SGTR: END-TO-END SCENE GRAPH GENERATION WITH TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Scene Graph Generation (SGG) remains a challenging visual understanding task due to its complex compositional property. Most previous works adopt a bottom-up two-stage or a point-based one-stage approach, which often suffers from overhead time complexity or sub-optimal design assumption. In this work, we propose a novel SGG method to address the aforementioned issues, which formulates the task as a bipartite graph construction problem. To solve the problem, we develop a transformer-based end-to-end framework that first generates the entity and predicate proposal set, followed by inferring directed edges to form the relation triplets. In particular, we develop a new entity-aware predicate representation based on a structural predicate generator to leverage the compositional property of relationships. Moreover, we design a graph assembling module to infer the connectivity of the bipartite scene graph based on our entity-aware structure, enabling us to generate the scene graph in an end-to-end manner. Extensive experimental results show that our design is able to achieve the state-of-the-art or comparable performance on two challenging benchmarks, surpassing most of the existing approaches and enjoying higher efficiency in inference. We hope our model can serve as a strong baseline for the Transformer-based scene graph generation.

1 INTRODUCTION

Inferring structural properties of a scene, such as the relationship between entities, is a fundamental visual understanding task. The visual relationship between two entities can be formally represented by a triple <*subject entity, predicate, object entity*>. Based on these visual relationships, a scene can be modeled in a form of graph structure, with entities as nodes and predicates as edges, termed scene graph. Such a graph provides a compact structural representation for a visual scene, which has potential applications in many vision tasks such as visual question answering Teney et al. (2017); Shi et al. (2019); Hildebrandt et al. (2020), image captioning Yang et al. (2019; 2021b) and image retrieval Johnson et al. (2015).

Different from the traditional vision tasks (*e.g.,* object detection), which focus on detecting individual instances, the main challenge of scene graph generation (SGG) lies in building an effective and efficient model for the pair-wise relations between the entities. The compositional property of visual relationships induces cubic complexity in terms of their constituents, which makes it difficult to learn a compact representation of the relationship concept for localization and/or classification.

Most previous work attempt to tackle this problem using two distinct design patterns: *bottom-up two-stage* Li et al. (2021); Yang et al. (2021a); Yao et al. (2021a); Desai et al. (2021); Chiou et al. (2021); Guo et al. (2021); Knyazev et al. (2021); Abdelkarim et al. (2021) and *point-based one-stage design* Liu et al. (2021); Dong et al. (2021). The former design typically first detects N entity proposals, followed by predicting the predicate categories of those entity combinations. While this strategy achieves high recalls in discovering relationship instances, its $\mathcal{O}(N^2)$ predicate proposals not only incur considerable computation cost but also produce substantial noise in context modeling. For the one-stage methods, entities and predicates are often extracted separately from the image in order to reduce the size of relationship proposal set. Nonetheless, they rely on a strong assumption

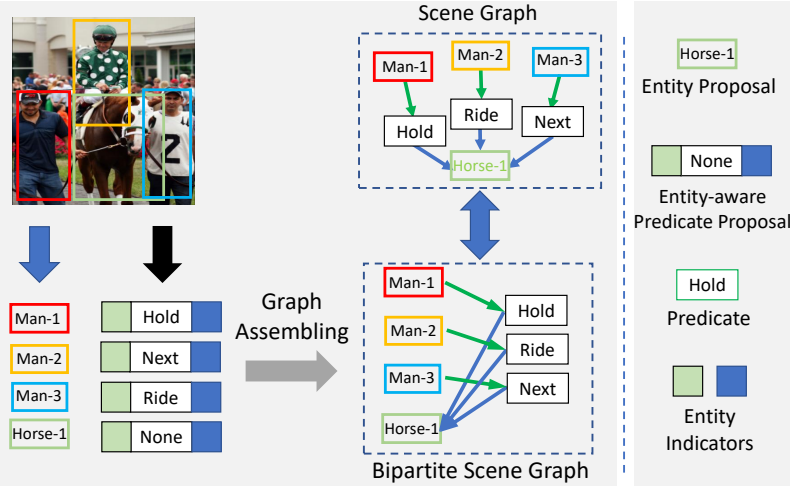

Figure 1: **The illustration of SGTR pipeline paradigm.** We formulate SGG as a bipartite graph construction process. First, the entity and predicate nodes are generated respectively. Then we assemble the bipartite scene graph from two types of nodes.

on the non-overlapping property of interaction regions, which severely restricts their application in modeling complex scenes[1].

In this work, we aim to tackle the aforementioned limitation by leveraging the compositional property of scene graphs. To this end, as illustrated in Fig. 1, we first formulate the SGG task as a bipartite graph construction problem, in which each relationship triplet is represented as two types of nodes (entity and predicate) linked by directed edges. Such a bipartite graph allows us to jointly generate entity/predicate proposals and their potential associations, yielding a rich hypothesis space for inferring visual relations. More importantly, we propose a novel entity-aware predicate representation that incorporates relating entity proposal information into each predicate node. This enriches the expressive power of predicates and therefore enables us to produce a relatively small number of high-quality predicate proposals. Moreover, such a representation encodes potential associations between each predicate and its subject/object entities, which can facilitate predicting the graph edges and leads to efficient generation of the visual relation triplets.

Specifically, we develop a new transformer-based end-to-end SGG model, dubbed Scene graph Generation TRansformer (SGTR), for constructing the bipartite graph. Our model consists of three main modules, including an *entity node generator*, a *predicate node generator* and a *graph assembling module*. Given an image, we first introduce two CNN+Transformer sub-networks as the entity and predicate generator to produce a set of entity and predicate nodes, respectively. In order to compute the entity-aware predicate representations, we design a structural predicate generator consisting of two parallel transformer decoders, which integrate each predicate feature with an entity indicator representation. After generating entity and predicate node representations, we then devise a differentiable *graph assembling* module to infer the directed edges of the bipartite graph, which exploits the entity indicator to predict the best grouping of the entity and predicate nodes.

We validate our method by extensive experiments on two SGG benchmarks: Visual Genome and OpenImages-V6 datasets. We report both prediction accuracy and efficiency with comparisons to the previous state-of-the-art methods. The results show that our method outperforms or achieves comparable performance on both benchmarks with high efficiency during inference. We hope this work can serve as a strong baseline for the transformer-based scene graph generation.

The main contribution of our work has three-folds:

- We propose a novel transformer-based end-to-end scene graph generation method with bipartite graph construction process, which inherits the advantages of both two-stage and one-stage methods.

[1] *e.g.,* two different relationships cannot have largely overlapped area – a phenomenon also discussed in the recent works on (HOI) Chen et al. (2021); Tamura et al. (2021)

- We develop the entity-aware structure for incorporating compositional property of visual relationship.
- Our method achieves the state-of-the-art or comparable performance on all metrics w.r.t the prior SGG methods but with fast inference.

2 RELATED WORKS

We categorize the related work of SGG/HOI into three directions: *Two-stage Scene Graph Generation*, *One-stage Scene Graph Generation* and *One-stage Human-Object Interaction*.

Two-stage Scene Graph Generation Two-stage SGG methods predict the relationships on densely connected entities pairs. Some works propose contextual modeling structure on dense relationship proposals Zellers et al. (2018); Xu et al. (2017); Li et al. (2017); Woo et al. (2018); Tang et al. (2019); Li et al. (2018); Yang et al. (2018); Qi et al. (2019); Yin et al. (2018); Wang et al. (2019); Lin et al. (2020); Zareian et al. (2020b;a;c); Yuren et al. (2020); Wang et al. (2020b); Khandelwal et al. (2021); Li et al. (2021). More recent works mainly address the long-tail recognition challenge in the SGG task by developing logits adjustment and training strategies Tang et al. (2020); Knyazev et al. (2017); Yan et al. (2020); Wang et al. (2020b); Suhail et al. (2021); Li et al. (2021); Yang et al. (2021a); Yao et al. (2021a); Desai et al. (2021); Chiou et al. (2021); Guo et al. (2021); Knyazev et al. (2021); Abdelkarim et al. (2021). These two-stage designs are capable of dealing with the complicated scenario encountered in SGG. However, as mentioned earlier in Sec. 1, the overhead relation proposal leads to large time complexity and unavoidable noise in context modeling. There are many two-stage-based works that propose heuristic designs to address these issues (*e.g.* proposal generation Yang et al. (2018), efficient context modeling Li et al. (2018); Tang et al. (2019); Qi et al. (2019); Yang et al. (2019); Wang et al. (2019); Li et al. (2021)). However, those complex two-stage approaches limit the end-to-end optimization and further development.

One-stage Scene Graph Generation Inspired by the fully convolutional one-stage object detection methods Tian et al. (2019); Carion et al. (2020); Sun et al. (2021), the SGG community starts to explore the one-stage design. These one-stage methods detect the relationship from image feature directly by the fully convolutional network Liu et al. (2021); Teng & Wang (2021) or CNN-Transformer Dong et al. (2021) architectures. The sparse proposal set allows these one-stage frameworks to perform efficiently. Neverless, with less instance-aware structure, those designs may have difficulty modeling the more complex compositing situations of visual relationships. Additionally, the node-edge consistency is disregarded in the majority of one-stage methods, since each triplet is predicted separately.

One-stege Human-Object Interaction The HOI is the similar sub-task of SGG. Recently, there is a trend of study on the one-stage framework for Human-Object Interaction Liao et al. (2020); Kim et al. (2020); Wang et al. (2020a); Zou et al. (2021); Chen et al. (2021); Tamura et al. (2021); Kim et al. (2021); Zhang et al. (2021). The Chen et al. (2021); Kim et al. (2021) introduce the interesting framework with a dual decoder structure that simultaneously extracts the human, object, and interaction, and then groups the components to produce final triplets. This decoding-grouping design provides a more suitable insight for modeling the compositional structure of interaction with the various combination, especially on one-stage scene graph generation, which has more complex composition characteristics. Inspired by their explorations, SGTR reformulates the scene graph generation as a bipartite graph construction. We further propose an entity-aware structure to explicitly model the association between entity and predicate, which allows us to achieve better results on SGG with an efficient pipeline.

3 PRELIMINARY

In the following of this section, we first introduce the problem setting of scene graph generation in Sec. 3.1. Then an overview of our approach is presented in Sec. 3.2

3.1 PROBLEM SETTING

The task of scene graph generation aims to parse an input into a scene graph $\mathcal{G}_{scene} = \{\mathcal{V}_e, \mathcal{E}_r\}$, where $\mathcal{V}_e$ is the node set denoting noun entities, and $\mathcal{E}_r$ is the edge set that represents predicates between pairs of subject and object entities. Specifically, each entity $v_i \in \mathcal{V}_e$ has a category label

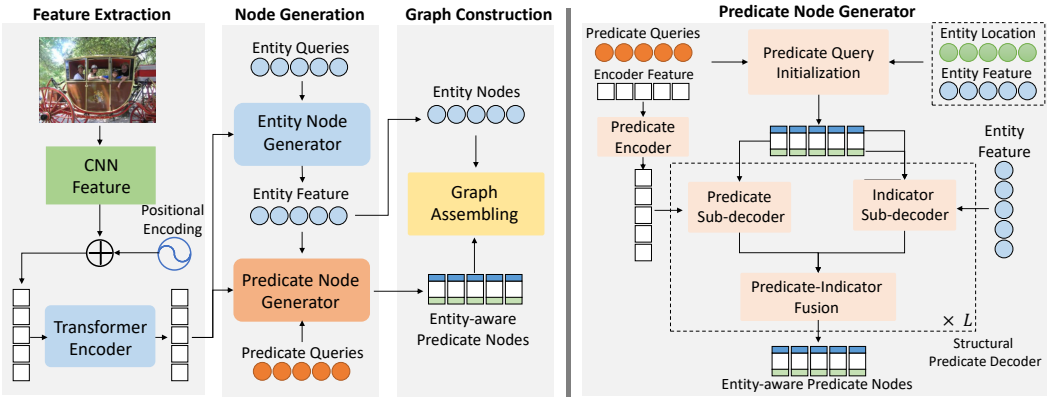

Figure 2: **Illustration of overall pipeline of our SGTR model. Left)** We use CNN backbone together with transformer encoder for image feature extraction. The entity/predicate node generator are introduced to produce the entity node and entity-aware predicate node. A graph assembling mechanism is developed to construct the final bipartite scene graph. **Right)** The predicate node generator consists of *a)* predicate query initialization, *b)* predicate encoder, and *c)* structural predicate decoder, which is designed to generate entity-aware predicate nodes.

from a set of entity classes $\mathcal{C}_e$ and a bounding box depicting its location in the image, while each edge $e_{i\to j} \in \mathcal{E}_r$ between a pair of nodes v_i and v_j is associated with a predicate label from a set of predicate classes $\mathcal{C}_p$ in this task.

One feasible way to generate the scene graph $\mathcal{G}_{scene}$ is extracting the relationship triplet set of the given image. We formulate the relationship triplet generation process as a bipartite graph construction taskLi et al. (2021). Specifically, the graph consists of two groups of nodes $\mathcal{V}_e, \mathcal{V}_p$, which correspond to entity representation and predicate representation respectively. These two groups of nodes are connected by two sets of directed edges $\mathcal{E}_{e\to p}, \mathcal{E}_{p\to e}$ representing the direction from the entities to predicates and vice versa. Hence the bipartite graph has a form as $\mathcal{G}_b = \{\mathcal{V}_e, \mathcal{V}_p, \mathcal{E}_{e\to p}, \mathcal{E}_{p\to e}\}$.

3.2 Model Overview

Our model defines a differentiable function $\mathcal{F}_{sgg}$ that takes an image $\boldsymbol{I}$ as the input and outputs the bipartite graph $\mathcal{G}_b$, denoted as $\mathcal{G}_b = \mathcal{F}_{sgg}(\boldsymbol{I})$, which allows end-to-end training. We propose to explicitly model the bipartite graph construction process by leveraging the compositional property of relationships. The bipartite graph construction is composed by two steps: *a) node (entity and predicate) generation*, and *b) directed edge connection*.

In *node generation* step, we extract the entity nodes and predicate nodes from the image with *entity node generator* and *predicate node generator* respectively. Specifically, the predicate node generator utilizes three parallel sub-decoders to update the predicate proposals. In *directed edge connection* step, we design the *graph assembling module* to assemble the bipartite scene graph from the entity and predicate proposals. An overview of our method is illustrated in Fig 2 and we will start with a detailed description of our model architecture below.

4 Model Architecture

Our model has a modular architecture consisting of four main submodules: (1) a **backbone network** to generate feature representation of the scene (Sec. 4.1); (2) a transformer-based **entity node generator** to predict entity proposals (Sec. 4.1), (3) a structural **predicate node generator** to decode predicate nodes (Sec. 4.2), (4) a differentiable **bipartite graph assembling** module to contruct the final bipartite graph by connecting the entity node and entity-aware predicate node(Sec. 4.3). The learning and inference are detailed in Sec. 4.4.

4.1 BACKBONE AND ENTITY NODE GENERATOR

We adopt an CNN backbone consists of a ResNet network, which produces a convolutional feature representation for the subsequent modules. Motivated by the Transformer-based detector, DETR Carion et al. (2020), we introduce a multi-layer Transformer encoder for entity node generator and predicate node generator. The output feature of the Transformer encoder is denoted as $\boldsymbol{Z} \in \mathbb{R}^{w \times h \times d}$, where w, h, d are the width, height, and channel of the feature map, respectively.

We adopt the decoder of DETR as the entity generator to decode the entity nodes from the learnable entity queries. Formally, we define the entity detector as a mapping function $\mathcal{F}_e$ from initial entity query $\boldsymbol{Q}_e \in \mathbb{R}^{N_e \times d}$ and the feature map $\boldsymbol{Z}$ to entity predicted localization $\boldsymbol{B}_e \in \mathbb{R}^{N_e \times 4}$ and class scores $\boldsymbol{P}_e \in \mathbb{R}^{N_e \times (\mathcal{C}_e+1)}$, along with its associated feature representation $\boldsymbol{H}_e \in \mathbb{R}^{N_e \times d}$ as follows,

$$\boldsymbol{B}_e, \boldsymbol{P}_e, \boldsymbol{H}_e = \mathcal{F}^d_{dec}(\boldsymbol{Z}, \boldsymbol{Q}_e) \tag{1}$$

where $\boldsymbol{B}_e = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{N_e}\}, \boldsymbol{b} = (x_c, y_c, w_b, h_b)$, x_c, y_c are the normalized center coordinates of the instance, w_b, h_b are the normalized width and height of each entity box.

4.2 PREDICATE NODE GENERATOR

Our predicate node generator aims to decode the entity-aware predicate representation by incorporating relating entity proposal information into each predicate node. This design enables us to encodes potential associations between each predicate and its subject/object entities, which can facilitate predicting the graph edges and leads to efficient generation of the visual relation triplets. As shown in Fig. 3, the predicate node generator is composed of three components: (1) a **predicate query initialization** module for initializing the entity-aware predicate query (in Sec. 4.2.2), (2) a **predicate encoder** for image feature extraction (in Sec. 4.2.1), and (3) a **structural predicate decoder** for decoding a set of entity-aware predicate nodes. (in Sec. 4.2.3).

4.2.1 PREDICATE NODE ENCODER

Based on the feature provided by the shared encoder, we introduce a lightweight extra predicate encoder to extract predicate-specific image features. Using a similar structure as the shared encoder, the predicate encoder has a form of multi-layer multi-head self-attention with the skip-connected feed-forward network. The resulting predicate-specific feature is denoted as $\boldsymbol{Z}^p \in \mathbb{R}^{w \times h \times d}$.

4.2.2 PREDICATE QUERY INITIALIZATION

A simple strategy for initializing the predicate proposal is adopting a fixed set of holistic learnable queries as same in the entity detector. However, such a holistic predicate query design ignores not only *the compositional property* of the visual relationships but also *entity candidate information*. The resulting representations are often less expressive for capturing the structured and diverse visual relationship.

To cope with this challenge, we develop a compositional representation for learning the entity-aware predicate nodes by decoupling the predicate query $\boldsymbol{Q}^e_p = \{\boldsymbol{Q}_{is}; \boldsymbol{Q}_{io}; \boldsymbol{Q}_p\} \in \mathbb{R}^{N_r \times 3d}$. It has three parts: *subject/object entity indicator* [2] $\boldsymbol{Q}_{is}, \boldsymbol{Q}_{io} \in \mathbb{R}^{N_r \times d}$ and *predicate representation* $\boldsymbol{Q}_p \in \mathbb{R}^{N_r \times d}$. We use the decoupled queries $\boldsymbol{Q}_{is}, \boldsymbol{Q}_{io}$ as entity indicator to explicitly modeling the predicate-entity association.

We dynamically generate this query with the entity-aware and scene-adaptive predicate representation $\boldsymbol{Q}^e_p$, from the initial query initial predicate queries $\boldsymbol{Q}_{init} \in \mathbb{R}^{N_r \times d}$ and entities representation $\boldsymbol{B}_e, \boldsymbol{H}_e$. Inspired by the previous work Yao et al. (2021b), we build the geometric-aware entity representation as to the key and value as follows: $\boldsymbol{K}_{init} = \boldsymbol{V}_{init} = (\boldsymbol{H}_e + \boldsymbol{G}_e) \in \mathbb{R}^{N_e \times d}, \boldsymbol{G}_e = \text{ReLU}(\boldsymbol{B}_e \boldsymbol{W}_g) \in \mathbb{R}^{N_e \times d}$, where $\boldsymbol{G}_e$ is the learnable geometric embedding of each entity query, $\boldsymbol{W}_g \in \mathbb{R}^{4 \times d}$ is the transformation parameters from bounding box location to embedding space. To generate the entity-aware predicate queries, a multi-head cross-attention is conducted between the initial predicate queries $\boldsymbol{Q}_{init} \in \mathbb{R}^{N_r \times d}$ and $\boldsymbol{K}_{init}$. We use $\mathcal{A}(q, k, v) = \text{FFN}(\text{MHA}(q, k, v))$ to denote the multi-head attention operation in the following sections for clarity. Thus we have $\boldsymbol{Q}^e_p = \mathcal{A}(\boldsymbol{Q}_{init}, \boldsymbol{K}_{init}, \boldsymbol{V}_{init})\boldsymbol{W}_e \in \mathbb{R}^{N_r \times 3d}$ and $\boldsymbol{W}_e \in \mathbb{R}^{d \times 3d}$. Finally, we split the $\boldsymbol{Q}^e_p$ into three

[2] The subscripts 's', 'o' stand for the subject and object entity, respectively.

decoupled queries $\boldsymbol{Q}_{is}, \boldsymbol{Q}_{io}, \boldsymbol{Q}_p$ To this end, we obtain the structural query which incorporates the entity information into the predicate query explicitly.

4.2.3 STRUCTURAL PREDICATE NODE DECODER

To leverage the compositional property, we develop a structural predicate node decoder to decode each component of the entity-aware predicate query $\boldsymbol{Q}_q^e$ in parallel. Our structural decoder consists of three modules: a) *predicate sub-decoder*; b) *entity indicator sub-decoders*; c) *predicate indicator fusion*. These two types of decoders take the encoder feature map $\boldsymbol{Z}^p$ and entity instance feature from entity generator $\boldsymbol{H}_e$ respectively. Based on the updated predicate and entity indicators, the *predicate-indicator fusion* is adopted to refine the association within the predicate node query.

We adopt the standard transformer decoder structure in this work, where each decoder layer consists of a multi-head self-attention layer, a multi-head cross-attention layer, and FFN layers. For notation clarity, we focus on the single decoder layer and omit layer number l while introducing the decoder of this section. The detailed notation of self-attention operation is also omitted. The iterative form will be discussed in the predicate-indicator fusion paragraph.

Predicate Sub-decoder. The predicate sub-decoder is designed to decode the predicate representation from the image feature map $\boldsymbol{Z}^p$, which utilizes the spatial context of the image for extracting predicate representation. We implement this decoding process using the cross-attention mechanism: $\widetilde{\boldsymbol{Q}}_p = \mathcal{A}(q = \boldsymbol{Q}_p, k = \boldsymbol{Z}^p, v = \boldsymbol{Z}^p)$. $\widetilde{\boldsymbol{Q}}_p$ is updated predicate representation.

Entity Indicator Sub-Decoders The entity indicator sub-decoders explicitly learn the representation of which entity associates with the predicate. Instead of relie on image feature, we leverage more accurate entity information of the given scene to conduct cross-attention between entity indicator of each predicate $\boldsymbol{Q}_{is}, \boldsymbol{Q}_{io}$ and entity feature $\boldsymbol{H}_e$ from the entity node generator, which explicitly modeling the association between entity and predicate. We denote the updated representation of the entities indicator as $\widetilde{\boldsymbol{Q}}_{is}, \widetilde{\boldsymbol{Q}}_{io}$, which are generated with standard cross-attention operation:

$$\widetilde{\boldsymbol{Q}}_{is} = \mathcal{A}(\boldsymbol{Q}_{is}, \boldsymbol{H}_e, \boldsymbol{H}_e), \quad \widetilde{\boldsymbol{Q}}_{io} = \mathcal{A}(\boldsymbol{Q}_{io}, \boldsymbol{H}_e, \boldsymbol{H}_e) \tag{2}$$

Predicate-Indicator Fusion The predicate sub-decoder owns a multi-layer self-attention design for modeling the relationships among all the predicates. However, it is necessary to encode the context between the predicate and its entity indicator for calibrating the features of each component. We explicitly fuse the current l-th decoder layer outputs $\widetilde{\boldsymbol{Q}}_p^l, \widetilde{\boldsymbol{Q}}_{is}^l, \widetilde{\boldsymbol{Q}}_{io}^l$ to update each component of as the query for next layer $\boldsymbol{Q}_p^{l+1}, \boldsymbol{Q}_{is}^{l+1}, \boldsymbol{Q}_{io}^{l+1}$. Specifically, we adopt simple fully connected layers for updating the predicate by fusing entity indicators representation as Eq. 3:

$$\boldsymbol{Q}_p^{l+1} = \left(\widetilde{\boldsymbol{Q}}_p^l + \left(\widetilde{\boldsymbol{Q}}_{is}^l + \widetilde{\boldsymbol{Q}}_{io}^l\right) \cdot \boldsymbol{W}_i\right) \cdot \boldsymbol{W}_p \tag{3}$$

where $\boldsymbol{W}_i, \boldsymbol{W}_p \in \mathbb{R}^{d \times d}$ is the transformation parameters for updating. For entity indicator, we simplicity adopt the previous layer output as input: $\boldsymbol{Q}_{is}^{l+1} = \widetilde{\boldsymbol{Q}}_{is}^l, \boldsymbol{Q}_{io}^{l+1} = \widetilde{\boldsymbol{Q}}_{io}^l$.

Based on the entity-aware predicate queries, we are able to predict the geometric and semantic prediction of the predicate node, and generate the location/category of its associated entity indicator.

$$\boldsymbol{P}_p = \text{Softmax}(\widetilde{\boldsymbol{Q}}_p \cdot \boldsymbol{W}_{cls}^p) \in \mathbb{R}^{N_r \times (\mathcal{C}_p + 1)}, \tag{4}$$

$$\boldsymbol{B}_p = \sigma(\widetilde{\boldsymbol{Q}}_p \cdot \boldsymbol{W}_{reg}^p) = \{(x_c^s, y_c^s, x_c^o, y_c^o)\} \in \mathbb{R}^{N_r \times 4} \tag{5}$$

where $\boldsymbol{P}_p$ is classification predictions of predicates, and $\boldsymbol{B}_p = \{(x_c^s, y_c^s, x_c^o, y_c^o)\}$ is the box center coordinates of its associated subject and object entities. The entity indicators are also translated as location prediction of entities $\boldsymbol{B}_s, \boldsymbol{B}_o \in \mathbb{R}^{N_r \times 4}$ and their classification predictions $\boldsymbol{P}_s, \boldsymbol{P}_o \in \mathbb{R}^{N_r \times (\mathcal{C}_e + 1)}$, which are similar to the entity detector. Finally, each predicate decoder layer produces the location and classification prediction of each entity-aware predicate query. With this multi-layer structure, the predicate decoder is able to gradually improve the quality for predicate and entity association.

4.3 BIPARTITE GRAPH ASSEMBLING

In the proposed SGTR, we convert the original scene graph into a bipartite graph structure which consists of N_e entity nodes and N_r predicate nodes, as shown in Fig. 3. The main goal of the assembling is to link the entity-aware predicate nodes to the proper entity node.

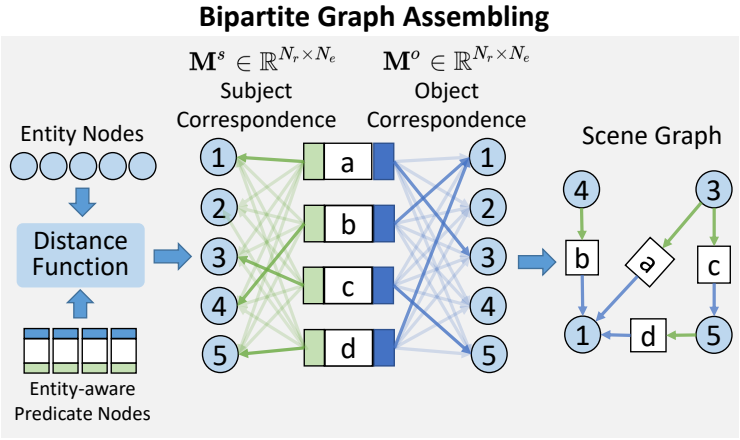

Figure 3: **The illustration of Bipartite Graph Assembling.**

To achieve this, we need to obtain the adjacency matrix between the N_e entity nodes and N_r predicate nodes, which can be encoded into a correspondence matrix $\boldsymbol{M} \in \mathbb{R}^{N_r \times N_e}$. Concretely, those correspondence score is defined by the distance between the entity indicators of predicates and the entity nodes. Taking the subject entity indicator as example, we have: $\boldsymbol{M}^s = d_{loc}(\boldsymbol{B}_s, \boldsymbol{B}_e) \cdot d_{cls}(\boldsymbol{P}_s, \boldsymbol{P}_e)$, where $d_{loc}(\cdot)$ and $d_{cls}(\cdot)$ are the distance function to measure the matching quality from different perspectives[3]. The correspondence of object entity $\boldsymbol{M}^o \in \mathbb{R}^{N_r \times N_e}$ is obtained following the same strategy. The empirical analysis of different distance measurements will be discussed in the experiment section. Based on the correspondence matrix, we keep the top-K links according to the matching score as the edge links for each predicate node,

$$\boldsymbol{R}^s = \mathcal{F}_{top}(\boldsymbol{M}^s, K) \in \mathbb{R}^{N_r \times K} \tag{6}$$

$$\boldsymbol{R}^o = \mathcal{F}_{top}(\boldsymbol{M}^o, K) \in \mathbb{R}^{N_r \times K} \tag{7}$$

where $\mathcal{F}_{top}$ is the top-K index selection operation, $\boldsymbol{R}^s$ and $\boldsymbol{R}^o$ are the index matrix of entity kept for each triplet from the two relationship roles of subject and object, respectively.

Based on the index matrix, we are able to generate the final relationship triplets as $\mathcal{T} = \{(\boldsymbol{b}_e^s, \boldsymbol{p}_e^s, \boldsymbol{b}_e^o, \boldsymbol{p}_e^o, \boldsymbol{p}_p, \boldsymbol{b}_p)\}$. The $\boldsymbol{b}_e^s, \boldsymbol{b}_e^o \in \mathbb{R}^{1\times 4}$ and $\boldsymbol{p}_e^s, \boldsymbol{p}_e^o \in \mathbb{R}^{1\times(\mathcal{C}_e+1)}$ are bounding box and classification prediction of subject and object entity respectively. $\boldsymbol{p}_p \in \mathbb{R}^{1\times(\mathcal{C}_p+1)}$ is classification predication of each predicate $\boldsymbol{P}_p$, and $\boldsymbol{b}_p \in \boldsymbol{B}_p$ is the centers of the predicate's associated entities. To this end, the graph assembling module generates the final scene graph as the output of SGTR.

4.4 Learning and Inference

Learning To train our SGTR model, we design a multi-task loss that consists of two components, including $\mathcal{L}^{enc}$ for entity generator and $\mathcal{L}^{pre}$ for predicate generator. The overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}^{enc} + \mathcal{L}^{pre}, \quad \mathcal{L}^{pre} = \mathcal{L}_i^{pre} + \mathcal{L}_p^{pre} \tag{8}$$

As we adopt a DETR-like detector, the $\mathcal{L}^{enc}$ follows a similar form with Carion et al. (2020), detailed loss equation is reported in the supplementary material. We mainly focus on $\mathcal{L}^{pre}$ in the remaining of this section.

To calculate the loss for the predicate node generator, we first obtain the matching matrix between the prediction and the ground truth, by adopting the Hungarian matching algorithm Kuhn (1955). We first convert the ground-truth of the visual relationships into a set of triplet representations in the similar form of $\mathcal{T}$, denoted as $\mathcal{T}^{gt}$. The cost of the set matching is defined as:

$$\mathcal{C} = \lambda_p \mathcal{C}_p + \lambda_e \mathcal{C}_e \tag{9}$$

[3] *e.g.,* cosine distance between the classification distribution, GIOU and L1 distance between the bounding box predictions, detailed illustration is presented in supplementary details.

#	EPN	SPD	GA	mR@50	mR@100	R@50	R@100
1	✓	✓	✓	**24.2**	**28.2**	**13.9**	**17.3**
2		✓	✓	12.0	15.9	22.9	26.3
3	✓		✓	11.4	15.1	21.9	24.9
4			✓	11.3	14.8	21.2	24.1
5	✓	✓		4.6	7.0	10.6	13.3

Table 1: **Ablation study on model components.** EPN: Entity-aware Predicate Node; SPD: Structural Predicate Decoder, GA: Graph Assembling.

The two components of the total cost correspond to the cost of the predicate, subject, and object entity, respectively. [4] The matching index $\boldsymbol{I}^{tri}$ between the triplet prediction and the ground truth is produced by: $\boldsymbol{I}^{tri} = \text{argmin}_{\mathcal{T},\mathcal{T}^{gt}}\mathcal{C}$, which is used for the following loss calculation of predicate node generator.

The two terms of $\mathcal{L}^{pre}$, that is, $\mathcal{L}_i^{pre}, \mathcal{L}_p^{pre}$ are used to supervise two types of sub-decoder in predicate node generator. For the entity indicator sub-decoder, we have $\mathcal{L}_i^{pre} = \mathcal{L}_{box}^i + \mathcal{L}_{cls}^i$, where $\mathcal{L}_{box}^i$ and $\mathcal{L}_{cls}^i$ are the localization loss (L1 and GIOU loss) and cross-entropy loss for entities indicator $\boldsymbol{P}_s, \boldsymbol{B}_s, \boldsymbol{P}_o, \boldsymbol{B}_o$. Similarly, for the predicate sub-decoder, we have $\mathcal{L}_p^{pre} = \mathcal{L}_{ent}^p + \mathcal{L}_{cls}^p$. The $\mathcal{L}_{ent}^p$ is the L1 loss of the location of the predicate's associated entities $\boldsymbol{B}_p$. The $\mathcal{L}_{cls}^p$ is the cross entropy of the predicate category $\boldsymbol{P}_p$.

Inference During model inference, we generate $K \cdot N_r$ visual relationship predictions after the assembling stage. We further remove the invalid self-connection edges during inference. We adopt a post-process operation to filter the self-connected triplets (subject and object entities are identical). Then, we rank the remaining predictions by the triplet score $\mathcal{S}_t$ and take the top N relationship triplet as final outputs. We have $\mathcal{S}^t = \{(s_s^t \cdot s_o^t \cdot s_p^t)\}$, where s_s^t, s_o^t and s_p^t are the classification probability of subject entity, object entity and predicate, respectively.

5 Experiments

5.1 Experiments Configuration

We evaluate our methods on Openimage V6 datasets Kuznetsova et al. (2020) and Visual Genome Krishna et al. (2017). We mainly adopt the data splits and evaluation metrics from the previous work Xu et al. (2017); Zellers et al. (2018); Li et al. (2021). In Openimage banchmark, the weighted evaluation metrics (wmAP$_{phr}$, wmAP$_{rel}$, score$_{wtd}$) are adopted by us for more class-balance evaluation. For Visual Genome dataset, we adopt the evaluation metric recall@K (R@K) and mean recall@K (mR@K) of SGDet, and also report the mR@100 on each long-tail category groups: *head*, *body* and *tail* as same as Li et al. (2021).

We use the ResNet-101 and DETR Carion et al. (2020) as backbone networks and entity detector. To increase the speed of convergence, we first train the entity detector on the target dataset, before the joint training with the predicate node generator. Differently, we don't fix the parameters of the detector in the SGG training phrase, the detection performance still can preserve or improve. In our predicate node generator, we use the 3 layers for predicate encoders, 6 layers decoder for predicate and entity indicator sub-decoder respectively, with $N_r = 150$ queries with $d = 256$ hidden dimensions. We set the $K = 40$ in training time and $K = 3$ in test time for the graph assembling module.

5.2 Ablation Study

Model Components As shown in Tab. 1, we ablate each module to demonstrate the effectiveness of our design on the validation set of Visual Genome.

• We find that using the holistic query for predicate rather than the proposed structural form decreases the performance by a margin of R@100 and mR@100 at **1.9** and **1.4** in line-2.

[4] We utilize the location and classification prediction to calculate the cost for each component, detailed formulations are presented in supplementary.

NPD	NED	mR@50	mR@100	R@50	R@100
3	3	10.6	13.3	23.4	27.4
6	6	**13.9**	**17.3**	**24.2**	28.2
12	12	13.7	17.0	24.0	**28.4**

Table 2: **Ablation study on number of predicate decoder layers.** NPD: number of predicate sub-decoder layers; NED: number of entity indicator sub-decoder layers;

GA	mR@50	mR@100	R@50	R@100
S	10.6	11.8	**24.4**	27.7
F	13.3	16.1	23.7	27.5
Ours	**13.9**	**17.3**	24.2	**28.2**

Table 3: **Ablation study on graph assembling**, S: spatial distance between the predicate and entity-based matching function proposed by AS-NetChen et al. (2021); F: feature similarity-based matching function proposed by HOTR Kim et al. (2021).

• Adopting the shared cross-attention between the image features and predicate/entity indicator instead of the structural predicate decoder leads to the sub-optimal performance as reported in line-3

• We further remove both entity indicators and directly decode the predicate node from the image feature. The result is reported in line-4, which decreases the performance by a marge of **4.2** and **2.5** on R@100 and mR@100.

• We also investigate the graph assembling mechanism by directly adopting the prediction of entity indicators as entity nodes for relationship prediction. The poor results are shown in line-5 demonstrate that the model struggles to tackle such a complex multi-task within a single structure, while proposed entity-prediction association modeling and graph assembling largely reduce the difficulty of optimization.

Graph Assembling Design We further investigate the effectiveness of our graph assembling design. Specifically, we adopt the differentiable entity-predicate pair matching function proposed by recent HOI methods Chen et al. (2021); Kim et al. (2021), as shown in Tab. 3. Comparison experiments are conducted on the validating set of Visual Genome by using different distance functions for the assembling module. In AS-NetChen et al. (2021), the matching is conducted based on the distance between entity bounding box and entity center predicted by interaction branch, which lacks the entity semantic information. The HOTR Kim et al. (2021) introduces a cosine similarity measurement between the predicate and entity in feature space. We implement this form for calculation the distance between the entity indicator $\widetilde{\mathbf{Q}}_{is}$, $\widetilde{\mathbf{Q}}_{io}$ and entity nodes $\mathbf{H}_e$. Compared with location-only Chen et al. (2021) similarity and feature-based Kim et al. (2021) similarity, our proposed assembling mechanism, taking both semantic and spatial information into the similarity measurement, is preferable. We also empirically observe that the feature-based Kim et al. (2021) similarity design has a slower and unstable convergence process.

Model Size To investigate the decoder layers' number of structural predicate decoder, we incrementally vary the number layer L of predicate and entity indicator decoder. The quantitative results are shown in Tab. 2. The results indicate that our model achieves the best performance while $L = 6$. We observe that the performance improvement is considerable when increasing the number of decoder layers from 3 to 6, and the performance will be saturate when $L = 12$.

5.3 Comparisons with State-of-the-Art Methods

We conduct experiments on Openimage benchmarks and VG datasets to demonstrate the effectiveness of our design. We compare our method with several state-of-the-art two-stage(*e.g.* EMB, VCTree-PCPL, VCTree-DLFE, BGNN Li et al. (2021) , VCTree-TDE) and one-stage methods(*e.g.* AS-Net, HOTR, FCSGG) on Visual Genome dataset. Since our backbone is different from what they reported, we reproduced the SOTA methods BGNN and its baseline RelDN with the same Res101 backbone for more fair comparisons. Furthermore, since there are only FCSGG for SGG specifically, we reproduce the result of several strong related one-stage HOI methods with similar entity-predicate

B	Models	mR@50	R@50	wmAP		$\text{score}_{\text{wtd}}$
				rel	phr	
X101-F	RelDN	37.20	**75.40**	33.21	31.31	41.97
	GPS-Net	38.93	74.74	32.77	33.87	41.60
	BGNN	40.45	74.98	33.51	34.15	42.06
R101	BGNN*†	39.41	74.93	31.15	31.37	40.00
	RelDN†	36.80	72.75	29.87	30.42	38.67
	HOTR†	40.09	52.66	19.38	21.51	26.88
	AS-Net†	35.16	55.28	25.93	27.49	32.42
	Ours	**42.61**	59.91	**36.98**	**38.73**	**42.28**

Table 4: **The Performance on Openimage V6.** † denotes results reproduced with the authors' code. The performance of ResNeXt-101 FPN is borrow from Li et al. (2021). * means using resampling strategy.

pairing mechanisms (AS-Net Kim et al. (2021), HOTR Kim et al. (2021)) by authors code for a more comprehensive comparison.

OpenImage V6 The performance on the OpenImage V6 dataset is reported in Tab. 4. We re-implement the SOTA one-stage and two-stage methods with the same ResNet-101 backbone. Our method outperforms two-stage SOTA method BGNN with a improvement of **2.28**. Specifically, our design has a significant improvement on weighted mAP metrics of relationship detection (rel) and phrase detection (phr) sub-tasks of **5.83** and **7.36** respectively, which indicates that leveraging the compositional property of the visual relationship is beneficial for the SGG task.

B	D	Method	mR@50/100	R@50/100	Head	Body	Tail	Time/Sec
⋆	⋆	FCSGG Liu et al. (2021)	3.6 / 4.2	21.3 / 25.1	-	-	-	0.12
X101-FPN	Faster-RCNN	RelDN Li et al. (2021)	6.0 / 7.3	31.4 / 35.9	-	-	-	0.65
		MotifsTang et al. (2020)	5.5 / 6.8	**32.1 / 36.9**	-	-	-	1.00
		VCTreeTang et al. (2020)	6.6 / 7.7	31.8 / 36.1	-	-	-	1.69
		BGNN*† Li et al. (2021)	10.7 / 12.6	31.0 / 35.8	34.0	12.9	6.0	1.32
		VCTree-TDETang et al. (2020)	9.3 / 11.1	19.4 / 23.2	-	-	-	-
		VCTree-PCPL† Chiou et al. (2021)	10.8 / 12.6	26.6 / 30.1	-	-	-	-
		VCTree-DLFE Chiou et al. (2021)	11.8 / 13.8	22.7 / 26.3	-	-	-	-
		VCTree-EBM Suhail et al. (2021)	9.7 / 11.6	20.5 / 24.7	-	-	-	-
		VCTree-BPLSA Guo et al. (2021)	13.5 / 15.7	21.7 / 25.5	-	-	-	-
		MOTIFS-VDS Yao et al. (2021a)	13.8 / 15.2	23.9 / 25.7	-	-	-	-
		DT2-ACBS Desai et al. (2021)	**22.0 / 24.4**	15.0 / 16.3	-	-	-	-
R101		BGNN*†	8.6 / 10.3	28.2 / 33.8	29.1	12.6	2.2	1.32
		RelDN†	4.4 / 5.4	**30.3 / 34.8**	**31.3**	2.3	0.0	0.65
	DETR	AS-Net† Chen et al. (2021)	6.12 / 7.2	18.7 / 21.1	19.6	7.7	2.7	0.33
		HOTR† Kim et al. (2021)	9.4 / 12.0	23.5 / 27.7	26.1	16.2	3.4	0.25
		Ours[1]	12.0 / 14.6	25.1 / 26.6	27.1	17.2	6.9	0.35
		Ours	12.0 / 15.2	24.6 / 28.4	28.2	18.6	7.1	0.35
		Ours*	**15.8 / 20.1**	20.6 / 25.0	21.7	**21.6**	**17.1**	0.35

Table 5: **The SGDet performance on test set of Visual Genome dataset.** † denotes results reproduced with the authors' code. $*$ denotes the bi-level resampling Li et al. (2021) is applied for this model. [1] denotes that our model uses the top-1 matching in graph assembling. $\star$ denotes the special backbone HRNetW48-5S-$\text{FPN}_{\times 2\text{-f}}$ and entities detector, CenterNetZhou et al. (2019).

Visual Genome As shown in Tab. 5, with same ResNet-101 backbone, we compare our method with two-stage method BGNN Li et al. (2021), and one-stage methods HOTR Kim et al. (2021). It shows that our method outperforms with a significant margin of **4.9** and **3.2** on mRecall@100 respectively.

• Benefitting from the sparse proposal set, SGTR has a more balanced foreground/background proposal distribution than the traditional two-stage design, where there exists a large number of negative samples due to exhausted entity pairing. Thus our method achieves competitive performance on the mean recall when equipped with the same backbone and learning strategy. We also list the

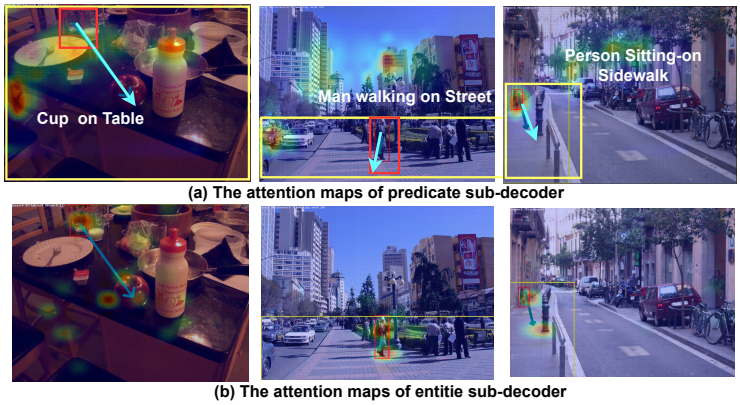

Figure 4: **The visualization on attention heatmap of structural predicate decoder.**

rest newly proposed works, which propose different training strategies for solving the long-tailed recognition. Our method achieves higher mR@100 performance with few overall performance drops when using the resampling strategy proposed in Li et al. (2021).

• We also find that the performance of our model on the head category is lower than the two-stage methods with the same backbone. The main reason lies in that the DETR detector performs weaker on small entities than traditional Faster-RCNN. Since the Visual Genome has a large proportion of relationships involved by small objects, our method performs sub-optimal on recognizing those relationships. The detailed analysis will be discussed in supplementary.

• We also compare the efficiency with previous methods according to the inference time (second per image) on NVIDIA GeForce Titan XP GPU with inference batch size of 1, with input size 600 x 1000. Our design obtains comparable inference time consuming with one-stage methods with the same backbone, which demonstrates the efficiency of our method.

• Moreover, the performance of our model with the recent advanced long-tailed training strategy Desai et al. (2021) is reported in supplementary.

5.4 Qualitative Results

As shown in Fig. 4, we visualize the attention weight of predicates sub-decoder, and entity sub-decoder from the validation set of Visual Genome dataset. By Comparing the heatmaps shown in Fig. 4 (a) and Fig. 4 (b), For the same triplet prediction, the predicate sub-decoder more focus on contextual representation around the entities of triplets. Entity sub-decoders focus on relationship-based entity regions. Thus, our compositional design allows the model to learn complementary information simultaneously and explicitly, improving prediction accuracy. We further demonstrate the other visualization results in supplementary(*e.g.* prediction of predicate decoder, comparison between two-stage method).

6 Conclusions

In this work, we propose a novel end-to-end CNN-Transformer-based scene graph generating approach (SGTR). In comparison to the prior approaches, our major contribution consists of two components: We formulate the SGG as a bipartite graph construction with three steps: entity and predicate nodes generation and directed edges connection. We develop the entity-aware representation for modeling the predicate nodes integrate with the entity indicators by structural predicate node decoder. Finally, the scene graph is constructed by the graph assembling module in an end-to-end manner. The extensive experimental results show that our SGTR outperforms or is competitive with previous state-of-the-art methods on the Visual Genome and Openimage V6 datasets.

Ethics Statement Our study doesn't have ethics risk as introduce in author guide. Our study doesn't involve the human subjects, practices to data set releases, potentially harmful insights, methodologies and applications, pontential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues.

Reproducibility Statement We have introduced all necessary details for reproducing our method in our submission. The datasets and data processing steps we use in the experiments are open-access for everyone.

REFERENCES

Sherif Abdelkarim, Aniket Agarwal, Panos Achlioptas, Jun Chen, Jiaji Huang, Boyang Li, Kenneth Church, and Mohamed Elhoseiny. Exploring long tail visual relationship recognition with large vocabulary. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15921–15930, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Mingfei Chen, Yue Liao, Si Liu, Zhiyuan Chen, Fei Wang, and Chen Qian. Reformulating hoi detection as adaptive set prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9004–9013, 2021.

Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. *arXiv preprint arXiv:2107.02112*, 2021.

Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. *arXiv preprint arXiv:2108.09668*, 2021.

Qi Dong, Zhuowen Tu, Haofu Liao, Yuting Zhang, Vijay Mahadevan, and Stefano Soatto. Visual relationship detection using part-and-sum transformers with composite queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3550–3559, 2021.

Yuyu Guo, Lianli Gao, Xuanhan Wang, Yuxuan Hu, Xing Xu, Xu Lu, Heng Tao Shen, and Jingkuan Song. From general to specific: Informative scene graph generation via balance adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16383–16392, 2021.

Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020.

Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

Siddhesh Khandelwal, Mohammed Suhail, and Leonid Sigal. Segmentation-grounded scene graph generation. *arXiv preprint arXiv:2104.14207*, 2021.

Bumsoo Kim, Taeho Choi, Jaewoo Kang, and Hyunwoo J Kim. Uniondet: Union-level detector towards real-time human-object interaction detection. In *European Conference on Computer Vision*, pp. 498–514. Springer, 2020.

Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 74–83, 2021.

Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W. Taylor, Aaron Courville, and Eugene Belilovsky. Graph Density-Aware Losses for Novel Compositions in Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2017.

Boris Knyazev, Harm de Vries, Catalina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Generative compositional augmentations for scene graph prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15827–15837, 2021.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision(IJCV)*, 2020.

Rongjie Li, Songyang Zhang, Bo Wan, and Xuming 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*, pp. 11109–11119, 2021.

Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1261–1270, 2017.

Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 335–351, 2018.

Yue Liao, Si Liu, Fei Wang, Yanjie Chen, Chen Qian, and Jiashi Feng. Ppdm: Parallel point detection and matching for real-time human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 482–490, 2020.

Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3746–3753, 2020.

Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11546–11556, 2021.

Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3957–3966, 2019.

Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8376–8384, 2019.

Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13936–13945, 2021.

Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14454–14463, 2021.

Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10410–10419, 2021.

Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6619–6628, 2019.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3716–3725, 2020.

Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2017.

Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. *arXiv preprint arXiv:2106.10815*, 2021.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4116–4125, 2020a.

Tzu-Jui Julius Wang, Selen Pehlivan, and Jorma Laaksonen. Tackling the unannotated: Scene graph generation with bias-reduced models. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020b.

Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8188–8197, 2019.

Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In *Advances in Neural Information Processing Systems*, pp. 560–570, 2018.

Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 5410–5419, 2017.

Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Pcpl: Predicate-correlation perception learning for unbiased scene graph generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 265–273, 2020.

Gengcong Yang, Jingyi Zhang, Yong Zhang, Baoyuan Wu, and Yujiu Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12527–12536, 2021a.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 670–685, 2018.

Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10685–10694, 2019.

Xuewen Yang, Yingru Liu, and Xin Wang. Reformer: The relational transformer for image captioning. *arXiv preprint arXiv:2107.14178*, 2021b.

Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, and Maosong Sun. Visual distant supervision for scene graph generation. *arXiv preprint arXiv:2103.15365*, 2021a.

Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021b.

Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 322–338, 2018.

Cong Yuren, Hanno Ackermann, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. Nodis: Neural ordinary differential scene understanding. *arXiv preprint arXiv:2001.04735*, 2020.

Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly supervised visual semantic parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3736–3745, 2020a.

Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2020b.

Alireza Zareian, Haoxuan You, Zhecan Wang, and Shih-Fu Chang. Learning visual commonsense for robust scene graph generation. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2020c.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840, 2018.

Aixi Zhang, Yue Liao, Si Liu, Miao Lu, Yongliang Wang, Chen Gao, and Xiaobo Li. Mining the benefits of two-stage and one-stage hoi detection. *arXiv preprint arXiv:2108.05077*, 2021.

Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical Contrastive Losses for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766*, 2018.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, et al. End-to-end human object interaction detection with hoi transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11825–11834, 2021.

7 SUPPLEMENTARY

7.1 QUERY REFINEMENT

After each layer of decoder structure, the triplet queries are updated with the decoder output. We design the fusing process that aggregate the representation between the two branches, to further improve the reprensetation of queries. For relationship predicate queries of next layer $\boldsymbol{Q}_{tp}^{l+1}$, we fuse the triplet entities hidden state with the pair-wise fusion function proposed in Zhang et al. (2018), which has been adopted in SGG task. The triplets entities queries $\boldsymbol{Q}_{to}^{l+1}, \boldsymbol{Q}_{ts}^{l+1}$ are updated with the triplet predicate decoder output $\boldsymbol{Q}_{tp}$, with the nears neighbors feature representation on encoder memory $\boldsymbol{Z}^t$, according to the center coordinates of predicted entities position $[x_c, y_c]$.

$$\boldsymbol{Q}_{tp}^{l+1} = \boldsymbol{Q}_{tp}^{l} + \text{ReLU}(\boldsymbol{W}_x \boldsymbol{Q}_{ts}^{l} + \boldsymbol{W}_y \boldsymbol{Q}_{to}^{l}) - ||\boldsymbol{Q}_{ts}^{l} - \boldsymbol{Q}_{to}^{l}||_2^2 \tag{10}$$

$$\boldsymbol{Q}_{ts}^{l+1} = \boldsymbol{Q}_{ts}^{l} + \boldsymbol{Z}^t(m,n) + \text{ReLU}(\boldsymbol{W}_{es} \boldsymbol{Q}_{tp}^{l}) \tag{11}$$

$$\boldsymbol{Q}_{to}^{l+1} = \boldsymbol{Q}_{to}^{l} + \boldsymbol{Z}^t(m,n) + \text{ReLU}(\boldsymbol{W}_{eo} \boldsymbol{Q}_{tp}^{l}) \tag{12}$$

$$m, n \leftarrow \underset{m,n \in [0,1] \times [0,1]}{\arg\min} (||[x_c, y_c] - [m, n]||_1) \tag{13}$$

7.2 MATCHING QUALITY CALCULATION FOR GRAPH ASSEMBLING

We take the matching quality of subject entities and predicates as example. For each factors of distance function is determined by the semantic outputs of entity detector and triplet decoder. The $d_{giou} \in \mathbb{R}^{N_r \times N_e}$, $d_{cos} \in \mathbb{R}^{N_r \times N_e}$, $d_{center} \in \mathbb{R}^{N_r \times N_e}$ are calculated by following process.

$$\boldsymbol{M}^s = d_{loc}(\boldsymbol{B}_s, \boldsymbol{B}_e) \cdot d_{cls}(\boldsymbol{P}_s, \boldsymbol{P}_e), \quad d_{loc}(\boldsymbol{B}_s, \boldsymbol{B}_e) = \frac{d_{giou}}{d_{center}} \tag{14}$$

$$d_{giou} = \max(\min(\text{GIOU}(\boldsymbol{b}_s, \boldsymbol{b}_e), 0), 1), \quad d_{cos} = \frac{\boldsymbol{p}_s \cdot \boldsymbol{p}_e^{\intercal}}{||\boldsymbol{p}_s|| \cdot ||\boldsymbol{p}_e||} \tag{15}$$

$$d_{center}(i,j) = ||[x_c, y_c]_i^s - [x_c, y_c]_i^e||_1 \tag{16}$$

where $[x_c, y_c]^s$ are the center coordinates of one box in $\boldsymbol{B}_s$.

7.3 TRIPLETS MATCHING COST

The triplets predication of the model is $\mathcal{T} = \{(\boldsymbol{b}_e^s, \boldsymbol{p}_e^s, \boldsymbol{b}_e^o, \boldsymbol{p}_e^o, \boldsymbol{p}_p, \boldsymbol{b}_p)\}$. The triplets matching cost $\mathcal{C} \in \mathbb{R}^{N_r \times N_{gt}}$ is composed by three part: predicate cost $\mathcal{C}_p$ and entity cost $\mathcal{C}_e$.

$$\mathcal{C} = \lambda_p \mathcal{C}_p + \lambda_e \mathcal{C}_e \quad \boldsymbol{I}^{tri} = \underset{\mathcal{T}, \mathcal{T}^{gt}}{\arg\min} \mathcal{C} \tag{17}$$

For the predicate cost $\mathcal{C}_p(i,j)$ between the i-th predicates prediction and j-th ground-truth relationship, it is computed according the predicate classification distribution.

$$\mathcal{C}_p(i,j) = \exp\left(\frac{\boldsymbol{p}_{p,i} \cdot \text{one-hot}(p_{p,j}^{gt})}{||\boldsymbol{p}_{p,i}|| \cdot ||\text{one-hot}(p_{p,j}^{gt})||} - 1\right) + \|\boldsymbol{b}_{p,i} - \boldsymbol{b}_{p,j}^{gt}\|_1 \tag{18}$$

where $\boldsymbol{p}_{p,i}$ is the i-th $\boldsymbol{p}_p$ of $\mathcal{T}$, and $p_{p,j}^{gt}$ is the predicate label of the j-th triplet in ground truth. Similarly, $\boldsymbol{b}_{p,i}$ is the i-th center coordinates(subject and object) $\boldsymbol{b}_p$ of the triplet prediction, $\boldsymbol{b}_{p,j}^{gt}$ is entity centers set of the j-th triplet in ground truth.

In entity cost $\mathcal{C}_e(i,j)$ between the i-th triplets prediction and j-th ground-truth relationship,, the calculation is given by:

$$\mathcal{C}_e(i,j) = + w_{giou} \cdot \prod_{\star=\{s,o\}} \exp\left(\max(\min(\text{GIOU}(\boldsymbol{b}^{\star}_{e,i}, \boldsymbol{b}^{\star}_{gt,j}), 0), 1)\right) \tag{19}$$

$$+ w_{l1} \cdot \sum_{\star=\{s,o\}} ||\boldsymbol{b}^{\star}_{e,i}, \boldsymbol{b}^{\star}_{gt,j}||_1 \tag{20}$$

$$+ w_{cls} \cdot \prod_{\star=\{s,o\}} \exp\left(\frac{\boldsymbol{p}^{\star}_{e,i} \cdot \text{one-hot}(p^{(*,gt)}_{e,j})}{||\boldsymbol{p}^{\star}_{e,i}|| \cdot ||\text{one-hot}(p^{(*,gt)}_{e,j})||} - 1\right) \tag{21}$$

where $\boldsymbol{b}^{\star}_{e,i}$ is the i-th entity box location come from the triplets prediction $\mathcal{T}$ after graph assembling, $\boldsymbol{p}^{\star}_{e,i}$ is the i-th entity classification prediction. $\boldsymbol{b}^{\star}_{gt,j}$ is the box location of j-th subject/object in the ground truth triplets, and $p^{(*,gt)}_{e,j}$ is entity(subject/object) class label of j-th ground truth triplets.

Based is cost function, we can obtain the matching of relationship prediction. We adopt the one-to-one Hungarian algorithm into an iterative many-to-one matching. Due to the label efficiency, the relationships can not be exhausted labeled in datasets. The one-to-one matching may lead to unstable training because many foreground relationships will be ignored. The model can not learn the proper NMS mechanism for prediction calibration. To circumvent this, we relax the matching threshold to prevent the NMS mechanism from learning. We iteratively execute T times of Hungarian minimum-cost bipartite graph matching.

7.4 Datasets and Implementation Details

7.4.1 Datasets and Metrics

Visual Genome Datasets For Visual Genome Krishna et al. (2017) dataset, we take the same split protocol as Xu et al. (2017); Zellers et al. (2018). The most frequent 150 object categories and 50 predicates are adopted for evaluation. To demonstrate the long-tailed recognition performance on VG dataset, we follow the protocol from Li et al. (2021) by dividing the categories into three disjoint groups. We we adopt the evaluation metric **recall@K(R@K)** and **mean recall@K (mR@K)** of SGDet, and also report the **mR@100 on each long-tail category groups**: *head*, *body* and *tail*.

Openimage V6 Datasets The Openimage datasets Kuznetsova et al. (2020) are large scale vision recognition datasets proposed by Google, and been used as SGG benchmarks in Zhang et al. (2019); Lin et al. (2020); Li et al. (2021); Teng & Wang (2021). We adopt the same data splits with the Li et al. (2021), which has 126,368 images used for training, 1813 and 5322 images for validation and test, respectively, with 301 object categories and 31 predicate categories.

The the weighted evaluation metrics (e.g. wmAP_{phr}, wmAP_{rel}, score_{wtd}) used in previous works Zhang et al. (2019); Lin et al. (2020); Li et al. (2021); Teng & Wang (2021). However, we argue that weighted scores are unfair when used to evaluate rare categories. Because it re-weights by multiplying the frequency of categories on per-class performance, low-frequency categories are disregarded, resulting in class unbalanced assessment metrics, even though this metric is more numerically stable, as cited in Zhang et al. (2019). In this work, we will report both weighted and initial performance (e.g. mAP_{phr}, mAP_{rel}, score) in our experiments, for more fair class balance evaluation metrics.

7.4.2 Implementation Details

We use the ResNet-101 and DETR Carion et al. (2020) as backbone networks and entities detectors, with six layers encoder and six layers decoder. The $N_e = 100$ entities queries with $d = 256$ hidden dimension are used as the proposals for feature aggregation. The same DETR detector parameters are use for all one-stage methods reproduced by us. In our triplets constructor, we use the 3 layers encoders. In triplets decoder, we adopt 12 layers decoder for predicate branch, and 5 layers decoder for entity branch, with $N_r = 150$ queries with $d = 256$ hidden dimensions. For two-stage methods, we use the Faster-RCNN detector with the ResNet-101 backbone.

To speedup the convergence, we first train the entities detector on the target dataset. Then, using this pre-trained detector, we train the relationship detector parts. The key difference between this work and previous work Kim et al. (2021); Li et al. (2021) is that we do not need to fix the parameters of the entities detector to avoid performance drop in SGG training. We keep the parameters of detector in training mode, that still can preserve the considerable performance, or obtain better performance in SGG training.