as baselines.

B. Unbiased Scene Graph Generation

Due to the long-tailed distribution and other language bias issues [37] in SGG, recent SGG work mainly focuses on unbiased SGG [16]. They can be roughly divided into four categories: 1) Class-aware Re-sampling: They balance the data distribution by oversampling samples of tail categories or undersampling of head ones, or a combination of both [13]. Specially, Desai et al. [38] resample both predicate and entity categories to balance the distribution of triplets. 2) Loss Re-weighting: They assign different weights to different categories to reduce the contributions of the head categories in loss calculation. These designed weights are often based on prior knowledge [15] or learnable predicate correlations [14]. 3) Biased-model-based: They separate unbiased predictions from biased trained models [16], [17]. For instance, Tang et al. [16] utilize counterfactual causality to separate biases from the biased model. 4) Noisy Label Correction: They reformulate SGG as a noisy label learning problem and improve annotation qualities. For example, Li et al. [20], [21] propose the NICE and NICEST strategy that can not only detect noisy samples but also reassign more high-quality labels. In this paper, we try to capture the possibility between the instance and multiple relation labels, which achieves a trade-off between the head and tail categories.

C. Knowledge Distillation

Knowledge Distillation (KD) is a training strategy inspired by the “teacher-student” learning behavior, in which the teacher model is the exporter of knowledge and the student model is the receiver [39], [40]. KD that can effectively learn a lightweight student model from a cumbersome teacher model has been widely used in numerous scene understanding applications, e.g., VQA [41], scene graph generation [42]. Traditional KD methods use knowledge from larger, better-performing teacher models to generate soft targets for student networks. Compared with conventional KD methods, self-knowledge distillation (self-KD) reduces the necessity of training a complicated teacher model in advance. It directly completes the distillation process with only the student network itself (i.e., the teacher model and student model share the same network), and it has shown conspicuous performance [43], [44], [45]. Since self-KD can capture the correlations between categories [46] and it does not require any additional networks, in this paper, we design two self-KD strategies (i.e., iterative and synchronous) to predict the label semantic distribution.

D. Label Smoothing and Label Confusion

Label smoothing (LS) strategy [47], [48] is a regularization method, which has been widely used in classification tasks to prevent over-confident prediction and improve generalization ability. It typically combines a weighted uniform noise distribution with original one-hot target labels to obtain some softer labels for training. However, such soft labels generated merely by adding noise cannot reflect the correlations between each subject-object instance and multiple predicate categories. Label confusion (LC) method [49] is proposed in text classification task. It trains a label encoder to learn label representations and calculates the similarity between instances and labels to generate a better label distribution for training. Among these solutions, [49] is the closest to our approach, but we do not need to design additional network encoders to capture correlations between instances and multiple predicates. In addition, some semi-supervised SGG methods also assign soft labels to unannotated samples [50], [51].

III. APPROACH

A. Scene Graph Definition

Given an image $I$, a scene graph is formally represented by $G = \{N = \{o_i; b_i\}; E = \{r_{ij}\}\}$, where the $N$ and $E$ denote the set of nodes and the set of edges, respectively. $o_i \in O$ denotes the object category of the $i$-th object, and its bounding box is $b_i \in B$. $r_{ij} \in R$ denotes the relation category between $i$-th object and $j$-th object. $B$, $O$, and $R$ represent the set of all predicted bounding boxes, object categories, and relation categories, respectively.

In this section, we first revisit the general SGG baseline in Section III-B, then present details of the proposed Label Semantic Knowledge Distillation (LS-KD) in Section III-C, and details of two self-KD strategies in Section III-D.

B. Revisit Scene Graph Generation Baselines

As shown in Fig. 2, the most common SGG framework typically consists of three modules: object proposal, object
classification and relation classification. These frameworks decompose the SGG target $P(G|I)$ into three steps:

$$P(G|I) = P(B|I)P(O|B, I)P(R|O, B, I),$$  \(1\)

where $P(B|I)$ is predicted by the object proposal module, $P(O|B, I)$ is predicted by the object classification module, and $P(R|O, B, I)$ is predicted by the relation classification module. In the following, we briefly introduce these modules.

1) Object Proposal Module: This module is to model $P(B|I)$. Given an image $I$, bounding box $b_i$ of each proposal is obtained by using a standard object detector (e.g., Faster R-CNN [8]). Visual feature $v_i$ corresponding to each proposal region is extracted with RoIAlign operation [52]. Similarly, we compute the union box $b_{ij}$ of the $i$-th and $j$-th detected object region and extract the union visual features $u_{ij}$ in the same way. The union region provides contextual information about the interactions between objects to aid relation classification.

2) Object Classification Module: It is composed of an object context encoder and an object classifier to model $P(O|B, I)$. The object context encoder $Enc_{rel}$ takes the object ROI features $\{v_i\}$ and bounding boxes $\{b_i\}$ as input and outputs the object context representation $\{f_i\}$:

$$f_i = Enc_{rel}(v_i \oplus b_i),$$  \(2\)

where $\oplus$ denotes concatenation operation. Some SGG work adopts bi-directional LSTMs [30], bi-directional TreeLSTMs [33] or fully connected layers [53], [54] to encode context information among objects. The object classifier utilizes the fully-connected layer $FC$ followed by a softmax operation with temperature $\tau$ to model $P(B|O, I)$ and predict object classes $\{o_i\}$:

$$FC(W, \sigma, f) = W^T f + \sigma,$$

$$\hat{y}_i = \text{Softmax}(FC(W_o, f_i, \sigma_o), \tau),$$  \(3, 4\)

where $W$ and $\sigma$ are learnable weight and bias of the fully-connected layer, $\hat{y}_i$ is the predicted label distribution of the object $o_i$.

3) Relation Classification Module: It consists of two parts: a relation context encoder and a relation classifier to model $P(R|O, B, I)$. The relation context encoder $Enc_{rel}$ takes the joint feature embeddings $\{x_i\}$ as input and outputs the refined object features $\{f'_i\}$:

$$x_i = v_i \oplus f_i \oplus e_i,$$

$$f'_i = Enc_{rel}(x_i),$$  \(5, 6\)

where $e_i$ is the word embedding of object class $o_i$. The pairwise refined object features mixed with union feature $u_{ij}$ to obtain the relation representation $f_{ij}$ between the $i$-th and $j$-th objects. The relation classifier is similar to object classifier but with different categories to predict the label distribution:

$$f_{ij} = [f'_i \oplus f'_j] \oplus u_{ij},$$

$$\hat{y}_{ij} = \text{Softmax}(FC(W_r, f_{ij}, \sigma_r), \tau),$$  \(7, 8\)

where $\oplus$ denotes element-wise multiplication, $\hat{y}_{ij}$ is the predicted label distribution of predicate $r_{ij}$. $W_r$ and $\sigma_r$ are parameters of $FC$ for relation classification.

4) Training Strategy: The object proposal module is frozen during training, while the two classification modules are trained using standard cross-entropy (XE) loss $L_o$ and $L_r$ with one-hot object and relation target labels, respectively, i.e.,

$$L_o = \text{XE}(y^o, \hat{y}^o) = -y^o \log(\hat{y}^o),$$

$$L_r = \text{XE}(y^r, \hat{y}^r) = -y^r \log(\hat{y}^r),$$  \(9, 10\)

where $y^o$ and $y^r$ denotes the object and relation one-hot target labels respectively, $\hat{y}^o$ and $\hat{y}^r$ is the predicted object and relation label distribution.

C. LS-KD Relation Classification

To alleviate the issues mentioned in Section I caused by the training relation classifier just with one-hot target labels, we design an LS-KD relation classification to replace the original relation classification module, as shown in Fig 3. Its targets are soft labels that consider the correlations between instances.
and multiple predicate categories. Taking triplet “man-on-chair” in Fig. 3 as an example, LS-KD can assign a portion of probability to predicate labels related to the “man-chair”, such as “sitting on”, “in” and “above” rather than give full probability to the one-hot label “on”. Compared to the basic relation classification, the LS-KD relation classification adds an extra LS-KD module.

The label semantic knowledge distillation module is constructed by a semantic filter block and a simulated label distribution computing block. The former filters the output of the teacher model classifier trained by self-KD, and the latter is designed to generate soft targets for student training. In this section, we will introduce the two blocks, and the training objective of LS-KD. The details of self-KD strategies will be described in Section III-D.

1) Semantic Filter: Inspired by the performance gains that statistical knowledge can bring to predicate prediction [30], in this block, we use the statistical knowledge in the dataset to filter less likely predicates output by the teacher classifier. Specifically, we first calculate the predicate-subject co-occurrence matrix $A^i$ and the predicate-object co-occurrence matrix $A^o$ before training. And due to the severe long-tailed predicate distribution in the dataset, we adopt a log function to soften the matrix as statistic knowledge $\log\log z_{ij}$.

$$b_{ij} = -\log \frac{\log(A^i + \epsilon)}{\sum_{c \in C} \log(A^i_{cj} + \epsilon)} - \log \frac{\log(A^o_{ij} + \epsilon)}{\sum_{c \in C} \log(A^o_{ij} + \epsilon)},$$  \hspace{1cm} (11)

where $C$ represents the predicate categories, $b_{ij}$ denotes the statistic knowledge between subject $i$ and object $j$. Then, the teacher model classifier takes the relation representations \{fi\} extracted by the relation encoder (cf. III-B) as an input and outputs predicate logits. Finally, the calculated statistic knowledge $b_{ij}$ is added to predicate logits to filter less probable predicates. After adopting softmax function to refined predicate logits $z_{ij}$, the Label Semantic Distributions (LSDs) is obtained. The LSD can dynamically capture the correlations among multiple predicate categories for each instance.

$$z_{ij} = FC(W, f_{ij}, \sigma) + b_{ij},$$ \hspace{1cm} (12)

$$y^t_{ij} = \text{Softmax}(z_{ij}, \tau),$$ \hspace{1cm} (13)

where $y^t_{ij}$ is the output LSD of the teacher model classifier.

2) SLD Computing: With the early supervision of the original single label, the LSD of the teacher model can reflect the correlations between each subject-object pair and multiple predicate categories. However, due to the long-tail predicate distribution, there is often a large probability gap between head and tail predicates, which can result in the excessive suppression of reasonable tail predicates. To address this issue, in this block, we adopt a controlling hyperparameter $\alpha$ to merge the original one-hot vector and LSD and then utilize a softmax function to generate the final Simulated Label Distribution (SLD). The SLD is used as the soft target label to replace the one-hot vector to supervise the training of the relation classifier. The hyperparameter $\alpha$ determines the proportion of the final fusion between the LSDs and the original one-hot target labels:

$$y^r_{ij} = \text{Softmax}(\alpha y^t_{ij} + y_{ij}, \tau),$$ \hspace{1cm} (14)

where $y^r_{ij}$ is the original one-hot vector of the predicate label, and $y^t_{ij}$ is the computed SLD. The learned simulated label distribution not only contributes to better representing the instance with multiple confusing predicate labels, but also it is more robust to missing-annotated triplets. For the missing-annotated subject-predicate-object triplets, a portion of the probability of the original one-hot background label is allocated to some reasonable foreground predicate labels (e.g., “wearing”, “in”, and other predicates for the “man-chair” pair in Fig. 4), thus the model can still learn some label semantic knowledge.

3) Training Objective: The supervision used in LS-KD for student model is the simulated label distribution $y^r_{ij}$. During training, we adopt KL-divergence [55] loss to measure the similarity between two probability distributions (the SLD $y^t_{ij}$ and the predicted relation label distribution $\hat{y}_{ij}$):

$$L_r = KL(y^r, \hat{y}) = \sum_{c} y^r_c \log(y^r_c),$$ \hspace{1cm} (15)

where $\hat{y}_{ij}$ is the output of the student model classifier.
labels. In the case of the “chair-table” example illustrated in Fig. 10, the maximum value of $y_{ij}$ is assigned to the “at” predicate, which corresponds to the ground truth (i.e., the best answer). In addition, our method also considers other reasonable predicates (i.e., a fair but generally good answer) predicates, such as “near”, “under”, and “in front of”, which are adaptively assigned proportional probabilities in the label distribution $y_{ij}$.

D. Self-Knowledge Distillation for LSD

Since the probabilistic output of the teacher model in self-KD can reflect the correlations of multiple predicate labels for each instance, it does not require the design of an extra model structure (i.e., the student and teacher models share the same network). In this paper, we explore two different self-KD strategies: iterative self-KD and synchronous self-KD, to prompt the teacher model to predict label semantic distributions. The pipelines are shown in Fig. 5.

1) Iterative Self-Knowledge Distillation: During the training, the student model becomes the teacher model itself iteratively. By iterative self-KD, the student model can incorporate its past predictions to serve more informative supervisions (i.e., label semantic knowledge) for training. Concretely, the student model of the $(t-1)$-th epoch becomes the teacher model during training in $(t)$-th epoch, and the parameters of the teacher model are frozen. The outputs of the teacher model are fused with one-hot target labels by hyperparameter $\alpha$ in the LS-KD module to generate soft labels. These soft labels are utilized to supervise the student model training with KL loss. Especially when training at the first epoch, we just utilize a cross-entropy loss with the one-hot label to train the student model.

2) Synchronous Self-Knowledge Distillation: During the training, the student model and the teacher model share the same relation encoder. However, they adopt two pseudo-siamese classifier heads (i.e., the networks with the same structure but different parameters): the teacher distill head and the student classifier head. These two heads take the same relation representations $\{f_{ij}\}$ as input. The difference is that the teacher distill head is trained by cross-entropy loss with one-hot labels. Similarly, the output of the teacher distill head is used in the LS-KD to generate soft target labels to supervise the training of the student classifier head by KL loss.

   a) Comparison with prior self-KD models: Our fusion strategy of the original one-hot label and the output of teacher model differs from all the standard self-KD methods [23], [24], [25]. More importantly, our motivation and main contributions are to model the correlation between multiple predicates for each sub-obj instance. To achieve the goal, we resorted to KD to generate soft labels for supervised training. In addition, to avoid designing additional model structures, we adopted self-KD strategy and proposed two variants.

IV. Experiments

A. Experimental Settings and Details

1) Datasets: We conducted the experiments on the most prevalent SGG benchmark: Visual Genome (VG) [7] that totally contains 108,073 images. Since the number of samples in most predicate categories is quite limited, in our experiments, we followed the widely-used data splits [1], which consists of 150 object categories and 50 predicate categories. The dataset is split with 70% of images as the training set as the validation set used for hyperparameter tuning and the other 30% as the testing set, respectively. Following the same protocol in [30], we sampled 5,000 images from the training set as the validation set used for hyperparameter tuning.

2) Tasks: We evaluated the model on the three SGG tasks [1]: 1) Predicate Classification (PredCls): Given the ground-truth object bounding boxes with ground-truth object classes, the model is required to predict the predicate categories of object pairs. 2) Scene Graph Classification (SGCls): Given the ground-truth object bounding boxes without classes, the model is required to predict both the object categories and predicate categories. 3) Scene Graph Generation (SGGen): Given an image, the model is required to detect all object bounding boxes, and predict categories of both the object and predicate.

3) Metrics: We evaluated all results on three metrics: 1) Recall@K (R@K): It calculates the fraction of the top $K$ confident predicted subject-relation-object triplets that match
and thus Mean is a comprehensive metric to measure the performance on different predicate categories [20].

| Models                  | PredCls | SGCS | SGGen |
|-------------------------|---------|------|-------|
|                         | mR@50/100 | R@50/100 | Mean | mR@50/100 | R@50/100 | Mean |
| MSDN [57]               | 15.9/17.5 | 64.4/86.6 | 41.2 | 9.3/9.7  | 38.4/38.7 | 24.3 | 6.1/7.2  | 31.5/38.6 | 20.5 |
| G-RCNN [31]             | 16.4/17.2 | 64.6/86.7 | 41.3 | 9.0/9.5  | 38.5/37.0 | 23.5 | 5.8/6.6  | 29.7/32.8 | 18.7 |
| DT2ACBS [38]            | 35.9/39.7 | 23.3/25.6 | 31.1 | 24.8/27.5 | 16.2/17.6 | 21.5 | 22.0/24.4 | 15.0/16.3 | 19.4 |

Motifs [30]              | 16.5 / 17.8 | 65.5 / 67.2 | 41.8 | 8.7 / 9.3 | 39.0 / 39.7 | 24.2 | 5.5 / 6.8 | 32.1 / 36.9 | 20.3 |
Motifs + TDE [16]        | 24.2 / 27.9 | 45.0 / 50.6 | 36.9 | 13.1 / 14.9 | 27.3 / 29.5 | 21.2 | 9.2 / 11.1 | 17.3 / 20.8 | 14.6 |
Motifs + PCPL [14]       | 24.3 / 26.1 | 54.7 / 56.5 | 40.4 | 12.0 / 12.7 | 35.3 / 36.1 | 24.0 | 10.7 / 12.6 | 27.8 / 31.7 | 20.7 |
Motifs + CogTree [18]    | 26.4 / 29.0 | 35.6 / 36.8 | 32.0 | 14.9 / 16.1 | 21.6 / 22.2 | 18.7 | 10.4 / 11.8 | 20.0 / 22.1 | 16.1 |
Motifs + DLPE [17]       | 26.9 / 28.8 | 52.5 / 54.2 | 40.6 | 15.2 / 15.9 | 32.3 / 33.1 | 24.1 | 11.7 / 13.8 | 25.4 / 29.4 | 20.1 |
Motifs + BPL-SA [19]     | 29.7 / 31.7 | 50.7 / 52.5 | 41.2 | 16.5 / 17.5 | 30.1 / 31.0 | 23.8 | 13.5 / 15.6 | 23.0 / 26.9 | 19.8 |
Motifs + LS-KD (Iter)    | 31.9 / 36.7 | 48.0 / 51.6 | 42.1 | 17.9 / 20.4 | 29.4 / 30.9 | 24.1 | 13.6 / 15.9 | 22.7 / 26.7 | 19.7 |
Motifs + LS-KD (Syn)     | 34.2 / 37.9 | 47.6 / 50.9 | 42.7 | 18.7 / 20.9 | 31.2 / 32.7 | 25.9 | 13.7 / 16.6 | 23.5 / 27.2 | 20.3 |
VCTree [33]              | 17.1 / 18.4 | 65.9 / 67.5 | 42.2 | 10.8 / 11.5 | 45.6 / 46.5 | 28.6 | 7.2 / 8.4  | 32.0 / 36.2 | 20.9 |
VCTree+TDE [16]          | 26.2 / 29.6 | 44.8 / 49.2 | 37.5 | 15.2 / 17.5 | 28.8 / 32.0 | 23.4 | 9.5 / 11.4 | 17.3 / 20.9 | 14.8 |
VCTree+PCPL [14]         | 22.8 / 24.5 | 56.9 / 58.7 | 40.7 | 15.2 / 16.7 | 40.6 / 41.7 | 28.4 | 10.8 / 12.6 | 26.6 / 30.3 | 20.1 |
VCTree+CogTree [18]      | 27.6 / 29.7 | 44.0 / 45.4 | 36.7 | 18.8 / 19.9 | 30.9 / 31.7 | 25.3 | 10.4 / 12.1 | 18.2 / 20.4 | 15.5 |
VCTree+DLPE [17]         | 25.3 / 27.1 | 51.8 / 56.5 | 39.1 | 19.9 / 20.1 | 33.5 / 34.6 | 26.8 | 11.8 / 13.8 | 22.7 / 26.3 | 18.7 |
VCTree+BPL-SA [19]       | 30.6 / 32.6 | 50.0 / 51.8 | 41.3 | 20.1 / 21.2 | 34.0 / 35.0 | 27.6 | 13.5 / 15.7 | 21.7 / 25.5 | 19.1 |
VCTree+LS-KD (Iter)      | 32.3 / 37.0 | 47.4 / 50.8 | 41.9 | 22.6 / 25.1 | 31.5 / 33.5 | 28.2 | 12.6 / 15.5 | 22.9 / 25.7 | 19.2 |
VCTree+LS-KD (Syn)       | 33.8 / 37.5 | 48.1 / 51.3 | 42.7 | 22.2 / 24.9 | 35.6 / 37.4 | 30.0 | 13.7 / 16.6 | 21.8 / 25.4 | 19.4 |

B. Comparison With State-of-the-Arts

1) Settings: We conducted LS-KD with iterative and synchronous self-KD on the two popular baselines (Motifs [30] and VCTree [33]), denoted as LS-KD (Iter) and LS-KD (Syn) respectively. In Table I, we listed the results of the various state-of-the-art SGG models for reference, and divided them into two groups: 1) The methods are not model-agnostic debiasing models, including MSDN [57], G-RCNN [31], and DT2ACBS [38]. 2) The methods are model-agnostic debiasing strategies, including TDE [16], PCPL [14], CogTree [18], DLPE [17], and BPL-SA [19]. Since LS-KD is model-agnostic, we mainly compared with methods in the second group with the same baselines. To further illustrate the performance of our model, we show recall statistics for each predicate of LS-KD, TDE [16], and Motifs [30] in Fig. 6 for comparisons. Besides, we report the R@100 metric of each predicate category group under the PredCls setting in Table II.

2) Results: From the results in Table I, we find that: 1) The Motifs baseline in Table I achieves the best R@K due to frequency bias. Nonetheless, as shown in Table II, the Motifs baseline suffers severe drops in tail predicates (e.g., 5.7% in Tail vs 42.3% in Avg under the PredCls setting on the R@100 metric). 2) LS-KD exceeds the performance of the state-of-the-art method (i.e., BPL-SA) under all three settings by a large margin in the mR@K metrics. Compared to SOTA unbiased methods, our proposed LS-KD can obtain comparable mR@K with lower sacrifices on R@K. 3) LS-KD achieves a better margin in the mR@K metrics. Compared to SOTA unbiased method (i.e., BPL-SA) under all three settings by a large margin in the mR@K metrics.

| Models                  | PredCls | R@50/100 |
|-------------------------|---------|----------|
|                         | Head    | Body     | Tail    | Avg    |
| Motifs [30]             | 80.5    | 29.3     | 5.7     | 42.3   |
| +TDE [16]               | 66.4    | 40.1     | 20.3    | 38.5   |
| +LS-KD                  | 67.7    | 29.5     | 37.3    | 44.8   |
Fig. 6. Recall statistics for each predicate category. The baseline model is Motifs [30].

TABLE III

| Methods                  | PredCls | mR@50/100 | R@50/100 | Mean  |
|--------------------------|---------|-----------|----------|-------|
| Motifs+Label Smooth (LS)|         | 24.6 / 27.9| 50.6 / 55.8| 39.7  |
| Motifs+Label Confusion (LC)|   | 25.1 / 28.6| 50.3 / 55.3| 39.8  |
| Motifs+LS-KD             |         | 34.2 / 37.9| 47.6 / 50.9| 42.7  |

Fig. 7. The mR-R curve of SOTA models under PredCls setting.

Though the Mean score of our method is not the highest in the SGGGen subtask, taking into account all three subtasks, our Mean is still SOTA. To the best of our knowledge, all SOTA debiasing SGG models have lower performance than the baseline for the Mean metric. In contrast, our LS-KD has the least performance drops on the head predicates (cf. R@K) and keeps the performance of tail predicates (cf. mR@K), which proves the superiority of LS-KD by synthesizing all predicates. 4) LS-KD (Syn) can achieve higher performance than LS-KD (Iter) on Mean, because the teacher head of LS-KD (Syn) is trained by cross-entropy loss with one-hot target labels, which is susceptible to the influence of long-tailed distributions. Furthermore, as shown in Fig. 6 and Table II, we can observe that our proposed LS-KD could achieve a decent trade-off among different predicate categories (e.g., 38.5% in TDE vs. 44.8% in LS-KD in the Avg metric under PredCls settings based on Motifs).

C. Comparison With Label Smoothing and Confusion

1) Settings: In Table III, we re-implemented the general Label Smoothing (LS) [58] and Label Confusion (LC) [49] methods under the PredCls setting for SGG. LS softens one-hot targets by adding uniform distribution. LC captures semantic overlap among labels by calculating the similarity between instances and labels, and generates soft targets for training.

TABLE IV

| $\alpha$ | Type | mR@50/100 | R@50/100 | Mean |
|----------|------|-----------|----------|------|
| 3 Iter   | -    | 34.7 / 38.9| 37.2 / 41.7| 38.1 |
| Syn      | 34.4 / 38.8| 39.3 / 43.4| 38.9 |
| 4 Iter   | -    | 34.2 / 37.9| 47.6 / 50.9| 42.7 |
| Syn      | 31.7 / 36.0| 51.7 / 54.2| 43.3 |
| 5 Iter   | -    | 32.1 / 35.5| 52.9 / 55.6| 44.0 |

For a fair comparison, all the $\alpha$ controlling the synthesis of the original one-hot vector and soft distribution were set to 4.

2) Results: From the results, we get the following findings:
1) By making the model less confident during training, the LS method can alleviate the over-fitting of the model to a certain extent, especially for the head predicate classes that account for a large proportion. LS only softens the targets by adding uniformly distributed noise. Thus, it cannot reflect the correlations among predicates labels, resulting in poor performance on Mean (39.7% in LS vs. 42.7% in LS-KD). 2) The LC method can achieve better results on mR@K, but it is inferior to our method in the performance on Mean (39.8% in LC vs. 42.7% in LS-KD) that considers all predicates synthetically. Besides, it needs to design additional networks to encode the representation of labels, and it takes a lot of time to calculate the similarity among all labels for each instance. 3) Both LS and LC methods can significantly improve the performance in the mR@K compared with the baseline performance, which means LS and LC are helpful even without using a teacher-student model. However, since our motivation is to generate soft labels that can reflect the correlations between subject-object pairs and predicates, we adopt KD whose output is a probability distribution. Meanwhile, as the training target is soft labels, it cannot depart from the multi-label paradigm.

D. Ablation Studies

1) Ablation Study on $\alpha$ in LS-KD: In Table IV, we set various hyperparameters $\alpha$ in LS-KD to control the degree of fusion between the LSD learned by the teacher model and the original one-hot vector. Following the work [49] using label confusion method in the text classification task, we also set $\alpha = 4$ as a moderate choice in our main experiments. To investigate the impact of different degrees of fusion on performance, we conducted experiments when $\alpha = 3$ and $\alpha = 5$.
the iteration interval, the higher the performance on mR@K. Observing the results, we can find that the larger R@K and mR@K. 0.5 iteration means the iteration period respectively, to test the impact of the iterative interval on the performance on mR@K, and the steeper the LSD, the better results, we find that the smoother the LSD, the better the performance on R@K.

We conducted experiments with different iterative intervals of iterative self-KD (α = 5). The larger α, the larger weight will be assigned to the one-hot vector when generating a soft label in the SLD module.

When the hyperparameter α is set to 5, the original one-hot vector takes a larger proportion than the other two values (i.e., 3 and 4), and the model makes over-confident predictions on head predicate with larger sample size, so the R@K is relatively high. When the hyperparameter α is set to 3, the LSD accounts for a greater proportion than the other two values (i.e., 4 and 5). If the instance is related to multiple labels, the sample will not excessively fit just one label, and the probability gap between head and tail predicate categories will decrease relatively. Therefore, mR@K is relatively high.

Also, since α is a hyperparameter for the trade-off between Recall and Mean Recall performance, we plot the mR-R curve of all the operating results in Fig. 7. As shown in Fig. 7, almost all prior debiasing methods are under this curve, which confirms the effectiveness of our proposed method.

2) Ablation Study on the Temperature τ in LS-KD: The temperature τ is used to control the smoothness of the output probability distribution of softmax function. Thus, we set three values of τ according to different magnitudes, which are 0.1, 1 and 10 respectively. Table V shows the performance under PredCls setting where we set different temperature τ of softmax function in LS-KD. The larger τ is, the smoother the output probability distribution of softmax is. From the results, we find that the smoother the LSD, the better the performance on mR@K, and the steeper the LSD, the better the performance on R@K.

3) Ablation Study on Iterative Interval t of Iterative Self-KD: We conducted experiments with different iterative intervals t of iterative self-KD (i.e., the interval of iterations to transform the student model into the teacher model). To be specific, we set t of iterative self-KD as 0.5, 1 and 2, respectively, to test the impact of the iterative interval on R@K and mR@K. 0.5 iteration means the iteration period is 0.5 epoch. Observing the results, we can find that the larger the iteration interval, the higher the performance on mR@K.

E. Qualitative Results

1) Generated Scene Graphs: Some generated scene graphs are visualized in Fig. 8, where the scene graphs with a yellow background are generated by Motifs-baseline, and those with a green background are by Motifs-Ours. Only the top-1 prediction and detected boxes overlapped with GT are shown. Comparing the results of the two methods, we can find that our method can detect more informative predicates, such as “laying on”, “sitting on”, and “carrying”. These predicates all have corresponding head predicates that have overlapping or overlaying semantics with them, such as “on” and “with”. The reason may be that we can reduce the margin of possibility between the head and tail predicates with similar semantics by capturing the correlation of the instance to multiple predicates. Thus, the over-confident predictions on head predicates under long-tailed distribution can be alleviated to some extent. Besides, for triplets that are hard to predict (e.g., “wheel-mounted on-board” and “sign-for-toilet”), Motifs-baselines are susceptible to frequency bias to make incorrect predictions (e.g., “wheel-on-board” and “sign-above-toilet”), while the predictions of our method are correct. It confirms that reducing the overfitting to the head predicates by training with soft labels is effective.
For example, for the “chair-at-table” triplet, the two predicate categories “under” and “in front of” capture the spatial position relationship between the “chair” and the “table” from different angles, and these more exact predicates are more valuable than “at”. In addition, for the “leaf-on-branch” triplet, predicates “growing on” and “covering” represent a dynamic relationship that is more informative than the static “on”. Besides, the predicate category with the max proportion is still GT (i.e., best answer), and the other reasonable tail predicates (i.e., a fair but generally good answer) are assigned proportional probabilities. Thus, the soft labels generated by LS-KD can avoid the over-fitting of the head predicates and promote unbiased prediction to a certain extent.

V. CONCLUSION AND LIMITATIONS

In this paper, we pointed out the drawbacks of the prevalent training paradigm in existing SGG works, i.e., training the predicate classifier with one-hot target labels. To alleviate these drawbacks, we presented a novel model-agnostic LS-KD for unbiased SGG. Two different self-KD strategies were designed in LS-KD to predict label semantic distribution (LSD): iterative self-KD and synchronous self-KD. LSD could not only reflect the correlations between subject-object instances and multiple predicate categories but also generate more robust “soft” pseudo-labels. Extensive results proved the generality and effectiveness of our method. Moving forward, we planned to: 1) integrate LS-KD into existing debiasing methods that aim to solve long-tailed distribution problems. 2) design better methods to capture correlations among predicate labels.

A. Limitations

Knowledge distillation may introduce extra computational overhead or increased GPU consumption during training. Nonetheless, the benefits of knowledge distillation are considered significant enough to justify the additional cost.

REFERENCES

[1] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, “Scene graph generation by iterative message passing,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3097–3106.

[2] C. Yan et al., “Task-adaptive attention for image captioning,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 1, pp. 43–51, Jan. 2022.
[3] Y. Peng and J. Chi, “Unsupervised cross-media retrieval using domain adaptation with scene graph,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 11, pp. 4368–4379, Nov. 2020.

[4] H. Su et al., “End-to-end video question-answer generation with generator-pretester network,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 11, pp. 4497–4507, Nov. 2021.

[5] L. Chen, X. Yan, J. Xiao, H. Zhang, S. Pu, and Y. Zhiang, “Counterfactual samples synthesizing for robust visual question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10797–10806.

[6] Y. Yang, Y. Zhiang, and Y. Pan, “Multiple knowledge representation for big data artificial intelligence: Framework, applications, and case studies,” Frontiers Inf. Technol. Electron. Eng., vol. 22, no. 12, pp. 1551–1558, Dec. 2021.

[7] R. Krishna et al., “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” Int. J. Comput. Vis., vol. 123, no. 1, pp. 32–73, May 2017.

[8] 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, pp. 1–9.

[9] N. Xu, A. Liu, Y. Wong, W. Nie, Y. Su, and M. Kankanhalli, “Scene graph inference via multi-scale context modeling,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 3, pp. 1031–1041, Mar. 2021.

[10] B. Zhao, Z. Mao, S. Fang, W. Zang, and Y. Zhiang, “Semantically similarity-wise dual-channel network for scene graph generation,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 7, pp. 4573–4583, Jul. 2022.

[11] X. Han et al., “Divide-and-conquer predictor for unbiased scene graph generation,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 12, pp. 8611–8622, Dec. 2022.

[12] Z. Fu et al., “DRAKE: Deep pair-wise relation alignment for knowledge-enhanced multilingual scene graph generation in social media posts,” IEEE Trans. Circuits Syst. Video Technol., early access, Dec. 21, 2022, doi: 10.1109/TCSTV.2022.3231437.

[13] R. Li, S. Zhang, B. Wan, and X. He, “Bipartite graph network with adaptive message passing for unbaised scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11104–11114.

[14] S. Yan et al., “PCPL: Predicate-correlation perception learning for unbiased scene graph generation,” in Proc. 28th ACM Int. Conf. Multimedia, Oct. 2020, pp. 265–273.

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

[16] 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. (CVPR), Jun. 2020, pp. 3713–3722.

[17] M.-J. Chiou, H. Ding, H. Yan, C. Wang, R. Zimmermann, and J. Feng, “Learning to compose dynamic tree structures for visual contexts,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 6612–6621.

[18] C. Tang, H. Xiao, X. He, S. Pu, and S.-F. Chang, “Counterfactual critic multi-agent training for scene graph generation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4612–4622.

[19] Q. Dong, Z. Tu, H. Liao, Y. Zhang, V. Mahadevan, and S. Soatto, “Visual relationship detection using part-and-sum transformers with composite queries,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 3530–3539.

[20] H. Liu, N. Yan, M. Mortazavi, and B. Bhanu, “Fully convolutional scene graph generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11541–11551.

[21] I. Misra, C. L. 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. (CVPR), Jun. 2016, pp. 2930–2939.

[22] A. Desai, T. Wu, S. Tripathi, and N. Vasconcelos, “Learning of visual relations: The devil is in the tails,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 15384–15393.

[23] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” Int. J. Comput. Vis., vol. 129, no. 6, pp. 1789–1819, Jun. 2021.

[24] L. Chen, C. Zhang, Z. Zhang, and S. Li, “Network learning via evolutionary knowledge distillation,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 4, pp. 2251–2263, Apr. 2022.

[25] L. Chen, Y. Zheng, and J. Xiao, “Rethinking data augmentation for robust visual question answering,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 95–112.

[26] X. Li, L. Chen, W. Ma, Y. Yang, and J. Xiao, “Integrating object-aware and interaction-aware knowledge for weakly supervised scene graph generation,” in Proc. ACM Int. Conf. Multimedia, Oct. 2022, pp. 4204–4213.

[27] S. Hahn and H. Choi, “Self-knowledge distillation in natural language processing,” 2019, arXiv:1908.01851.

[28] S. Yun, J. Park, K. Lee, and J. Shin, “Regularizing class-wise predictions via self-knowledge distillation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 1387–1396.

[29] K. Kim, B. Ji, D. Yoon, and S. Hwang, “Self-knowledge distillation with progressive refinement of targets,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 6547–6556.

[30] J. Tang et al., “Understanding and improving knowledge distillation,” 2020, arXiv:2002.03532.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[32] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8697–8710.

[33] B. Guo, S. Han, X. Han, H. Huang, and T. Lu, “Label confusion learning to enhance text classification models,” 2020, arXiv:2012.04987.

[34] T. J. Wang, S. Pehlivan, and J. Laaksonen, “Tackling the unannotated: Scene graph generation with bias-reduced models,” 2020, arXiv:2008.07812.
Wenxiao Wang received the Ph.D. degree in computer science and technology from Zhejiang University, China, in 2022. He is a Distinguished Research Fellow with the School of Software Technology, Zhejiang University. His research interests include deep learning and computer vision.

Jian Shao received the Ph.D. degree in signal and information processing from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with the College of Computer Science, Zhejiang University, Hangzhou, China. His research interests include unstructured data management, cross media computing, and cognitive decision service.

An-An Liu (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Tianjin University, China. He is currently a Professor with the School of Electrical and Information Engineering, Tianjin University. He was a Visiting Professor with the SESAME Centre, National University of Singapore; and a Visiting Scholar with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests include computer vision and machine learning.

Yi Yang (Senior Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He was a Professor and the Director of the ReLER Labortory, Australian Artificial Intelligence Institute (AAI), University of Technology Sydney, Australia. He was a Post-Doctoral Research with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Distinguished Professor with Zhejiang University. His current research interests 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.

Long Chen received the B.Eng. degree in electrical information engineering from the Dalian University of Technology in 2015 and the Ph.D. 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 Post-Doctoral Research Scientist with Columbia University and a Senior Researcher with Tencent AI Laboratory. His research interests include computer vision, machine learning, and multimedia.

Lin Li received the B.S. degree from the College of Software, Qufu Normal University, Qufu, China, in 2019. She is currently pursuing the Ph.D. degree with the College of Computer Science, Zhejiang University, Hangzhou, China. Her current research interests include deep learning and computer vision.

Jun Xiao received the Ph.D. 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 B.Eng. degree in computer science from Zhejiang University, Hangzhou, China, in 2022, where she is currently pursuing the master’s degree with the College of Computer Science. Her research interests include scene graph generation, visual question answering, and deep learning.