# HYDRA-SGG: HYBRID RELATION ASSIGNMENT FOR ONE-STAGE SCENE GRAPH GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

DETR introduces a simplified one-stage framework for scene graph generation (SGG) but faces challenges of sparse supervision and false negative samples. The former occurs because each image typically contains fewer than 10 relation annotations, while DETR-based SGG models employ over 100 relation queries. Each ground truth relation is assigned to only one query during training. The latter arises when one ground truth relation may have multiple queries with similar matching scores, leading to suboptimally matched queries being treated as negative samples. To address these, we propose Hydra-SGG, a one-stage SGG method featuring a Hybrid Relation Assignment. This approach combines a One-to-One Relation Assignment with an IoU-based One-to-Many Relation Assignment, increasing positive training samples and mitigating sparse supervision. In addition, we empirically demonstrate that removing self-attention between relation queries leads to duplicate predictions, which actually benefits the proposed One-to-Many Relation Assignment. With this insight, we introduce Hydra Branch, an auxiliary decoder without self-attention layers, to further enhance One-to-Many Relation Assignment by promoting different queries to make the same relation prediction. Hydra-SGG achieves state-of-the-art performance on multiple datasets, including VG150 (**16.0** mR@50), Open Images V6 (**50.1** weighted score), and GQA (**12.7** mR@50). Our code and pre-trained models will be released.

## 1 INTRODUCTION

A scene graph is a data structure that describes the entities (objects) in a scene and the relations between these objects [3]. Scene graph generation (SGG) has attracted significant research attention [19, 46, 3, 40] due to its ability to enhance machines' semantic comprehension of visual content. It has been widely adopted in various downstream applications, such as robotic vision and interaction [44, 54, 72, 23], image synthesis and manipulation [71, 9, 21], visual question answering [20, 60, 38], and video understanding [77, 70, 62, 68].

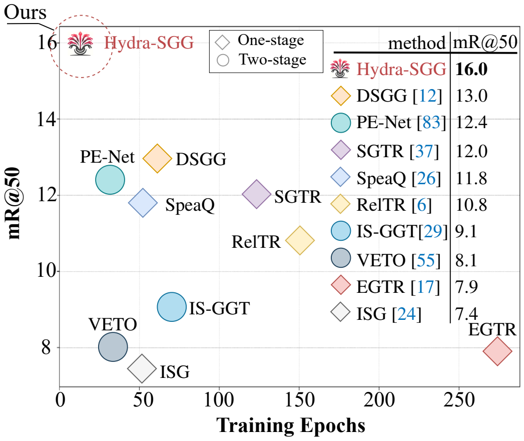


Figure 1: Comparison with other SGG methods in mR@50 and training epochs on VG150 [67].

Mainstream SGG methods [76, 58, 46, 83, 55, 29] work in a *two-stage* fashion. First, an off-the-shelf object detector extracts all entities within an image. Then, the extracted entities are permuted, yielding $N(N-1)$ entity pairs for $N$ detected entities. These entity pairs are used to predict the relationships between the corresponding entities. However, the two-stage methods face a critical limitation: they predict relations for all entity pairs, even though many pairs do not participate in any relations. This incurs heavy computational overhead and time-consuming inference, especially in complex scenes with numerous entities and intricate interactions.

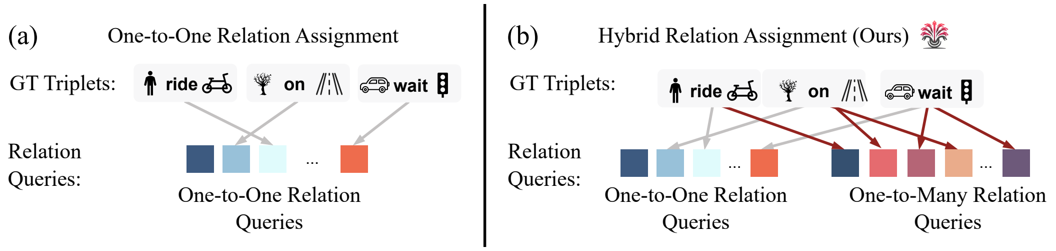

(a)
One-to-One Relation Assignment
GT Triplets:
ride
on
wait
Relation Queries:
...
One-to-One Relation Queries
(b)
Hybrid Relation Assignment (Ours)
GT Triplets:
ride
on
wait
Relation Queries:
...
One-to-One Relation Queries
...
One-to-Many Relation Queries

Figure 2: (a) Previous DETR-based SGG methods such as RelTR [6] and SGTR [37] match each GT relation with only one query. (b) Our Hybrid Relation Assignment utilizes both One-to-One and One-to-Many assignments, generating more positive samples and thus accelerating training.

Recently, the simplicity of DETR-based SGG methods [6, 37] has led to an ongoing paradigm shift apart from the two-stage framework. Specifically, a sequence of visual tokens, mapped from the input image, interacts with a predefined set of relation queries in a Transformer decoder for simultaneous object detection and relation prediction. Then, One-to-One set matching (*i.e.*, Hungarian matching [28]) is used to assign ground truth labels to the predictions (Fig. 2a). This set-prediction framework allows DETR-based SGG models to eliminate hand-designed components, such as Non-Maximum Suppression (NMS), which are commonly used in traditional two-stage methods.

Unfortunately, one significant drawback of DETR-based SGG models is their slow convergence. For instance, one-stage RelTR [6] requires 150 training epochs to converge, while a recent two-stage method, PE-Net [83], only needs 32 epochs. This drawback can be attributed to the sparse relation supervision induced by Hungarian matching, which assigns each ground truth to **only one** relation query. The sparsity of relation annotations per image further exacerbates this issue. For instance, VG150 [67] `train` averages only 5.5 ground truth relation triplets per image [40]. This means that each image provides a mere 5.5 positive relation queries to optimize the loss functions. In the case of RelTR [6], which has 200 predefined relation queries, only 2.75% of these queries are positive samples per optimization step. Consequently, relation queries require more optimization steps to learn due to the limited number of positive samples for training. Furthermore, approximately 50% [26] of the plausible but suboptimally matched queries (*i.e.*, queries with correct subject-object classification and IoU $>$ 0.6 for both boxes) are simply treated as `no-relation` due to the One-to-One constraint of Hungarian matching. This constraint discards valuable supervisory signals by treating suboptimal yet informative queries as negative samples. These queries, while not the best matches for ground truth labels, may capture informative relational cues that could contribute to model learning. Simply assigning these queries as negative samples may introduce false negatives, which in turn leads to label noise and performance degradation [42].

To improve the training efficacy of DETR-based SGG models, we introduce Hydra-SGG[1], a comprehensive framework that addresses the slow convergence problem in one-stage DETR-based SGG models. The cores of Hydra-SGG are Hybrid Relation Assignment and Hydra Branch. Specifically, Hybrid Relation Assignment synergizes One-to-One and One-to-Many Relation Assignment strategies, providing over **50%** more positive queries per training step than previous arts such as RelTR [6]. To further enhance this assignment strategy, we introduce an auxiliary branch called Hydra Branch. This branch is specifically designed to encourage different queries to predict duplicated relations by removing self-attention in the decoder, creating a synergistic effect with our Hybrid Relation Assignment. The branch shares all other parameters with the original decoder, and this intentional design choice leads to improved supervision signals during training (§3.3).

Hydra-SGG makes three main contributions to the field of SGG: **First**, we propose a comprehensive framework that effectively addresses the slow convergence problem in one-stage DETR-based SGG models through a hybrid query-label assignment and synergistic architectural design. **Second**, our Hybrid Relation Assignment strategy significantly increases relation supervision signals by mining false negative samples to increase positive samples during training, providing over 50% more positive samples per training step. **Third**, the proposed Hydra Branch complements our assignment strategy by encouraging duplicate relation predictions during training, while maintaining

[1] Hydra is an abbreviation for "**Hy**brid **R**elation **A**ssignment". The name is chosen because it reflects the multi-branch structure of our model, which resembles the multiple heads of the Hydra in Greek mythology.

inference efficiency as it shares parameters with the original decoder and is only used during training. Hydra-SGG achieves state-of-the-art performance with remarkable training efficiency, converging 10× faster than existing one-stage SGG counterparts [6, 37, 17].

Extensive experiments on three challenging SGG benchmarks, VG150 [67], Open Images V6 [30], and GQA [15], demonstrate the effectiveness of our Hydra-SGG. It achieves **16.0** mR@50 on VG150 `test` in only **12** epochs (Fig. 1), surpassing the previous state-of-the-art one-stage method SGTR [37] by **+4.0** and two-stage method PE-Net [83] by **+3.6**. The code will be released.

## 2 Related Work

**Two-stage SGG.** Lu *et al*. [46] first proposed SGG task and designed a *two-stage* pipeline. Later, many SGG models have been built on it, using various techniques such as recurrent neural networks or its variants [76, 58, 67, 64], visual translation embedding [80, 16], graph neural networks [74, 56, 66, 74, 69], external or internal knowledge integration [5, 11, 75, 1, 39, 73], attention mechanisms [22, 49, 8], and Transformer-based models [47, 6, 37, 55, 7, 29, 53].

Despite the promising performance, two-stage SGG methods rely heavily on manually designed modules, such as NMS, anchor generation, and entity pairing generation modules [76, 58, 8]. In addition, their designs typically involve separate stages for detection, pairing, and relation classification, resulting in a complicated pipeline that cannot be trained in a fully end-to-end manner. Furthermore, these methods predict dense entity pairs in inference, leading to high time complexity and computational burden. In contrast, this paper proposes Hydra-SGG, a one-stage SGG method that offers significant advantages over two-stage methods. Hydra-SGG enables end-to-end training and surpasses the performance of two-stage methods.

**One-stage SGG.** *One-stage* methods have gained increasing attention for their simplicity and end-to-end training ability. Early works in this line adopt CNN-based one-stage detectors [43] or query-based sparse R-CNN [61] for direct relationship prediction. Recently, the DETR [2] framework significantly advances SGG and Human-Object Interaction [6, 37, 57, 86, 41, 25, 78, 26, 17]. This framework enables end-to-end training by associating ground truth labels with output queries. Existing DETR-based SGG methods focus on different aspects: SGTR [37] and DSGG [12] explore query designs for relation feature extraction, SpeaQ [26] investigates relation-specific query grouping strategies to improve the specialization and discrimination of queries, while other works focus on architectural innovations [6] and lightweight frameworks [17].

However, DETR-based SGG models, while eliminating NMS, face a critical challenge of slow convergence due to sparse relation supervision, requiring significantly more training epochs than two-stage counterparts (*e.g*., 150 for RelTR [6] vs. 32 for PE-Net [83]). Our proposed Hydra-SGG specifically addresses this limitation through a query-label assignment and architectural design while maintaining the advantages of DETR-based approaches. Hydra-SGG achieves remarkably fast convergence with only 12 epochs, surpassing both one-stage counterparts and two-stage methods to achieve state-of-the-art performance across multiple SGG datasets (16.0 mR@50 on VG150, 50.1 weighted score on Open Images V6, and 12.7 mR@50 on GQA).

**DETR for Object Detection.** The introduction of DETR by Carion *et al*. [2] marks a significant shift away from traditional CNN-based object detection models [52, 50, 51, 42, 13], adopting an end-to-end trainable approach with a novel application of the Transformer architecture and bipartite matching. While DETR streamlines the detection process, it requires 500 epochs to converge [84]. This spurs a wave of research that focuses on improving low training efficacy, including enhancing training signal [4, 14, 18, 85], the adoption of anchor boxes [45, 48, 65], and the leverage of efficient attention mechanism [84, 10]. For instance, Co-DETR [85] introduces extra groups of object queries and DN-DETR [31] utilizes noisy queries to increase positive samples.

In this paper, we present Hydra-SGG, a framework that addresses the sparse relation supervision challenge inherent in DETR-based SGG models. Our method introduces Hybrid Relation Assignment, which significantly increases the number of positive samples per training iteration, effectively mitigating the sparse relation supervision issue and accelerating the model's convergence.

# 3 HYDRA-SGG

§3.1 introduces our baseline model, which employs One-to-One Relation Assignment. Next, §3.2 describes Vanilla Hydra-SGG, which enhances the baseline by introducing a Hybrid Relation Assignment strategy. Finally, §3.3 details the complete Hydra-SGG model, which includes Hydra Branch (`HydraBranch`), an auxiliary decoder that promotes One-to-Many Relation Assignment. §3.4 performs a statistical analysis to validate our design of Hydra Branch, demonstrating its effectiveness in enhancing the One-to-Many assignment strategy.

## 3.1 ONE-TO-ONE RELATION ASSIGNMENT BASELINE

**Problem Formulation.** Given an input image, SGG models aim to generate a scene graph in the form of relation triplet: $\langle e_{\text{sub}}, \rho, e_{\text{obj}} \rangle$, where each entity $e_{\text{sub}}, e_{\text{obj}} \in \mathcal{E}$ is represented by a category label $c \in \mathcal{C}$ (*e.g.*, `cat`, `people`, `car`) and a bounding box $b$, and $\rho \in \mathcal{P}$ is a specific relation type (*e.g.*, `on`, `have`, `ride`). In this paper, we distinguish between the terms "relation" and "relation triplet". A relation refers to the interaction or relationship between entities, while a relation triplet represents the complete structure containing the subject $e_{\text{sub}}$, object $e_{\text{obj}}$, and their relation $\rho$.

**Baseline Architecture.** Our baseline model is composed of a `Backbone` model, an `Encoder`, and a `RelDecoder` (Relation Decoder). The relation queries interact with image tokens extracted from the input image by `Backbone` and `Encoder`. The updated queries are processed by box regression, entity classification, and relation classification heads to generate the final predictions.

- **`Backbone`:** `Backbone` maps an input image into a feature map that has a $H \times W \times C$ dimension, where $H$ and $W$ denote the spatial size of the feature map, while $C$ is the channel dimension (*e.g.*, 2048 or 1024).
- **`Encoder`:** `Encoder` captures comprehensive spatial and contextual information across the image. Before feeding the feature map into `Encoder`, a 1×1 conv layer is employed to reduce the dimension of the feature map. The enhanced image tokens are denoted as $\boldsymbol{F} \in \mathbb{R}^{HW \times 256}$.
- **`RelDecoder`:** In `RelDecoder`, we introduce relation queries $\boldsymbol{Q}_{\text{rel}} \in \mathbb{R}^{N \times 512}$ (🟦🟦🟦⬜···🟥), which are formed by concatenating subject queries $\boldsymbol{Q}_{\text{sub}} \in \mathbb{R}^{N \times 256}$ and object queries $\boldsymbol{Q}_{\text{obj}} \in \mathbb{R}^{N \times 256}$ along the channel dimension (*i.e.*, 256). Here $N$ denotes the number of queries. `RelDecoder`$(\boldsymbol{Q}_{\text{sub}}, \boldsymbol{Q}_{\text{obj}}, \boldsymbol{F})$ works as follows:

$$\begin{aligned} \tilde{\boldsymbol{Q}}_{\text{sub}}, \tilde{\boldsymbol{Q}}_{\text{obj}} &= \texttt{SA}([\boldsymbol{Q}_{\text{sub}}, \boldsymbol{Q}_{\text{obj}}]) \in \mathbb{R}^{2N \times 256}, \\ \bar{\boldsymbol{Q}}_{\text{sub}} &= \texttt{CA}(\boldsymbol{F}, \tilde{\boldsymbol{Q}}_{\text{sub}}) \in \mathbb{R}^{N \times 256}, \\ \bar{\boldsymbol{Q}}_{\text{obj}} &= \texttt{CA}(\boldsymbol{F}, \tilde{\boldsymbol{Q}}_{\text{obj}}) \in \mathbb{R}^{N \times 256}. \end{aligned} \tag{1}$$

  Specifically, the subject queries $\boldsymbol{Q}_{\text{sub}}$ and object queries $\boldsymbol{Q}_{\text{obj}}$ first interact in a self-attention layer (`SA`) to obtain updated queries $\tilde{\boldsymbol{Q}}_{\text{sub}}$ and $\tilde{\boldsymbol{Q}}_{\text{obj}}$. Subsequently, these updated queries interact with image tokens in cross-attention layers (`CA`). The output subject and object queries are fed into independent box regression heads and classification heads, producing box predictions $\bar{b}_{\text{sub}}, \bar{b}_{\text{obj}} \in \mathbb{R}^{N \times 4}$ and entity class predictions $\bar{p}_{\text{sub}}, \bar{p}_{\text{obj}} \in \mathbb{R}^{N \times |\mathcal{C}|}$, respectively. The relation predictions $\bar{p}_{\text{rel}} \in \mathbb{R}^{N \times |\mathcal{P}|}$ are obtained by a relation prediction head. For example, with 300 relation queries (*i.e.*, $N = 300$) and 50 relation classes (*i.e.*, $|\mathcal{P}| = 50$), the output dimension of $\bar{p}_{\text{rel}}$ would be 300 × 50. Finally, the outputs of `RelDecoder` are combined to generate the predicted relation triplets $\bar{\boldsymbol{R}} = \{\langle(\bar{p}_{\text{sub}}, \bar{b}_{\text{sub}}), \bar{p}_{\text{rel}}, (\bar{p}_{\text{obj}}, \bar{b}_{\text{obj}})\rangle_n\}_{n=1}^{N}$. Note that the $i$-th subject query is concatenated only with the $i$-th object query, and there is no permutation process.
- **One-to-One Relation Assignment Loss.** In One-to-One Relation Assignment Loss $\mathcal{L}_{\text{o2o}}$ [6, 37], Hungarian matching (`HM`) is employed to find the optimal correspondence between the predicted relation triplets $\bar{\boldsymbol{R}}$ and the ground truth triplets $\boldsymbol{R}$. The loss function can be formulated as:

$$\mathcal{L}_{\text{o2o}} = \mathcal{L}_{\texttt{HM}}(\bar{\boldsymbol{R}}, \boldsymbol{R}). \tag{2}$$

  The classification losses (including entity and relation) and box regression losses are computed between matched predictions and ground truth labels, which is identical with the previous models [6, 37]. The Hungarian matching hyperparameters are the same as RelTR [6]. A detailed

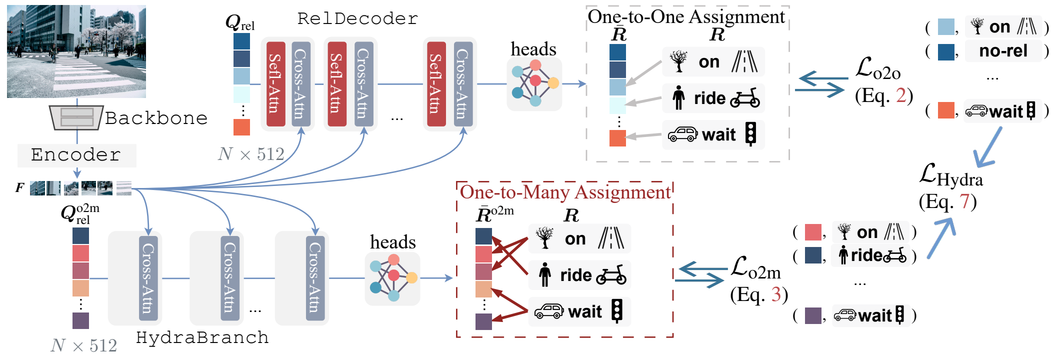


Figure 3: Overall pipeline of Hydra-SGG: For simplicity, FFN inside the Transformer layer are omitted. Hydra-SGG incorporates two Transformer decoders: `HydraBranch` and `RelDecoder`. `HydraBranch` shares its parameters with `RelDecoder` but removes self-attention layers. Hydra-SGG combines One-to-One and One-to-Many assignments in a synergy, generating more supervision signals.

implementation of the loss function, training strategy, and model architecture is provided in Appendix. The evaluation of our baseline model is given in §4.3.

### 3.2 Vanilla Hybrid Relation Assignment

VG150 [67] contains an average of only 5.5 ground truth relation triplets per image in `train`. Our baseline adopts a One-to-One Relation Assignment and only assigns approximately 5 out of 300 relation queries to match the ground truth relation triplets for each image (*i.e.*, $N = 300$). This severe sparse supervision reduces the training efficacy, as only about 2% of queries in each training step match with ground truth labels for learning, while the remaining 98% are treated as negative samples. Consequently, the model requires a significantly higher number of training steps to learn effectively, leading to slow convergence.

To enrich the relation supervision signals, we propose a Hybrid Relation Assignment. Specifically, we first devise a One-to-Many Relation Assignment (`o2m`) and then embed it into the baseline to cooperate with One-to-One Relation Assignment. Given a ground truth triplet $\boldsymbol{r} = \langle (c_{\text{sub}}, b_{\text{sub}}),\ c_{\text{rel}}, (c_{\text{obj}}, b_{\text{obj}}) \rangle \in \boldsymbol{R}$ and a predicted triplet $\bar{\boldsymbol{r}} = \langle (\bar{p}_{\text{sub}},\ \bar{b}_{\text{sub}}),\ \bar{p}_{\text{rel}}, (\bar{p}_{\text{obj}},\ \bar{b}_{\text{obj}}) \rangle \in \bar{\boldsymbol{R}}$, the One-to-Many Assignment score $\mathcal{S}_{\texttt{o2m}}$ is calculated by:

$$\mathcal{S}_{\texttt{o2m}}(\boldsymbol{r}, \bar{\boldsymbol{r}}) = \bar{p}_{\text{sub}}^{[c_{\text{sub}}]} + \bar{p}_{\text{obj}}^{[c_{\text{obj}}]} + \text{IoU}(b_{\text{sub}}, \bar{b}_{\text{sub}}) + \text{IoU}(b_{\text{obj}}, \bar{b}_{\text{obj}}). \tag{3}$$

$\mathcal{S}_{\texttt{o2m}}$ combines the predicted class probabilities and IoU scores for the subject and object bounding boxes. The notations $\bar{p}_{\text{sub}}^{[c_{\text{sub}}]}$ and $\bar{p}_{\text{obj}}^{[c_{\text{obj}}]}$ represent the probabilities of the ground truth classes for the subject and object, respectively. For example, if the ground truth subject class $c_{\text{sub}}$ is 3, we would extract the probability corresponding to class 3 from $\bar{p}_{\text{sub}}$.

Given $M$ ground truth relation triplets and $N$ predicted relation triplets, the final One-to-Many Relation Assignment score matrix, with dimensions $M \times N$, is computed by applying Eq. 3 to each pair of ground truth and predicted relation triplets. We keep the results with scores greater than a threshold $T$ and select the top 6 queries for each ground truth (see details in §4.3).

The proposed Vanilla Hydra-SGG applies both One-to-One and One-to-Many Relation Assignment strategies to the same set of predicted relation triplets $\bar{\boldsymbol{R}}$. These triplets are obtained from the updated queries $\boldsymbol{Q}_{\text{rel}}$ after passing through the model. By combining these two assignment strategies, we can formulate the vanilla version loss function as:

$$\mathcal{L}_{\text{vanilla}} = \mathcal{L}_{\text{o2o}}(\bar{\boldsymbol{R}}, \boldsymbol{R}) + \mathcal{L}_{\text{o2m}}(\bar{\boldsymbol{R}}, \boldsymbol{R}). \tag{4}$$

Our Vanilla Hydra-SGG provides richer relation supervision signals compared to the baseline by harmonizing the supervision of two assignment strategies. Specifically, it increases the number of positive samples by 65.5% in VG150 [67] `train` and 58.7% in `val` (Fig. 4a, b).Vanilla Hydra-SGG significantly improves training efficacy and performance (§4.3).

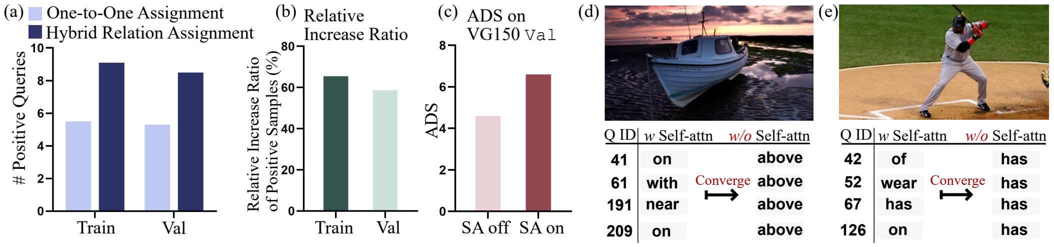


Figure 4: (a) The average number of positive samples of VG150 `train` and `val` for One-to-One and Hybrid Relation Assignment. (b) The percentage increase in positive samples achieved by Hybrid Relation Assignment compared to the One-to-One baseline. (c) ADS on VG150 `val`. (d)-(e) The visualizations show that for the same group of queries that previously predicted different relations, removing the self-attention layers causes them to make identical predictions. The Q ID column represents the ID of each relation query.

### 3.3 Complete Hydra-SGG

Each `RelDecoder` layer contains self- and cross-attention layers, and a point-wise feed-forward network (FFN). The self-attention layer enables **inter-query** interactions [63], while cross-attention and FFN do not explicitly support query interactions. Previous studies indicate that self-attention helps inhibit duplicate predictions [2, 6]. In this work, we further find that the self-attention layer in `RelDecoder` is crucial to reduce duplicated relation predictions.

We introduce a Diversity Score (DS) to quantify the impact of the self-attention layer on the diversity of relation predictions. Specifically, DS is defined as the number of distinct relation categories predicted by the model for a single image. For example, if the model predicts relations for 5 queries as (`sit`, `sit`, `on`, `on`, `has`), DS would be 3, as there are 3 distinct relation categories (`sit`, `on`, `has`). We calculate DS using the trained Vanilla Hydra-SGG model, with and without self-attention in the `RelDecoder`. Let $\mathrm{DS}_{\mathrm{on}}^{(k)}$ and $\mathrm{DS}_{\mathrm{off}}^{(k)}$ represent the DS for the $k$-th image with self-attention enabled and disabled, respectively. The average DS (ADS) across the dataset is then computed as:

$$\mathrm{ADS}_{\mathrm{on}} = \frac{1}{K}\sum\nolimits_{k=1}^{K} \mathrm{DS}_{\mathrm{on}}^{(k)}, \quad \mathrm{ADS}_{\mathrm{off}} = \frac{1}{K}\sum\nolimits_{k=1}^{K} \mathrm{DS}_{\mathrm{off}}^{(k)}, \tag{5}$$

where $K$ is the total number of images. A higher $\mathrm{ADS}_{\mathrm{on}}$ compared to $\mathrm{ADS}_{\mathrm{off}}$ would indicate that self-attention promotes diverse relation predictions. As shown in Fig. 4c, $\mathrm{ADS}_{\mathrm{off}}$ and $\mathrm{ADS}_{\mathrm{on}}$ of VG150 [67] `val` are 4.6 and 6.6, respectively. Fig. 4d, e illustrate how the self-attention layer enables diverse relation predictions. Without self-attention, queries converge to the same relation. These examples demonstrate self-attention's role in reducing duplicate relations.

The above findings reveal critical interactions between self-attention and One-to-Many Relation Assignment: **i) Conflict with Self-Attention:** Applying One-to-Many Relation Assignment strategy to self-attention-updated relation queries in `RelDecoder` of Vanilla Hydra-SGG causes a mismatch in optimization objectives, as self-attention promotes diversity while One-to-Many Relation Assignment assigns one ground truth to multiple queries. **ii) Benefit without Self-Attention:** Conversely, removing self-attention leads to more duplicated relation predictions, potentially synergizing with our One-to-Many Relation Assignment.

To address this potential conflict and fully harness the benefits of both the self-attention layers and the One-to-Many Relation Assignment strategy, we propose `HydraBranch` (Hydra Branch, Fig. 3), an auxiliary decoder that shares parameters with `RelDecoder` but removes the self-attention layers. This multi-branch architecture decouples the learning objectives: the self-attention layers in the main `RelDecoder` can focus on promoting diversity in relation predictions, while `HydraBranch` facilitates One-to-Many Relation Assignment. This separation enables each branch to optimize its specific function without compromising the other.

`HydraBranch` operates with the main `RelDecoder` in parallel, processing the same input but without the influence of self-attention layers. Specifically, the initial One-to-Many relation queries $\boldsymbol{Q}_{\mathrm{rel}}^{\mathrm{o2m}}$ (🟦🟥🟥🟧···🟪) are set to $\boldsymbol{Q}_{\mathrm{rel}}$ (*i.e.*, $\boldsymbol{Q}_{\mathrm{rel}}^{\mathrm{o2m}} = \boldsymbol{Q}_{\mathrm{rel}}$) before being sent into `HydraBranch`. The process of $\texttt{HydraBranch}(\boldsymbol{Q}^{\text{o2m}}_{\text{sub}}, \boldsymbol{Q}^{\text{o2m}}_{\text{obj}}, \boldsymbol{F})$ is as follows:

$$\bar{\boldsymbol{Q}}^{\text{o2m}}_{\text{sub}} = \texttt{CA}(\boldsymbol{F}, \boldsymbol{Q}^{\text{o2m}}_{\text{sub}}) \in \mathbb{R}^{N\times 256}, \quad \bar{\boldsymbol{Q}}^{\text{o2m}}_{\text{obj}} = \texttt{CA}(\boldsymbol{F}, \boldsymbol{Q}^{\text{o2m}}_{\text{obj}}) \in \mathbb{R}^{N\times 256}. \tag{6}$$

The One-to-Many predicted relation triplets $\bar{\boldsymbol{R}}^{\text{o2m}} = \{\langle(\bar{p}^{\text{o2m}}_{\text{sub}}, \bar{b}^{\text{o2m}}_{\text{sub}}), \bar{p}^{\text{o2m}}_{\text{rel}}, (\bar{p}^{\text{o2m}}_{\text{obj}}, \bar{b}^{\text{o2m}}_{\text{obj}})\rangle_n\}_{n=1}^{N}$ are derived in the same manner as $\bar{\boldsymbol{R}}$ and share prediction heads with $\bar{\boldsymbol{R}}$. Compared with Eq. 1, subject and object queries do not interact in a self-attention layer before they are sent into subsequent cross-attention layers. We apply One-to-Many Relation Assignment described in Eq. 3 to $\bar{\boldsymbol{R}}^{\text{o2m}}$ and $\boldsymbol{R}$. Hybrid Relation Assignment Loss is then given by:

$$\mathcal{L}_{\text{Hydra}} = \mathcal{L}_{\texttt{o2o}}(\bar{\boldsymbol{R}}, \boldsymbol{R}) + \mathcal{L}_{\texttt{o2m}}(\bar{\boldsymbol{R}}^{\text{o2m}}, \boldsymbol{R}). \tag{7}$$

By incorporating HydraBranch, the complete version of Hydra-SGG achieves **10.6** and **16.0** on mR@20 and mR@50, respectively, in just 12 training epochs, further boosting the performance compared to the vanilla version (§4.3). Note that HydraBranch is used only in training and discarded in inference, thus bringing no extra parameters or delay.

### 3.4 Statistical Analysis of Hydra Branch

We conduct a statistical analysis to investigate how Hydra Branch enhances our One-to-Many assignment strategy through query embeddings. Our study examined the Euclidean distances between query pairs within each image across 5,000 images from VG150 val. For each image containing 300 queries, we computed 44,850 (300×299/2) pairwise distances, comparing scenarios with and without self-attention layers. The analysis reveals that removing self-attention results in a decrease in mean query distance from 7.55 to 7.23 (a 5% reduction). Statistical validation through a paired t-test yielded a t-statistic of 54.502 ($p < 1e\text{-}3$), indicating high statistical significance. Furthermore, Cohen's d effect size of 0.771 approaches the threshold for a large effect (0.8). These results empirically demonstrate that Hydra Branch effectively promotes query similarity by removing self-attention, thereby facilitating our One-to-Many assignment.

## 4 Experiment

### 4.1 Experimental Setup

**Datasets.** We conduct experiments on three datasets:

- **Visual Genome (VG150)** [27, 67] contains 150 entity and 50 relation categories. It is split into 57,723 training, 5,000 validation, and 26,446 testing images.
- **Open Images V6** [30] features 288 entity and 30 relation categories, including 126,368 training, 1,813 validation, and 5,322 testing images with relation annotations.
- **GQA** [15] encompasses 200 entity and 100 relation types, with a split of 52,623 training, 5,000 validation, and 8,209 testing images annotated for SGG tasks.

**Evaluation Metrics.** We focus on the scene graph detection (SGDet) setting on VG150 [27], GQA [15] and Open Images V6 [30] datasets. For VG150 and GQA, we report Recall@$k$ (R@$k$), mean Recall@$k$ [5, 58] (mR@$k$), and F-Recall [79] performance. mR@K calculates R@K for each predicate individually, then averages these values. It is important to note that recall metrics are more influenced by dataset bias [3], whereas mean recall provides a more holistic evaluation of the model's performance. F-Recall is the harmonic average of Recall and mean Recall. For Open Images V6, we follow evaluation protocols [30]: Recall@50, the weighted mean Average Precision (wmAP) for relationship detection (wmAP*rel*), and phrase detection (wmAP*phr*). The overall score, denoted as score$_{wtd}$, is calculated as a weighted average of these metrics: $0.2 \times \text{R@50} + 0.4 \times \text{wmAP}_{rel} + 0.4 \times \text{wmAP}_{phr}$.

**Competitors.** Hydra-SGG is compared with methods from two categories: (1) Two-stage SGG methods, including MOTIFS [76], VCTree-TDE [59], BGNN [36], PE-Net [83], IS-GGT [29], VETO [55], UniVRD [82], DRM [32], CFA [34], and SHA [8]; (2) One-stage methods including SGTR [37], SSR-CNN [61], ISG [24], RelTR [6], DSGG [12], SpeaQ [26], and EGTR [17].

**Reproducibility.** Hydra-SGG is implemented in PyTorch and trained/tested on eight NVIDIA RTX 4090 GPUs, each with 24GB of memory. Our code and trained models will be released.

Table 1: SGDet evaluation on VG150 [27] `test` (§4.2). $^+$: detector pre-trained on VG150. FPS (Frames Per Second) indicates inference speed. F-Recall of Hydra-SGG is calculated based on the best results.

| Method | Backbone | # Epoch | FPS | # Param | R@20/50/100 | mR@20/50/100 | F@20/50/100 |
|---|---|---|---|---|---|---|---|
| *Two-stage methods* | | | | | | | |
| MOTIFS [76] [CVPR2018] | ResNeXt101-FPN | - | - | 369.9M | 25.1 / 32.1 / 36.9 | 4.1 / 5.5 / 6.8 | 7.1 / 9.2 / 11.7 |
| VCTree-TDE [59] [CVPR2020] | ResNeXt101-FPN | - | - | 361.3M | 14.3 / 19.6 / - | 6.3 / 9.3 / 11.1 | 8.8 / 12.4 / - |
| BGNN [36] [CVPR2021] | ResNeXt101-FPN | - | - | 341.9M | 23.3 / 31.0 / 35.8 | 7.5 / 10.7 / 12.7 | 11.3 / 15.5 / 19.0 |
| PE-Net [83] [CVPR2023] | ResNeXt101-FPN | 32$^+$ | - | - | - / 30.7 / 35.2 | - / 12.4 / 14.5 | - / 17.7 / 21.2 |
| IS-GGT [29] [CVPR2023] | ResNet101 | 70 | - | - | - / - / - | - / 9.1 / 11.3 | - / - / - |
| VETO [55] [ICCV2023] | ResNeXt101-FPN | 33$^+$ | - | - | - / 27.5 / 31.5 | - / 8.1 / 9.5 | - / 12.5 / 14.6 |
| UniVRD [82] [ICCV2023] | CLIP ViT-B | - | - | - | - / - / - | - / 9.6 / 12.1 | - / - / - |
| DRM [32] [CVPR2024] | ResNeXt101-FPN | - | - | - | - / 34.0 / 38.9 | - / 9.0 / 11.2 | - / 14.2 / 17.4 |
| *One-stage methods* | | | | | | | |
| SGTR [37] [CVPR2022] | ResNet101 | 123$^+$ | - | 117.1M | - / 25.1 / 26.6 | - / 12.0 / 14.6 | - / 16.2 / 18.9 |
| SSR-CNN [61] [CVPR2022] | ResNet101 | - | - | 274.3M | 25.8 / 32.7 / 36.9 | 6.1 / 8.4 / 10.0 | 9.9 / 13.4 / 15.7 |
| ISG [24] [NeurIPS2022] | ResNet101 | 52 | - | 93.5M | - / 29.5 / 32.1 | - / 7.4 / 8.4 | - / 11.8 / 13.3 |
| RelTR [6] [TPAMI2023] | ResNet50 | 150 | 6.5 | 63.7M | 21.2 / 27.5 / 30.7 | 6.8 / 10.8 / 12.3 | 10.3 / 15.5 / 17.6 |
| DSGG [12] [CVPR2024] | - | 60 | - | - | - / 32.9 / 38.5 | - / 13.0 / 17.3 | - / 18.6 / 23.9 |
| SpeaQ [26] [CVPR2024] | ResNet101 | 52 | - | - | - / 32.9 / 36.0 | - / 11.8 / 14.1 | - / 17.4 / 20.3 |
| EGTR [17] [CVPR2024] | ResNet50 | 275$^+$ | 7.7 | 42.5M | 23.5 / 30.2 / 34.3 | 5.5 / 7.9 / 10.1 | 8.9 / 12.5 / 15.6 |
| *Ours* | | | | | | | |
| Hydra-SGG | ResNet50 | 12 | 5.3 | 67.6M | 21.9 / 28.6 / 33.4<br>±0.1 / ±0.2 / ±0.3 | **10.3 / 15.9 / 19.4**<br>±0.2 / ±0.2 / ±0.2 | **14.0 / 20.5 / 24.7** |

Table 2: Evaluation on Open Images V6 [30] `test` (§4.2). $^+$: detector pre-trained on Open Images V6.

| Method | Backbone | # Epoch | # Param | R@50 | wmAP$_{rel}$ | wmAP$_{phr}$ | score$_{wtd}$ |
|---|---|---|---|---|---|---|---|
| *Two-stage methods* | | | | | | | |
| Motifs [76] [CVPR2018] | ResNeXt101-FPN | - | 369.9M | 71.6 | 29.9 | 31.6 | 38.9 |
| BGNN [36] [CVPR2021] | ResNeXt101-FPN | - | 341.9M | 75.0 | 35.5 | 34.2 | 42.1 |
| PE-Net [83] [CVPR2023] | ResNeXt101-FPN | - | - | 76.5 | 35.4 | 34.9 | 44.9 |
| *One-stage methods* | | | | | | | |
| SGTR [37] [CVPR2022] | ResNet101 | 123$^+$ | 117.1M | 59.9 | 37.0 | 38.7 | 42.3 |
| RelTR [6] [TPAMI2023] | ResNet50 | 150 | 63.7M | 71.7 | 34.2 | 37.5 | 43.0 |
| EGTR [17] [CVPR2024] | ResNet50 | 275$^+$ | 42.5M | 75.0 | 42.0 | 41.9 | 48.6 |
| *Ours* | | | | | | | |
| Hydra-SGG | ResNet50 | 7 | 67.6M | 76.0$_{\pm 0.2}$ | **42.8**$_{\pm 0.2}$ | **44.1**$_{\pm 0.2}$ | **50.0**$_{\pm 0.2}$ |

## 4.2 Quantitative Comparison Result

**VG150** [67] `test`. Table 1 reports the comparison results on VG150 `test`. Hydra-SGG demonstrates outstanding performance on challenging SGDet, achieving mR@20 and mR@50 scores of **10.6** and **16.0**, respectively, setting a new state-of-the-art. Hydra-SGG achieves this performance with significantly shorter training time, requiring only **12** epochs. This training time is substantially shorter compared to SpeaQ [26], EGTR [17], and DSGG [12], with reductions of 40, 263, and 48 epochs, respectively. Despite the shorter training, Hydra-SGG outperforms these methods by **+4.2**, **+8.1**, and **+3.0** on the mR@50. Furthermore, it even outperforms the state-of-the-art two-stage model, PE-Net [83], by a margin of **3.6** on the mR@50 metric. We speculate that the significant improvement in mR can be attributed to our One-to-Many Assignment strategy, which ensures balanced supervision across both rare and common relations by allocating a fixed number of six queries per relation category (§ 3.2). For instance, in an image containing ten "on" relations and two "sit" relations, while One-to-One Assignment would allocate ten queries to "on" and only two to "sit", our approach assigns six queries to each, resulting in a 60% increase for common relations and a substantial 300% increase for rare relations.

Table 3: Evaluation on GQA [15] `test` (§4.2).

| Method | Backbone | R@50/100 | mR@50/100 |
|---|---|---|---|
| SHA [8] [CVPR2022] | ResNeXt101-FPN | 25.5 / 29.1 | 6.6 / 7.8 |
| VETO [55] [ICCV2023] | ResNeXt101-FPN | 26.1 / 29.0 | 7.0 / 8.1 |
| CFA [34] [ICCV2023] | ResNeXt101-FPN | - | 10.8 / 12.6 |
| *Ours* | | | |
| Hydra-SGG | ResNet50 | 23.1 / 26.8<br>±0.3 / ±0.3 | **12.5 / 15.6**<br>±0.2 / ±0.3 |

**Open Images V6** [30] `test`. As shown in Table 2, Hydra-SGG achieves SOTA in only **7** epochs while SGTR [37], RelTR [6] and EGTR [17] require 119, 150, and 275 epochs. Although Open Images V6 contains more than 120,000 images, its scenes are not as complex as those in VG150 [67]. Training Hydra-SGG for 7 epochs is sufficient to achieve good performance.

**GQA** [15] `test`. Our experimental results demonstrate that Hydra-SGG achieves SOTA mean Recall performance on GQA. As shown in Table 3, Hydra-SGG achieves 12.7 and 15.9 for mR@50 and mR@100 respectively, surpassing previous best results. Our model employs the relatively lightweight ResNet50 as its backbone, in contrast to the heavier ResNeXt101 used by other methods.

Table 4: A set of ablative experiments about on VG150 [27] `test` (§4.3). The adopted hyperparameters are marked in red.

| Method | mR@20 | mR@50 | # Epoch |
|---|---|---|---|
| Baseline | 8.7 | 12.9 | 50 |
| Vanilla Hydra-SGG | 9.9 (+1.2) | 14.9 (+2.0) | 12 |
| Hydra-SGG | 10.6 (+1.9) | 16.0 (+3.1) | 12 |

(a) Key Components

| Method | Total training time | mR@20 | mR@50 |
|---|---|---|---|
| RelTR [6] | 50.0h (150 epochs) | 6.8 | 10.8 |
| Baseline | 16.7h (50 epochs) | 8.7 | 12.9 |
| Hydra-SGG | **12.0**h (12 epochs) | **10.6** | **16.0** |

(b) Training Time

| $T$ | mR@20 | mR@50 | mR@100 |
|---|---|---|---|
| 0.3 | 10.4 | 14.9 | 18.9 |
| 0.4 | **10.6** | **16.0** | **19.7** |
| 0.5 | 9.4 | 15.5 | 19.6 |
| 0.6 | 10.3 | 14.9 | 19.1 |

(c) Threshold $T$

| # Epoch | mR@20 | mR@50 | mR@100 |
|---|---|---|---|
| 10 | 10.0 | 15.4 | 18.6 |
| 12 | 10.6 | 16.0 | 19.7 |
| 21 | **11.6** | **16.1** | 19.7 |
| 24 | 10.4 | 15.2 | **19.9** |

(d) Training Epoch

| # Query | mR@20 | mR@50 | mR@100 |
|---|---|---|---|
| 100 | 8.8 | 14.1 | 17.8 |
| 200 | 9.9 | 15.3 | 18.4 |
| 300 | **10.6** | **16.0** | **19.7** |
| 400 | 10.6 | 15.5 | 18.9 |

(e) Number of Relation Queries

| Loss Ratio (1-to-1 : 1-to-M) | mR@50 | mR@100 |
|---|---|---|
| 1 : 0.5 | 14.6 | 18.9 |
| 1 : 0.8 | 15.6 | 19.4 |
| 1 : 1 | **16.0** | **19.7** |
| 1 : 1.5 | 15.4 | 18.7 |
| 1 : 2 | 15.6 | 19.5 |
| 1 : 3 | 15.4 | 19.3 |

(f) Loss Weights Between One-to-One (1-to-1) and One-to-Many (1-to-M) Losses.

| Method | mR@50 | mR@100 |
|---|---|---|
| TDE [59][CVPR2020] | 9.2 | 11.1 |
| NICE [33][CVPR2022] | 10.4 | 12.7 |
| IETrans [79][ECCV2022] | 12.5 | 15.0 |
| CFA [34][ICCV2023] | 12.3 | 14.6 |
| VETO [55][ICCV2023] | 10.6 | 13.8 |
| NICEST [35][TPAMI2024] | 10.4 | 12.4 |
| Hydra-SGG | **16.0** | **19.7** |

(g) Comparisons with Unbiasing Methods.

**Comparisons with Unbiasing Methods.** As shown in Table 4g, Hydra-SGG achieves very competitive performance on VG150 compared to these specialized debiasing methods, including TDE [59], NICE [33], IETrans [79], CFA [34], VETO [55], and NICEST [35].

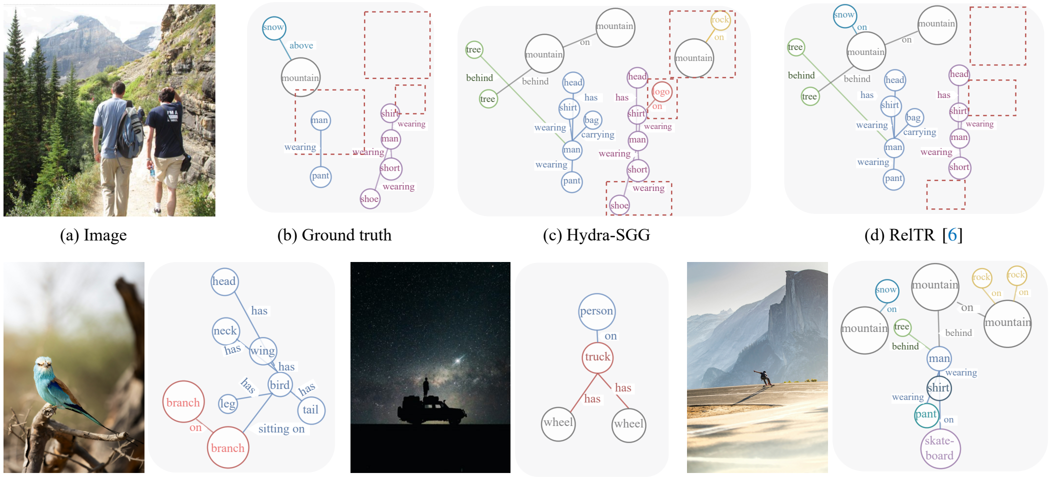
### 4.3 Diagnostic Experiment

**Key Components.** We first investigate the effectiveness of our core Hybrid Relation Assignment (§3.2) and Hydra Branch (§3.3) in Table 4a. The first row gives the score of our baseline model (§3.1). The second row corresponds to the result of Vanilla Hydra-SGG, which directly applies Hybrid Relation Assignment into `RelDecoder`. The third row lists the results of the complete Hydra-SGG model. Our results show that taking the proposed Hybrid Relation Assignment in the baseline model improves the mean recall and achieves 9.9/14.9 mR@20/mR@50. Furthermore, the integration of Hydra Branch with Hybrid Relation Assignment yields a synergistic effect, boosting performance substantially. Specifically, Hydra Branch increases mR@20 from 9.9 to 10.6 and mR@50 from 14.9 to 16.0, highlighting the complementary nature of these two strategies.

**Training Efficacy.** In Table 4b, we compare the training time costs on VG150 [67]. All models are trained on eight NVIDIA RTX 4090 GPUs with a ResNet50 backbone. Our model achieves 16.0 mR@50 in 12 hours, whereas RelTR takes 50 hours to achieve 6.8 mR@50. This demonstrates the superior training efficacy of Hydra-SGG.

**Threshold $T$.** We next study the influence of the threshold $T$ in One-to-Many Relation Assignment in Table 4c. The threshold $T$ controls the quality and quantity of positive relation queries. A higher value of $T$ keeps only the high-quality positive relation queries but at the cost of reducing their number. Conversely, lowering $T$ yields more positive samples, but their quality may decrease. This trade-off between quality and quantity directly affects the final performance of the model. Experimental results show optimal performance at $T = 0.4$, yielding mR@20 of 10.6 and mR@50 of 16.0. As we increase $T$ to 0.5 and 0.6, the performance gradually decreases. This suggests that while maintaining high-quality positive samples is important, having a sufficient number of them is also crucial for the model to learn effectively.

**Training Epochs.** We further investigate the influence of training epochs on the performance. As presented in Table 4d, Hydra-SGG achieves state-of-the-art mean Recalls in just 12 epochs, demonstrating its fast convergence. We observe further improvements when increasing the epochs to 21, *i.e.*, 10.6 mR@20 and 16.0 mR@50 respectively. However, the performance decreases at epoch 24, with mR@20 and mR@50 dropping to 10.4 and 15.2, respectively. This reduction in performance can be attributed to overfitting.



(a) Image (b) Ground truth (c) Hydra-SGG (d) RelTR [6]

(e) Usplash Image 1 (f) Usplash Image 2 (g) Usplash Image 3

Figure 5: Qualitative results §4.4. (a)-(d) compare Hydra-SGG and RelTR [6] on a VG150 [27] `val` image. We use the same color for each entity category, and the color of a predicate matches that of its subject. Differences are highlighted with red dashed rectangles ⬚. (e)-(g) show scene graphs generated by Hydra-SGG from images sourced from Unsplash, a platform for freely-usable images. These images are real-world, "in the wild" scenarios, demonstrating our model's capability to handle diverse and unseen visual content.

**Number of Relation Queries.** Lastly, we studied the effect of the number of relation queries on model performance, as shown in Table 4e. The performance with 200 queries and 400 queries is inferior to that with 300 queries. The model achieved mR@20/50 of 9.9/15.3 with 200 queries, and 10.6/15.5 with 400 queries. In contrast, with 300 queries, the model reached its highest performance, with 10.6 mR@20 and 16.0 mR@50. We hypothesize that using 200 queries may be insufficient to capture the diversity of relationships within the data. On the other hand, using 400 queries could introduce excessive noise and false positives, thereby reducing the overall precision of the model.

**Impact of Loss Weights.** As shown in Table 4f, we conducted an ablation study to investigate the effect of different loss weights between One-to-One and One-to-Many losses. With a 1:1 loss ratio, the model achieves the best results (mR@50: 16.0, mR@100: 19.7). When the One-to-Many loss weight is reduced (*e.g.*, 1:0.5), the model fails to fully utilize the additional positive samples available. Conversely, increasing this weight excessively (*e.g.*, 1:3) leads to an overemphasis on One-to-Many triplets, which typically have lower quality compared to One-to-One triplets. This quality difference arises from the fundamental nature of the assignments: One-to-One assignment matches each relation label with its optimal query exclusively, while One-to-Many assignment matches a relation label with multiple queries meeting certain criteria.

### 4.4 Qualitative Comparison Result

In Fig. 5a-d, we visualize the generated scene graph results on the image of VG150 [27] `val`. Both Hydra-SGG and RelTR [6] detect plausible relations, but these relations were not annotated, so these reasonable predictions become false negatives. Our Hydra-SGG detects more fine-grained relations such as `rock-on-mountain` and `logo-on-shirt`, while RelTR fails to detect such relations. In Fig. 5e-g, we visualize generated scene graph results on unseen Unsplash images.

## 5 Conclusion

This paper introduces Hydra-SGG, a one-stage DETR-based scene graph generation (SGG) model that addresses the slow convergence issue in existing DETR-based SGG models. The key contributions of our work are as follows: **i)** We propose a Hybrid Relation Assignment, which combines One-to-One and One-to-Many Relation Assignment strategies to increase relation supervision signals. **ii)** We introduce a Hydra Branch, an auxiliary decoder that encourages relation queries to predict duplicate relations, further enhancing the proposed One-to-Many Relation Assignment. These innovations work in synergy to significantly accelerate the learning process. Consequently, Hydra-SGG achieves state-of-the-art results on VG150, GQA, and Open Images V6 in just 12, 12, and 7 training epochs, demonstrating remarkable performance and efficiency.

## REFERENCES

[1] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving visual relationship detection using semantic modeling of scene descriptions. In *ECCV*, pp. 53–68, 2017.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pp. 213–229. Springer, 2020.

[3] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE TPAMI*, 45(1): 1–26, 2021.

[4] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In *ICCV*, pp. 6633–6642, 2023.

[5] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, pp. 6163–6171, 2019.

[6] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE TPAMI*, 2023.

[7] 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 *ICCV*, pp. 3550–3559, 2021.

[8] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *CVPR*, pp. 19427–19436, 2022.

[9] Azade Farshad, Yousef Yeganeh, Yu Chi, Chengzhi Shen, Böjrn Ommer, and Nassir Navab. Scenegenie: Scene graph guided diffusion models for image synthesis. In *ICCV*, pp. 88–98, 2023.

[10] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *ICCV*, pp. 3621–3630, 2021.

[11] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, pp. 1969–1978, 2019.

[12] Zeeshan Hayder and Xuming He. Dsgg: Dense relation transformer for an end-to-end scene graph generation. In *CVPR*, pp. 28317–28326, 2024.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017.

[14] Zhengdong Hu, Yifan Sun, Jingdong Wang, and Yi Yang. Dac-detr: Divide the attention layers and conquer. In *NeurIPS*, 2024.

[15] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pp. 6700–6709, 2019.

[16] Zih-Siou Hung, Arun Mallya, and Svetlana Lazebnik. Contextual translation embedding for visual relationship detection and scene graph generation. *IEEE TPAMI*, 43(11):3820–3832, 2020.

[17] Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. Egtr: Extracting graph from transformer for scene graph generation. In *CVPR*, pp. 24229–24238, 2024.

[18] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. In *CVPR*, pp. 19702–19712, 2023.

[19] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, pp. 3668–3678, 2015.

[20] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, pp. 2989–2998, 2017.

[21] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pp. 1219–1228, 2018.

[22] Deunsol Jung, Sanghyun Kim, Won Hwa Kim, and Minsu Cho. Devil's on the edges: Selective quad attention for scene graph generation. In *CVPR*, pp. 18664–18674, 2023.

[23] Franklin Kenghagho Kenfack, Feroz Ahmed Siddiky, Ferenc Balint-Benczedi, and Michael Beetz. Robotvqa—a scene-graph-and deep-learning-based visual question answering system for robot manipulation. In *IROS*, pp. 9667–9674. IEEE, 2020.

[24] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. In *NeurIPS*, 2022.

[25] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *CVPR*, pp. 74–83, 2021.

[26] Jongha Kim, Jihwan Park, Jinyoung Park, Jinyoung Kim, Sehyung Kim, and Hyunwoo J Kim. Groupwise query specialization and quality-aware multi-assignment for transformer-based visual relationship detection. In *CVPR*, 2024.

[27] 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. *ICCV*, 123:32–73, 2017.

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

[29] Sanjoy Kundu and Sathyanarayanan N Aakur. Is-ggt: Iterative scene graph generation with generative transformers. In *CVPR*, pp. 6292–6301, 2023.

[30] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 128(7):1956–1981, 2020.

[31] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, pp. 13619–13627, 2022.

[32] Jiankai Li, Yunhong Wang, Xiefan Guo, Ruijie Yang, and Weixin Li. Leveraging predicate and triplet learning for scene graph generation. In *CVPR*, 2024.

[33] Lin Li, Long Chen, Yifeng Huang, Zhimeng Zhang, Songyang Zhang, and Jun Xiao. The devil is in the labels: Noisy label correction for robust scene graph generation. In *CVPR*, pp. 18869–18878, 2022.

[34] Lin Li, Guikun Chen, Jun Xiao, Yi Yang, Chunping Wang, and Long Chen. Compositional feature augmentation for unbiased scene graph generation. In *ICCV*, pp. 21685–21695, 2023.

[35] Lin Li, Jun Xiao, Hanrong Shi, Hanwang Zhang, Yi Yang, Wei Liu, and Long Chen. Nicest: Noisy label correction and training for robust scene graph generation. *IEEE TPAMI*, 2024.

[36] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, pp. 11109–11119, 2021.

[37] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, pp. 19486–19496, 2022.

[38] Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *CVPR*, pp. 2928–2937, 2022.

[39] Yikang Li, Wanli Ouyang, Xiaogang Wang, and Xiao'ou Tang. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*, pp. 1347–1356, 2017.

[40] Yuanzhi Liang, Yalong Bai, Wei Zhang, Xueming Qian, Li Zhu, and Tao Mei. Vrr-vg: Refocusing visually-relevant relationships. In *ICCV*, pp. 10403–10412, 2019.

[41] Yue Liao, Aixi Zhang, Miao Lu, Yongliang Wang, Xiaobo Li, and Si Liu. Gen-vlkt: Simplify association and enhance interaction understanding for hoi detection. In *CVPR*, pp. 20123–20132, 2022.

[42] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.

[43] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *CVPR*, pp. 11546–11556, 2021.

[44] Rui Liu, Xiaohan Wang, Wenguan Wang, and Yi Yang. Bird's-eye-view scene graph for vision-language navigation. In *ICCV*, pp. 10968–10980, 2023.

[45] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022.

[46] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, pp. 852–869, 2016.

[47] Yichao Lu, Himanshu Rai, Jason Chang, Boris Knyazev, Guangwei Yu, Shashank Shekhar, Graham W Taylor, and Maksims Volkovs. Context-aware scene graph generation with seq2seq transformers. In *ICCV*, pp. 15931–15941, 2021.

[48] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pp. 3651–3660, 2021.

[49] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *CVPR*, pp. 3957–3966, 2019.

[50] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pp. 7263–7271, 2017.

[51] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[53] Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. In *ECCV*, pp. 422–439. Springer, 2022.

[54] Kunal Pratap Singh, Jordi Salvador, Luca Weihs, and Aniruddha Kembhavi. Scene graph contrastive learning for embodied navigation. In *ICCV*, pp. 10884–10894, 2023.

[55] Gopika Sudhakaran, Devendra Singh Dhami, Kristian Kersting, and Stefan Roth. Vision relation transformer for unbiased scene graph generation. In *ICCV*, pp. 21882–21893, 2023.

[56] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *CVPR*, pp. 13936–13945, 2021.

[57] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. In *CVPR*, pp. 10410–10419, 2021.

[58] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pp. 6619–6628, 2019.

[59] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, pp. 3716–3725, 2020.

[60] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *CVPR*, pp. 1–9, 2017.

[61] Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. In *CVPR*, pp. 19437–19446, 2022.

[62] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *ICCV*, pp. 13688–13697, 2021.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[64] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *CVPR*, pp. 8188–8197, 2019.

[65] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, pp. 2567–2575, 2022.

[66] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Linknet: Relational embedding for scene graph. In *NeurIPS*, 2018.

[67] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pp. 5410–5419, 2017.

[68] Li Xu, Haoxuan Qu, Jason Kuen, Jiuxiang Gu, and Jun Liu. Meta spatio-temporal debiasing for video scene graph generation. In *ECCV*, pp. 374–390. Springer, 2022.

[69] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pp. 670–685, 2018.

[70] Jingkang Yang, Wenxuan Peng, Xiangtai Li, Zujin Guo, Liangyu Chen, Bo Li, Zheng Ma, Kaiyang Zhou, Wayne Zhang, Chen Change Loy, et al. Panoptic video scene graph generation. In *CVPR*, pp. 18675–18685, 2023.

[71] Ling Yang, Zhilin Huang, Yang Song, Shenda Hong, Guohao Li, Wentao Zhang, Bin Cui, Bernard Ghanem, and Ming-Hsuan Yang. Diffusion-based scene graph to image generation with masked contrastive pre-training. *arXiv preprint arXiv:2211.11138*, 2022.

[72] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

[73] Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, and Maosong Sun. Visual distant supervision for scene graph generation. In *ICCV*, pp. 15816–15826, 2021.

[74] 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 *ECCV*, pp. 322–338, 2018.

[75] Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*, pp. 1974–1982, 2017.

[76] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pp. 5831–5840, 2018.

[77] Pengpeng Zeng, Haonan Zhang, Lianli Gao, Jingkuan Song, and Heng Tao Shen. Video question answering with prior knowledge and object-sensitive learning. *IEEE TIP*, 31:5936–5948, 2022.

[78] 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. In *NeurIPS*, 2021.

[79] Ao Zhang, Yuan Yao, Qianyu Chen, Wei Ji, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Fine-grained scene graph generation with data transfer. In *ECCV*, pp. 409–424. Springer, 2022.

[80] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, pp. 5532–5540, 2017.

[81] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *ICLR*, 2023.

[82] Long Zhao, Liangzhe Yuan, Boqing Gong, Yin Cui, Florian Schroff, Ming-Hsuan Yang, Hartwig Adam, and Ting Liu. Unified visual relationship detection with vision and language models. In *ICCV*, pp. 6962–6973, 2023.

[83] Chaofan Zheng, Xinyu Lyu, Lianli Gao, Bo Dai, and Jingkuan Song. Prototype-based embedding network for scene graph generation. In *CVPR*, pp. 22783–22792, 2023.

[84] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.

[85] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *ICCV*, pp. 6748–6758, 2023.

[86] 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 *CVPR*, pp. 11825–11834, 2021.

# APPENDIX

For a better understanding of the main paper, we provide additional details in this supplementary material, which is organized as follows:

- §A details the implementation details.
- §B provides the pseudo code of Hydra-SGG.
- §C shows failure cases of Hydra-SGG in VG150 [67].
- §D discusses our limitations, societal impact, and directions of future work.

## A IMPLEMENTATION

Inspired by RelTR [6], we also set relation queries to be composed of subject and object queries. We adopt the same training techniques and attention mechanisms as previous works [31, 81, 6] in Hydra-SGG. In particular, we use anchor boxes as embeddings to accelerate training and employ deformable attention [84] respectively. We input noised box-label pairs to generate both entity and relation denoising queries. To prevent information leakage, we employ attention masks. Given the absence of an attention map in the deformable decoder, we simplify the model by omitting the convolutional mask head in RelTR [6]. In particular, we concatenate the output subject and object queries and input them into an MLP. This simplification not only reduces computational complexity but also maintains performance.

Both One-to-One and One-to-Many losses are composed of box regression loss, entity classification loss, and relation classification loss as previous works [2, 37, 6, 17, 57]. Specifically, we use Focal loss [42] for both relation and detection, as in previous works [45, 31, 81]. In the post-processing stage, the predictions are first ranked by the relation probability. Then we use the relation index to find corresponding subject and object predictions to compose the prediction triplets. Finally, we follow the same process as RelTR [6] that removes predictions where the subject and object are the same, as such data does not exist in VG150 [67], but we do not use this process in Open Images V6 [30].

For training, we adopt the same data augmentation techniques as RelTR [6] but discard random cropping since some triplets could be incomplete. The default training epochs are 12 for VG150 [67] and 7 for Open Images V6 [30]. For VG150, the learning rate is scaled by 0.1 at epoch 11, while for Open Images V6, it is scaled at epoch 6. Open Images V6 [30] contains more than 120,000 training images, but most scenes in the dataset are simpler than those in VG150 [67], therefore training for 7 epochs is enough to achieve state-of-the-art performance. Extended training on Open Images V6 may improve the performance, but considering the dataset size, the marginal benefits are limited.

Due to the inherent bias in SGG datasets, numerous unbiasing algorithms have been developed [59, 34, 33, 55, 83]. While adopting these unbiasing techniques typically leads to significant performance improvements, we opt for a fair comparison by evaluating Hydra-SGG against methods that do not employ such techniques.

## B PSEUDO CODE

The pseudo-code of Hydra-SGG is given in Algorithm S1. Our code and pre-trained models will be made publicly available.

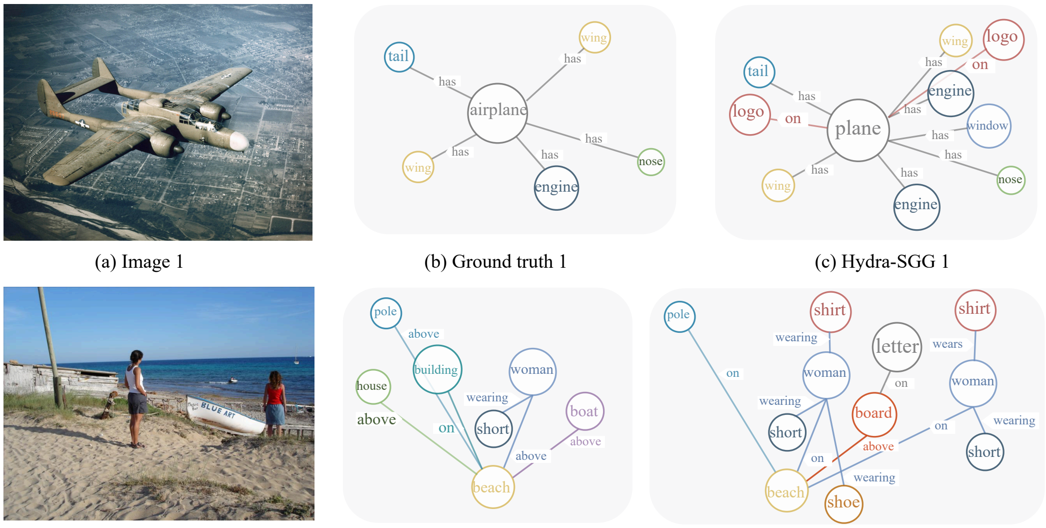
## C FAILURE CASES

In this section, we present failure cases of scene graphs generated by Hydra-SGG on VG150 [67] `val`. In the first row (Fig. S1a-c), Hydra-SGG successfully recognizes the airplane in the image, but the predicted label is ‘plane’. Although ‘plane’ is a plausible label, the ground truth label is ‘airplane’, which leads to all predicted relationships being incorrect since they don’t match the ground truth. However, from a semantic perspective, the model correctly identifies these relationships and even recognizes more fine-grained relationships that are not annotated in the ground truth. For instance, the ground truth only marks the right engine of the airplane, but the model correctly identifies

**Algorithm 1** Hydra-SGG: PyTorch-like Pseudo-code

```
# RelDecoder shares parameters with Hydra Branch
# memory: image tokens (L x D)
# Query_rel: relation queries of RelDecoder (N x D)
# Query_rel_o2m: relation queries of HydraBranch (N x D)
# Loss_o2o: apply box regression, entity, and relation classification losses between
    matched queries and GTs by Hungarian matching.
# Loss_o2m: apply box regression, entity, and relation classification losses between
    matched queries and GTs by One-to-Many Relation Assignment.

# Hydra SGG forward
def forward(sample, target):
    # extract and enhance features from the input
    memory = Encoder(Backbone(sample)) # (L x D)

    # init relation queries for RelDecoder and Hydra Branch
    Query_rel = Query_rel_o2m = self.init_queries()

    Query_rel_bar = RelDecoder(Query_rel, memory) # (N x D)
    Query_rel_o2m_bar = HydraBranch(Query_rel_o2m, memory) # (N x D)

    loss_1 = Loss_o2o(Query_rel_bar, targets)
    loss_2 = Loss_o2m(Query_rel_o2m_bar, targets)
    loss_hydra = loss_1 + loss_2
    loss_hydra.backward()

# RelDecoder
def RelDecoder(Query_rel, memory):
    Query_rel = Self_attn(Query_rel) # (N x D)
    Query_rel = Cross_attn(Query_rel, memory) # (N x D)
    Query_rel = FFN(Query_rel)
    return Query_rel

# Hydra Branch
def HydraBranch(Query_rel_o2m, memory):
    Query_rel_o2m = Cross_attn(Query_rel_o2m, memory) # (N x D)
    Query_rel_o2m = FFN(Query_rel_o2m)
    return Query_rel_o2m
```

Algorithm S1: Hydra-SGG core implementation.



(a) Image 1

(b) Ground truth 1

(c) Hydra-SGG 1

(d) Image 2

(e) Ground truth 2

(f) Hydra-SGG 2

Figure S1: Failure cases of Hydra-SGG on VG150 [67] `val`.

both engines. In addition, the model correctly identifies the logo and windows on the airplane, which are reasonable and correct predictions but are considered incorrect because they are not annotated in the ground truth. In the second row (Fig. S1d-f), the ground truth annotation uses a single box to label two women, which is an unreasonable annotation. The model correctly predicts two women and identifies that both women are wearing shirts and shorts, which are annotations absent in the ground truth but are semantically accurate.

## D Discussion

**Limitation Analysis.** A significant limitation of the current Hydra-SGG is its inability to predict object and relation categories beyond a predefined closed set. The algorithm can only make predictions for object and relation classes that have been explicitly defined and included in the training dataset. This constraint means that the model lacks the capability to recognize or infer novel object types or relationship categories that were not present during the training phase. Such a limitation restricts the model's applicability in real-world scenarios, where encountering previously unseen objects or relations is common. In addition, since the current One-to-Many Relation Assignment rules are manually designed, they may lack flexibility and potentially introduce some noise to the training process.

**Societal Impact.** Hydra-SGG has the potential to significantly enhance autonomous driving systems, thereby improving road safety and efficiency. For instance, when integrated into self-driving vehicles, Hydra-SGG can provide a deeper understanding of complex traffic scenarios. Beyond merely identifying vehicles, pedestrians, and road signs, it can comprehend the spatial relationships and potential interactions between these elements. This advanced scene understanding could enable autonomous vehicles to more accurately predict the behavior of other road users, leading to safer and more efficient navigation in diverse traffic conditions.

**Future Work.** A promising direction for future research is to extend the current model towards open vocabulary SGG. This advancement would allow Hydra-SGG to recognize and predict objects and relationships beyond the predefined closed set of categories used in training.

As mentioned in the limitations, since the current One-to-Many Relation Assignment rules are manually designed, they may introduce noise during training that affects learning. Therefore, another direction for future work could focus on designing better one-to-many strategies to further improve performance. For instance, we could develop a feature composition module to generate new relation triplets, or we can incorporate a relationship refine module to transfer coarse-grained relations into fine-grained relations.