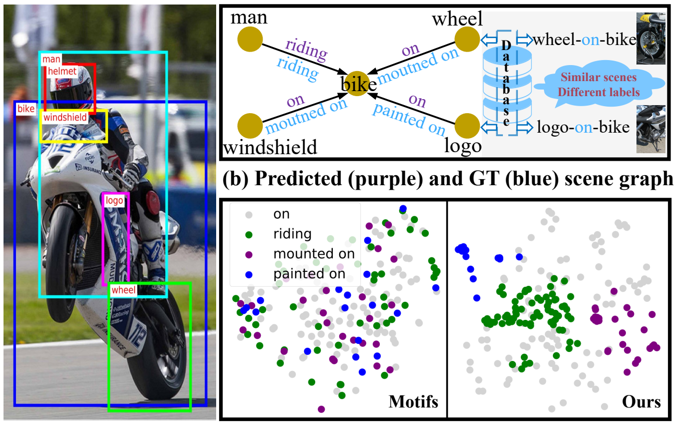
# Adaptive Feature Learning for Unbiased Scene Graph Generation

Jiarui Yang, Chuan Wang, Liang Yang, Yuchen Jiang, Angelina Cao

***Abstract*—Scene Graph Generation (SGG) aims to detect all objects and identify their pairwise relationships in the scene. Recently, tremendous progress has been made in exploring better context relationship representations. Previous work mainly focuses on contextual information aggregation and uses de-biasing strategies on samples to eliminate the preference for head predicates. However, there remain challenges caused by indeterminate feature training. Overlooking the label confusion problem in feature training easily results in a messy feature distribution among the confused categories, thereby affecting the prediction of predicates. To alleviate the aforementioned problem, in this paper, we focus on enhancing predicate representation learning. Firstly, we propose a novel Adaptive Message Passing (AMP) network to dynamically conduct information propagation among neighbors. AMP provides discriminating representations for neighbor nodes under the view of de-noising and adaptive aggregation. Furthermore, we construct a feature-assisted training paradigm alongside the predicate classification branch, guiding predicate feature learning to the corresponding feature space. Moreover, to alleviate biased prediction caused by the long-tailed class distribution and the interference of confused labels, we design a Bi-level Curriculum learning scheme (BiC). The BiC separately considers the training from the feature learning and de-biasing levels, preserving discriminating representations of different predicates while resisting biased predictions. Results on multiple SGG datasets show that our proposed method AMP-BiC has superior comprehensive performance, demonstrating its effectiveness.**

***Index Terms*—Unbiased Scene Graph Generation, Adaptive Message Passing, Bi-Level Unbiased Training, and Feature Enhancement.**

## I. Introduction



**(b) Predicted (purple) and GT (blue) scene graph**

**(a) Input Image** **(c) Distribution of learned features**

Fig. 1. Scattered distribution of predicate representations caused by label confusion. (a) Image with ground-truth object bounding boxes and labels. (b) Left: Confused SG generated by Motifs [1] (colored with purple) and GT (colored with blue). Right: examples explain the label confusion that similar scenes have different labels. (c) Feature distribution with t-SNE. Tangled distribution with inseparable semantic-similar predicates (left) and distinguishable representations (right, generated by the proposed AMP-BiC).

SCENE Graph (SG) is a graphical formulation including all of the objects and their pairwise relationships in the scene, defining a comprehensive and multi-level scene understanding. An interaction relationship in SG is represented by a triplet as <*Subject, Predicate, Object*>, and all relationships are formulated as a graph structure to describe the visual scene. The generated SG provides not only visual context but also interaction knowledge, thus benefiting multiple visual tasks including VQA [2, 3], image retrieval [4–6], and image captioning [7–10].

Despite the promise of graph-level scene perception, the primary challenge in SGG remains the development of effective feature learning. It aims to aggregate and generate more representative predicate features, achieving accurate and insightful scene graph understanding. To conduct information aggregation, recent work [11–18] has explored various message passing networks to effectively capture neighboring and global information. With the fact that some predicate categories hold few samples, some model-agnostic unbiased prediction methods [19–23] have been proposed to boost the performance of tail predicates. However, these approaches overlook the *label confusion* problem inherent in scene graph datasets, which appears between general (e.g. “on” and “in”) and specific (e.g. “mounted on” and “painted on”) predicates. As shown in Fig. 1 (b), relationships of the object-pair

This work was supported in part by the National Key Research and Development Program of China (NO.2022YFB3102200), and in part by the National Natural Science Foundation of China (No. 62102413, 62376088, 62132006, 62376151). (Corresponding author: Chuan Wang)

Jiarui Yang is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China, and also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China (yangjiarui@iie.ac.cn)

Chuan Wang is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China, and also with Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen, 518060, China (wangchuan@iie.ac.cn)

Liang Yang is with the School of Artificial Intelligence, Hebei University of Technology, 300401, Tianjin, China (yangliang@vip.qq.com)

Yuchen Jiang is with the School of Cyber Science and Technology, Shenzhen Campus, Sun Yat-sen University, Shenzhen 518107, China (jiangych39@mail.sysu.edu.cn)

Angelina Cao is with the Montgomery Blair High School-Magnet(STEM) Program, MD 200850, US(Angelinascao@gmail.com)

Manuscript received 9 May 2023; revised 14 October 2023 and 9 January 2024; accepted 7 February 2024

"wheel-bike" have different annotations in similar scenes, i.e., "mounted on" in Fig. 1 (a) and "on" in the image from the database.

Under the label confusion problem, the aggregation of information between nodes inevitably suffers severe interference coming from noise. For similar scenes, the model struggles to provide a clear prediction, leading to a messy feature distribution among all confused categories without distinction. As shown in Fig. 1 (c), the messy feature distribution occurs not only between head and tail predicates (e.g. "on" and "painted on") but also among tail predicates (e.g. "mounted on" and "painted on"). Although unbiased prediction methods pay more attention to the training weight or time for tail predicates, they still fail to address the issue of messy features for confused predicates during the supervised training process. Other work [24, 25] uses label correction, which transfers noisy ground-truth (general) predicates to high-quality (specific) ones, to alleviate label confusion. However, this direct modification requires complex manual design, introducing more noise and label uncertainty.

Based on the aforementioned considerations, in this paper, we provide an effective and general solution, termed AMP-BiC, from the view of feature learning for complex scene graph understanding. AMP-BiC simultaneously achieves both the *discriminated information propagation and aggregation during message passing* and the *de-confusion and de-bias during training*. Specifically, we first design an Adaptive Message Passing (AMP) network, which aims to dynamically conduct information propagation and aggregation among neighbors. The AMP network is formulated as a GNN-based graph de-noising process, consisting of noise separation and residual reinforcement modules. The noise separation module constructs confidence-guided attention to adaptively suppress noisy nodes and aggregate valid neighbor information. The residual reinforcement module introduces a weighted residual connection between the initial and the aggregated features, thus conducting the aggregation via adaptively controlling the aggregation weight. The proposed AMP network simultaneously achieves smoothing noisy nodes and preserving the discriminative features.

Then, we design a feature-assisted training paradigm aiming to preserve the distinguishability of predicate representations. Since there are hardly direct constraints on predicate features for a classifier-alone scheme, we strip off a predicate feature training branch along with the predicate classification branch. The feature training branch is incorporated with a predicate prototype-based contrastive learning, thus forcing learned predicate representations to be pulled toward their corresponding prototypes, making representations separable.

Based on the designed feature-assisted training paradigm, we further design a Bi-level Curriculum (BiC) learning strategy to simultaneously achieve de-confusing and de-biasing. BiC divides the entire training process into two levels: *feature-level* and *predicate level*. At the feature level, we bring in a curriculum coefficient that continuously controls the loss weights of the feature learning branch (*large* → *small*) and the classifier learning branch (*small* → *large*) along with training. At the predicate level, we introduce a curriculum coefficient array, whose element represents the weight of a predicate corresponding to its sample frequency and varies throughout the training. That is, for head predicates: zero → small, whereas for tail predicates: large → larger. By linking these two levels, as training starts, feature training (*feature-level*) and tail predicate learning (*predicate-level*) are implemented with large weights. So confusing scenes marked with head predicates hardly interfere with tail predicates' learning. As the training proceeds, the feature training of the tail predicates gradually reaches the concave point. Thus, gradually increasing the weights of head predicates improves their corresponding performance meanwhile having little impact on the performance of tail predicates, thereby solving both the label confusion and unbiased prediction problems.

The main contributions are summarized as follows:

- We propose an Adaptive Message Passing (AMP) network to dynamically propagate and aggregate neighbor and initial features between entities and predicates, which effectively conducts graph de-noising thus keeping features representative in complex scene graphs.
- We design a predicate feature-assisted training paradigm incorporated with prototype-based contrastive learning to directly perform the discriminating feature learning.
- A novel Bi-level Curriculum (BiC) learning scheme is presented, which has the strong ability to simultaneously solve intertwined label confusion and biased learning.
- We achieve comprehensive performance superiority on multiple SGG datasets, proving that better feature representations significantly improve the performance of predicate classification.

## II. Related Work

**Message Passing in Scene Graph Generation.** SGG aims to use a comprehension graph structure to represent a scene image. Early works pay more attention to exploring different networks, such as GNN [26–29], CRF [30], and RNN/LSTM [1, 31], to model the message passing and aggregation mechanisms between the entities and predicates. Recently, more researchers have considered contextual information. [11, 12] constrain the rationality of the distribution of the scene graph structure by introducing a global energy value. [13, 32] capture better global contextual information and dependencies through Transformer [33]. [14, 15, 18] delve deep into the attributes of nodes on a graph-based network and fully utilize their contextual information to achieve better feature representations. However, they ignore the differences and complexity of the scenes. Continuous fusion easily makes the representations between nodes more similar, thus resulting in an oversmoothing problem [34]. HL-Net [16] is the first work to consider the heterophily in the scene graph. It allows the aggregated weights and teleport probability to be negative (passing the high-frequency signals), and conducts the predicate message passing similar to APPNP [35]. HL-Net [16] successfully preserves the specificity of node features to a certain extent. However, differentiating the sign of aggregation coefficients based on object category may not handle the propagation of confused object pairs. Having two objects or predicates of the same category does little to help in identifying relationships. In this paper, we argue that the ultimate purpose of message passing is to make features discriminating for classification. Enhancing node discrimination during contextual aggregation is not a good choice due to the widespread noise. Being different from previous work, we propose to remove the noise in the scene graph and aggregate in-context information from relevant nodes as much as possible. Additionally, we introduce an adaptive residual aggregation mechanism, which maintains node discrimination by adaptively adjusting the aggregated weight of its clean feature.

**Long-tailed Distribution in Scene Graph Generation.** In the long-tailed class distribution, it is difficult to handle tail predicates without an effective unbiased method. [31, 36] claim using the mean Recall evaluation metric and [19] proposes the first solution for unbiased SGG model prediction. Recent works mainly utilize re-sample [15, 20, 37, 38] or re-weight [22, 23, 39, 40] with some auxiliary means like predicate correlation learning [22, 40], and group learning [20] to alleviate biased prediction. PCPL [40] and PCL [22] utilize the correlations between predicates obtained during training to encourage the network to predict more informative predicates. GCL [20] used a group re-sampling strategy, achieving unbiased prediction by giving more average training time for tail predicates. Following this strategy, MEET [21] makes classifiers among different groups mutually exclusive, thereby effectively preventing the performance degradation of the head predicates. However, these methods ignore the problem of label confusion, which hurts the comprehensive performance.

NARE [41] first proposes the concept of implicit and explicit predicates in SGG and points out that many explicit annotations can be labeled as implicit annotations, *i.e.*, label confusion. It first trains on implicit predicates and then refines the labels of explicit predicates. However, this method is complex and not general, once the types of predicates become numerous, like GQA-LT [42], it is difficult to distinguish them manually. NICE [24] tackles the label confusion problem more aggressively. It takes noisy ground truth labels into account and achieves unbiased prediction by correcting noisy labels. This approach requires multiple steps of re-training, which is cumbersome and has limited applications. IETrans [25] is the most recent work that considers the label confusion problem. It transfers the general predicates to informative ones based on the confusion matrix. In this paper, we propose a novel method to alleviate label confusion at the feature learning level. We first design a feature-assisted training paradigm for directly training distinctive features for samples with different categories. Based on this paradigm, we further design a bi-level curriculum learning scheme, which is quite simple and effective, to enhance the feature training of tail predicates and reduce the interference of confusing labels and data bias, thus achieving de-confusion and de-biasing simultaneously.

## III. Approach

The task of scene graph generation is to parse an image $I$ into a scene graph $\mathcal{G} = \{\mathcal{E}_{sub}, \mathcal{P}, \mathcal{E}_{obj}\}$, where $\mathcal{E}_{sub}$ and $\mathcal{E}_{obj}$ represent the set of subject and object entities, respectively. $\mathcal{P}$ denotes the set of predicates for all entity pairs. Typically, a two-stage SGG model is implemented as follows. The first stage plays the role of entity detection and detects the entities through an object detector (*e.g.*, Faster R-CNN [43]). It outputs three key variables of objects, including initial visual feature $\mathbf{v} \in \mathbb{R}^{d_v \times 1}$, spatial feature $\mathbf{s} \in \mathbb{R}^{d_s \times 1}$ of object bounding box, and category vector $\mathbf{l}_e \in \mathbb{R}^{d_e \times 1}$. Given an entity pair $(e_i, e_j)$, the initial entity feature $\mathbf{e}_{init} \in \mathbb{R}^{d_e \times 1}$and predicate feature $\mathbf{p}_{init} \in \mathbb{R}^{d_p \times 1}$ are generated as,

$$\begin{aligned} (\mathbf{e}_{init})_i &= f_e(\mathbf{v}_i \oplus \mathbf{s}_i), \\ (\mathbf{p}_{init})_{i\to j} &= f_u(\mathbf{u}_{ij}) + f_v((\mathbf{e}_{init})_i \oplus (\mathbf{e}_{init})_j), \end{aligned} \tag{1}$$

where $\oplus$ is a concatenate operation and $\mathbf{u}_{ij}$ denotes the convolutional feature of the union region covering entities $e_i$ and $e_j$. $f_e$, $f_u$ and $f_v$ are three fully-connected operations. The second stage identifies the relationships that exist between all entity pairs composited from the first stage. Each relationship is denoted as a triplet <*subject entity, predicate, object entity*>, and the predicate $\mathbf{l}_p \in \mathcal{P}$ contains *background* category that means no relation.

**Overview.** In this paper, we stick to the idea that better features make better classifiers while presenting an effective and general solution from the feature learning view for complex scene graph understanding, termed AMP-BiC. AMP-BiC performs feature aggregation within the Adaptive Message Passing (AMP) network and utilizes the Bi-level Curriculum (BiC) learning strategy for learning.

Taking initial entity and predicate representations as input, we first construct two graphs: *Entity-to-Entity* graph and *Entity-to-Predicate* graph. The *Entity-to-Entity* graph conducts adaptive message propagation between neighboring nodes and the initial node, thus ensuring discriminating entity feature representations. The *Entity-to-Predicate* graph subsequently takes updated entity features as input and adaptively aggregates the subject and object features to their corresponding predicate features under a bipartite graph formulation. The adaptive aggregation in the AMP network is implemented on both graphs to achieve de-noising and enhancing features during message passing.

To well-explore discriminating representations of predicates, a stripped predicate feature training branch is implemented after the adaptive message passing network. It is deployed in parallel with the predicate classification and updated by a contrastive loss function based on predicate prototypes.

Furthermore, the Bi-level Curriculum learning is designed to simultaneously concatenate the feature-assisted training paradigm to long-tailed class distribution. By dynamically linking the change curves of the two curriculum coefficients, the learning of the AMP network simultaneously reduces interference of confusing labels and achieves bias removal.

In the following, The Adaptive Message Propagation mechanism is introduced in Section III-A. The feature-assisted training paradigm is illustrated in Section III-B. The Bi-level Curriculum learning is presented in Section III-C. An overview is illustrated in Fig. 2.

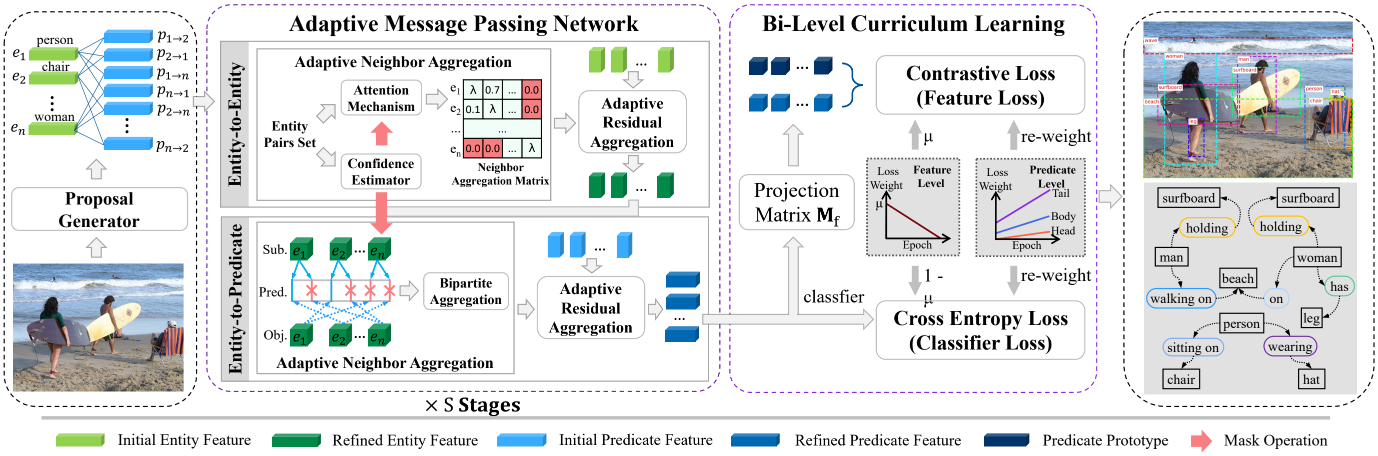


Fig. 2. Overview of the proposed AMP-BiC. The network contains (a) an object detector module to generate relationship proposals, including initial entity and predicate representations. (b) The AMP network contains Entity-to-Entity and Entity-to-Predicate graphs, aggregating features for entities and predicates with the adaptive aggregation mechanism including Adaptive Neighbor Aggregation (ANA) and Adaptive Residual Aggregation (ARA). (c) Predicate feature-assisted training paradigm updated with prototype-based contrastive loss. The bi-level curriculum learning scheme is presented to dynamically adjust loss weight on the feature level and the predicate level, to alleviate label confusion and bias problems simultaneously.

## A. *Adaptive Message Passing Network*

In this section, we first regard the scene graph message passing as a graph signal de-noising process and generalize the paradigm of previous graph-based methods. Then, we propose a general improvement scheme for this paradigm: *Adaptive Neighbor Aggregation* and *Adaptive Residual Aggregation*. Lastly, we apply these improvements to the two sequential graphs (*Entity-to-Entity* and *Entity-to-Predicate*) conducting message passing.

*1) Generalization of GNN-based Message Passing:* When generating initial entity proposals, the scene graph defaults to a directed complete graph. All we need is to conduct effective propagation and aggregation in the graph, introducing noise suppression and related information aggregation. The formulation of the message passing process in SGG can be written as a general graph signal de-noising [44, 45]:

$$\arg\min_{\mathbf{X}\in\mathbb{R}^{N\times D}} L_1(\mathbf{X}) = \lambda\|\mathbf{X}-\mathbf{X}_{\text{init}}\|_F^2 + (1-\lambda)tr(\mathbf{X}^T\tilde{\mathbf{L}}\mathbf{X}), \quad (2)$$

where $\mathbf{X}$ is either entity or predicate features, N is the number of entities or predicates, and $\mathbf{X}_{\text{init}}$ is the initial feature extracted from the object detector. The first term is the optimization goal, which guides the noisy signal $\mathbf{X}$ to be close to clean signal $\mathbf{X}_{\text{init}}$. The second term is the Laplacian regularization that guides $\mathbf{X}$'s smoothness over the scene graph $\mathcal{G}$. $\tilde{\mathbf{L}}$ is the normalized graph laplacian matrix and $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$. $\tilde{\mathbf{A}} = \mathbf{I} + \mathbf{A}$ denotes the adjacency matrix with self-loop and $\tilde{\mathbf{D}}$ is $\tilde{\mathbf{A}}$'s degree matrix. For simplicity, we set $\tilde{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ to represent normalized adjacency matrix, thus $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}$. Since the initial state of the scene graph is a complete graph, the confidence score [15] or attention mechanism [14, 36] are always used to obtain the normalized adjacency matrix $\tilde{\mathbf{A}}$.

Setting a stepsize $\gamma$ to 0.5, the update of $\mathbf{X}$ is written as:

$$\begin{aligned}\mathbf{X}^{k+1} &= \mathbf{X}^k - \gamma\frac{\partial L_1(\mathbf{X}^k)}{\partial \mathbf{X}^k} \\ &= \mathbf{X}^k - 2\gamma[\lambda(\mathbf{X}^k - \mathbf{X}_{\text{init}}) + (1-\lambda)(\mathbf{I}-\tilde{\mathbf{A}})\mathbf{X}^k] \\ &= (1-\lambda)\tilde{\mathbf{A}}\mathbf{X}^k + \lambda\mathbf{X}_{\text{init}}.\end{aligned} \quad (3)$$

Eq. 3 shows a general paradigm for SGG message passing. For example, in BGNN [15], $\lambda = 0$; in RU-Net [18], $\lambda = \frac{1}{3}$; in HL-Net [16], $\lambda > 1$. From this paradigm, we conclude that the optimization of Eq. 2 is a simple aggregation of three components including initial($\mathbf{X}_{\text{init}}$), current(self-loop of $\tilde{\mathbf{A}}$) and neighbor(no self-loop of $\tilde{\mathbf{A}}$) features.

**Consideration.** Analyzing this paradigm, we argue that this updated formulation may not be appropriate for complex scene graph structures. Firstly, the selection of neighbor nodes (elements in $\tilde{\mathbf{A}}$) should be prudent to avoid over-smoothing during message passing in GNN. Direct confidence-based gating may increase the probability of removing valid node pairs, and a single attention mechanism may aggregate some unnecessary information. Secondly, fixing the weight ($\lambda$) for the initial residual ($\mathbf{X}_{init}$) is not proper. For the condition that the noise around the node is small, we should aggregate more neighboring features. For other conditions, since the node noise may be large, a large weight for initial features can suppress the impact of noise. As analyzed above, we improve these two terms of Eq. 2 with an adaptive mechanism, respectively, realizing feature de-noising and enhancement.

*2) Adaptive Neighbor Aggregation (ANA):* To effectively bring information into aggregation, the adjacency matrix $\tilde{\mathbf{A}}$ in Eq. 3 is vital for conducting noise suppression and positive information aggregation. Therefore, we introduce the Adaptive Neighbor Aggregation (ANA) module, which is implemented with a Confidence Estimator (CE) to remove unrelated (low-confidence) edges and an Attention Mechanism (AM) to calculate weights for aggregation.

The Confidence Estimator (CE) module is formulated as a fully connected structure followed by a gate filter. It suppresses information flows holding less contribution in the message propagation. Concretely, for computing the confidence score of the predicate from the entity $e_i$ to the entity $e_j$, we take visual features $\mathbf{v}_i$ and $\mathbf{v}_j$, spatial features $\mathbf{s}_i$ and $\mathbf{s}_j$ as input. After feature embedding with the fully connected layer, we use a gate filter to predict the confidences of predicates. The operation is described as:

$$c_{i\to j} = f_c(\mathbf{v}_i \oplus \mathbf{v}_j \oplus \mathbf{s}_i \oplus \mathbf{s}_j), \tag{4}$$

where $f_c$ is the CE operation. Then predicates holding low confidence are masked thereby preventing message aggregation for unrelated nodes.

For information aggregation in the graph, each entity node provides varying contributions to its neighbors, under different conditions including location, semantic and realistic similarities, etc. Therefore, using the same weight to aggregate surrounding information may hold back the aggregation effectiveness. For example, two entities that are closer together often need to aggregate more information than those farther apart. To this end, we follow the concept of the attention mechanism [33] and present a multi-head self-attention mechanism to capture the correlation between entities.

Specifically, to construct multiple heads, for the $h^{th}$ head, we generate queries $\mathbf{Q}_h$ and keys $\mathbf{K}_h$ based on the current layer entity features $\mathbf{E} \in \mathbb{R}^{N\times d_e}$ as:

$$\mathbf{Q}_h = \mathbf{E}\mathbf{W}_h^Q, \quad \mathbf{K}_h = \mathbf{E}\mathbf{W}_h^K, \tag{5}$$

where $N$ is the number of entities, $\mathbf{W}_h^Q, \mathbf{W}_h^K \in \mathbb{R}^{d_e\times d_e/8}$ are linear projections, and $h = 1, 2, ..., H$ denotes the indication of the head. $H$ is set as 8. To model the correlation between entities under the $h^{th}$ head, we construct the attention weight matrix $\mathbf{A}_h \in \mathbb{R}^{N\times N}$ as:

$$\mathbf{A}_h = \text{softmax}(\frac{\mathbf{Q}_h\mathbf{K}_h^T}{\sqrt{d_e/8}}, \text{mask} = \{(i,j)|c_{i\to j} < \rho\}), \tag{6}$$

where $\sqrt{d_e/8}$ is a scaling factor. Based on the confidence $c_{i\to j}$ estimated from Eq. 4, we use the threshold $\rho$ to mask unrelated edges. By averaging the attention weights from all heads, we obtain the final attention map:

$$\hat{\mathbf{A}} = \frac{1}{H}\sum_{h=1}^{H}\mathbf{A}_h. \tag{7}$$

Compared with the normalized adjacency matrix $\tilde{\mathbf{A}}$ depicted in Eq. 3, the attention map $\hat{\mathbf{A}}$ not only considers the rationality of the existence of the relationship but also aggregates more valid contextual information.

*3) Adaptive Residual Aggregation (ARA):* Then we improve the first term in Eq. 2. We argue that the fixed weight for residual is not proper due to the complexity of the scene graph. Consider the Frobenius norm in Eq. 2, the derivative of each element is independent due to $\nabla_{x_{ij}}\|\mathbf{X}\|_F^2 = 2x_{ij}$. For adaptive residuals, we consider using the $\ell_{2,1}$-norm since $\ell_{2,1}$ is "sample-specific" [46]. $\ell_{2,1}$-norm encourages the rows of $\mathbf{X}$ to be zero, *i.e.*, $x_i - (x_{init})_i \to 0$, which also meets our optimization objective, and it allows each column to be non-zero. To the end, we rewrite Eq. 2, replacing F-norm with $\ell_{2,1}$-norm:

$$\begin{aligned}\underset{\mathbf{X}\in\mathbb{R}^{N\times D}}{\arg\min}\, L_2(\mathbf{X}) =& \lambda\|\mathbf{X} - \mathbf{X}_{\text{init}}\|_{2,1} \\ &+ (1-\lambda)tr(\mathbf{X}^T(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{X}).\end{aligned} \tag{8}$$

However, we found that $h(\mathbf{X}) = \lambda\|\mathbf{X} - \mathbf{X}_{\text{init}}\|_{2,1}$ is actually locally non-differentiable since $\nabla_{x_{ij}}h(\mathbf{X}) = \lambda\frac{x_{ij}-(x_{init})_{ij}}{\|x_i-(x_{init})_i\|_2}$ has an undefined derivative when $x_i - (x_{init})_i = 0$. To this end, we optimize it by proximal gradient descent [47]. First conducting gradient descent on the second term of Eq. 8 with stepsize $\gamma = 0.5$:

$$\begin{aligned}\mathbf{Y}^k &= \mathbf{X}^k - \gamma(1-\lambda)\nabla tr((\mathbf{X}^k)^T(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{X}^k) \\ &= \lambda\mathbf{X}^k + (1-\lambda)\tilde{\mathbf{A}}\mathbf{X}^k.\end{aligned} \tag{9}$$

Then the optimization of Eq. 8 is transformed into a proximal mapping process:

$$\begin{aligned}\mathbf{X}^{k+1} &= \text{prox}_{h,\gamma}(\mathbf{Y}^k) \\ &= \underset{\mathbf{X}\in\mathbb{R}^{N\times D}}{\arg\min}\, \frac{1}{2\gamma}\|\mathbf{X} - \mathbf{Y}^k\|_2^2 + \lambda\|\mathbf{X} - \mathbf{X}_{\text{init}}\|_{2,1}.\end{aligned} \tag{10}$$

Setting $\mathbf{Z} = \mathbf{X} - \mathbf{X}_{\text{init}}$, we further transform Eq. 10 into:

$$\mathbf{X}^{k+1} = \mathbf{X}_{\text{init}} + \underset{\mathbf{Z}}{\arg\min}(\|\mathbf{Z} - (\mathbf{Y}^k - \mathbf{X}_{\text{init}})\|_2^2 + \lambda\|\mathbf{Z}\|_{2,1}). \tag{11}$$

The second term of Eq. 11 is a $\ell_{2,1}$-norm regularized least squares regression problem [48] as its solution [46, 49]:

$$\mathbf{Z}_i = \frac{\mathbf{Y}_i^k - (\mathbf{X}_{\text{init}})_i}{\|\mathbf{Y}_i^k - (\mathbf{X}_{\text{init}})_i\|_2}\max(\|\mathbf{Y}_i^k - (\mathbf{X}_{\text{init}})_i\|_2 - \frac{\lambda}{2}, 0). \tag{12}$$

At last, we obtain the message passing form of Eq. 8:

$$\begin{aligned}\mathbf{X}_i^{k+1} &= (\mathbf{X}_{\text{init}})_i + \beta(\mathbf{Y}_i^k - (\mathbf{X}_{\text{init}})_i) \\ &= (1-\beta)(\mathbf{X}_{\text{init}})_i + \beta[\lambda\mathbf{X}_i^k + (1-\lambda)(\tilde{\mathbf{A}}\mathbf{X}^k)_i],\end{aligned} \tag{13}$$

where $\beta = \max(1 - \frac{\lambda}{2\|\mathbf{Y}_i^k-(\mathbf{X}_{init})_i\|_2}, 0)$.

The second term of the Eq. 13 performs adaptive neighbor aggregation, in which the normalized adjacency matrix $\tilde{\mathbf{A}}$ is equal to $\hat{\mathbf{A}}$ shown in Eq. 7. Compared with Eq. 3, $\beta$ in this formula embodies sample-specific residual aggregation. Except for the anti-over-smoothing effect of the residual module, $\beta$ can also distinguish noise in a complex scene graph environment. Features with noisy nodes will lead $\mathbf{Y}^k$ to be elusive during neighbor features aggregation. In this case, $\beta$ is at a high value, which means low weight for residual connection, thus the noise is gradually smoothed. In contrast, normal features mean high weight for residual connection to tackle the over-smoothing problem and retain more accurate and clear information.

*4) Overall Message Passing Flow:* After discussing the general message aggregation method, we describe the overall message passing on two graphs including *Entity-to-Entity* and *Entity-to-Predicate*, as shown in Fig. 2.

The *Entity-to-Entity* graph first evaluates the confidence score for each relationship proposal and calculates the normalized adjacency matrix $\hat{\mathbf{A}}^k$ by ANA module. Then we adopt the Eq. 13 to adaptively aggregate neighbor features and initial features, which attenuates the effects of noise while maintaining node differentiation:

$$\mathbf{e}_i^{k+1} = (1-\beta)(\mathbf{e}_{\text{init}})_i + \beta\big(\lambda \mathbf{e}_i^k + (1-\lambda)(\hat{\mathbf{A}}^k \mathbf{e}^k)_i\big). \tag{14}$$

The *Entity-to-Predicate* graph is constructed as a bipartite graph following previous methods [14–16, 32]. It treats the subject node and the object node as different sources when aggregating features to predicate:

$$\begin{aligned}(\mathbf{p}^s_{\text{aggr}})_{i\to j} = \mathbf{M}_{e\to p}\big(&f_{\text{mp}}(\tilde{\mathbf{e}}_i^K \oplus \mathbf{p}^s_{i\to j})\tilde{\mathbf{e}}_i^K \\ &- f_{\text{mp}}(\tilde{\mathbf{e}}_j^K \oplus \mathbf{p}^s_{i\to j})\tilde{\mathbf{e}}_j^K\big).\end{aligned} \tag{15}$$

$s$ is the stage indication during the message passing and $K$ is the number of graph layers in the current stage. $\mathbf{M}_{e\to p} \in \mathbb{R}^{d_e \times d_p}$ denotes the projection matrix mapping from entity feature space to predicate feature space. $f_{mp}$ is a gate mechanism that contains a two-layer MLP and a *tanh* output activation function layer. $\oplus$ means concatenate operation. $\tilde{e}$ denotes the refined entity features after Entity-to-Entity graph. We utilize the subtraction operation to distinguish the subject and the object. Again, we use Eq. 13 to perform message passing:

$$\begin{aligned}\mathbf{p}^{s+1}_{i\to j} =& (1-\beta)(\mathbf{p}_{\text{init}})_{i\to j} \\ &+ \beta\big(\lambda \mathbf{p}^s_{i\to j} + (1-\lambda)(\mathbf{p}^s_{\text{aggr}})_{i\to j}\big).\end{aligned} \tag{16}$$

Noted that in the *Entity-to-Predicate* graph, we only perform message aggregation on edges whose confidence score $c_{i\to j} \geq \rho$. After $S$ stages, we obtain the final predicate features.

### B. Predicate Feature-Assisted Training Paradigm

Considering that predicate learning always suffers from confused labeling, learned representations of predicates may be inclined to be indistinguishable as the training progressed (Fig. 1(c) left), resulting in confused classification results (Fig. 1(b) left). To alleviate the cluttered feature distribution, a straightforward idea is to directly constrain the learning of features. Ideally, different predicates should have an obvious dividing line in feature space. To this end, we design a feature-assisted training paradigm shown in Fig. 2. We construct the feature learning branch as a stripped network from the predicate classification network. To make the feature distribution of different predicate categories as scattered as possible, we introduce the contrastive learning mechanism incorporating the guidance from predicate prototypes. The proposed prototype-based contrastive learning loss function aims to pull features of the same predicate toward their corresponding prototype.

Concretely, we extract the prototype of each predicate from the pre-trained GloVe [50] with a 300-dimensional feature vector. The cosine similarity is utilized to measure the distance between output features from the AMP network and prototypes. The loss function is defined as:

$$\mathcal{L}_f(\mathbf{p}_{i\to j}) = -log \frac{\exp(\|\mathbf{M}_f \mathbf{p}_{i\to j}\|_2 \cdot \|\mathbf{q}_{gt}\|_2/\tau)}{\sum_{j\in\mathcal{B}} \exp(\|\mathbf{M}_f \mathbf{p}_{i\to j}\|_2 \cdot \|\mathbf{q}_j\|_2/\tau)}. \tag{17}$$

$\|\cdot\|_2$ represents $\ell^2$-norm, $\mathbf{q}_{gt}$ denotes prototypes, and $\mathbf{M}_f$ is a projection matrix mapping from $\mathbb{R}^{d_p}$ to $\mathbb{R}^{300}$. $\mathcal{B}$ is the set of the ground truth predicates that appear in a mini-batch. $\tau$ is a scalar temperature parameter set with 0.1.

### C. Bi-Level Curriculum Learning Strategy

The long-tailed distribution of predicates has been ubiquitous throughout the SGG datasets (*e.g.* VG [51], GQA [42]). Although the proposed feature-assisted training paradigm relieves the cluttered feature distribution, biased prediction still brings under-trained situations for tail predicates and cannot solve label confusion. Therefore, it is desperate to adequately train features of the tail predicates, while reducing the impact of confusing labels. In this section, we present a novel solution that introduces the bi-level curriculum learning scheme.

The bi-level curriculum learning scheme is conducted as: when training starts, the model emphasizes the training of features and emphatically learns better features for tail predicates; subsequently, the model gradually increases the loss weights of head predicates and emphasizes the training of the classifier. In this way, features of tail predicates achieve to be close to the "concave point" due to the double reinforcement. Meanwhile, the interference label is double-weakened and hardly interferes with the training of the tail predicates.

Concretely, in the bi-level curriculum learning, *the first level* considers joint learning of features and classifiers (*feature-level*). In the initial training stage, since learned features are inaccurate, it makes little contribution to training the classifier. Therefore, the feature training branch occupies a larger weight. As the training goes on, feature representations have been enough learned, so we gradually reduce the weight of this part and increase the weight of the classifier. *The second level* is from the perspective of the long-tailed distribution (*predicate-level*). At this level, we separately consider the importance of predicates based on their sample size. To be specific, we set the initial loss weights of *head-body-tail* predicates [15] to $w^{(0)}$. The growth rate of the loss weight of each predicate is related to the number of samples in the train set:

$$\begin{aligned} w_l^{(0)} &= \begin{cases} \dfrac{\bar{P}_{tail}}{\bar{P}_{head}}, & l \textit{ is a head predicate} \\ \dfrac{\bar{P}_{tail}}{\bar{P}_{body}}, & l \textit{ is a body predicate} \\ 1.0, & l \textit{ is a tail predicate} \end{cases} \\ w_l^{(n)} &= w_l^{(0)} + \frac{n}{n_{\max}}\sqrt{\frac{P_b}{P_l}}. \end{aligned} \tag{18}$$

$\frac{\bar{P}_{tail}}{\bar{P}_{head}}$ represents the ratio of the average occurrence frequency of tail predicates and head predicates. It indicates the initial weight of the head predicates during the training process, with the same rule applying to $\frac{\bar{P}_{tail}}{\bar{P}_{body}}$. $P_l$ denotes the probability of occurrence of predicate $l$, and $b$ is a benchmark predicate (*e.g.* the most frequent tail predicate). $(0)$ and $(n)$ is the first epoch (initial weight) and $n^{th}$ epoch during training, respectively. The maximal epoch number is set with 15 in the experiment. (In addition, we always set the weight of the *background* category to 1.) Different from the previous static re-weighting, our method is dynamic.

From the view of combining two levels, *predicate-level* is successfully linked to *feature-level*. The feature representations of tail predicates are double-emphasized at the beginning of training. At this stage, the long-tail distribution seems to disappear, since only the tail predicates are involved during training.

TABLE I
COMPREHENSIVE COMPARISON ON VG150 WITH STATE-OF-THE-ART UNBIASED METHODS BASED ON RESNEXT-101-FPN BACKBONE. † DENOTES RESULTS REPRODUCED WITH OUR CODE. THE BEST AND SECOND-BEST ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

| **Baseline Models** | **De-bias Methods** | PredCls | | | SGCls | | | SGDet | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | R@50 / 100 | mR@50 / 100 | hR@50 / 100 | R@50 / 100 | mR@50 / 100 | hR@50 / 100 | R@50 / 100 | mR@50 / 100 | hR@50 / 100 |
| JM-SGG [12] | | 70.8 / 71.7 | 24.9 / 28.0 | 36.8 / 40.3 | 43.4 / 44.2 | 13.1 / 14.7 | 20.1 / 22.1 | 29.3 / 32.2 | 9.8 / 11.8 | 14.7 / 17.3 |
| Seq2Seq-RL [13] | | 66.4 / 68.5 | 26.1 / 30.5 | 37.5 / 42.2 | 38.3 / 39.0 | 14.7 / 16.2 | 21.2 / 22.7 | 30.9 / 34.4 | 9.6 / 12.1 | 14.6 / 17.9 |
| Motifs [1] | Baseline | 66.0 / 67.9 | 14.6 / 15.8 | 23.9 / 25.6 | 39.1 / 39.9 | 8.0 / 8.5 | 13.3 / 14.0 | 32.1 / 36.9 | 5.5 / 6.8 | 9.4 / 11.5 |
| | TDE [19] | 46.2 / 51.4 | 25.5 / 29.1 | 32.9 / 37.2 | 27.7 / 29.9 | 13.1 / 14.9 | 17.8 / 19.9 | 16.9 / 20.3 | 8.2 / 9.8 | 11.0 / 13.2 |
| | PCL [22] | 55.0 / 57.3 | 33.6 / 35.8 | 41.7 / 44.1 | 34.2 / 35.2 | 18.2 / 19.1 | 23.8 / 24.8 | 29.0 / 33.4 | 14.2 / 16.6 | 19.1 / 22.2 |
| | NICE [24] | 55.1 / 57.2 | 29.9 / 32.3 | 38.8 / 41.3 | 33.1 / 34.0 | 16.6 / 17.9 | 22.1 / 23.5 | 27.8 / 31.8 | 12.2 / 14.4 | 17.0 / 19.8 |
| | GCL [20] | 42.7 / 44.4 | 36.1 / 38.2 | 39.1 / 41.1 | 26.1 / 27.1 | 20.8 / 21.8 | 23.2 / 24.2 | 18.4 / 22.0 | 16.8 / 19.3 | 17.6 / 20.6 |
| | HML [52] | 47.1 / 49.1 | 36.3 / 38.7 | 41.0 / 43.3 | 26.1 / 27.4 | 20.8 / 22.1 | 23.2 / 24.5 | 17.6 / 21.1 | 14.6 / 17.3 | 16.0 / 19.0 |
| | DeC [53] | 59.2 / 60.6 | 18.3 / 20.3 | 28.0 / 30.4 | 34.6 / 35.9 | 11.8 / 12.3 | 17.6 / 18.3 | 27.7 / 30.8 | 9.0 / 10.4 | 13.6 / 15.5 |
| | IETrans [25] | 54.7 / 56.7 | 30.9 / 33.6 | 39.5 / 42.2 | 32.5 / 33.4 | 16.8 / 17.9 | 22.2 / 23.3 | 26.4 / 30.6 | 12.4 / 14.9 | 16.9 / 20.0 |
| | MEET [21] | 67.4 / 72.7 | 25.3 / 33.5 | 36.8 / **45.9** | 40.5 / 43.2 | 19.0 / 23.7 | **25.9** / **30.6** | 27.9 / 33.3 | 8.5 / 11.8 | 13.0 / 17.4 |
| | CFA [54] | 54.1 / 56.6 | 35.7 / 38.2 | **43.0** / 45.6 | 34.9 / 36.1 | 17.0 / 18.4 | 22.8 / 24.3 | 27.4 / 31.8 | 13.2 / 15.5 | 17.8 / 20.8 |
| | BiC† | 47.4 / 49.5 | 37.4 / 40.2 | 41.8 / 44.4 | 33.0 / 34.4 | 19.0 / 21.0 | 24.1 / 26.1 | 24.3 / 28.3 | 17.2 / 19.9 | **20.1** / **23.3** |
| VCTree [31] | Baseline | 65.4 / 67.2 | 16.7 / 18.2 | 26.6 / 28.6 | 46.7 / 47.6 | 11.8 / 12.5 | 18.8 / 19.8 | 31.9 / 36.2 | 7.4 / 8.7 | 12.0 / 14.0 |
| | TDE [19] | 47.2 / 51.6 | 25.4 / 28.7 | 33.0 / 36.9 | 25.4 / 27.9 | 12.2 / 14.0 | 16.5 / 18.6 | 19.4 / 23.2 | 9.3 / 11.1 | 12.6 / 15.0 |
| | PCL [22] | 53.4 / 56.2 | 32.9 / 35.7 | 40.7 / 43.7 | 38.4 / 39.5 | 25.2 / 26.3 | 30.4 / 31.6 | 27.6 / 31.9 | 14.8 / 17.4 | 19.3 / 22.5 |
| | NICE [24] | 55.0 / 56.9 | 30.7 / 33.0 | 39.4 / 41.8 | 37.8 / 39.0 | 19.9 / 21.3 | 26.1 / 27.6 | 27.0 / 30.8 | 11.9 / 14.1 | 16.5 / 19.3 |
| | GCL [20] | 40.7 / 42.7 | 37.1 / 39.1 | 38.8 / 40.8 | 27.7 / 28.7 | 22.5 / 23.5 | 24.8 / 25.8 | 17.4 / 20.7 | 15.2 / 17.5 | 16.2 / 19.0 |
| | HML [52] | 47.0 / 48.8 | 36.9 / 39.2 | 41.3 / 43.5 | 27.0 / 28.4 | 25.0 / 26.8 | 26.0 / 27.6 | 17.6 / 21.0 | 13.7 / 16.3 | 15.3 / 18.4 |
| | IETrans [25] | 53.0 / 55.0 | 30.3 / 33.9 | 38.6 / 41.9 | 32.9 / 33.8 | 16.5 / 18.1 | 22.0 / 23.6 | 25.4 / 29.3 | 11.5 / 14.0 | 15.8 / 18.9 |
| | MEET [21] | 62.0 / 69.8 | 25.5 / 34.5 | 36.1 / 46.1 | 35.4 / 39.2 | 14.5 / 18.6 | 20.6 / 25.2 | 26.4 / 31.2 | 8.2 / 11.5 | 12.5 / 16.8 |
| | CFA [54] | 54.7 / 57.5 | 34.5 / 37.2 | 42.3 / 45.2 | 42.4 / 43.5 | 19.1 / 20.8 | 26.3 / 28.1 | 27.1 / 31.2 | 13.1 / 15.5 | 17.7 / 20.7 |
| | BiC† | 56.0 / 58.7 | 35.3 / 38.0 | **43.3** / **46.1** | 37.2 / 38.7 | 27.9 / 29.7 | **31.9** / **33.6** | 22.6 / 26.5 | 16.9 / 19.7 | **19.3** / **22.6** |
| Transformer [33] | Baseline† | 65.1 / 66.8 | 16.1 / 17.7 | 25.8 / 28.0 | 38.4 / 39.1 | 9.2 / 10.0 | 14.8 / 15.9 | 31.2 / 35.6 | 7.2 / 8.4 | 11.7 / 13.6 |
| | CogTree [55] | 38.4 / 39.7 | 28.4 / 31.0 | 32.7 / 34.8 | 22.9 / 23.4 | 15.7 / 16.7 | 18.6 / 19.5 | 19.5 / 21.7 | 11.1 / 12.7 | 14.1 / 16.0 |
| | HML [52] | 45.6 / 47.8 | 33.3 / 35.9 | 38.5 / 41.0 | 22.5 / 23.8 | 19.1 / 20.4 | 20.7 / 22.0 | 15.4 / 18.6 | 15.0 / 17.7 | 15.2 / 18.1 |
| | IETrans [25] | 51.8 / 53.8 | 30.8 / 34.7 | 38.6 / 42.2 | 32.6 / 33.5 | 14.7 / 19.1 | 22.7 / 24.3 | 25.5 / 29.6 | 12.5 / 15.0 | 16.8 / 19.9 |
| | CFA [54] | 59.2 / 61.5 | 30.1 / 33.7 | 39.9 / 43.5 | 36.3 / 37.3 | 15.7 / 17.2 | 21.9 / 23.5 | 27.7 / 32.1 | 12.3 / 14.6 | 17.0 / 20.1 |
| | BiC† | 53.4 / 56.0 | 34.6 / 37.2 | **42.0** / **44.7** | 33.0 / 34.0 | 19.7 / 21.0 | **24.7** / **26.0** | 23.3 / 27.3 | 16.7 / 19.1 | **19.5** / **22.5** |
| AMP | Baseline† | 64.6 / 67.0 | 18.4 / 20.1 | 28.6 / 30.9 | 38.4 / 39.3 | 10.8 / 11.6 | 16.9 / 17.9 | 31.7 / 35.6 | 7.6 / 9.3 | 12.3 / 14.7 |
| | BiC† | 51.7 / 54.0 | 38.3 / 40.8 | **44.0** / **46.5** | 30.7 / 32.2 | 21.8 / 23.3 | **25.5** / **27.0** | 22.2 / 26.8 | 16.6 / 19.2 | **18.9** / **22.4** |

Therefore, the model is equivalent to pre-train a model for tail predicates. Although there are continuous interference tags involved as the training goes on, it can be treated as a fine-tuning process and will not cause significant interference to the performance of tail predicates. Under this setting, it simultaneously solves the label confusion and unbiased prediction problem.

With the whole model containing the feature loss and the classification loss, the overall loss function is written as:

$$\begin{aligned}\mathcal{L}^{(n)}(\mathbf{p}_{i\to j}) =& \mu_f^{(n)}(w_l^{(n)}\mathcal{L}_f(\mathbf{p}_{i\to j})) \\ &+ (1-\mu_f^{(n)})(w_l^{(n)}\mathcal{L}_c(\mathbf{p}_{i\to j})).\end{aligned} \tag{19}$$

The feature-level curriculum coefficient $\mu_f^n = 1 - n/n_{max}$ and $\mathcal{L}_c$ is the standard cross entropy loss for multi-class classification.

## IV. EXPERIMENTS

### A. Experimetal Setup

**Dataset.** We evaluate our proposed AMP-BiC on three common-used SGG datasets, *i.e.*, Visual Genome (VG) [51], Open Image (OI) V6 [56], and GQA-LT [42]. For VG, following previous work [1, 31, 57], we adopt the most popular pre-processed VG150 including 108k images, the most frequent 150 object classes, and 50 predicate categories. For OI V6, we follow the same data pre-processing and evaluation protocols utilized in [14–16, 18, 58], including 602 object classes and 30 predicate categories. Furthermore, we also conduct the experiments on a more challenging dataset GQA-LT [42]. GQA-LT has more object classes (1703) and predicate categories (310) with a more extreme long-tailed distribution.

**Tasks and Evaluation Metrics.** The scene graph generation is divided into three sub-tasks with the acquisition of objects: Predicate Classification (PredCls) which takes ground truth object detection as inputs; Scene Graph Classification (SGCls) gives ground truth bounding boxes; and Scene Graph Detection (SGDet) detects the whole scene graph from the scratch. To estimate the performance, we use Recall@K (R@K), mean Recall@K (mR@K), and harmonic Recall (hR@K) [25, 59] as evaluation metrics. The hR@K is defined as the harmonic mean of R@K and mR@K and effectively reflects the overall performance of the model. We also report the mR@K on each long-tail category group followed by [15], including *head* (more than 10k), *body* (0.5k ∼ 10k), and *tail* (less than 0.5k).

**Implementation Details:** For the object detector selection, we utilize the pre-trained Faster R-CNN [43] with ResNeXt-101-FPN [60] and freeze the model parameters during training. Unlike the most commonly used fine-tuning operation in recent works, we do not fine-tune the object categories on our proposed AMP network in SGCls and SGDet tasks. Due to the annotation sparsity of SGG datasets, we set confidence in a low threshold ($\rho = 0.1$) to just remove some "impossible" edges. The scale hyper-parameter $\lambda$ for adaptive neighbor aggregation is set to 0.5. Empirically, the whole message passing network contains 3 stages ($S = 3$), and the number of layers of *Entity-to-Entity* graph K is set to 3. The model is trained with 15 epochs ($n_{max} = 15$) and divided into three stages: Start (epoch 1), Middle (epoch 8), and End (epoch 15). The initial learning rate is $1.0 \times 10^{-3}$ with being decayed by a factor of 10 at the $9^{th}$ epoch and $12^{th}$ epoch. The batch size is set to be 12 for PredCls and SGCls; and 8 for SGDet, respectively. Our

TABLE II
COMPREHENSIVE COMPARISON ON VG150 BETWEEN PURE MESSAGE PASSING NETWORK WITH/WITHOUT OUR PROPOSED FEATURE-ASSISTED TRAINING PARADIGM. † DENOTES RESULTS REPRODUCED WITH OUR CODE. THE BEST IS HIGHLIGHTED IN BOLD.

| **Models** | **PredCls** | | | **SGCls** | | | **SGDet** | | |
|---|---|---|---|---|---|---|---|---|---|
| | R@50 / 100 | mR@50 / 100 | hR@50 / 100 | R@50 / 100 | mR@50 / 100 | hR@50 / 100 | R@50 / 100 | mR@50 / 100 | hR@50 / 100 |
| MSDN [27] | 64.6 / 66.6 | 15.9 / 17.5 | 25.5 / 27.7 | 38.4 / 39.8 | 9.3 / 9.7 | 15.0 / 15.6 | 31.9 / 36.6 | 6.1 / 7.2 | 10.2 / 12.0 |
| VTransE [61] | 65.7 / 67.6 | 14.7 / 15.8 | 24.0 / 25.6 | 38.6 / 39.4 | 8.2 / 8.7 | 13.5 / 14.3 | 29.7 / 34.3 | 5.0 / 6.0 | 8.6 / 10.2 |
| GPS-Net [15] | 65.2 / 67.1 | 15.2 / 16.6 | 24.7 / 26.6 | 37.8 / 39.2 | 8.5 / 9.1 | 13.9 / 14.8 | 31.1 / 35.9 | 6.7 / 8.6 | 11.0 / 13.9 |
| BGNN† [15] | 65.4 / 67.2 | 16.9 / 18.3 | **26.9 / 28.8** | 38.3 / 39.4 | 10.2 / 10.8 | **16.1 / 17.0** | 30.7 / 35.7 | 7.0 / 8.9 | **11.4 / 14.2** |
| Motifs [1] | 66.0 / 67.9 | 14.6 / 15.8 | 23.9 / 25.6 | 39.1 / 39.9 | 8.0 / 8.5 | 13.3 / 14.0 | 32.1 / 36.9 | 5.5 / 6.8 | 9.4 / 11.5 |
| Motifs+FAT | 64.6 / 66.4 | 19.3 / 20.8 | **29.7 / 31.2** | 37.9 / 38.8 | 9.7 / 10.3 | **15.4 / 16.3** | 31.5 / 36.5 | 7.9 / 9.5 | **12.6 / 15.1** |
| VCTree [31] | 65.4 / 67.2 | 16.7 / 18.2 | 26.6 / 28.6 | 46.7 / 47.6 | 11.8 / 12.5 | 18.8 / 19.8 | 31.9 / 36.2 | 7.4 / 8.7 | 12.0 / 14.0 |
| VCTree+FAT | 64.3 / 66.4 | 19.9 / 21.5 | **30.4 / 32.5** | 44.9 / 45.8 | 13.3 / 14.1 | **20.5 / 21.6** | 30.6 / 34.8 | 9.2 / 10.6 | **14.1 / 16.2** |
| Transformer† [33] | 65.1 / 66.8 | 16.1 / 17.7 | 25.8 / 28.0 | 38.4 / 39.1 | 9.2 / 10.0 | 14.8 / 15.9 | 31.2 / 35.6 | 7.2 / 8.4 | 11.7 / 13.6 |
| Transformer+FAT | 64.0 / 65.4 | 19.3 / 20.6 | **29.7 / 31.7** | 38.3 / 39.1 | 10.5 / 11.2 | **16.5 / 17.4** | 30.6 / 35.6 | 8.1 / 9.7 | **12.8 / 15.2** |
| AMP | 64.6 / 67.0 | 18.4 / 20.1 | 28.6 / 30.9 | 38.4 / 39.3 | 10.8 / 11.6 | 16.9 / 17.9 | 31.7 / 35.6 | 7.6 / 9.3 | 12.3 / 14.7 |
| AMP-FAT | 65.1 / 66.6 | 20.8 / 22.6 | **31.5 / 33.7** | 38.0 / 39.1 | 12.3 / 13.2 | **18.6 / 19.7** | 31.2 / 35.3 | 9.9 / 11.1 | **15.0 / 16.9** |

TABLE III
SGDET COMPREHENSIVE COMPARISON ON OPEN IMAGE V6 DATASET.

| | **Models** | mR@50 | R@50 | wmAP | | $score_{wtd}$ |
|---|---|---|---|---|---|---|
| | | | | rel | phr | |
| **Biased** | RelDN [58] | 33.98 | 73.08 | 32.16 | 33.39 | 40.84 |
| | VCTree [31] | 33.91 | 74.08 | 34.16 | 33.11 | 40.21 |
| | Motifs [1] | 32.68 | 71.63 | 29.91 | 31.59 | 38.93 |
| | GPS-Net [14] | 35.26 | 74.81 | 32.85 | 33.98 | 41.69 |
| | HL-Net [16] | - | **76.50** | **35.10** | 34.70 | 43.20 |
| | AMP | **36.57** | 75.08 | 35.03 | **35.94** | **43.40** |
| **Unbiased** | Unbiased [19] | 35.47 | 69.30 | 30.74 | 32.80 | 39.27 |
| | SGTR [32] | 42.61 | 59.91 | **36.98** | **38.73** | 42.28 |
| | BGNN+BLS [15] | 40.45 | 74.98 | 33.51 | 34.15 | 42.06 |
| | PCL [22] | 41.63 | 74.75 | 34.65 | 34.98 | 42.80 |
| | AMP-BiC | **43.97** | **75.65** | 34.97 | 35.87 | **43.39** |

TABLE IV
PREDCLS COMPREHENSIVE COMPARISON ON OPEN IMAGE V6 DATASET.

| **Models** | mR@50 | R@50 | wmAP | | $score_{wtd}$ |
|---|---|---|---|---|---|
| | | | rel | phr | |
| VCTree [31] | 48.41 | 91.36 | 95.13 | 82.04 | 90.55 |
| Motifs [1] | 47.87 | 91.43 | 95.12 | 81.49 | 90.31 |
| Transformer [33] | 46.86 | 91.75 | 95.47 | 82.16 | 90.74 |
| AMP | 48.53 | **91.95** | **95.56** | **82.37** | **90.88** |
| AMP-BiC | **54.69** | 91.35 | 94.39 | 81.20 | 89.91 |

model is trained in an end-to-end manner due to the curriculum learning scheme. All our experiments are conducted using RTX A5000 GPUs.

### *B. Performance Comparisons*

In this section, we perform a comparison between our proposed feature-assisted training and the existing state-of-the-art methods of scene graph generation. We divide the proposed method into three types: 1) pure adaptive message passing network (AMP); 2) only use feature-assisted training paradigm (network-FAT); 3) use bi-level curriculum learning scheme (network-BiC).

**Quantitative analysis on Visual Genome.** In Table I, our proposed AMP-BiC achieves new state-of-the-art performance. Adaptive feature aggregation and training allow our model to effectively balance the performance of head predicates and tail predicates. While achieving SOTA performance on the majority of sub-tasks in terms of mR@K, it also maintains competitive R@K performance.

Table II is a comprehensive comparison between pure message passing networks with/without the feature-assisted training paradigm. Compared with another adaptive-style message passing network, BGNN [15], AMP shows huge improvement with average increments of 7.2% and 5.4% in mR@100 and hR@100. The results indicate that our model can retain the original features, thus alleviating the feature smoothing caused by scene noise and long-tailed distribution, and improving the prediction accuracy of tail predicates. Furthermore, compared with four baseline networks (Motifs [1], VCTree [31], Transformer [33], and our AMP), the performance of hR@100 is further improved by 23.2%, 30.1%, 12.8%, and 11.4% after using feature-assisted training (FAT) paradigm. FAT significantly improves the performance of mR@100 while maintaining the R@100 performance. This improvement indirectly proves that direct training on features can effectively prevent informative predicates from being disturbed by meaningless head predicates under the effect of long-tailed distribution.

Table I shows the superiority of the bi-level curriculum learning method compared with the state-of-the-art unbiased method, like GCL [20], NICE [24], and IETrans [25], under three common-used baseline networks, VCTree [31], Motifs [1], and Transformer [33]. BiC achieves superior comprehensive performance on most tasks. In particular, compared with the SOTA model CFA [54], BiC average outperforms with 0.7%, 12.5%, and 10.7% increment on hR@100 across PredCls, SGCls, and SGDet tasks. The remarkable overall performance improvement demonstrates that our proposed BiC strategy effectively differentiates between head and tail predicates. This ensures a balanced focus for the model and prevents an undue emphasis on either tail or head predicates, which could result in a significant decline in performance for the other component. In addition, we also conduct the BiC

TABLE V

PREDCLS COMPREHENSIVE COMPARISON ON GQA-LT DATASET. † DENOTES RESULTS REPRODUCED WITH OUR CODE.

| **Models** | mR@50 / 100 | R@50 / 100 | Head(16) | Body(46) | Tail(248) |
|---|---|---|---|---|---|
| Motifs† [1] | 2.93 / 4.38 | 48.76 / 55.53 | **36.65** | 11.72 | 0.92 |
| Motifs+BiC† | 10.01 / 11.80 | 36.93 / 42.69 | 29.14 | 28.17 | 7.65 |
| VCTree† [31] | 3.58 / 5.57 | **49.13 / 56.17** | 35.92 | 14.08 | 2.03 |
| VCTree+BiC† | 9.57 / 11.19 | 36.11 / 42.10 | 29.51 | 27.82 | 6.92 |
| Transformer† [33] | 3.23 / 4.69 | 47.14 / 53.86 | 35.52 | 11.60 | 1.42 |
| Transformer+BiC† | 10.19 / 12.25 | 38.20 / 44.10 | 28.78 | 25.00 | 8.82 |
| AMP† | 3.63 / 4.73 | 45.04 / 50.91 | 36.24 | 13.22 | 1.12 |
| AMP+BiC† | **14.16 / 17.38** | 37.70 / 43.69 | 28.22 | **29.39** | **14.45** |

method on our proposed AMP network, which exhibits greater performance enhancements, especially in PredCls with an absolute improvement of 1.4% compared with VCTree-BiC. We attribute the performance improvements to the ability of the AMP network to effectively aggregate scene information, thus providing discriminative representations and powerful classifiers for all predicates. In the three subtasks, the noise in the scene information increases progressively, *i.e.*, noise: SGDet > SGCls > PredCls. For the AMP network, less noise implies better adaptive feature aggregation. Therefore, in terms of performance improvement, PredCls > SGCls > SGDet is consistent with our experimental results.

**Quantitative analysis on Open Image V6.** We further verify the generalizability of our proposed method on OI V6, which has better annotation quality compared with the Visual Genome. Following [15], we use the mean Recall@50 (mR@50), Recall@50 (R@50), weighted mean AP of relationships (wmAPrel), and weighted mean AP of phrase (wmAPphr) as evaluation metrics. Following standard evaluation metrics of Open Images, the weight metric $score_{wtd}$ is computed as: $score_{wtd} = 0.2 \times R@50 + 0.4 \times wmAP_{rel} + 0.4 \times wmAP_{phr}$.

As shown in Tables III and IV, AMP and AMP-BiC both achieve superior performance on mR@50 and $score_{wtd}$. On the SGDet sub-task, compared with biased methods [1, 14, 16, 31, 58], AMP obtains the best mR@50 and $score_{wtd}$ results. When BiC is adopted, mR@50 has a huge improvement, and $score_{wtd}$ still achieves the best performance compared with unbiased methods [15, 19, 22, 32]. On the PredCls subtask, AMP shows the best overall performance compared with the three baseline models, and AMP-BiC shows significant improvement in mR@50. The results further demonstrate that our proposed feature-enhancement method has good generalization, both on message passing and unbiased prediction.

**Quantitative analysis on GQA-LT.** GQA-LT is a more challenging dataset due to its huge number of entity and predicate categories and extreme long-tail distribution. Compared with the VG150, GQA-LT is more fine-grained with dense relationship labeling. Therefore, GQA-LT inevitably has a more serious label confusion problem. For example, “parked alongside"(52), “parked along"(65), “parked at"(84), “parked beside"(44), “parked in"(281), “parked on"(855) are similar predicates. Although ’alongside’ and ’beside’ represent different states of ’parked’, human always prefers a more general expression ’on/in’, disturbing the model to predict more specific results. In addition, almost every image in GQA-LT has nearly a hundred relation annotations, far more than VG150. Therefore, GQA-LT is more suitable for testing the performance of the scene graph generated by the model.

TABLE VI

ABLATION STUDY OF THE MODEL COMPONENTS.

| Module | | | | PredCls | SGCls | SGDet |
|---|---|---|---|---|---|---|
| **ANA** | **ARA** | **FAT** | **BiC** | R/mR/hR@100 | R/mR/hR@100 | R/mR/hR@100 |
| **Baseline** | | | | 67.4 / 17.9 / 28.3 | 39.1 / 9.8 / 15.7 | **36.6** / 7.9 / 13.0 |
| ✓ | | | | **67.5** / 19.4 / 30.1 | 39.2 / 10.7 / 16.8 | 35.8 / 8.8 / 14.1 |
| ✓ | ✓ | | | 67.0 / 20.1 / 30.9 | **39.3** / 11.6 / 17.9 | 35.6 / 9.3 / 14.7 |
| ✓ | ✓ | ✓ | | 66.6 / 22.6 / 33.7 | 39.1 / 13.2 / 19.7 | 35.3 / 11.1 / 16.9 |
| ✓ | ✓ | ✓ | ✓ | 54.0 / **40.8** / **46.5** | 32.2 / **23.3** / **27.0** | 26.8 / **19.2** / **22.4** |

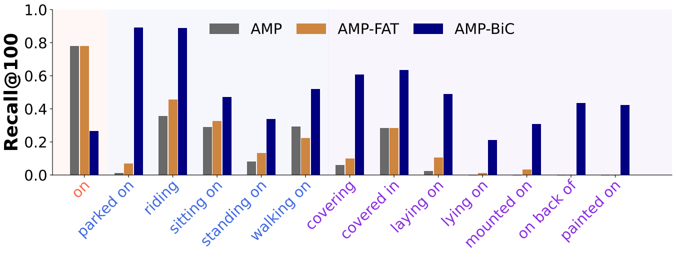


Fig. 3. The performance of several predicate classes on Recall@100. All selected predicates are semantically close to “on”. Although AMP-BiC drops for “on”, the superior recalls for the others show our effectiveness in enhancing representation.

As shown in Table V, after applying the BiC unbiased method, all baseline methods produce a huge improvement in body and tail parts with absolute average boosting of 14.94% and 8.09%, and normally, 7.17% drop on the head. Compared between the three baselines and AMP, we find that AMP achieves equal or even better Recall@100 performance in different long-tailed parts (head-body-tail) but shows poor performance on all Recall. The whole Recall performance depends on the head predicates. Interestingly, AMP has realized superior Recall performance in the head part, which implies that AMP produces poor performance on most frequent predicates but achieves even higher on second-frequent ones. This evidence supports the capability of our feature augmentation-based method in alleviating model bias toward predicting relationships.

## C. Ablation Study

We further conduct a detailed ablation study over components in our network on VG150. *Baseline* uses a pure message passing network without Adaptive Neighbor Aggregation (ANA) and Adaptive Residual Aggregation (ARA). As shown in Table VI, we observe that Adaptive Neighbor Aggregation averagely improves the performance of the classifier by 4.5% on both mean Recall and Recall, and it is further improved by 2.5% with Adaptive Residual Aggregation. We attribute these performance improvements to the well-de-noising capability of ANA and the anti-over-smoothing function of ARA. Feature-assisted training paradigm, which continues to improve by 7.3% average, proves the effectiveness of direct training for features. After further adopting the bi-level curriculum

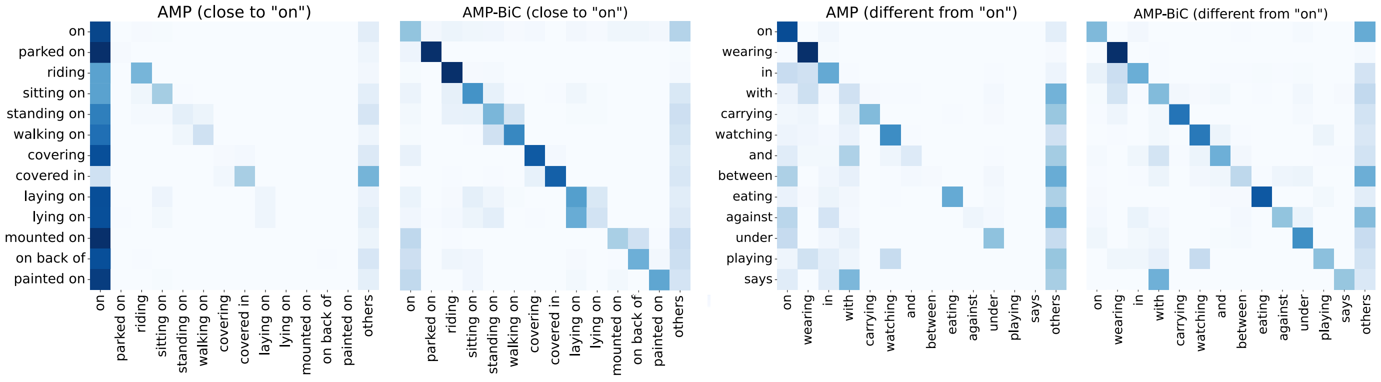


Fig. 4. Confusion matrices for predicted results of AMP and AMP-BiC. The horizontal axis lists predicted predicates and the vertical axis represents ground truth predicates. The selected predicates are semantically close to “on" in the left two matrices and different from “on" in the right two matrices. Comparing the first and third matrices demonstrates the existence of label confusion. Comparing the first and second matrices shows the de-confusion ability of BiC.

TABLE VII
PERFORMANCE ON DIFFERENT SOLUTIONS FOR UNBIASED PREDICTION.

| Method | PredCls | SGCls | SGDet |
|---|---|---|---|
| | mR@50/100 | mR@50/100 | mR@50/100 |
| Re-sample | 30.2 / 34.3 | 17.0 / 19.3 | 13.3 / 15.7 |
| Re-weight | 34.3 / 36.8 | 19.1 / 21.1 | 14.7 / 17.3 |
| Curriculum | 36.8 / 38.4 | 19.5 / 21.9 | 15.3 / 17.9 |
| BiC | **38.3 / 40.8** | **21.8 / 23.3** | **16.5 / 19.2** |

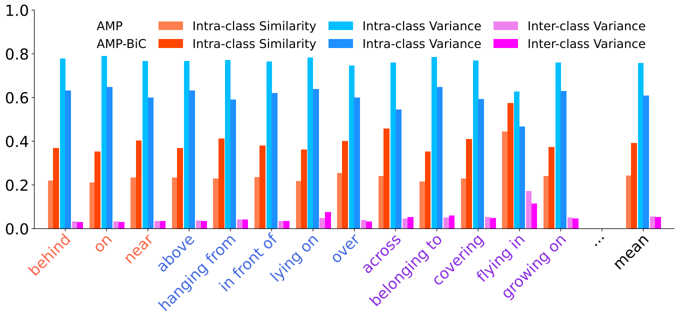


Fig. 5. Comparison of feature distribution between AMP and AMP-BiC. 13 predicates are selected to visualize, including 3 head predicates (orange), 5 body predicates (blue), and 5 tail predicates (purple). While the inter-class variance remains stable, BiC significantly increases intra-class similarity and decreases intra-class variance, resulting in clustered feature distribution.

scheme, which double-emphasizes the feature training of tail predicates, the representations of tail predicates become powerful and manifest directly in the huge improvement of mR@100 with 80.5%. Although bi-level curriculum learning significantly reduces the performance of R@100, we should note that the predicate “on" accounts for nearly 30% samples in the whole dataset, showing its decisive impact on the final Recall performance. Considering that “on" is an uninformative predicate that can be replaced by a more precise description in most cases, high-accuracy of “on" does not make a lot of sense.

In Fig. 3, we show the R@100 performance of AMP, AMP-FAT, and AMP-BiC on predicates that are semantically close to “on". The performance of AMP and AMP-FAT is mostly contributed by “on", but is less satisfactory on other informative predicates. In contrast, AMP-BiC has a huge improvement in almost all other informative predicates, demonstrating a significant effect on capturing informative predicates. For example, in a smart home scene, the machine should have the ability to distinguish between “walking on" and “standing on". Simply classifying both as “on" will make intelligent products confusing. Fig. 4 further compares the confusion matrices of the predicted results of AMP and AMP-BiC, including predicates semantically similar (left two) and different (right two) to “on". Compared with the first and third matrices of AMP, misclassifying semantic-similar predicates to “on" is easier than misclassifying semantic-different ones. This phenomenon proves the existence of the label confusion problem between general (e.g., “on") and specific predicates (semantically similar to “on"). The label confusion further exacerbates the biased prediction under long-tailed distribution. Our proposed de-biased method BiC effectively alleviates this problem. As shown in the second matrix and the fourth matrix in Fig. 4, predicates that are semantically similar and different to “on" have roughly the same probability of being misclassified as “on".

Moreover, we conducted two comparative experiments to demonstrate the effectiveness of bi-level curriculum learning. 1) We compare BiC with two conventional unbiased methods *re-sample*, *re-weight*, and the predicate-level only curriculum scheme (feature-assisted paradigm is not used). We adopt the BLS algorithm proposed by BGNN [15] for *re-sample* and use the weight of each predicate at $n_{max}$ epoch for *re-weight*. As shown in Table VII, the BiC achieves the best performance, which proves the effectiveness of the feature-assisted paradigm. 2) We quantitatively analyze the output features of the two different models (AMP and AMP-BiC). Firstly, we take out all predicted features with their ground truth predicates and conduct $\ell^2$ normalization. Then we calculate the center vector of each predicate category $\phi_l$. We use three metrics to evaluate the feature cluster performance, intra-class similarity $\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{p}_i\|_2 \cdot \phi_l$, intra-class variance $\frac{1}{n-1}\sum_{i=1}^{n}(\|\mathbf{p}_i\|_2 - \phi_l)^2$, and inter-class variance $\frac{1}{C-1}\sum_{j\neq l}(\phi_l - \phi_j)^2$. $C$ is the number of categories. The

TABLE VIII
PERFORMANCE ON DIFFERENT VALUE CHOICE OF $\lambda$.

| $\lambda$ | PredCls | SGCls | SGDet |
|---|---|---|---|
| | R/mR/hR@100 | R/mR/hR@100 | R/mR/hR@100 |
| 0.1 | 53.7 / 40.1 / 45.9 | **32.5** / 22.9 / 26.9 | **28.2** / 18.0 / 22.0 |
| 0.3 | **54.4** / 40.5 / 46.4 | 31.6 / **23.9** / **27.2** | 27.0 / 18.8 / 22.2 |
| 0.5 | 54.0 / 40.8 / **46.5** | 32.2 / 23.3 / 27.0 | 26.8 / 19.2 / **22.4** |
| 0.7 | 53.2 / **40.9** / 46.2 | 31.8 / 23.2 / 26.8 | 26.5 / **19.3** / 22.3 |

TABLE IX
PERFORMANCE ON DIFFERENT VALUE CHOICE OF $\rho$.

| $\rho$ | PredCls | | |
|---|---|---|---|
| | R@50 / 100 | mR@50 / 100 | hR@50 / 100 |
| 0.0 | 64.5 / 66.9 | 18.1 / 19.7 | 28.3 / 30.4 |
| 0.1 | 64.6 / 67.0 | 18.4 / 20.1 | **28.6 / 30.9** |
| 0.5 | 64.2 / 66.6 | 17.7 / 19.6 | 27.7 / 30.2 |
| 0.9 | 64.7 / 67.0 | 18.2 / 20.0 | 28.4 / 30.7 |

TABLE X
PERFORMANCE ON DIFFERENT GRAPH LAYERS (K) AND MESSAGE PASSING STAGES (S).

| $S-K$ | PredCls | | | |
|---|---|---|---|---|
| | R/mR/hR@100 | Head(7) | Body(21) | Tail(22) |
| 1 - 1 | 53.2 / 39.1 / 45.1 | 51.5 | 35.6 | 38.5 |
| 2 - 1 | **55.1** / 39.6 / 46.1 | 54.8 | 33.9 | 40.2 |
| 3 - 1 | 53.5 / 40.6 / 46.2 | 52.6 | 38.8 | 38.5 |
| 3 - 2 | 54.6 / 40.3 / 46.4 | 54.1 | 37.2 | 38.9 |
| 3 - 3 | 54.0 / **40.8** / **46.5** | **56.3** | 30.8 | **45.3** |
| 4 - 3 | 54.5 / 40.4 / 46.4 | 55.8 | **40.2** | 35.7 |

TABLE XI
ANALYSIS ON PROTOTYPE GENERATION METHODS.

| Prototype | PredCls | SGCls | SGDet |
|---|---|---|---|
| | R/mR/hR@100 | R/mR/hR@100 | R/mR/hR@100 |
| GloVe | 54.0 / **40.8** / **46.5** | 32.2 / **23.3** / **27.0** | **26.8** / **19.2** / **22.4** |
| Trainable | **56.1** / 38.9 / 45.9 | **34.5** / 22.2 / 27.0 | 25.4 / 18.9 / 21.7 |

results in Fig. 5 show that although the inter-class variance nearly does not change, the intra-class similarity significantly increases while the intra-class variance reduces. This means in a limited space, the distribution of the same predicate features is more clustered, thus helping the classification.

### *D. Model Analysis*

Variants of our proposed method were investigated for more insights on VG150: the value of $\lambda$ for the proportion of self-feature; the number of graph layers in AMP; selection of the prototypes; and the effectiveness of bi-level curriculum scheme for solving label confusion problem.

**Value of $\lambda$.** We experiment with the value of $\lambda$. A high value means the representation of the node comes more from itself and vice versa. Under the adaptive aggregation mechanism, the value of $\lambda$ should not have a significant impact on the performance. The experimental result shown in Table VIII confirms our conjecture, and we find 0.5 is a better choice.

**Value of $\rho$.** We also experiment with the value of $\rho$. A high value means masking more aggregated nodes by using spatial and visual information from object-pair. The results shown in Table IX indicate that the influence of varying $\rho$ values on performance is relatively small. This could be due to the inherent capability of the attention mechanism to perform adaptive learning. The role of $\rho$ is merely to further constrain the scope of attention computation by introducing spatial and visual information from object-pair.

**Number of graph layers.** We further explore the effect of the stages of message passing network ($S$) and the layers of *Entity-to-Entity* graph ($K$) on performance. As shown in Table X, we find that the number of graph layers has less impact on performance. This is mainly because the AMP is an anti-over-smoothing method, and the number of layers will not cause performance degradation. In addition, the BiC training strategy further ensures the distinction between predicates, and features will not become similar due to the increase in the number of layers. 3 stages of message passing network and 3 layers of *Entity-to-Entity* graph achieve better performance.

**Selection of the prototypes.** We evaluate two ways to select prototypes. The first way is in consideration of the good generality and expressivity of the pre-trained model. Directly obtaining the prototype from GloVe [50] is a simple and effective way. The dimension of a prototype is set as 300. For predicates holding multiple words, the summarization is conducted. The second way is with the consideration of updating prototypes along with training. Since SGG is a specific task, general word vectors may not be suitable as prototypes. During the training process, weighted updating with features obtained in the current batch is provided as $\mathbf{q}_l^{next} = \frac{\mathbf{q}_l \times COUNT(l) + \mathbf{M}_f \tilde{\mathbf{p}}_l}{COUNT(l)+1}$. $\mathbf{q}_l$ denotes the current prototype for predicate $l$, and $\tilde{\mathbf{p}}_l$ denotes the predicate feature output by the SGG model. $COUNT(l)$ means the number of samples for a specific predicate that have appeared before the current mini-batch. $\mathbf{M}_f$ is a projection matrix mapping from $\mathbb{R}^{d_p}$ to $\mathbb{R}^{300}$. The comparison results are shown in Table XI. We find that the first way has advantages over the second one in both convenience and performance. We attribute this phenomenon to unstable training. Since some tail predicates have few samples, predicate features are easily affected by interference labels. Consequently, updating prototypes during training brings in a lot of noise and confusion.

**Effectiveness of bi-level curriculum scheme.** We discuss how BiC handles the label confusion problem based on mR@100 performance of *head*, *body*, and *tail* predicate categories across the Start, Middle, and End stages during training. As shown in Table XII, the performance of the *head* predicates increases with the dynamic change of bi-level weights during training. But the *body* and *tail* parts have maintained good performance and even further improved, especially on *tail* part. The results well proved that the proposed bi-level curriculum learning is more capable of obtaining separated feature distribution for less dominant predicate categories when trained first. Even though the weight of the head predicates gradually increases, it does not hurt the performance of the tail parts. In this way, the performance of tail classes will be mainly affected by the richness of training samples rather than category imbalance

TABLE XII
MR@100 RESULTS OF *head*, *body*, AND *tail* PREDICATES AT DIFFERENT EPOCHS DURING TRAINING.

| **Training Stage** | **PredCls** | | | | | **SGCls** | | | | | **SGDet** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Head | Body | Tail | mR@100 | R@100 | Head | Body | Tail | mR@100 | R@100 | Head | Body | Tail | mR@100 | R@100 |
| Start (1$^{st}$ epoch) | 5.7 | **41.2** | 45.2 | 38.0 | 10.9 | 10.0 | 19.9 | 18.4 | 12.2 | 17.8 | 2.4 | 17.7 | 16.8 | 6.4 | 15.2 |
| Middle (8$^{th}$ epoch) | 53.7 | 29.8 | 37.4 | 36.5 | 52.5 | 28.8 | **19.9** | 20.9 | 21.6 | 28.3 | 15.9 | **21.6** | 17.4 | 15.9 | 19.0 |
| End (15$^{th}$ epoch) | **56.3** | 30.8 | **45.3** | **40.8** | **54.0** | **33.8** | 17.8 | **25.4** | **23.3** | **32.2** | **27.3** | 16.6 | **19.0** | **19.2** | **26.8** |

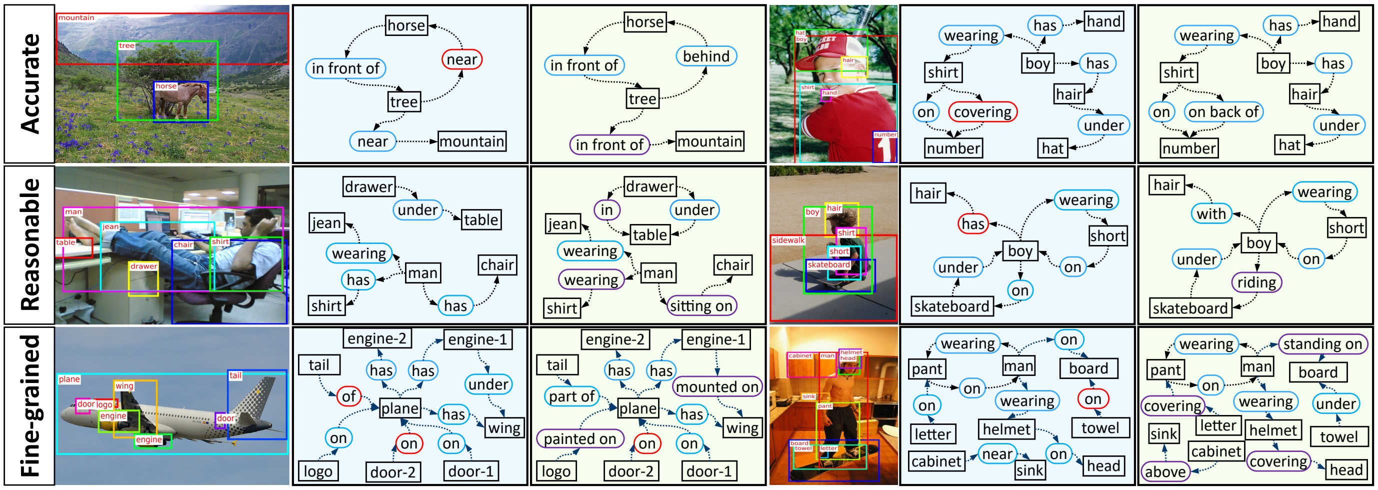


Fig. 6. Visualization results of scene graphs generated by Baseline (blue background) and AMP-BiC (green background) on the PredCls task. The quality of predicted predicates is marked in three levels: red (false), blue (correct), and purple (better).

TABLE XIII
COMPARISON OF MODEL PARAMETERS, TRAINING TIME, AND INFERENCE SPEED ON SGDET SUB-TASK.

| **Models** | parameters (M) | training time (batch / s) | inference speed (image / s) |
|---|---|---|---|
| Motifs [1] | 368.404 | 1.176 | 0.369 |
| VCTree [31] | 431.039 | 5.863 | 0.595 |
| Transformer [33] | 331.899 | 0.997 | 0.322 |
| AMP | 325.846 | 1.458 | 0.355 |

and label confusion problems.

**Analysis on computational complexity.** The AMP network mainly contains two components: ANA and ARA. ANA is essentially a multi-head self-attention architecture with a mask condition (constrained by a confidence estimator) to calculate the adjacency matrix $\hat{\mathbf{A}}$. The computational complexity of ANA is $O(n^2dh + d^2)$, where $n, d, h$ represent the number of instances, feature dimension, and the number of attention heads respectively. ARA is a fusion process among initial, neighbor, and self features. For relationship feature aggregation, each relationship requires $O(d^2)$ computations. Given that there are $O(n^2)$ relationships in total, the overall computational complexity is $O(n^2d^2)$. For instance feature aggregation, the computational complexity is $O(n^2d)$. In conclusion, the overall computational complexity of the model is $O(n^2d^2)$. In addition, we show some quantitative results in Table XIII, regarding the model's parameter size, training time, and inference speed. Our model demonstrates good computational efficiency.

### E. Qualitative Analysis

We visualize several PredCls results in Fig. 6, which show that our approach relieves the label confusion problem and predicts semantically informative predicates rather than vaguely biased ones. Compared with the baseline, our proposed AMP-BiC obtains accurate, reasonable, and fine-grained results. As shown in the top row of Fig. 6, we correctly predict the predicate *"on back of"* rather than *"covering"*, *"behind"* rather than *near*, which is attributed to the fact that they are distant from each other in feature space even if semantically similar. In the second row, the results show that AMP-BiC makes a more reasonable prediction of <*boy, riding, skateboard*> rather than <*boy, on, skateboard*>, <*man, sitting on, chair*> rather than *man, on, chair*>. This reasonability mainly derives from the fact that the bi-level curriculum scheme alleviates the label confusion problem. In the bottom row, the results prove that AMP-BiC is more capable of predicting fine-grained scene graphs in complex scenes. Vague head predicates are usually not predicted by the model when more precise descriptors are available.

## V. Conclusion

In this paper, we emphasize the importance of feature training as better features make better classifiers. We provide an effective and general solution from the feature learning view for SGG. We first propose a novel adaptive message passing network to adaptively aggregate neighbor features and initial features. Then, we design a feature-assisted training paradigm to directly learn more discriminative features. To adapt this paradigm to the environment of long-tailed class distribution, we further design a bi-level curriculum learning scheme, which effectively solves the label confusion and unbiased prediction simultaneously. The results suggest that our proposed method contributes to the enhancement of the comprehensive performance in scene graph generation.

## References

[1] 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.

[2] D. Teney, L. Liu, and A. van Den Hengel, "Graph-structured representations for visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1–9.

[3] Z. Zhu, J. Yu, Y. Wang, Y. Sun, Y. Hu, and Q. Wu, "Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visual question answering," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020, pp. 1097–1103.

[4] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3668–3678.

[5] S. Wang, R. Wang, Z. Yao, S. Shan, and X. Chen, "Cross-modal scene graph matching for relationship-aware image-text retrieval," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 1508–1517.

[6] F. Liu, X. Deng, C. Zou, Y.-K. Lai, K. Chen, R. Zuo, C. Ma, Y.-J. Liu, and H. Wang, "Scenesketcher-v2: Fine-grained scene-level sketch-based image retrieval using adaptive gcns," *IEEE Transactions on Image Processing*, vol. 31, pp. 3737–3751, 2022.

[7] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 685–10 694.

[8] X. Li and S. Jiang, "Know more say less: Image captioning based on scene graphs," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2117–2130, 2019.

[9] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang, "Unpaired image captioning via scene graph alignments," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 323–10 332.

[10] X. Hua, X. Wang, T. Rui, F. Shao, and D. Wang, "Adversarial reinforcement learning with object-scene relational graph for video captioning," *IEEE Transactions on Image Processing*, vol. 31, pp. 2004–2016, 2022.

[11] 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 Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 936–13 945.

[12] M. Xu, M. Qu, B. Ni, and J. Tang, "Joint modeling of visual objects and relations for scene graph generation," *Annual Conference on Neural Information Processing Systems*, pp. 7689–7702, 2021.

[13] Y. Lu, H. Rai, J. Chang, B. Knyazev, G. Yu, S. Shekhar, G. W. Taylor, and M. Volkovs, "Context-aware scene graph generation with seq2seq transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 931–15 941.

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

[15] 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 Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 109–11 119.

[16] X. Lin, C. Ding, Y. Zhan, Z. Li, and D. Tao, "Hl-net: Heterophily learning network for scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 476–19 485.

[17] J. Chen, A. Agarwal, S. Abdelkarim, D. Zhu, and M. Elhoseiny, "Reltransformer: A transformer-based long-tail visual relationship recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 507–19 517.

[18] X. Lin, C. Ding, J. Zhang, Y. Zhan, and D. Tao, "Ru-net: Regularized unrolling network for scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 457–19 466.

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

[20] X. Dong, T. Gan, X. Song, J. Wu, Y. Cheng, and L. Nie, "Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 427–19 436.

[21] G. Sudhakaran, D. S. Dhami, K. Kersting, and S. Roth, "Vision relation transformer for unbiased scene graph generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 882–21 893.

[22] L. Tao, L. Mi, N. Li, X. Cheng, Y. Hu, and Z. Chen, "Predicate correlation learning for scene graph generation," *IEEE Transactions on Image Processing*, vol. 31, pp. 4173–4185, 2022.

[23] C. Chen, Y. Zhan, B. Yu, L. Liu, Y. Luo, and B. Du, "Resistance training using prior bias: toward unbiased scene graph generation," *Thirty-Sixth AAAI Conference on Artificial Intelligence*, pp. 212–220, 2022.

[24] L. Li, L. Chen, Y. Huang, Z. Zhang, S. Zhang, and J. Xiao, "The devil is in the labels: Noisy label correction for robust scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 869–18 878.

[25] A. Zhang, Y. Yao, Q. Chen, W. Ji, Z. Liu, M. Sun, and T.-S. Chua, "Fine-grained scene graph generation with data transfer," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 409–424.

[26] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1–26, 2023.

[27] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1261–1270.

[28] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: an efficient subgraph-based framework for scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 335–351.

[29] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3957–3966.

[30] W. Cong, W. Wang, and W.-C. Lee, "Scene graph generation via conditional random fields," *arXiv preprint arXiv:1811.08075*, pp. 1–10, 2018.

[31] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6619–6628.

[32] R. Li, S. Zhang, and X. He, "Sgtr: End-to-end scene graph generation with transformer," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19 464–19 474, 2022.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Annual Conference on Neural Information Processing Systems*, pp. 5998–6008, 2017.

[34] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *the Eighth International Conference on Learning Representations*, 2020, pp. 1–37.

[35] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," *The Seventh International Conference on Learning Representations*, pp. 1–15, 2019.

[36] T. Chen, W. Yu, R. Chen, and L. Lin, "Knowledge-embedded routing network for scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6163–6171.

[37] Y. Guo, L. Gao, X. Wang, Y. Hu, X. Xu, X. Lu, H. T. Shen, and J. Song, "From general to specific: Informative scene graph generation via balance adjustment," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 16 383–16 392.

[38] A. Desai, T.-Y. Wu, S. Tripathi, and N. Vasconcelos, "Learning of visual relations: The devil is in the tails," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 404–15 413.

[39] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging knowledge graphs

to generate scene graphs," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 606–623.

[40] 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.

[41] A. Goel, B. Fernando, F. Keller, and H. Bilen, "Not all relations are equal: Mining informative labels for scene graph generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 596–15 606.

[42] S. Abdelkarim, A. Agarwal, P. Achlioptas, J. Chen, J. Huang, B. Li, K. Church, and M. Elhoseiny, "Exploring long tail visual relationship recognition with large vocabulary," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 15 921–15 930.

[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Annual Conference on Neural Information Processing Systems*, pp. 91–99, 2015.

[44] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1202–1211.

[45] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[46] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the International Conference on Machine Learning*, 2010, pp. 663–670.

[47] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[48] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine learning*, vol. 73, no. 3, pp. 243–272, 2008.

[49] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l2, 1-norm minimization," *arXiv preprint arXiv:1205.2631*, pp. 1–10, 2012.

[50] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[51] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[52] Y. Deng, Y. Li, Y. Zhang, X. Xiang, J. Wang, J. Chen, and J. Ma, "Hierarchical memory learning for fine-grained scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 266–283.

[53] T. He, L. Gao, J. Song, and Y.-F. Li, "State-aware compositional learning toward unbiased training for scene graph generation," *IEEE Transactions on Image Processing*, vol. 32, pp. 43–56, 2022.

[54] L. Li, G. Chen, J. Xiao, Y. Yang, C. Wang, and L. Chen, "Compositional feature augmentation for unbiased scene graph generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 685–21 695.

[55] J. Yu, Y. Chai, Y. Wang, Y. Hu, and Q. Wu, "Cogtree: Cognition tree loss for unbiased scene graph generation," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2020, pp. 1274–1280.

[56] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[57] 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.

[58] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 535–11 543.

[59] S. Khandelwal and L. Sigal, "Iterative scene graph generation," in *Annual Conference on Neural Information Processing Systems*, 2022, pp. 1–14.

[60] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[61] 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.