SGTR: End-to-end Scene Graph Generation with Transformer

Rongjie Li[1,3,4] Songyang Zhang [1,3,4] Xuming He[1,2]
[1]School of Information Science and Technology, ShanghaiTech University
[2]Shanghai Engineering Research Center of Intelligent Vision and Imaging
[3]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
[4]University of Chinese Academy of Sciences
{lirj2, zhangsy1, hexm}@shanghaitech.edu.cn

Abstract

Scene Graph Generation (SGG) remains a challenging visual understanding task due to its compositional property. Most previous works adopt a bottom-up two-stage or a point-based one-stage approach, which often suffers from high time complexity or sub-optimal designs. In this work, we propose a novel SGG method to address the aforementioned issues, formulating 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 that leverages 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]

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 typically represented by a triple <*subject entity, predicate, object entity*>. Based on the visual relationships, a scene can be modeled as a graph structure, with entities as nodes and predicates as edges, referred to as scene graph. The scene graph provides a compact structural representation for a visual scene, which has potential applications in many vision tasks such as visual question answering [8, 25, 31], image captioning [42, 43] and image retrieval [9].

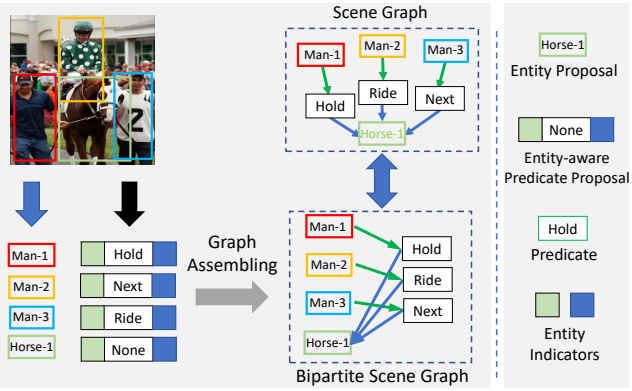

Figure 1. **The illustration of SGTR pipeline paradigm.** We formulate SGG as a bipartite graph construction process. First, the entity and predicate nodes are generated, respectively. Then we assemble the bipartite scene graph from two types of nodes.

Different from the traditional vision tasks (*e.g.,* object detection) that focus on entity instances, the main challenge of scene graph generation (SGG) lies in building an effective and efficient model for the relations between the entities. The compositional property of visual relationships induces high 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 works attempt to tackle this problem using two distinct design patterns: *bottom-up two-stage* [1, 4, 5, 7, 14, 18, 40, 44] and *point-based one-stage design* [6, 23]. The former typically first detects N entity proposals, followed by predicting the predicate categories of those entity combinations. While this strategy achieves high recalls in discovering relation instances, its $\mathcal{O}(N^2)$ predicate proposals not only incur considerable computation cost but also produce substantial noise in context modeling. In the one-stage methods, entities and predicates are often extracted separately from

[1]This work was supported by Shanghai Science and Technology Program 21010502700. Code is available: https://github.com/Scarecrow0/SGTR

the image in order to reduce the size of relation proposal set. Nonetheless, they rely on a strong assumption of the non-overlapping property of interaction regions, which severely restricts their application in modeling complex scenes[2].

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 relevant entity proposal information into each predicate node. This enriches the predicate representations 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 lead 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. To compute the entity-aware predicate representations, we design a structural predicate generator consisting of three parallel transformer decoders, which fuses the 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. With end-to-end training, our SGTR learns to infer a sparse set of relation proposals from both input images and entity proposals adaptively, which can mitigate the impact of noisy object detection.

We validate our method by extensive experiments on two SGG benchmarks: We validate our method by extensive experiments on two SGG benchmarks: Visual Genome and OpenImages-V6 datasets, with comparisons to previous state-of-the-art methods. The results show that our method outperforms or achieves comparable performance on both benchmarks and with high efficiency during inference.

The main contribution of our work has three-folds:

- We propose a novel transformer-based end-to-end scene graph generation method with a bipartite graph construction process that inherits the advantages of both two-stage and one-stage methods.
- We develop an entity-aware structure for exploiting the compositional properties of visual relationships.
- Our method achieves the state-of-the-art or comparable performance on all metrics w.r.t the prior SGG methods and with more efficient inference.

[2] *e.g.,* two different relationships cannot have largely overlapped area – a phenomenon also discussed in the recent works on (HOI) [3, 28]

2. Related Works

We categorize the related work of SGG/HOI according to three research 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 between densely connected entity pairs. Based on dense relationship proposals, many previous works focus on modeling contextual structure [10, 18–20, 22, 24, 30, 35–38, 41, 46–51]. Recent studies develop logit adjustment and other training strategies to address the long-tail recognition in the SGG task [1, 4, 5, 7, 13, 14, 18, 26, 29, 35, 39, 40, 44]. The two-stage design is capable of handling complex scenarios encountered in SGG.

However, as discussed in Sec. 1, the dense relation proposal generation often leads to high time complexity and unavoidable noise in context modeling. Many two-stage works propose heuristic designs to address these issues (*e.g.*, proposal generation [41], efficient context modeling [18, 19, 24, 30, 36, 43]). However, these sophisticated designs often rely on the specific properties of the downstream tasks, which limits the flexibility of their representation learning and is difficult to achieve end-to-end optimization.

One-stage Scene Graph Generation Inspired by the fully convolutional one-stage object detection methods [2, 27, 33], the SGG community starts to explore the one-stage design. The fully convolutional network [23, 32] or CNN-Transformer [6] architecture is used in the one-stage methods to detect the relationship from image features directly. These one-stage frameworks typically can perform efficiently due to their sparse proposal set. Nonetheless, without explicit entity modeling, those designs may struggle to capture the complex visual relationships associated with real-world scenarios. Moreover, the majority of one-stage methods ignore entity-relation consistency as they predict each relationship independently rather than a valid graph structure with consistent node-edge constraint.

One-stage Human-Object Interaction Our work is also related to the Human-Object Interaction (HOI) task. There has been a recent trend toward studying the one-stage framework for Human-Object Interaction [3, 11, 12, 21, 28, 34, 52, 54]. In particular, [3, 12] introduce an intriguing framework based on

a dual decoder structure that simultaneously extracts the human, object, and interaction and then groups the components into final triplets. This decoding-grouping approach provides a divide-and-conquer strategy for detecting the human and interacted object. Inspired by this design, we propose the bipartite graph construction method in our SGTR for the more general SGG task. To further improve the association modeling between entity and predicate, we propose a predicate node generator with an entity-aware structure and a graph assembling mechanism. With such a design, the SGTR is able to handle the complex composition of relationships and achieve strong performance on SGG benchmarks.

3. Preliminary

In this section, we first introduce the problem setting of scene graph generation in Sec. 3.1, and then present an overview of our approach 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 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 possible way to generate the scene graph $\mathcal{G}_{scene}$ is by extracting the relationship triplet set from the given image. In this work, we formulate the relationship triplet generation process as a bipartite graph construction task [18]. Specifically, our 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 $\mathbf{I}$ as the input and outputs the bipartite graph $\mathcal{G}_b$, denoted as $\mathcal{G}_b = \mathcal{F}_{sgg}(\mathbf{I})$, 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 consists of two steps: *a) node (entity and predicate) generation*, and *b) directed edge connection*.

In the *node generation* step, we extract the entity nodes and predicate nodes from the image with an *entity node generator* and a *predicate node generator*, respectively. The predicate node generator augments the predicate proposals with entity information based on three parallel sub-decoders. In the *directed edge connection* step, we design a *graph assembling module* to generate 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. Our Approach

Our model consists of four main submodules: (1) a **backbone network** for generating feature representation of the scene (Sec. 4.1); (2) a transformer-based **entity node generator** for predicting entity proposals (Sec. 4.1); (3) a structural **predicate node generator** for decoding predicate nodes (Sec. 4.2); (4) a **bipartite graph assembling** module for constructing final bipartite graph via connecting entity nodes and entity-aware predicate nodes (Sec. 4.3). The model learning and inference are detailed in Sec. 4.4.

4.1. Backbone and Entity Node Generator

We adopt a ResNet as the backbone network, which first produces a convolutional feature representation for the subsequent modules. Motivated by the Transformer-based detector, DETR [2], we then use a multi-layer Transformer encoder to augment the convolutional features. The resulting CNN+transformer feature is denoted as $\mathbf{Z} \in \mathbb{R}^{w\times h\times d}$, where w, h, d are the width, height, and channel of the feature map, respectively.

For the entity node generator, we adopt the decoder of DETR to produce N_e entity nodes from a set of learnable entity queries. Formally, we define the entity decoder as a mapping function $\mathcal{F}_e$, which takes initial entity query $\mathbf{Q}_e \in \mathbb{R}^{N_e\times d}$ and the feature map $\mathbf{Z}$ as inputs, and outputs the entity locations $\mathbf{B}_e \in \mathbb{R}^{N_e\times 4}$ and class scores $\mathbf{P}_e \in \mathbb{R}^{N_e\times(\mathcal{C}_e+1)}$, along with their associated feature representations $\mathbf{H}_e \in \mathbb{R}^{N_e\times d}$ as follows,

$$\mathbf{B}_e, \mathbf{P}_e, \mathbf{H}_e = \mathcal{F}_e(\mathbf{Z}, \mathbf{Q}_e) \tag{1}$$

where $\mathbf{B}_e = \{\mathbf{b}_1, \cdots, \mathbf{b}_{N_e}\}, \mathbf{b} = (x_c, y_c, w_b, h_b)$, 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 generate an entity-aware predicate representation by incorporating relevant entity proposal information into each predicate node. Such a design enables us to encode potential associations between each predicate and its subject/object entities, which can facilitate predicting the graph edges and lead to efficient generation of the visual relation triplets.

As shown in Fig. 2, 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).

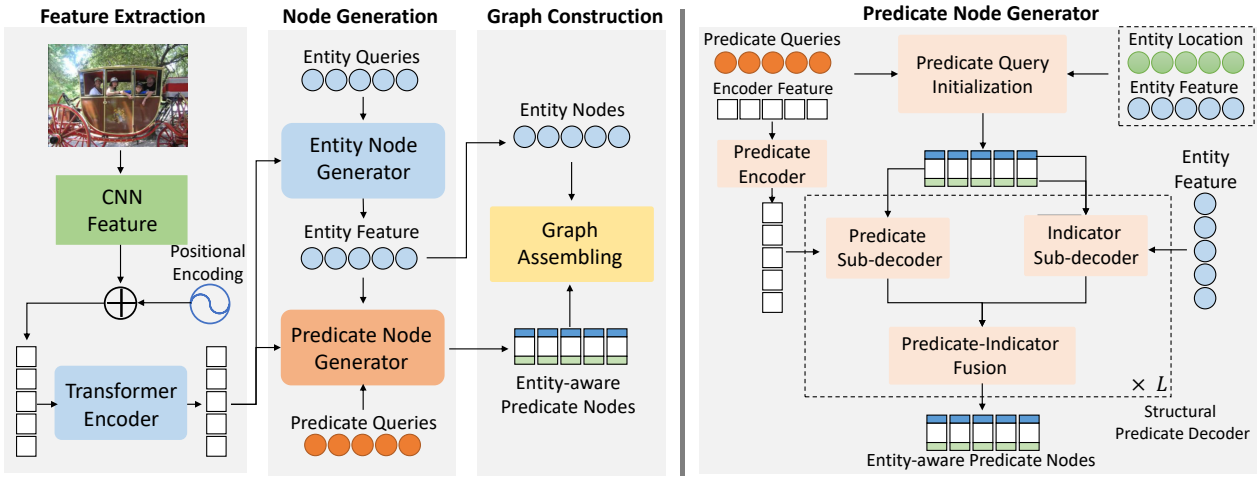

Figure 2. **An illustration of overall pipeline of our SGTR model. Left)** We use a CNN backbone together with a transformer encoder for image feature extraction. The entity and predicate node generators are introduced to produce the entity node and entity-aware predicate node. A graph assembling mechanism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

4.2.1 Predicate Encoder

Based on the CNN+transformer features $\mathbf{Z}$, we introduce a lightweight predicate encoder to extract predicate-specific image features. Our predicate encoder, which has a similar structure to the backbone Transformer encoder, employs a form of multi-layer multi-head self-attention via the skip-connected feed-forward network. The resulting predicate-specific feature is denoted as $\mathbf{Z}^p \in \mathbb{R}^{w \times h \times d}$.

4.2.2 Predicate Query Initialization

A simple strategy for initializing the predicate queries is to adopt a set of learnable vectors as in the DETR [2]. However, such a holistic vector-based query design ignores not only the *compositional property* of the visual relationships but also the *entity candidate* information. The resulting representations are not expressive enough for capturing the structured and diverse visual relationships.

To cope with this challenge, we introduce a compositional query representation that decouples predicate queries, denoted as $\mathbf{Q}_p^e \in \mathbb{R}^{N_r \times 3d}$, into three components $\{\mathbf{Q}_{is}; \mathbf{Q}_{io}; \mathbf{Q}_p\}$, where *subject/object entity indicator* $\mathbf{Q}_{is}, \mathbf{Q}_{io} \in \mathbb{R}^{N_r \times d}$ [3] and *predicate representation* $\mathbf{Q}_p \in \mathbb{R}^{N_r \times d}$. Concretely, we generate the predicate query $\mathbf{Q}_p^e$ in an entity-aware and scene-adaptive manner using a set of initial predicate queries $\mathbf{Q}_{init} \in \mathbb{R}^{N_r \times d}$ and entities representation $\mathbf{B}_e, \mathbf{H}_e$. To achieve this, we first build a geometric-aware entity representation as in [45], which defines a set of key and value vectors $\in \mathbb{R}^{N_e \times d}$ as follows:

$$\mathbf{K}_{init} = \mathbf{V}_{init} = (\mathbf{H}_e + \mathbf{G}_e), \mathbf{G}_e = \text{ReLU}(\mathbf{B}_e \mathbf{W}_g), \quad (2)$$

where $\mathbf{G}_e \in \mathbb{R}^{N_e \times d}$ is a learnable geometric embedding of entity proposals, $\mathbf{W}_g \in \mathbb{R}^{4 \times d}$ is a transformation from bounding box locations to the embedding space.

Given the augmented entity representations, we then compute the predicate queries $\mathbf{Q}_p^e$ using a multi-head cross-attention operation on the initial predicate queries $\mathbf{Q}_{init}$ and $\mathbf{K}_{init}$. For clarity, we use $\mathcal{A}(q, k, v) = \text{FFN}(\text{MHA}(q, k, v))$ to denote the multi-head attention operation. As such, we have $\mathbf{Q}_p^e = \mathcal{A}(\mathbf{Q}_{init}, \mathbf{K}_{init}, \mathbf{V}_{init})\mathbf{W}_e$, where $\mathbf{W}_e \in \mathbb{R}^{d \times 3d} = [\mathbf{W}_e^{is}, \mathbf{W}_e^{io}, \mathbf{W}_e^{p}]$ are the transformation matrices for the three sub-queries $\mathbf{Q}_{is}, \mathbf{Q}_{io}, \mathbf{Q}_p$, respectively. In this way, we obtain a structural query that incorporates the entity information into the predicate query. The sub-queries $\mathbf{Q}_{is}, \mathbf{Q}_{io}$ are referred to as entity indicators as they will be used to capture predicate-entity associations below.

4.2.3 Structural Predicate Node Decoder

Given the predicate query $\mathbf{Q}_q^e$, we now develop a structural predicate node decoder that leverages the compositional property and decodes all the predicate triplets from the entity/predicate feature maps.

Our structural decoder consists of three modules: a) *predicate sub-decoder*; b) *entity indicator sub-decoders*; c) *predicate indicator fusion*. The two types of decoders take the encoder feature map $\mathbf{Z}^p$ and entity features $\mathbf{H}_e$, respectively and update the three components of the predicate query independently. Based on the updated predicate query components, the *predicate-indicator fusion* refines the entire predicate queries, aiming to improve the entity-predicate association within each compositional query.

Specifically, we adopt the standard transformer decoder structure below. For notation clarity, we focus on a single decoder layer and omit layer number l within each sub-decoder,

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

as well as the notation of the self-attention operation.

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

Entity Indicator Sub-Decoders. The entity indicator sub-decoders refine the entity indicators associated with the predicate queries. Instead of relying on image features, we leverage more accurate entity features in the given scene. Specifically, we perform cross-attention operation between entity indicators $\mathbf{Q}_{is}, \mathbf{Q}_{io}$ and entity proposal features $\mathbf{H}_e$ from the entity node generator, aiming to enhance the representation of the entity associations.We denote the updated representation of the entities indicator as $\widetilde{\mathbf{Q}}_{is}, \widetilde{\mathbf{Q}}_{io}$, which are generated with standard cross-attention operation:

$$\widetilde{\mathbf{Q}}_{is} = \mathcal{A}(\mathbf{Q}_{is}, \mathbf{H}_e, \mathbf{H}_e), \quad \widetilde{\mathbf{Q}}_{io} = \mathcal{A}(\mathbf{Q}_{io}, \mathbf{H}_e, \mathbf{H}_e) \quad (3)$$

Predicate-Indicator Fusion To encode the contextual relation between each predicate query and its entity indicators, we perform a predicate-indicator fusion to calibrate the features of three components in the query. We explicitly fuse the current l-th decoder layer outputs $\widetilde{\mathbf{Q}}_p^l, \widetilde{\mathbf{Q}}_{is}^l, \widetilde{\mathbf{Q}}_{io}^l$ to update each component of as the query for next layer $\mathbf{Q}_p^{l+1}, \mathbf{Q}_{is}^{l+1}, \mathbf{Q}_{io}^{l+1}$. Specifically, we adopt fully connected layers for updating the predicate by fusing entity indicator representations as Eq. 4:

$$\mathbf{Q}_p^{l+1} = \left(\widetilde{\mathbf{Q}}_p^l + \left(\widetilde{\mathbf{Q}}_{is}^l + \widetilde{\mathbf{Q}}_{io}^l\right) \cdot \mathbf{W}_i\right) \cdot \mathbf{W}_p \quad (4)$$

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

Based on the refined predicate queries, we are able to generate the geometric and semantic predictions of the predicate node, as well as the location and category of its associated entity indicator as follows,

$$\mathbf{P}_p = \text{Softmax}(\widetilde{\mathbf{Q}}_p \cdot \mathbf{W}_{cls}^p) \in \mathbb{R}^{N_r \times (\mathcal{C}_p + 1)}, \quad (5)$$

$$\mathbf{B}_p = \sigma(\widetilde{\mathbf{Q}}_p \cdot \mathbf{W}_{reg}^p) = \{(x_c^s, y_c^s, x_c^o, y_c^o)\} \in \mathbb{R}^{N_r \times 4} \quad (6)$$

where $\mathbf{P}_p$ are the class predictions of predicates, and $\mathbf{B}_p = \{(x_c^s, y_c^s, x_c^o, y_c^o)\}$ are the box center coordinates of its subject and object entities. The entity indicators are also translated as location prediction of entities $\mathbf{B}_s, \mathbf{B}_o \in \mathbb{R}^{N_r \times 4}$ and their classification predictions $\mathbf{P}_s, \mathbf{P}_o \in \mathbb{R}^{N_r \times (\mathcal{C}_e + 1)}$, which are similar to the entity generator.

Overall, each predicate decoder layer produces the locations and classifications for all the entity-aware predicate queries. Using the multi-layer structure, the predicate decoder is able to gradually improve the quality of predicate and entity association.

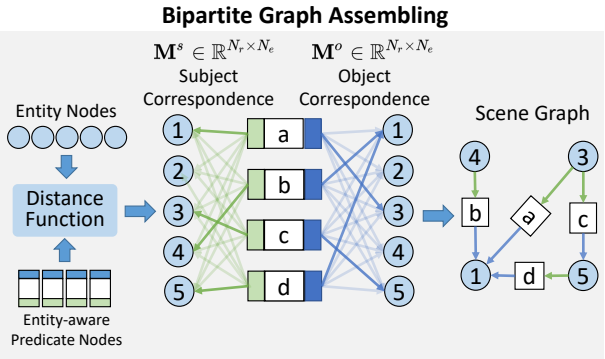

Figure 3. **The illustration of Bipartite Graph Assembling.**

4.3. Bipartite Graph Assembling

In our formulation, 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 graph assembling is to link the entity-aware predicate nodes to the proper entity node.

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 $\mathbf{M} \in \mathbb{R}^{N_r \times N_e}$. Concretely, we define the correspondence matrix by the distance between the entity indicators of predicate nodes and the entity nodes. Taking the subject entity indicator as example, we have: $\mathbf{M}^s = d_{loc}(\mathbf{B}_s, \mathbf{B}_e) \cdot d_{cls}(\mathbf{P}_s, \mathbf{P}_e)$, where $d_{loc}(\cdot)$ and $d_{cls}(\cdot)$ are the distance function to measure the matching quality from different dimensions[4]. The correspondence matrix of object entity $\mathbf{M}^o \in \mathbb{R}^{N_r \times N_e}$ is obtained following the same strategy. The empirical analysis of different distance metrics will be discussed in the experiment section. Based on the correspondence matrix, we keep the top-K links according to the matching scores as the edge links for each predicate node:

$$\mathbf{R}^s = \mathcal{F}_{top}(\mathbf{M}^s, K) \in \mathbb{R}^{N_r \times K} \quad (7)$$

$$\mathbf{R}^o = \mathcal{F}_{top}(\mathbf{M}^o, K) \in \mathbb{R}^{N_r \times K} \quad (8)$$

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

Using the index matrix $\mathbf{R}^s$ and $\mathbf{R}^o$, we are able to generate the final relationship triplets as $\mathcal{T} = \{(\mathbf{b}_e^s, \mathbf{p}_e^s, \mathbf{b}_e^o, \mathbf{p}_e^o, \mathbf{p}_p, \mathbf{b}_p)\}$. Here $\mathbf{b}_e^s, \mathbf{b}_e^o \in \mathbb{R}^{1 \times 4}$ and $\mathbf{p}_e^s, \mathbf{p}_e^o \in \mathbb{R}^{1 \times (\mathcal{C}_e + 1)}$ are bounding boxes and class predictions of its subject and object entity respectively, $\mathbf{p}_p \in \mathbb{R}^{1 \times (\mathcal{C}_p + 1)}$ is the class prediction of each predicate $\mathbf{P}_p$, and $\mathbf{b}_p \in \mathbf{B}_p$ are the centers of the predicate's entities. In the end, the graph assembling module generates the final scene graph as the output of our SGTR model.

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

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 the entity generator and $\mathcal{L}^{pre}$ for predicate generator. The overall loss function is formulated as:

$$\mathcal{L} = \mathcal{L}^{enc} + \mathcal{L}^{pre}, \quad \mathcal{L}^{pre} = \mathcal{L}^{pre}_{i} + \mathcal{L}^{pre}_{p} \tag{9}$$

As we adopt a DETR-like detector, the $\mathcal{L}^{enc}$ follows a similar form as [2], and the detailed loss equation is reported in the supplementary material. We mainly focus on $\mathcal{L}^{pre}$ in the remaining parts 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 [16]. We then convert the ground-truth of the visual relationships into a set of triplet representations in as similar form as $\mathcal{T}$, denoted as $\mathcal{T}^{gt}$. The cost of the set matching is defined as:

$$\mathcal{C} = \lambda_p \mathcal{C}_p + \lambda_e \mathcal{C}_e \tag{10}$$

The two components in the total cost correspond to the costs of predicate and subject/object entity, respectively [5]. The matching index $\mathbf{I}^{tri}$ between triplet predictions and ground truths is produced by: $\mathbf{I}^{tri} = \text{argmin}_{\mathcal{T},\mathcal{T}^{gt}}\mathcal{C}$, which is used for following loss calculation of predicate node generator.

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

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-processing operation to filter out 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 denote the output as $\mathcal{S}^t = \{(s^t_s \cdot s^t_o \cdot s^t_p)\}$, where s^t_s, s^t_o and s^t_p 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 [17] and Visual Genome [15]. We mainly adopt the data splits and evaluation metrics from the previous work [18, 38, 51]. For the Openimage benchmark, the weighted evaluation metrics (wmAP$_{phr}$, wmAP$_{rel}$, score$_{wtd}$) are used for more class-balanced evaluation. For the 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 [18].

We use the ResNet-101 and DETR [2] as backbone networks and entity detector, respectively. To speedup training convergence, we first train entity detector on the target dataset, followed by joint training with predicate node generator. The predicate node generator uses 3 layers of transformer encoder for predicate encoders and 6 layers of transformer decoder for predicate and entity indicator sub-decoders, whose hidden dimensions d is 256. Our predicate decoder uses N_r=150 queries. We set K=40 in training and K=3 during test for graph assembling module. For more implementation details please refer to the supplementary.

[5] We utilize the location and classification predictions to calculate cost for each component. Detailed formulations are presented in supplementary.

#	EPN	SPD	GA	mR@50	mR@100	R@50	R@100
1	✓	✓	✓	**13.9**	**17.3**	**24.2**	**28.2**
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.

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.

• 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 margin 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 shown in line-5 demonstrate that the model struggles to tackle such complex multi-tasks within a single structure, while proposed entity-prediction association modeling and graph assembling 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 [3, 12], as shown

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-Net [3]; F: feature similarity-based matching function proposed by HOTR [12].

in Tab. 3. Comparison experiments are conducted on the validation set of Visual Genome by using different distance functions for the assembling module. In AS-Net [3], the grouping 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 [12] 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 [3] similarity and feature-based [12] 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 [12] similarity design has a slower and more unstable convergence process.

Model Size To investigate the model complexity of the structural predicate node decoder, we incrementally vary the number of layers L in the 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 performance will be saturated when $L = 12$.

Entity Detector As we adopt different entity detectors compared to previous two-stage designs, we conduct experiments to analyze the influence of detectors on the SGTR. The detailed results are presented in the supplementary.

5.3. Comparisons with State-of-the-Art Methods

We conduct experiments on Openimage-V6 benchmark and VG dataset to demonstrate the effectiveness of our design. We compare our method with several state-of-the-art two-stage(*e.g.*, VCTree-PCPL, VCTree-DLFE, BGNN [18], VCTree-TDE, DT2-ACBS [5]) 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 ResNet-101 backbone for more fair comparisons. Furthermore, since FCSGG [23] is the only published one-stage method for SGG, we reproduce the result of several strong one-stage HOI methods with similar entity-predicate pairing mechanisms (AS-Net [3], HOTR [12]) using their released code for a more comprehensive comparison.

B	Models	mR@50	R@50	wmAP		score$_{wtd}$
				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 [18]. * means using resampling strategy.

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 the two-stage SOTA method BGNN with an improvement of **2.28**. Specifically, our design has a significant improvement on weighted mAP metrics of relationship detection (wmAP$_{rel}$) and phrase detection (wmAP$_{phr}$) 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.

Visual Genome As shown in Tab. 5, with the same ResNet-101 backbone, we compare our method with the two-stage method BGNN [18], and the one-stage methods HOTR [12], AS-Net [3]. It shows that our method outperforms HOTR with a significant margin of **4.9** and **3.2** on mRecall@100. Furthermore, our method achieves considerable improvement when compared with the two-stage methods, and the detailed performance is presented in the supplementary.

• 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, when equipped with the same backbone and learning strategy as before, our method achieves competitive performance in mean recall. We also list several newly proposed works, which develops various training strategies for long-tailed recognition. Our method achieves higher mR@100 performance with less overall performance drop when using the resampling strategy proposed in [18]. We refer the reader to the supplementary for more experiments on our model using advanced long-tail training strategies.

• We find that the performance of our model in the head

B	D	Method	mR@50/100	R@50/100	Head	Body	Tail	Time/Sec
⋆	⋆	FCSGG [23]	3.6 / 4.2	21.3 / 25.1	-	-	-	0.12
X101-FPN	Faster-RCNN	RelDN [18]	6.0 / 7.3	31.4 / 35.9	-	-	-	0.65
		Motifs [29]	5.5 / 6.8	**32.1 / 36.9**	-	-	-	1.00
		VCTree [29]	6.6 / 7.7	31.8 / 36.1	-	-	-	1.69
		BGNN*† [18]	10.7 / 12.6	31.0 / 35.8	34.0	12.9	6.0	1.32
		VCTree-TDE [29]	9.3 / 11.1	19.4 / 23.2	-	-	-	≥1.69
		VCTree-DLFE [4]	11.8 / 13.8	22.7 / 26.3	-	-	-	≥1.69
		VCTree-EBM [26]	9.7 / 11.6	20.5 / 24.7	-	-	-	≥1.69
		VCTree-BPLSA [7]	13.5 / 15.7	21.7 / 25.5	-	-	-	≥1.69
		DT2-ACBS [5]	**22.0 / 24.4**	15.0 / 16.3	-	-	-	∼0.63
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† [3]	6.12 / 7.2	18.7 / 21.1	19.6	7.7	2.7	0.33
		HOTR† [12]	9.4 / 12.0	23.5 / 27.7	26.1	16.2	3.4	0.25
		Ours$^{\diamond}$	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 [18] is applied for this model. $\diamond$ denotes that our model uses $K = 1$ for top-K matching in graph assembling (more ablative experiments for K are presented in the supplementary). $\star$ denotes the special backbone HRNetW48-5S-FPN$_{\times 2\text{-f}}$ and entities detector, CenterNet [53].

category is lower than the two-stage methods with the same backbone. The main reason is that the DETR detector performs weaker on small entities than the traditional Faster-RCNN. Since the visual genome has a large proportion of relationships involving small objects, our method performs sub-optimal in recognizing those relationships. The detailed limitation analysis is presented in the supplementary.

• We compare the efficiency of SGTR with previous methods according to the inference time (seconds/image) on the NVIDIA GeForce Titan XP GPU with a batch size of 1 and an input size of 600 x 1000. Our design obtains comparable inference time as the one-stage methods using the same backbone, which demonstrates the efficiency of our method.

5.4. Qualitative Results

As shown in Fig. 4, we visualize the attention weight of the predicates sub-decoder and entity sub-decoder on images from the validation set of the Visual Genome dataset. By comparing the heatmaps in Fig. 4 (a) and Fig. 4 (b), we note that for the same triplet prediction, the predicate sub-decoder focuses more on the contextual regions around the entities of triplets while the entity sub-decoders put more attention on the entity regions. Therefore, our design allows the model to learn the compositional property of visual relationships more effectively, which improves prediction accuracy. More visualization results are reported in the supplementary (including analysis of graph assembling, comparison between two-stage methods, etc.).

6. Conclusions

In this work, we propose a novel end-to-end CNN-Transformer-based scene graph generating approach (SGTR).

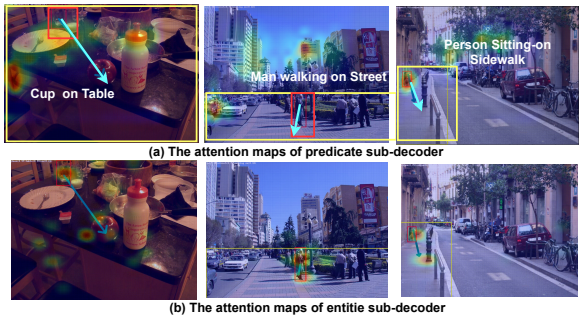

Figure 4. **The visualization on attention heatmap of structural predicate decoder.** The predicate sub-decoder focus on contextual representation around the entities of triplets. Entity indicator sub-decoders focus on relationship-based entity regions.

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 edge connection. We develop the entity-aware representation for modeling the predicate nodes, which is integrated with the entity indicators by the structural predicate node decoder. Finally, the scene graph is constructed by the graph assembling module in an end-to-end manner. 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.

Potential Negative Societal Impact One possible negative impact is that SGG may serve as a base module for surveillance abuse and collecting private information.

References

[1] 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*, pages 15921–15930, 2021. 1, 2

[2] 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*, pages 213–229. Springer, 2020. 2, 3, 4, 6

[3] 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*, pages 9004–9013, 2021. 2, 6, 7, 8

[4] 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. 1, 2, 8

[5] 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. 1, 2, 7, 8

[6] 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*, pages 3550–3559, 2021. 1, 2

[7] 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*, pages 16383–16392, 2021. 1, 2, 8

[8] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020. 1

[9] 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*, pages 3668–3678, 2015. 1

[10] Siddhesh Khandelwal, Mohammed Suhail, and Leonid Sigal. Segmentation-grounded scene graph generation. *arXiv preprint arXiv:2104.14207*, 2021. 2

[11] 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*, pages 498–514. Springer, 2020. 2

[12] 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*, pages 74–83, 2021. 2, 6, 7, 8

[13] 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. 2

[14] 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*, pages 15827–15837, 2021. 1, 2

[15] 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. 6

[16] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[17] 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. 6

[18] 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*, pages 11109–11119, 2021. 1, 2, 3, 6, 7, 8

[19] 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)*, pages 335–351, 2018. 2

[20] 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*, pages 1261–1270, 2017. 2

[21] 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*, pages 482–490, 2020. 2

[22] 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*, pages 3746–3753, 2020. 2

[23] 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*, pages 11546–11556, 2021. 1, 2, 7, 8

[24] 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 Con-*

ference on Computer Vision and Pattern Recognition, pages 3957–3966, 2019. 2

[25] 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*, pages 8376–8384, 2019. 1

[26] 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*, pages 13936–13945, 2021. 2, 8

[27] 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*, pages 14454–14463, 2021. 2

[28] 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*, pages 10410–10419, 2021. 2

[29] 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*, pages 3716–3725, 2020. 2, 8

[30] 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*, pages 6619–6628, 2019. 2

[31] 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*, pages 1–9, 2017. 1

[32] Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. *arXiv preprint arXiv:2106.10815*, 2021. 2

[33] 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*, pages 9627–9636, 2019. 2

[34] 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*, pages 4116–4125, 2020. 2

[35] 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*, 2020. 2

[36] 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*, pages 8188–8197, 2019. 2

[37] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In *Advances in Neural Information Processing Systems*, pages 560–570, 2018. 2

[38] 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)*, pages 5410–5419, 2017. 2, 6

[39] 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*, pages 265–273, 2020. 2

[40] 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*, pages 12527–12536, 2021. 1, 2

[41] 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)*, pages 670–685, 2018. 2

[42] Xuewen Yang, Yingru Liu, and Xin Wang. Reformer: The relational transformer for image captioning. *arXiv preprint arXiv:2107.14178*, 2021. 1

[43] 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*, pages 10685–10694, 2019. 1, 2

[44] 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*, 2021. 1, 2

[45] 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*, 2021. 4

[46] 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)*, pages 322–338, 2018. 2

[47] Cong Yuren, Hanno Ackermann, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. Nodis: Neural ordinary differential scene understanding. *arXiv preprint arXiv:2001.04735*, 2020. 2

[48] 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)*, 2020. 2

[49] 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*, pages 3736–3745, 2020. 2

[50] 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)*, 2020. 2

[51] 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*, pages 5831–5840, 2018. 2, 6

[52] 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. 2

[53] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 8

[54] 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*, pages 11825–11834, 2021. 2