Divide-and-Conquer Predictor for Unbiased Scene Graph Generation

Xianjing Han, Xingning Dong, Xuemeng Song, *Senior Member, IEEE*, Tian Gan, Yibing Zhan, Yan Yan, Liqiang Nie, *Senior Member, IEEE*

***Abstract*—Scene Graph Generation (SGG) aims to detect the objects and their pairwise predicates in an image. Existing SGG methods mainly fulfil the challenging predicate prediction task that involves severe long-tailed data distribution with a single classifier. However, we argue that this may be enough to differentiate predicates that present obvious differences (*e.g.*, *on* and *near*), but not sufficient to distinguish similar predicates that only have subtle differences (*e.g.*, *on* and *standing on*). Towards this end, we divide the predicate prediction into a few sub-tasks with a Divide-and-Conquer Predictor (DC-Predictor). Specifically, we first develop an offline pattern-predicate correlation mining algorithm to discover the similar predicates that share the same object interaction pattern. Based on that, we devise a general pattern classifier and a set of specific predicate classifiers for DC-Predictor. The former works on recognizing the pattern of a given object pair and routing it to the corresponding specific predicate classifier, while the latter aims to differentiate similar predicates in each specific pattern. In addition, we introduce the Bayesian Personalized Ranking loss in each specific predicate classifier to enhance the pairwise differentiation between head predicates and their similar ones. Experiments on VG150 and GQA datasets show the superiority of our model over state-of-the-art methods.**

***Index Terms*—Scene Graph Generation, Vision and Language, Divide-and-Conquer, Bayesian Personalized Ranking.**

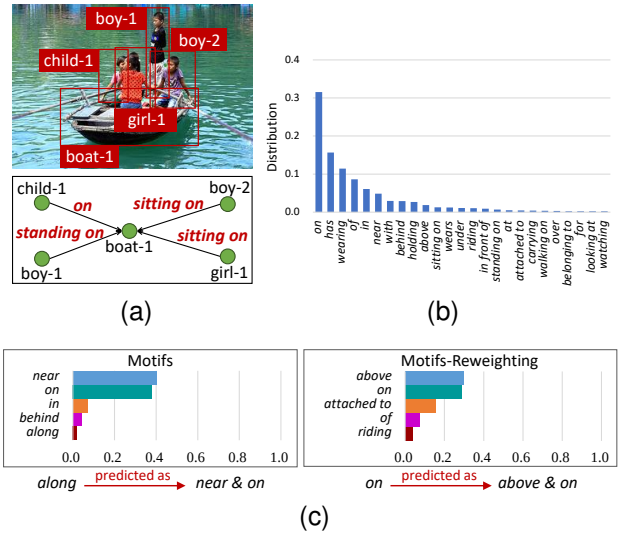


Fig. 1. The long-tailed data distribution in SGG dataset and the biased predictions of the recent SGG methods. (a) An example of SGG. (b) Data distribution of the 25 most frequent predicate classes in VG dataset. And (c) the predicted predicate distributions of samples with ground truth predicates *along* and *on* by the biased Motifs [10] and unbiased Motifs-Reweighting [11], respectively.

I. Introduction

Scene Graph Generation (SGG) aims to abstract the image with a set of detected objects and their pairwise relationships (*i.e.*, predicates), as shown in Fig. 1a. Due to its huge benefits to many high-level visual-language understanding tasks [1], [2], such as image captioning [3], [4] and visual question answering [5], [6], SGG has gained increasing research attention [7]–[9], recently. One key issue faced by existing methods is the long-tailed data distribution.

Specifically, as shown in Fig. 1b, since a few head predicate classes occupy the majority of training samples, the model tends to yield the head predicates (*e.g.*, *on* and *near*) rather than tail ones (*e.g.*, *standing on* and *along*), and thus results in the biased prediction. Thereby, some unbiased SGG methods, such as the re-sampling [12] and re-weighting based methods [13], [14], have been proposed. Although these methods have achieved compelling progress, they improve the performance on the tail predicates by largely sacrificing the performance on the head ones.

Based on the predicate prediction of both biased and unbiased SGG methods, we observe that predicates are more likely to be predicted as the ones that share the same object interaction patterns, *i.e.*, the similar spatial (*e.g.*, *lying on* and *on*) or semantic (*e.g.*, *made of* and *has*) relations between two objects. For instance, Fig. 1c shows the predicted predicate distributions of samples with ground truth predicates *along* and *on* by the biased Motifs [10] and unbiased Motifs-Reweighting [11] model, respectively. In the biased prediction, the tail predicate *along* is mostly predicted as the head predicate *near* and *on*, while in the unbiased prediction, the

This work was supported by the National Key Research and Development Project of New Generation Artificial Intelligence under Grant 2018AAA0102502, in part by the National Natural Science Foundation of China under Grant 62002090, Grant 61772310, Grant 61702300, and Grant U1936203, in part by the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project under Grant 2021ZD0111700, in part by the Natural Science Foundation of Shandong Province under Grant ZR2019JQ23, in part by the Shandong Provincial Key Research and Development Program under Grant 2019JZZY010118, and in part by the Innovation Teams in Colleges and Universities in Jinan under Grant 2018GXRC014. (Corresponding authors: Xuemeng Song and Liqiang Nie)

X. Han, X. Dong, X. Song, and T. Gan are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: hanxianjing2018@gmail.com, pass1463365882@gmail.com, sxmustc@gmail.com, gantian@sdu.edu.cn).

Y. Zhan is with the Institution of JD Explore Academy, Beijing 100176, China (e-mail: zhanyibing@jd.com)

Y. Yan is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: yyan34@iit.edu).

L. Nie is with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China (e-mail: nieliqiang@gmail.com).

head predicate on tends to be predicted as the tail predicate $above$ and $attached\ to$. The reason for the observation may be that when predicates share similar object interaction patterns, the differences among their samples are too subtle to be distinguished, which aggravates the predicate prediction. Therefore, we argue that one key to improve the biased predicate prediction is how to differentiate the various subtle differences among similar predicates. However, faced with the challenging predicate prediction task, existing biased/unbiased methods try to differentiate all the predicates with only a single classifier. In fact, a single classifier could be adequate to differentiate the predicates that have obvious differences (*e.g.*, on and $near$), but may be far from sufficient to distinguish the predicates with subtle differences (*e.g.*, on and $standing\ on$).

In the light of this, we propose to divide the predicate prediction into a few sub-tasks, *i.e.*, distinguishing the subtle differences among a subset of similar predicates that share the same object interaction pattern. In particular, as shown in Fig. 2, we propose a Divide-and-Conquer Network (DCNet), where the key component lies in the Divide-and-Conquer Predicate Predictor (DC-Predictor). Specifically, to facilitate the dividing of the task, we first develop an offline pattern-predicate correlation mining algorithm, to excavate the object interaction patterns shared by predicates and build the pattern-predicate correlation. Based on the uncovered correlation, we introduce a general pattern classifier and a set of specific predicate classifiers for DC-Predictor. The former works on recognizing the general pattern of a given object pair and routing it to the corresponding specific classifier, while the latter trained with specific data aims to distinguish the subtle differences among similar predicates in each specific pattern. In addition, as the head predicates (*e.g.*, on) are most probably to be the sub-optimal prediction for their similar tail predicates (*e.g.*, $lying\ on$ and $covering$), we adopt the Bayesian Personalized Ranking (BPR) [15] loss in each specific predicate classifier to enhance the pairwise differentiation between head predicates and their similar ones. It is worth noting that DC-Predictor can be applied to the predicate predictors of various SGG models to enhance their similar predicate differentiation ability.

The contributions of this paper are three-fold:

- Towards unbiased SGG, we devise a model-agnostic DC-Predictor, which consists of two key components: pattern-predicate correlation mining and divide-and-conquer predicate classification. As far as we know, we are the first to highlight the subtle different differentiation among predicates, and fulfil the predicate prediction task that suffers from long-tailed data distribution in a divide-and-conquer manner.
- To the best of our knowledge, we are the first to focus on the pairwise differentiation between the head and tail predicates with the BPR loss.
- We conduct experiments on VG150 [16] and GQA [17] datasets, and the results indicate the superiority of our DC-Predictor in unbiased SGG. We release the source codes and trained model on GitHub[1].

[1]https://github.com/hanxjing/DCNet.

II. Related Work

Scene Graph Generation. One major challenge of SGG [16] is the predicate prediction for object pairs. Early studies mainly resort to the language prior [18] or multi-modal features [19]–[22] to address the challenge, but ignore the effect of the contextual information. Toward this end, recent studies mainly employ the message passing method [16], recurrent sequential structured networks [10], [23], graph neural networks [24]–[27], attention mechanism [7], [8], [28], [29], and multi-scale representations [30] to encode the object with rich contextual information. Though these efforts have gained improvement on the overall recall of the predicates, the predicted predicates are often trivial and less informative due to the long-tailed data distribution.

Toward this end, Chen *et al*. [31] and Tang *et al*. [23] introduced the mean recall of all the predicates to evaluate the unbiased SGG. Thereby, some unbiased methods [11], [12], [32] emerged. For example, Tang *et al*. [11] employed the counterfactual causality to disentangle the bias from the prediction. Yan *et al*. [13] utilized the learned predicate independence to assign the classification loss weight. In addition, Yu *et al*. [14] and Zhao *et al*. [33] noticed the correlation among predicates. In particular, Yu *et al*. employed the predicate correlation tree to filter the interference of the obviously irrelevant predicate prediction. Though these unbiased methods improve the recall of tail predicates, they sacrifice much performance on head predicates and could hardly improve the predicate differentiating ability of the model. Therefore, in this work, we aim to excavate the patterns shared by predicates and based on that explore the predicate correlations, which can be used as the beneficial prior like language prior [18], [34] to promote the similar predicates differentiation. Despite there are some human activity recognition works [35], [36] also exploit the latent pattern among the data, they mainly aim to achieve a pattern-balanced training by sampling the pattern data, which are different from our method that excavates the pattern to facilitate the distinguishing of the predicates that share the same pattern.

Bayesian Personalized Ranking. BPR [15] method is first employed in the recommender system [37], [38] to cope with the implicit feedback. Due to the success of BPR in the pairwise preference modeling, it has been widely used in various domains, such as multi-model searching [39], [40] and fashion compatibility modeling [41], [42]. For example, in compatibility modeling, Song *et al*. [42] exploited the pairwise preference to the bottom clothing for the given top clothing with BPR loss. Inspired by this, we adopted BPR loss to enhance the pairwise differentiation between head predicates and their similar predicates in SGG.

III. Methodology

In this section, we first formulate the problem and then introduce our proposed DCNet. As shown in Fig. 2, our scheme follows the common SGG pipeline [10], [14], [23], comprising three components: 1) object proposal network for the object detection; 2) object classification network for the object class refinement; and 3) predicate classification network

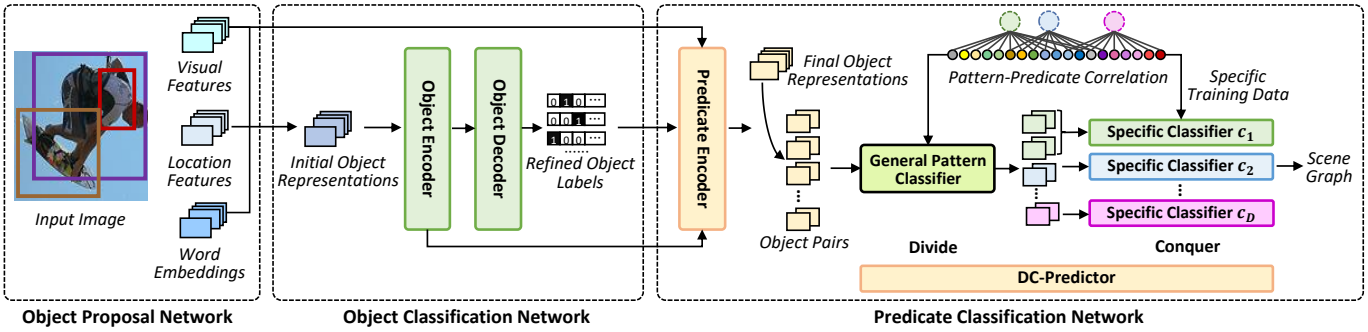


Fig. 2. Schematic illustration of the proposed DCNet, comprising three components: 1) object proposal network, 2) object classification network, and 3) predicate classification network. We devise a DC-Predictor in predicate classification network, where we first excavate the patterns among predicates and build the pattern-predicate correlation to uncover the similar predicates. Based on that, we devise a general pattern classifier and a set of specific predicate classifiers. The former is employed to divide the object pairs into their corresponding specific classifiers, while the latter is used to distinguish the subtle differences among similar predicates in each specific pattern.

to assign the predicate for each object pair. To boost the predicate classification, we first develop an offline pattern-predicate correlation mining algorithm to uncover the pattern-predicate correlation, and then we devise the DC-Predictor to handle the predicate classification task in a divide-and-conquer manner.

A. Problem Formulation

SGG aims to detect the objects and their pairwise predicates within an image. Formally, for a given image, we first detect a set of objects $\mathcal{O} = \{o_i\}_{i=1}^{N}$ in the image and then obtain the object classes by an object classification network. Thereafter, for each object pair (o_i, o_j), we predict their predicate p_{ij} by a predicate classification network, where $p_{ij} \in \mathcal{P}$, and $\mathcal{P} = \{p_m\}_{m=1}^{M}$ stands for the set of predicate classes. Ultimately, we can generate a scene graph as a set of triplets $\{(o_i, p_{ij}, o_j) | o_i \in \mathcal{O}, o_j \in \mathcal{O}, p_{ij} \in \mathcal{P}\}$.

B. Object Proposal Network

We choose the pre-trained Faster R-CNN [43], which is commonly used as an object proposal network in SGG methods, to detect the objects in a given image. With Faster R-CNN, each detected object o_i can be represented with a visual feature $\mathbf{v}_i$, a location feature $\mathbf{b}_i$ (*i.e.*, the coordinates of the object bounding box), and a word embedding $\mathbf{e}_i$ of the initial detected object class, which can provide the additional semantic information of the object.

C. Object Classification Network

Following existing studies [10], [14], we devise the object classification network with an encoder and a decoder. The encoder targets at encoding the object contextual information into the representation of each object to promote the object classification, while the decoder works on predicting the refined object class. Similar to [14], we adopt the Transformer-based [44] object encoder, due to its capability in capturing the contextual information resided in the input.

*1) **Transformer-Based Object Encoder**:* We first derive the input of the object encoder as follows,

$$\mathbf{x}_i = f_o([\mathbf{b}_i, \mathbf{v}_i, \mathbf{e}_i]), \tag{1}$$

where $\mathbf{x}_i$ denotes the initial representation of the object o_i. $f_o(\cdot)$ represents a fully-connected layer followed by a sigmoid function. We then feed all the initial object representations $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ into the Transformer-based encoder [44] as follows,

$$\mathbf{X}' = \mathrm{MHA}(\mathbf{X}), \tag{2}$$

where $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_N]$ are the refined object representations. $\mathrm{MHA}(\cdot)$ denotes the multi-head self-attention in Transformer [44].

*2) **Object Decoder**:* As for each object o_i, we can derive its refined object class $\mathbf{o}'_i$ by the object decoder f_d as follows,

$$\mathbf{o}'_i = f_d(\mathbf{x}'_i), \tag{3}$$

where $f_d(\cdot)$ consists of a fully-connected layer followed by a softmax function. We then can obtain the refined word embedding $\mathbf{e}'_i$ according to the maximum element of the refined object class $\mathbf{o}'_i$.

D. Predicate Classification Network

Predicate classification network consists of a Transformer-based relationship encoder and a newly proposed DC-Predictor. The former is used to compile the contextual information among objects, while DC-Predictor is introduced to predict the predicate for each object pair.

*1) **Transformer-Based Predicate Encoder**:* To boost the predicate prediction, we adopt another Transformer-based encoder to exploit the interaction context among objects in an image, which can benefit the predicate prediction. We first derive the input object representation $\tilde{\mathbf{x}}_i$ of the relationship encoder as follows,

$$\tilde{\mathbf{x}}_i = f_r([\mathbf{v}_i, \mathbf{x}'_i, \mathbf{e}'_i]), \tag{4}$$

where $f_r(\cdot)$ represents a fully-connected layer followed by a sigmoid function. $\mathbf{v}_i$ is incorporated to retain the original object visual feature in predicate prediction. We then feed

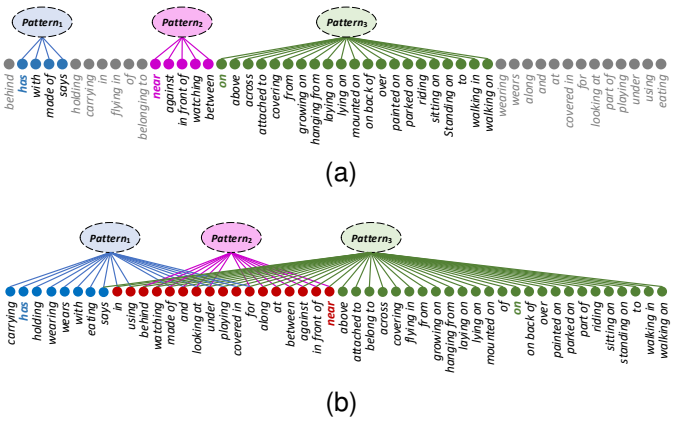


Fig. 3. The excavated patterns and pattern-predicate correlation in VG150 dataset, where the parent predicate in each pattern is annotated in the distinct color. (a) Excavate Patterns. (b) Pattern-Predicate Correlation.

all the object representations $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_N]$ into the Transformer-based predicate encoder, which shares the same structure with the Transformer-based object encoder, to get the final object representations $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \cdots, \hat{\mathbf{x}}_N]$.

2) ***Predicate Decoder****:* As for the predicate decoder, in this work, rather than classifying all the predicates with a single classifier, we propose to use multiple specific predicate classifiers to handle the predicate classification in a divide-and-conquer manner. Specifically, we first devise a pattern-predicate correlation mining algorithm to discover the similar predicates, and then introduce the DC-predictor for divide-and-conquer predicate classification.

E. Pattern-Predicate Correlation Mining

In this work, we propose to use multiple specific predicate classifiers to predict the predicates, where each specific classifier differentiates the similar predicates that share the same object interaction pattern. To achieve this goal, we first devise a pattern-predicate correlation mining algorithm to discover the similar predicates.

As aforementioned, affected by the long-tailed data distribution, biased SGG models tend to predict the tail predicates with their similar head predicates. According to the biased prediction, existing efforts [14], [45] represent the similarity correlation among predicates as a tree, where a tail predicate can be the child node of one head predicate. However, we argue that one tail predicate can be similar to multiple head ones (see Fig. 1c), and the predicate correlation can be interlaced. Therefore, we aim to use the prediction of the biased SGG model to excavate latent object interaction patterns shared by predicates, and utilize these patterns as anchors to uncover the interlaced pattern-predicate correlation. The pattern-predicate correlation mining consist of two parts: pattern excavation and pattern-predicate correlation construction.

1) ***Pattern Excavation****:* Motifs [10] is a typical biased SGG method. Given the detected objects in an image, Motifs first adopts the BiLSTM-based [46] object encoder to refine the object classes, and then adopts the BiLSTM-based predicate encoder to gather the object contextual information to predict the predicate $\mathbf{p}_{ij}$ for each object pair (o_i, o_j). Affected by the long-tailed data distribution, Motifs tends to predict the tail predicate samples as the head predicates that share the same object interaction patterns with them. Using the biased predicate prediction of Motifs, we can obtain D patterns $\mathcal{S} = \{s_d\}_{d=1}^{D}$, each covering a parent predicate, denoted as p'_d, and several child predicates. In particular, based on the the biased predicate prediction of Motifs, for each predicate p, we calculate the distribution of the predicted predicate frequency $\mathbf{f} = [f_1, f_2, \cdots, f_M] \in \mathbb{R}^M$ with all its object pair samples. M is the total number of predicate classes. We then define the parent-child relation among the predicates with the following condition,

$$p \Rightarrow p', \text{ if } (f^{1st} > \lambda f^{2nd}) \wedge (p \neq p'), \tag{5}$$

where f^{1st} and f^{2nd} are the first and second largest elements in $\mathbf{f}$ and p' is the corresponding predicate of f^{1st}. λ is the hyperparameter to control the strictness of the condition. If $(f^{1st} > \lambda f^{2nd}) \wedge (p \neq p')$, we deem predicate p is similar to p' and treat it as a child predicate of p' with the notation "$\Rightarrow$". After conducting the above process for all predicates, we treat each parent predicate p' and all its associated child predicates as a whole, and assume that they follow a specific object interaction pattern.

In addition, as the prediction of the biased SGG model may be unstable, we repeat the above process twice, including the training of the biased SGG model and the latent pattern excavation. We then only retain the output patterns that cover exactly the same predicates (*i.e.*, the parent predicate and its child predicates) in the two implementations. Fig. 3a illustrates the final result of pattern excavation in VG150 dataset.

2) ***Pattern-Predicate Correlation Construction****:* Based on the above pattern excavation, we obtain a few patterns and their strongly correlated predicates. Then, based on the samples of predicates in each pattern, we can train a pattern classifier to further find the weakly related predicates for each pattern to construct the complete pattern-predicate correlation. Specifically, for each pattern s_d, we first group all the object pairs of its covered predicates, and label them with pattern label s_d. Based on these training data, we then train the biased SGG model [10], and use it to predict the pattern probability distribution for each object pair sample. Let $\mathbf{s}_{ij} = [s_{ij}^1, s_{ij}^2, \cdots, s_{ij}^D]$ denote the pattern probability distribution for the object pair (o_i, o_j). For each predicate p, we calculate its overall pattern probability distribution, termed as $\mathbf{q}_p = [q_p^1, q_p^2, \cdots, q_p^D]$, by averaging the pattern probability distributions of all its object pair samples, where q_p^d refers to the probability that the predicate p is related to the pattern s_d. Thereafter, we define the predicate p as related to pattern s_d as follows,

$$p \rightarrow s_d, \text{ if } q_p^d > \mu, \tag{6}$$

where μ is the predefined threshold. Notably, each predicate can be related to multiple patterns. Ultimately, we build the pattern-predicate correlation as a bipartite graph G shown in Fig. 3b, where the nodes of one side are the learned patterns and that of the other side are predicates, while the edges represent the affiliation between the patterns and predicates.

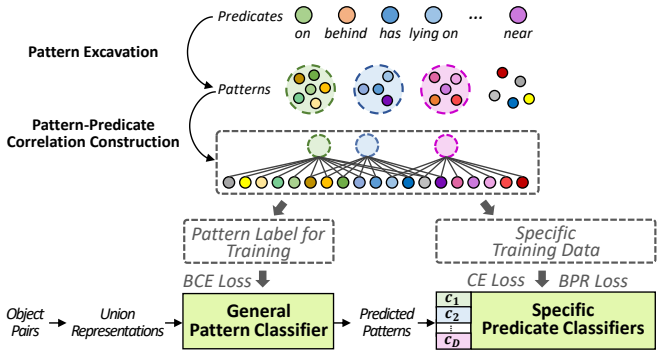


Fig. 4. The framework of our DC-Predictor.

F. Divide-and-Conquer Predicate Classification.

1) ***DC-Predictor****:* To guide the model to target at the differentiation among similar predicates, in DC-Predictor, we devise a general pattern classifier $\mathcal{S}$ and a set of specific predicate classifiers $\{\mathcal{C}_d\}_{d=1}^{D}$. Notably, this general pattern classifier is not the one used in the pattern-predicate correlation construction, and it is trained jointly with the specific predicate classifiers. As shown in Fig. 4, for each object pair, we first get their union representation $\mathbf{g}_{ij}$ by their final object representations and union visual feature $\mathbf{u}_{ij}$. We then predict its pattern probability $\mathbf{s}_{ij} \in \mathbb{R}^D$ by the general pattern classifier $\mathcal{S}$. According to the pattern with the largest predicted probability, we select the corresponding specific predicate classifier to obtain the predicted predicate probability $\mathbf{p}_{ij} \in \mathbb{R}^M$, which can be formulated as follows,

$$\begin{cases} \mathbf{g}_{ij} = \mathbf{W}_g[\mathbf{W}_f\hat{\mathbf{x}}_i; \mathbf{W}_b\hat{\mathbf{x}}_j] \otimes \mathbf{u}_{ij}, \\ \mathbf{s}_{ij} = \mathcal{S}(\mathbf{g}_{ij}), \\ \mathbf{p}_{ij} = \mathcal{C}_d(\mathbf{g}_{ij}|\mathbf{s}_{ij}), \end{cases} \tag{7}$$

where the subscript $d = \arg\max(\mathbf{s}_{ij})$. Similar to [10], we leverage $\mathbf{W}_f$ and $\mathbf{W}_b$ to project the forward and backward relation of object pairs, respectively. $\mathbf{W}_g$ denotes the linear transformation, and $\otimes$ is the element-wise product. Both $\mathcal{S}$ and $\mathcal{C}_d$ are composed of a fully-connected layer followed by a softmax function, whose output dimensions are the number of patterns and predicate classes, respectively. Notably, as the optimization targets of $\mathcal{S}$ and $\mathcal{C}_d$ are different, the argmax operation in determining the specific classifier $\mathcal{C}_d$ would not make the model indifferentiable.

2) ***Optimization****:* As shown in Fig. 4, according to the pattern-predicate correlation, we can obtain the pattern labels of object pairs to train the general pattern classifier and get the specific training object pairs to train the specific predicate classifiers.

General Pattern Classifier. We utilize the binary cross-entropy loss to optimize the general pattern classifier as follows,

$$\mathcal{L}_{bce} = -\sum_{d=1}^{D} \tilde{s}_{ij}^d \log(s_{ij}^d) + (1 - \tilde{s}_{ij}^d) \log(1 - s_{ij}^d). \tag{8}$$

where $\tilde{s}_{ij} = [s_{ij}^1, s_{ij}^2, \cdots, s_{ij}^d] \in \{0, 1\}^D$ refers to the pattern label of the object pair (o_i, o_j), which is derived according to the associated pattern(s) of its ground truth predicate label in bipartite graph G. Since some predicates can be correlated multiple patterns, $\tilde{s}_{ij}$ could be a multi-hot vector.

Specific Predicate Classifier.

Regarding the optimization for each specific predicate classifier, in addition to the cross-entropy loss, we also adopt the BPR loss to promote the classifier differentiate the subtle differences among similar predicates. In particular, we first adopt the cross-entropy loss for each specific predicate classifier as follows,

$$\mathcal{L}_{ce}^d = \frac{1}{|\mathcal{N}_d|} \sum_{(o_i, o_j) \in \mathcal{N}_d} log(\hat{p}_{ij}), \tag{9}$$

where $\mathcal{N}_d$ is the specific training dataset of pattern s_d, comprising all the object pairs of predicates covered by the pattern s_d. $\hat{p}_{ij}$ denotes the predicted probability that the object pair (o_i, o_j) belongs to its ground truth predicate class.

To enhance the pairwise differentiation between the head predicates (*e.g.*, *on*) and their similar predicates (*e.g.*, *sitting on* and *parking on*), we adopt the BPR loss [15] for each specific predicate classifier as follows,

$$\mathcal{L}_{bpr}^d = \frac{1}{|\mathcal{N}_d^t|} \sum_{(o_i, o_j) \in \mathcal{N}_d^t} -\log \sigma(\hat{p}_{ij} - p'_{ij}), \tag{10}$$

where $\mathcal{N}_d^t \in \mathcal{N}_d$ denotes the set of object pair samples that belong to the tail predicate classes. In particular, we regard the predicate class that covers less than $1,000$ samples as a tail predicate. p'_{ij} is the predicted probability of the parent predicate p'_d in pattern s_d. $\sigma(\cdot)$ is the sigmoid function. The underlying philosophy is that according to our pattern-predicate correlation mining algorithm, the parent predicate of each pattern should be a head predicate. The BPR loss propels the predicted probability corresponding to the ground truth predicate to be larger than that towards the head predicate.

Ultimately, we reach the final object function as follows,

$$\mathcal{L} = \mathcal{L}_{bce} + \sum_{d=1}^{D} (\mathcal{L}_{ce}^d + \alpha \mathcal{L}_{bpr}^d), \tag{11}$$

where α is the hyperparameter to weigh the BPR loss.

IV. Experiments

A. Experiment Settings

Dataset. We evaluate our DCNet on Visual Genome [47] and GQA [17] datasets. **Visual Genome** dataset is the most widely used benchmark for scene graph generation. Following prior studies [10], [16], [48], we adopt the most frequently used version VG150 [16], which contains 108K images with the most frequent 150 object categories and 50 predicate classes. We adopt the same experimental settings with [10], [11] to split the dataset by attributing 70% of the images for training, 30% for testing, and randomly sampling 5K images from the training set for validation. **GQA** dataset is a widely used vision-and-language benchmark with rich object relationship annotations. Similar to [49], we normalize GQA dataset to adapt the SGG by pruning the poor-quality and unnatural annotations. We keep the top 200 object categories

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON SGDET, SGCLS, AND PREDCLS TASKS WITH VG150 DATASET IN TERMS OF MR@50/100, R@50/100, AND MEAN. † REPRESENTS THE CORRESPONDING METHOD EMPLOYS FASTER R-CNN WITH VGG-16. $_u$ DENOTES THE CORRESPONDING METHOD TARGETS AT THE UNBIASED SGG. $^\diamondsuit$ DENOTES THE CORRESPONDING METHOD ADOPT THE RESAMPLING METHOD. THE OPTIMAL RESULTS FROM THE SAME OBJECT REPRESENTATION METHODS (*i.e.*, MOTIFS AND VCTREE) ARE IN BOLD. THE GLOBAL OPTIMAL RESULTS OF THE SAME FASTER R-CNN ARE UNDERLINED.

Model	SGDet			SGCls			PredCls		
	mR@50/100	R@50/100	Mean	mR@50/100	R@50/100	Mean	mR@50/100	R@50/100	Mean
IMP†	3.8 / 4.8	20.7 / 24.5	13.5	5.8 / 6.0	34.6 / 35.4	20.5	9.8 / 10.5	59.3 / 61.3	35.2
Motifs†	5.7 / 6.6	27.2 / 30.3	17.5	7.7 / 8.2	35.8 / 36.5	22.1	14.0 / 15.3	65.2 / 67.1	40.4
KERN†	6.4 / 7.3	27.1 / 29.8	17.7	9.4 / 10.0	36.7 / 37.4	23.4	17.7 / 19.2	65.8 / 67.6	32.0
VCTree†	6.9 / 8.0	27.9 / 31.3	18.5	10.1 / 10.8	38.1 / 38.8	24.5	17.9 / 19.4	66.4 / 68.1	43.0
GPS-Net†	8.7 / 9.8	28.4 / 31.7	19.7	11.8 / 12.6	39.2 / 40.1	25.9	21.3 / 22.8	66.9 / 68.8	45.0
Schemata†	-	-	-	10.1 / 10.9	39.1 / 39.8	25.0	19.1 / 20.7	66.9 / 68.4	43.8
PCPL† $_u$	9.5 / 11.7	14.6 / 18.6	13.6	18.6 / 19.6	27.6 / 28.4	23.6	35.2 / 37.8	50.8 / 52.6	44.1
VTransE+	5.0 / 6.0	29.7 / 34.3	18.8	8.2 / 8.7	38.6 / 39.4	23.7	14.7 / 15.8	65.7 / 67.6	41.0
CogTree $_u$	11.1 / 12.7	19.5 / 21.7	16.3	15.7 / 16.7	22.9 / 23.4	19.7	28.4 / 31.0	38.4 / 39.7	34.4
BGNN $^\diamondsuit_u$	10.7 / 12.6	31.0 / 35.8	22.5	14.3 / 16.5	37.4 / 38.5	26.7	30.4 / 32.9	59.2 / 61.3	46.0
Motifs-Baseline	5.8 / 7.8	**32.5** / **37.3**	20.9	8.0 / 8.5	**39.3** / **40.1**	24.0	4.6 / 15.8	**66.1** / **68.0**	41.1
Motifs-Reweight $_u$	8.6 / 10.1	26.4 / 30.7	19.0	11.6 / 13.0	34.2 / 35.9	23.7	20.3 / 23.2	57.3 / 61.0	40.5
Motifs-Resample $^\diamondsuit_u$	8.2 / 9.7	30.5 / 35.4	21.0	11.0 / 11.8	37.9 / 38.8	24.9	18.5 / 20.0	64.6 / 66.7	42.5
Motifs-EBM $_u$	7.7 / 9.3	31.7 / 36.3	21.3	10.2 / 11.0	39.2 / 40.0	25.1	18.0 / 19.5	65.2 / 67.3	42.5
Motifs-TDE $_u$	8.2 / 9.8	16.9 / 20.3	13.8	13.1 / 14.9	27.7 / 29.9	21.4	25.5 / 29.1	46.2 / 51.4	38.1
Motifs-CogTree $_u$	10.4 / 11.8	20.0 / 22.1	16.1	14.9 / 16.1	21.6 / 22.2	18.7	26.4 / 29.0	35.6 / 36.8	32.0
Motifs-DC (ours)	**11.8** / **14.0**	30.1 / 34.3	**22.6**	**15.3** / **16.4**	**39.3** / 40.0	**27.8**	**27.9** / **30.2**	59.7 / 61.4	**44.8**
VCTree-Baseline	5.7 / 6.9	**31.5** / **36.2**	20.1	7.5 / 7.9	40.5 / 41.4	24.3	14.9 / 16.1	**66.2** / **68.1**	41.3
VCTree-EBM $_u$	7.7 / 9.1	31.4 / 35.9	21.0	12.5 / 13.5	**44.7** / **45.8**	**29.1**	18.2 / 19.7	64.0 / 65.8	41.9
VCTree-TDE $_u$	9.3 / 11.1	19.4 / 23.2	15.8	12.2 / 14.0	25.4 / 27.9	19.9	25.4 / 28.7	47.2 / 51.6	38.2
VCTree-CogTree $_u$	10.4 / 12.1	18.2 / 20.4	15.3	**18.8** / **19.9**	30.9 / 31.7	25.3	27.6 / 29.7	44.0 / 45.4	36.7
VCTree-DC (ours)	**12.1** / **13.8**	29.6 / 33.7	**22.3**	16.5 / 17.4	38.5 / 39.2	27.9	**27.7** / **29.8**	60.5 / 62.4	**45.1**
DCNet (ours)	12.9 / 16.1	29.7 / 34.0	23.2	17.8 / 19.1	37.0 / 37.9	27.8	29.2 / 31.5	59.3 / 61.0	45.3
DCNet* (ours)	14.3 / 17.3	28.6 / 32.9	23.3	21.2 / 22.2	36.0 / 36.8	29.1	33.4 / 35.6	57.3 / 59.1	46.4

and top 100 predicate categories by frequency. After the processing, we totally obtain 57,623 images with an average of 14.8 objects and 4.7 relationship triplets per image. We use the same dataset split strategy of VG150 for GQA dataset. Compared to VG150, the GQA dataset is more challenging as the objects and predicates categories are more numerous and diverse.

Evaluation Tasks. Following the previous studies [9], [16], [31], [48], we adopt the following three tasks to evaluate the performance: 1) Predicate Classification (**PredCls**) predicts the predicate class of each object pair with both ground truth bounding boxes and object labels. 2) Scene graph classification (**SGCls**) predicts both the object classes and the predicate class of each object pair with the ground truth bounding boxes. And 3) scene graph detection (**SGDet**) generates scene graphs only based on the given image.

Evaluation Metrics. We use the recall of the triplets to evaluate the aforementioned three tasks as [23], [31]. There are two types of recall which are widely adopted in SGG: recall@K (**R@K**) [16], [18] and mean recall@K (**mR@K**) [23], [31]. R@K indicates the correctness of generated relationships on the whole, which could be easily dominated by head classes owing to the long-tailed data distribution. mR@K, defined as the average R@K of all predicate classes, which validates the unbiased scene graph generation and is mainly affected by plenty of tail predicate classes. We set $K \in \{50, 100\}$ In this work. we also use the **Mean** of R@50, R@100, mR@50, and mR@100 to complementary evaluate the model's capability of generating correct and unbiased scene graphs.

Implementation Details. We use a pre-trained Faster R-CNN with ResNeXt-101-FPN provided by [11] to detect the objects and derive their features. We adopt the word embedding with dimension 200 provided by Glove [50]. In addition, we employ SGD with a momentum of 0.9 as the optimizer. For all three tasks, the batch size and initial learning rate are consistently set to be 10 and 0.001, respectively. In pattern-predicate correlation construction, the hyperparameters λ and μ are set as 2 and 0.2, respectively. Ultimately, we obtain 3 and 4 object interaction patterns in VG150 and GQA datasets, and the parent predicates in these patterns are $[on,\ has,\ near]$ and $[on,\ near,\ in,\ holding]$, respectively. Accordingly, we deploy 3 and 4 specific predicate classifiers in VG150 and GQA datasets, respectively. The hyperparameter α to weigh the BPR loss is set to 0.02. We adopt the same warm-up and decayed strategy as [11], and each training lasts for 50,000 steps. All our experiments are conducted via the RTX2080 Ti GPU.

B. Compared Methods

On the one hand, in order to evaluate the whole framework of our DCNet, we compare it with state-of-the-art methods, including re-conducted IMP+ [16] by Zellers *et al*. [10], Motifis [10], KERN [31], VCTree [23], GPS-Net [9], Schemata [51], PCPL [13], VTransE+ [52], CogTree [14], BGNN [12], EBM [49], TDE [11] and the debiasing methods, re-weighting and re-sampling. On the other hand, to evaluate the effectiveness of our proposed DC-Predictor, we transplant it to two

TABLE II
DETAILED PERFORMANCE COMPARISON OF DIFFERENT METHODS ON PREDCLS, SGCLS, AND SGDET TASKS ON GQA DATASET WITH RESPECT TO MR@50/100 (%), R@50/100 (%), AND MEAN. THE GLOBAL OPTIMAL RESULTS ARE UNDERLINED. NOTE THAT ALL THE METHODS ARE IMPLEMENTED WITH THE SAME OBJECT DETECTOR, *i.e.*, A PRE-TRAINED FASTER R-CNN WITH RESNEXT-101-FPN. THE OPTIMAL RESULTS ARE UNDERLINED.

Model	SGDet			SGCls			PredCls		
	mR@50/100	R@50/100	Mean	mR@50/100	R@50/100	Mean	mR@50/100	R@50/100	Mean
Motifs	6.4 / 7.9	28.9 / 33.1	19.1	7.0 / 8.2	34.2 / 34.9	21.1	16.4 / 17.1	65.3 / 66.8	41.4
Motifs-DC (ours)	9.4 / 10.7	28.3 / 32.1	20.1	9.9 / 10.4	32.5 / 33.1	21.5	21.4 / 22.5	61.3 / 62.7	42.0
VCTree	6.5 / 7.4	27.3 / 30.9	18.0	7.9 / 8.3	34.1 / 34.8	21.3	16.5 / 17.4	63.8 / 65.7	40.9
VCTree-DC (ours)	8.0 / 9.5	25.8 / 29.1	18.1	10.5 / 11.2	32.2 / 32.8	21.7	22.2 / 23.5	61.3 / 62.9	42.5
DCNet-w/o-DC	6.9 / 8.1	27.1 / 30.8	18.2	8.4 / 8.9	33.2 / 34.1	21.1	17.1 / 17.6	63.0 / 64.9	40.7
DCNet (ours)	9.9 / 11.9	27.9 / 31.6	20.3	12.9 / 13.4	32.4 / 33.0	22.9	23.9 / 25.1	57.6 / 59.1	41.4

baseline models: Motifis [10] and VCTree [23], and denote the variant models as Motifis-DC and VCTree-DC, respectively. For fairness, we also introduce a variant of our model, which adopts the same re-weighting method with Cogtree as an enhancement of our method, denoted as DCNet*.

Table I shows the performance of different methods on VG150 dataset. The results of the Motifs-Baseline and VCTree-Baseline are provided in [11], where the Faster R-CNN detector is employed as our model does. From Table I, we have the following observations:

1) Motifs-DC and VC-Tree-DC achieve the best performance on mR@50/100 in most scenarios, compared with corresponding models that adopt the same object encoding methods, respectively. This indicates that our DC-Predictor is able to improve the tail predicate prediction of the baseline methods (*i.e.*, motifs and VCTree), which may be attributed to its capability of differentiating the subtle differences among similar predicates.

2) Compared with both debiasing methods (*i.e.*, re-weighting and re-sampling) and unbiased models (*e.g.*, TDE, CogTree, and EBM), Motifs-DC and VC-Tree-DC achieve the best performance on Mean in most scenarios. It reflects that our DC-Predictor is able to largely increase the mean recall by slightly sacrificing the overall recall[2]. For example, in the SGDet task, compared with the recent unbiased SGG method CogTree, our Motifs-DC improves the mR@100 of Motifs-Baseline by 6.2% and loses R@100 by 3.0%, while Motifs-CogTree improves the mR@100 by 4.0% but loses R@100 by 15.2%. One possible reason is that investigating the subtle differences may be more effective than directly ignoring the irrelevant predicate prediction as CogTree does to boost the unbiased SGG.

3) Overall, DCNet outperforms the baseline methods in terms of both mR@50/100 and Mean in most scenarios, which demonstrates the superiority of our model over existing methods. In addition, unsurprisingly, equipped with the widely adopted re-weighting method among SGG methods [13], [53], DCNet* surpasses DCNet.

To evaluate the effectiveness of our DC-Predictor on GQA dataset, we conduct experiments based on three different object encoding methods, including Motifis [10], VCTree [23], and DCNet-w/o-DC (*i.e.*, Transformer-based encoder). The results is shown in Table II, and we observe that: 1) Motifs-DC, VC-Tree-DC, and DCNet achieve the best perfor-mance on mR@50/100, compared with corresponding models that adopt the same object encoding methods, respectively. This indicates that our DC-Predictor is able to improve the tail predicate prediction of the base-line methods. 2) Compared with the improve of DC-Predictor in VG150 dataset, that on GQA dataset is relatively smaller. One possible reason is that the predicate category in GQA dataset is more than that in VG dataset, and thus improving the prediction of each predicate (*i.e.*, mR@50/100) in GQA dataset is more challenging.

To provide more detailed comparison, followed [14], we compare our models and the baseline methods with respect to R@100 of the 35 most frequent predicates of VG150 dataset in Fig. 5. In particular, Fig. 5a shows the comparison between our Motifs-DC and the biased Motifs-Baseline, while Fig. 5b shows that between our DCNet and the unbiased CogTree. We find that Motifs-DC has an obvious increase over the recall of the tail predicates compared with Motifs-Baseline, and a slight drop on the recall of the first head predicate on. Meanwhile, we observe that DCNet is comparable to CogTree in terms of the recall of the tail predicates, while significantly outperforming CogTree pertaining to the recall of the head predicates (*e.g.*, on, has, and $wearing$). These observations confirm that our model can largely improve the mean recall with limited sacrifice on the overall recall.

C. Ablation Study

Ablation Study on DC-Predictor. To verify the effectiveness of each component in DCNet, we compare our model with the following derivatives:

1) DCNet-w/o-DC: To evaluate the proposed DC-Predictor, we replace it with one fully-connected layer as the predicate classifier, which is optimized with the cross-entropy loss.

2) DCNet-w/o-MC: To investigate the effectiveness of the multiple classifiers manner, we remove the general pattern classifier as well as the specific predicate classifiers and replace them with one fully-connected layer as the predicate classifier, which is optimized with both cross-entropy loss and BPR loss.

3) DCNet-w/o-BPR: We remove the BPR loss in each specific predicate classifier.

Table III presents the ablation study results. Firstly, we observe that compared with DCNet-w/o-DC, DCNet gains obvious improvements on the mR@50/100 and Mean, which demonstrates the effectiveness of our DC-Predictor in unbiased predicate prediction. Secondly, compared with DCNet, both DCNet-w/o-MC and DCNet-w/o-BPR have an obvious

[2]The decrease on the overall recall is hard to avoid when pursuing the increase on the mean recall of all predicates on a biased dataset.

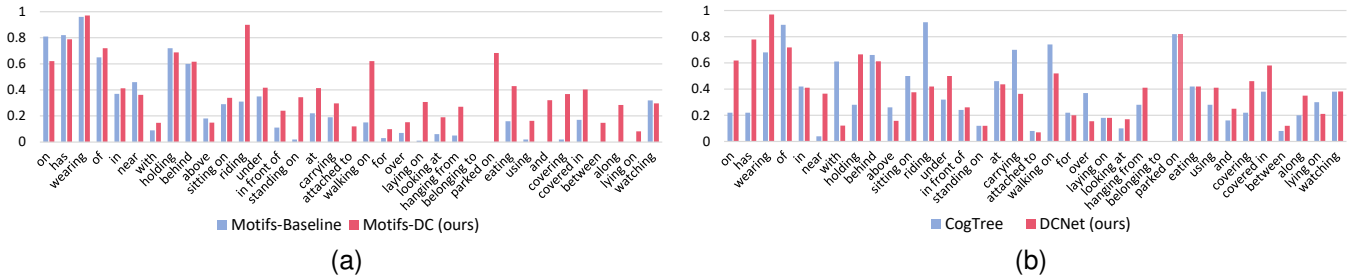


Fig. 5. R@100 of the 35 most frequent predicates in PredCls task on VG150 dataset. (a) Comparison between Motifs-Baseline and Motifs-DC. (b) Comparison between CogTree and DCNet. The results of CogTree are reported according to their paper.

TABLE III
ABLATION STUDY OF DCNET IN TERMS OF MR@50/100, R@50/100, AND MEAN IN SGDET, SGCLS, AND PREDCLS TASKS ON VG150 DATASET.

Model	SGDet			SGCls			PredCls		
	mR@50/100	R@50/100	Mean	mR@50/100	R@50/100	Mean	mR@50/100	R@50/100	Mean
DCNet-w/o-DC	8.1 / 9.5	31.9 / 36.2	21.4	9.7 / 10.3	**40.1** / **40.9**	25.2	17.0 / 18.4	**65.4** / **67.2**	42.0
DCNet-w/o-MC	8.5 / 9.9	31.7 / 36.2	21.6	10.6 / 11.4	39.9 / 40.8	25.6	17.6 / 19.2	65.2 / 67.0	42.3
DCNet-w/o-BPR	8.8 / 10.4	**32.0** / **36.5**	21.9	10.7 / 11.4	39.9 / 40.7	25.7	18.7 / 20.1	65.0 / 66.7	42.6
DCNet (Ours)	**12.9** / **16.1**	29.7 / 34.0	**23.2**	**17.8** / **19.1**	37.0 / 37.9	**27.8**	**29.2** / **31.5**	59.3 / 61.0	**45.3**

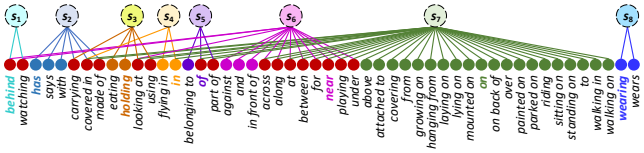


Fig. 6. Pattern-predicate bipartite graph with the unfiltered patterns.

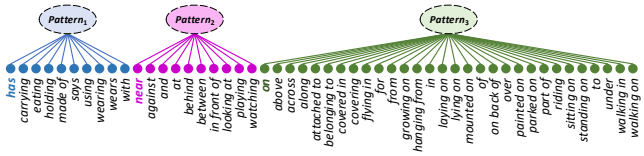


Fig. 7. Pattern-predicate tree.

decrease on the mR@50/100 and Mean. This implies the necessity of using both specific predicate classifiers and the pairwise predicate differentiation to exploit the subtle differences among similar predicates in each pattern.

Ablation Study on Pattern-Predicate Correlation Mining. We also validate the effectiveness of the key steps in pattern-predicate correlation mining with the following derivations:

- DCNet-w/o-Filter: To evaluate the necessity, we conduct our DCNet based on the pattern-predicate bipartite graph with the unfiltered patterns, as shown in Fig. 6.
- DCNet-w/o-Overlap: To demonstrate the superiority of the pattern-predicate bipartite graph that supports one predicate with multiple patterns, we modify the predicate correlation from the pattern-predicate bipartite graph to the pattern-predicate tree, as shown in Fig. 7, where each predicate only connects with its most related pattern.

Table IV shows the performance of different derivations of DCNet with respect to the PredCls task, from which we have following observations:

1) DCNet outperforms DCNet-w/o-Filter, which demonstrates the necessity of filtering the unstable pattern. One possible reason is that the patterns with limited significance may be noise and complicate the predicate correlation, which

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT DERIVATIONS OF DCNET IN TERMS OF R@50/100, MR50/100, AND MEAN IN PREDCLS TASK.

Model	PredCls		
	mR@50/100	R@50/100	Mean
DCNet-w/o-Filter	25.2 / 27.3	57.7 / 59.5	42.4
DCNet-w/o-Overlap	26.9 / 29.1	**59.4** / **61.0**	44.1
DCNet	**29.2** / **31.5**	59.3 / **61.0**	**45.3**

may increase the difficulty of distinguishing the belonging patterns for the object pair samples and thus mislead the model to distinguish uncorrelated predicates.

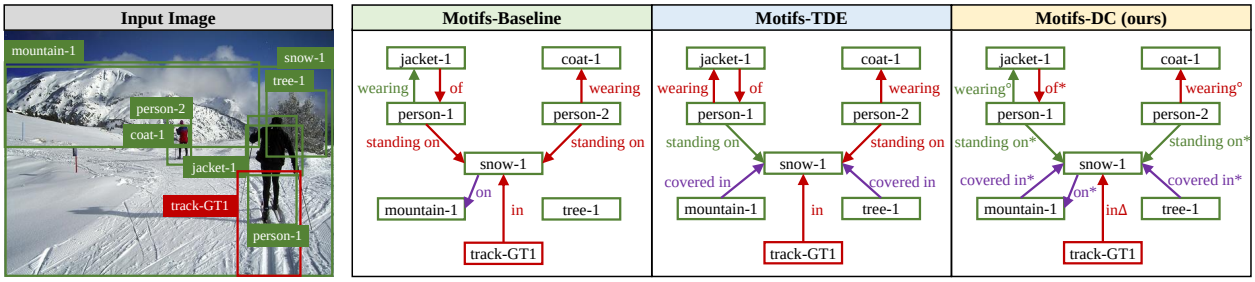
2) DCNet surpasses DCNet-w/o-Overlap, indicating that it is more reasonable to depict the predicate correlation with the pattern-predicate bipartite graph rather than the tree.

D. Qualitative Results

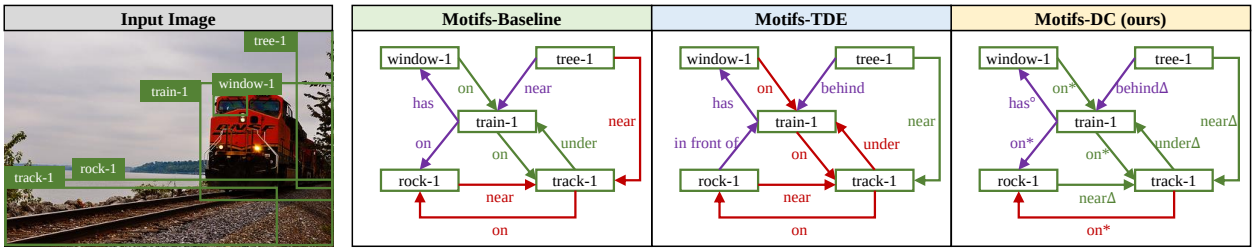
We present the intuitive comparison of DCNet and other state-of-the-art SGG methods, including the biased SGG model Motifs [10] and the debiasing SGG model TDE [11]. As shown in Fig. 8, we distinguish the predicted predicates from different specific predicate classifiers with different marks. We observe that: 1) compared with the scene graphs generated by Motifs-Baseline, the scene graphs generated by Motifs-TDE and Motifs-CG contain more informative relationships (tail predicates), *e.g.*, *person standing on snow* and *mountaion covered in snow* in Example 1. 2) Though Motifs-TDE boosts the tail predicate prediction, it overly focus on the tail predicates and loses performance on the predicting of the general head predicates (*e.g.*, *on* and *has*). For example, *train on track* in Example 2, *pole near bike* in Example 3, and *person wearing jacket* in Example 4 are captured by our Motifs-DC but are not captured by Motifs-TDE. These qualitative results reflect the comprehensive improvement of our method in SGG.

E. Parameter Analyses

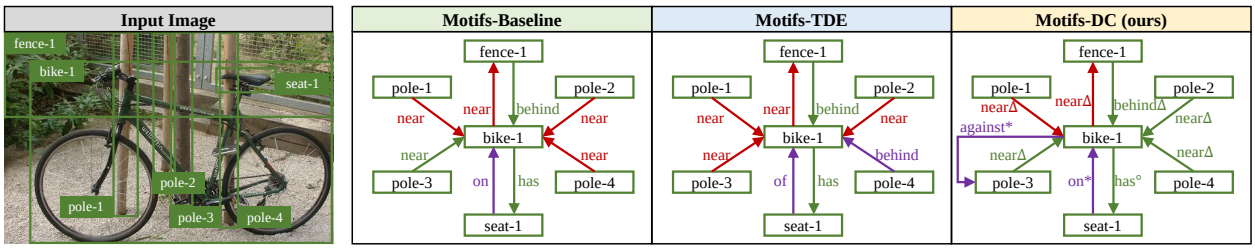
To investigate the effect of the hyperparameter λ in pattern excavation, we illustrate the performance of DCNet with



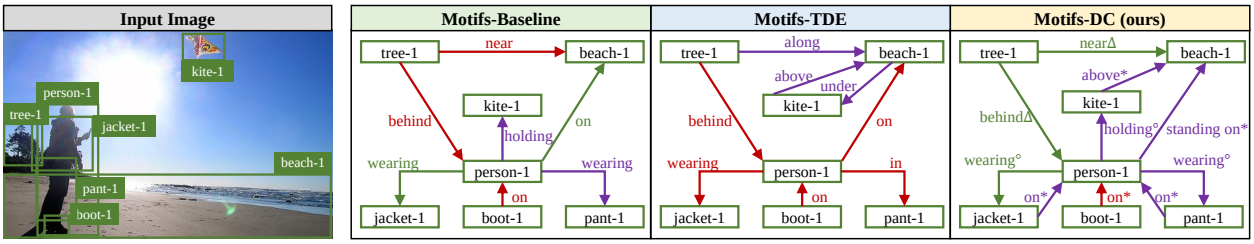
(a) Example 1.



(b) Example 2.



(c) Example 3.



(d) Example 4.

Fig. 8. Generated scene graph examples by Motifs-Baseline, Motifs-TDE, and Motifs-DC in SGDet task with R@20. Green boxes are correctly detected objects with IOU larger than 0.5, while red ones refer to the misclassified objects. Green edges indicate that the corresponding relationships are captured by the top 20 predicted places, while the red ones refer to the uncaught ground truth relationships. Purple edges denote the reasonable captured relationships, but are not annotated as the ground truth. The predicted predicates from the specific predicate classifiers of father predicate *on*, *near*, and *has* are marked with *, △, and °, respectively.

different value of λ in Table V. We found that the number of the excavated patterns increases with the decrease of λ, and the performance of predicate prediction (*i.e.*, mR@50/100, R@50/100, and Mean) is decrease with the increase of the excavated patterns. The underlying reason may be that the increase of the excavated patterns makes the pattern recognition task in the general pattern classifier more challenging, which may result more error propagation in the general pattern classifier and thus affect the following specific predicate prediction.

TABLE V
THE NUMBER OF EXCAVATED PATTERNS, R@50/100, MR@50/100, AND MEAN OF DCNET IN PREDCLS TASK WITH RESPECT TO THE HYPERPARAMETER PARAMETER λ ON VG150 DATASET.

λ	# Pattern	PreCls		
		mR@50/100	R@50/100	Mean
1.2	6	26.1 / 28.0	58.5 / 59.8	43.1
1.6	4	26.7 / 28.4	59.2 / 60.6	43.7
2.0	3	**29.2 / 31.5**	**59.3 / 61.0**	**45.3**
2.4	3	**29.2 / 31.5**	**59.3 / 61.0**	**45.3**

To investigate the detailed influence of the pairwise predicate preference modeling, we illustrate the performance of DCNet with different BPR loss weights (*i.e.*, α) in Table VI. We find that with the growing of the value of α, the performance of mR@50/100 gradually increases and then plateaus, while that of R@50/100 continuously decreases until being stable. One possible reason is that the BPR loss favors the tail predication. Thus, the larger weight encourages the model to slightly favor the tail predicates, leading the increase on

TABLE VI
R@50/100, MR@50/100, AND MEAN OF DCNET IN PREDCLS TASK WITH RESPECT TO THE TRADE-OFF PARAMETER α ON VG150 DATASET.

α	PreCls				
	mR@50	mR@100	R@50	R@100	Mean
0.01	28.2	30.3	**60.7**	**62.3**	**45.4**
0.02	29.2	31.5	59.3	61.0	45.3
0.03	29.1	32.1	58.7	60.4	45.1
0.04	**30.4**	**32.7**	56.6	58.2	44.5
0.05	30.0	32.4	56.7	58.2	44.3

TABLE VII
MR@50/100 AND PARAMS IN PREDCLS TASK OF MOTIFS AND MOTIFS-DC. PARAMS DENOTED THE NUMBER OF PARAMETERS.

Method	mR@50	mR@100	Params (millions)
Motifs	5.8	7.8	367.266M
Motifs-DC (ours)	11.8	14.0	367.704M

the mR@50/100 as well as the decrease on the R@50/100. Overall, the performance in terms of the Mean is relatively stable.

F. Efficiency Analysis

To verify the efficiency of our DC-Predictor, we compare the number of parameters of Motifs [10] and Motifs-DC, which is equipped with our DC-Predictor. As shown in Table VII, compared with Motifs, Motifs-DC has limited parameter addition, demonstrating that our DC-Predictor can efficiently boost other SGG methods.

V. CONCLUSION AND FUTURE WORK

In this work, we present a model-agnostic DC-Predictor to promote the unbiased SGG. In particular, we first develop a pattern-predicate correlation mining algorithm to discover the similar predicates that share the same object interaction patterns. Based on that, we devise a general pattern classifier and a set of specific predicate classifiers. We use the general pattern classifier to recognize the pattern for each object pair sample and rout it to the corresponding specific predicate classifier, which is able to distinguish the subtle differences among similar predicates in each pattern. Experiments on two datasets indicate that our model is able to obviously increase the mean recall with the slighter loss on the overall recall of predicates, thus materially boost the unbiased SGG. In addition, considering that the error propagation produced in the general pattern classifier may affect the following specific predicate prediction, we plan to improve the general pattern classifier in the future.

REFERENCES

[1] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. S. Davis, "Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1247–1257.

[2] Y. Peng and J. Chi, "Unsupervised cross-media retrieval using domain adaptation with scene graph," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4368–4379, 2019.

[3] S. Chen, Q. Jin, P. Wang, and Q. Wu, "Say as you wish: Fine-grained control of image caption generation with abstract scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9962–9971.

[4] C. Yan, Y. Hao, L. Li, J. Yin, A. Liu, Z. Mao, Z. Chen, and X. Gao, "Task-adaptive attention for image captioning," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 32, no. 1, pp. 43–51, 2021.

[5] H. Luo, G. Lin, Z. Liu, F. Liu, Z. Tang, and Y. Yao, "Segeqa: Video segmentation based visual attention for embodied question answering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9667–9676.

[6] H.-T. Su, C.-H. Chang, P.-W. Shen, Y.-S. Wang, Y.-L. Chang, Y.-C. Chang, P.-J. Cheng, and W. H. Hsu, "End-to-end video question-answer generation with generator-pretester network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4497–4507, 2021.

[7] C. Han, F. Shen, L. Liu, Y. Yang, and H. T. Shen, "Visual spatial attention network for relationship detection," in *Proceedings of the 26th ACM international conference on multimedia*, 2018, pp. 510–518.

[8] A. Kolesnikov, A. Kuznetsova, C. Lampert, and V. Ferrari, "Detecting visual relationships using box attention," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 1749–1753.

[9] X. Lin, C. Ding, J. Zeng, and D. Tao, "Gps-net: Graph property sensing network for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3746–3753.

[10] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5831–5840.

[11] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3716–3725.

[12] R. Li, S. Zhang, B. Wan, and X. He, "Bipartite graph network with adaptive message passing for unbiased scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 109–11 119.

[13] S. Yan, C. Shen, Z. Jin, J. Huang, R. Jiang, Y. Chen, and X.-S. Hua, "Pcpl: Predicate-correlation perception learning for unbiased scene graph generation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 265–273.

[14] J. Yu, Y. Chai, Y. Hu, and Q. Wu, "Cogtree: Cognition tree loss for unbiased scene graph generation," *arXiv preprint arXiv:2009.07526*, 2020.

[15] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[16] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5410–5419.

[17] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6700–6709.

[18] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *European conference on computer vision*, 2016, pp. 852–869.

[19] S. Woo, D. Kim, D. Cho, and I. S. Kweon, "Linknet: Relational embedding for scene graph," *arXiv preprint arXiv:1811.06410*, 2018.

[20] Z. Cui, C. Xu, W. Zheng, and J. Yang, "Context-dependent diffusion network for visual relationship detection," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1475–1482.

[21] W. Liao, B. Rosenhahn, L. Shuai, and M. Ying Yang, "Natural language guided visual relationship detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 444–453.

[22] Y. Bin, Y. Yang, C. Tao, Z. Huang, J. Li, and H. T. Shen, "Mr-net: exploiting mutual relation for visual relationship detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8110–8117.

[23] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu, "Learning to compose dynamic tree structures for visual contexts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6619–6628.

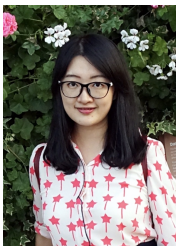
[24] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 670–685.

[25] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, "Attentive relational networks for mapping images to scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3957–3966.
[26] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging knowledge graphs to generate scene graphs," in *European Conference on Computer Vision*, 2020, pp. 606–623.
[27] H. Zhou, C. Zhang, and C. Hu, "Visual relationship detection with relative location mining," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 30–38.
[28] A. Zareian, H. You, Z. Wang, and S.-F. Chang, "Learning visual commonsense for robust scene graph generation," *arXiv preprint arXiv:2006.09623*, 2020.
[29] A. Zareian, S. Karaman, and S.-F. Chang, "Weakly supervised visual semantic parsing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3736–3745.
[30] N. Xu, A.-A. Liu, Y. Wong, W. Nie, Y. Su, and M. Kankanhalli, "Scene graph inference via multi-scale context modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1031–1041, 2020.
[31] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6163–6171.
[32] V. S. Chen, P. Varma, R. Krishna, M. Bernstein, C. Re, and L. Fei-Fei, "Scene graph prediction with limited labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2580–2590.
[33] B. Zhao, Z. Mao, S. Fang, W. Zang, and Y. Zhang, "Semantically similarity-wise dual-branch network for scene graph generation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
[34] K. Ye and A. Kovashka, "Linguistic structures as weak supervision for visual scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8289–8299.
[35] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 5, pp. 1747–1756, 2019.
[36] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, "Making sense of spatio-temporal preserving representations for eeg-based human intention recognition," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 3033–3044, 2019.
[37] X. He, H. Zhang, M. Y. Kan, and T. S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 549–558.
[38] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, p. 144–150.
[39] Y. Chen, S. Gong, and L. Bazzani, "Image search with text feedback by visiolinguistic attention learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3001–3011.
[40] M. Portaz, H. Randrianarivo, A. Nivaggioli, E. Maudet, C. Servan, and S. Peyronnet, "Image search using multilingual texts: a cross-modal learning approach between image and text," *arXiv preprint arXiv:1903.11299*, 2019.
[41] X. Han, Z. Wu, Y. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *Proceedings of the ACM International Conference on Multimedia*, 2017, pp. 1078–1086.
[42] X. Song, F. Feng, X. Han, X. Yang, W. Liu, and L. Nie, "Neural compatibility modeling with attentive knowledge distillation," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018, pp. 5–14.
[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.
[45] Y. Zhou, S. Sun, C. Zhang, Y. Li, and W. Ouyang, "Exploring the hierarchy in relation labels for scene graph generation," *arXiv preprint arXiv:2009.05834*, 2020.
[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[47] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332
[48] W. Wang, R. Wang, S. Shan, and X. Chen, "Exploring context and visual pattern of relationship for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8188–8197.
[49] M. Suhail, A. Mittal, B. Siddiquie, C. Broaddus, J. Eledath, G. Medioni, and L. Sigal, "Energy-based learning for scene graph generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 936–13 945.
[50] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
[51] S. Sharifzadeh, S. M. Baharlou, and V. Tresp, "Classification by attention: Scene graph classification with prior knowledge," *arXiv preprint arXiv:2011.10084*, 2020.
[52] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5532–5540.
[53] R. Yu, A. Li, V. I. Morariu, and L. S. Davis, "Visual relationship detection with internal and external linguistic knowledge distillation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1974–1982.

Xianjing Han received the B.E. degree from Northeastern University of China in 2017. She currently is a Ph.D. student from the School of Computer Science and Technology at Shandong University, supervised by Liqiang Nie and Xuemeng Song. Her research interest comprises multimedia computing and computer vision. She has published several papers in the top venues, such as ACM SIGIR, MM and IEEE TIP.

Xingning Dong received the B.E. degree from the School of Computer Science and Communication Engineering, Jiangsu University, in 2020, where he is currently pursuing the master's degree with the School of Computer Science and Technology, Shandong University. His research interests include computer vision, multimodal pretraining, and scene understanding.

Xuemeng Song received the B.E. degree from University of Science and Technology of China in 2012, and the Ph.D. degree from the School of Computing, National University of Singapore in 2016. She is currently an assistant professor of Shandong University, Jinan, China. Her research interests include the information retrieval and social network analysis. She has published several papers in the top venues, such as ACM SIGIR, MM and TOIS. In addition, she has served as reviewers for many top conferences and journals.



Tian Gan is currently an Associate Professor with the School of Computer Science and Technology, Shandong University. She received her B.S. from East China Normal University in 2010, and the Ph.D. degree from National University of Singapore, Singapore, in 2015. She was a Research Scientist in Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR). Her research interests include social media marketing, video understanding, and multimedia computing.

Yibing Zhan obtained his bachelor's degree and doctor's degree from the information science and technology school at the University of Science and Technology of China in 2012 and 2018. After graduating with a doctor's degree, from 2018 to 2020, Yibing Zhan served as an associate researcher in the school of computer science of Hangzhou Dianzi University. Now, Yibing Zhan works in the JD Explore Academy as an algorithm scientist and head of graph neural networks. He mainly explores graph models and multimodal learning tasks, such as cross-modal retrieval, scene graph generation, and graph neural networks. He has published many scientific papers in top conferences and journals, including CVPR, ACM mm, AAAI, IJCV, and IEEE TMM.

Yan Yan is currently a Gladwin Development Chair Assistant Professor in the Department of Computer Science at Illinois Institute of Technology. He was an assistant professor at the Texas State University, a research fellow at the University of Michigan and the University of Trento. He received his Ph.D. in Computer Science at the University of Trento. His research interests include computer vision, machine learning and multimedia.

Liqiang Nie, IAPR Fellow, is currently the dean with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen campus). He received his B.Eng. and Ph.D. degree from Xi'an Jiaotong University and National University of Singapore (NUS), respectively. His research interests lie primarily in multimedia content analysis and information retrieval. Dr. Nie has co-/authored more than 100 CCF-A papers and 5 books, with 15k plus Google Scholar citations. He is an AE of IEEE TKDE, IEEE TMM, IEEE TCSVT, ACM ToMM, and Information Science. Meanwhile, he is the regular area chair or SPC of ACM MM, NeurIPS, IJCAI and AAAI. He is a member of ICME steering committee. He has received many awards over the past three years, like ACM MM and SIGIR best paper honorable mention in 2019, the AI 2000 most influential scholars 2020, SIGMM rising star in 2020, MIT TR35 China 2020, DAMO Academy Young Fellow in 2020, SIGIR best student paper in 2021, first price of the provincial science and technology progress award in 2021 (rank 1), and provincial youth science and technology award in 2022. Some of his research outputs have been integrated into the products of Alibaba, Kwai, and other listed companies.