

# Panoptic Scene Graph Generation

Jingkang Yang<sup>1</sup>, Yi Zhe Ang<sup>1</sup>, Zujin Guo<sup>1</sup>,  
 Kaiyang Zhou<sup>1</sup>, Wayne Zhang<sup>2</sup>, and Ziwei Liu<sup>1</sup>

<sup>1</sup> S-Lab, Nanyang Technological University, Singapore

{jingkang001,yizhe.ang,gu00008,kaiyang.zhou,ziwei.liu}@ntu.edu.sg

<sup>2</sup> SenseTime Research, Shenzhen, China

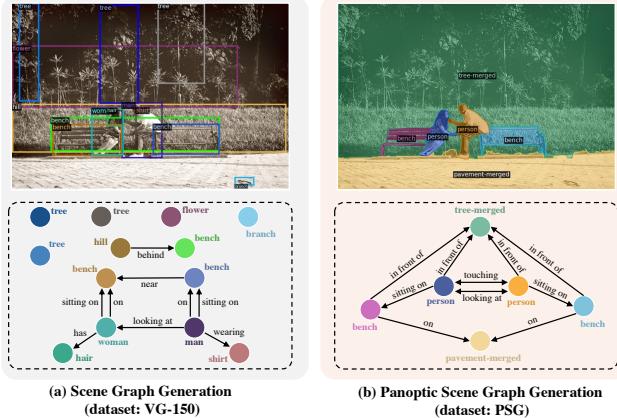
wayne.zhang@sensetime.com

**Abstract.** Existing research addresses scene graph generation (SGG)—a critical technology for scene understanding in images—from a detection perspective, *i.e.*, objects are detected using bounding boxes followed by prediction of their pairwise relationships. We argue that such a paradigm causes several problems that impede the progress of the field. For instance, bounding box-based labels in current datasets usually contain redundant classes like hairs, and leave out background information that is crucial to the understanding of context. In this work, we introduce *panoptic scene graph generation (PSG)*, a new problem task that requires the model to generate a more comprehensive scene graph representation based on panoptic segmentations rather than rigid bounding boxes. A high-quality *PSG dataset*, which contains 49k well-annotated overlapping images from COCO and Visual Genome, is created for the community to keep track of its progress. For benchmarking, we build four two-stage baselines, which are modified from classic methods in SGG, and two one-stage baselines called PSGTR and PSGFormer, which are based on the efficient Transformer-based detector, *i.e.*, DETR. While PSGTR uses a set of queries to directly learn triplets, PSGFormer separately models the objects and relations in the form of queries from two Transformer decoders, followed by a prompting-like relation-object matching mechanism. In the end, we share insights on open challenges and future directions. We invite users to explore the PSG dataset on our project page <https://psgdataset.org/>, and try our codebase <https://github.com/Jingkang50/OpenPSG>.

## 1 Introduction

The goal of scene graph generation (SGG) task is to generate a graph-structured representation from a given image to abstract out objects—grounded by bounding boxes—and their pairwise relationships [5,65]. Scene graphs aim to facilitate the understanding of complex scenes in images and has potential for a wide range of downstream applications, such as image retrieval [27,52,50], visual reasoning [1,53], visual question answering (VQA) [21], image captioning [16,8], structured image generation and outpainting [26,13,66], and robotics [14,2].

Since the introduction of SGG [27], this problem has been addressed from a detection perspective, *i.e.*, using bounding boxes to detect objects followed by



**Fig. 1: Scene graph generation (a. SGG task) vs. panoptic scene graph generation (b. PSG task).** The existing SGG task in (a) uses bounding box-based labels, which are often inaccurate—pixels covered by a bounding box do not necessarily belong to the annotated class—and cannot fully capture the background information. In contrast, the proposed PSG task in (b) presents a more comprehensive and clean scene graph representation, with more accurate localization of objects and including relationships with the background (known as stuff), i.e., the trees and pavement.

the prediction of their pairwise relationships [35, 64, 42]. We argue that such a bounding box-based paradigm is not ideal for solving the problem, and would instead cause a number of issues that impede the progress of the field. Firstly, bounding boxes—as labeled in current datasets [35]—only provide a coarse localization of objects and often contain noisy/ambiguous pixels belonging to different objects or categories (see the bounding boxes of the two persons in Fig. 1-a). Secondly, bounding boxes typically cannot cover the full scene of an image. For instance, the pavement region in Fig. 1-a is crucial for understanding the context but is completely ignored. Thirdly, current SGG datasets often include redundant classes and information like `woman-has-hair` in Fig. 1-a, which is mostly deemed trivial [42]. Furthermore, inconsistent and redundant labels are also observed in current datasets, e.g., the trees and benches in Fig. 1-a are labeled multiple times, and some extra annotations do not contribute to the graph (see isolated nodes). Using such labels for learning might confuse the model.

Ideally, the grounding of objects should be clear and precise, and a scene graph should not only focus on salient regions and relationships in an image but also be comprehensive enough for scene understanding. We argue that as compared to bounding boxes, panoptic segmentation [33] labels would be a better choice for constructing scene graphs. To this end, we introduce a new problem, *panoptic scene graph generation*, or PSG, with a goal of generating scene graph representations based on panoptic segmentations rather than rigid bounding boxes.

To help the community keep track of the research progress, we create a new PSG dataset based on COCO [43] and Visual Genome (VG) [35], which contains 49k well-annotated images in total. We follow COCO’s object annotation schema of 133 classes; comprising 80 thing classes and 53 stuff (background) classes. To construct predicates, we conduct a thorough investigation into existing VG-based datasets, *e.g.*, VG-150 [64], VrR-VG [42] and GQA [24], and summarize 56 predicate classes with minimum overlap and sufficient coverage of semantics. See Fig. 1–b for an example of our dataset. From Fig. 1, it is clear that the panoptic scene graph representation—including both panoptic segmentations and the scene graph—is much more informative and coherent than the previous scene graph representation.

For benchmarking, we build four two-stage models by integrating four classic SGG methods [64,71,58,54] into a classic panoptic segmentation framework [32]. We also turn DETR [4], an efficient Transformer-based detector, into a one-stage PSG model dubbed as PSGTR, which has proved effective for the PSG task. We further provide another one-stage baseline called PSGFormer, which extends PSGTR with two improvements: 1) modeling objects and relations separately in the form of queries within two Transformer decoders, and 2) a prompting-like interaction mechanism. A comprehensive comparison of one-stage models and two-stage models is discussed in our experiments.

In summary, we make the following contributions to the SGG community:

- **A New Problem and Dataset:** We discuss several issues with current SGG research, especially those associated with existing datasets. To address them, we introduce a new problem that combines SGG with panoptic segmentation, and create a large PSG dataset with high-quality annotations.
- **A Comprehensive Benchmark:** We build strong two-stage and one-stage PSG baselines, and evaluate them comprehensively on our new dataset, so that the PSG task is solidified in its inception. We find that one-stage models, despite having a simplified training paradigm, have great potential for PSG as it achieves competitive results on the dataset.

## 2 Related Work

**Scene Graph Generation** Existing scene graph generation (SGG) methods have been dominated by the two-stage pipeline that consists of object detection and pairwise predicate estimation. Given bounding boxes, early work predicts predicates using conditional random fields [27,11] or casts predicate prediction into a classification problem [75,34,49]. Inspired by knowledge graph embeddings, VTransE [74] and UVTransE [25] are proposed for explicit predicate modeling. Follow-up works have investigated various variants based on, *e.g.*, RNN and graph-based modeling [64,71,58,67,44,9,38], energy-based models [54], external knowledge [58,18,69,70,46], and more recently language supervision [77,68]. Recent research has shifted the attention to problems associated with the SGG datasets, such as the long-tailed distribution of predicates [57,12],

**Table 1: Comparison between classic SGG datasets and PSG dataset.** #PPI counts predicates per image. DupFree checks whether duplicated object groundings are cleaned up. Spvn indicates whether the objects are grounded by bounding boxes or segmentations.

Dataset	#Image	#ObjCls	#RelCls	#PPI	DupFree	Spvn	Source
VG [35]	108K	34K	40K	21.4	✗	BBox	COCO & Flickr
VG-150 [64]	88K	150	50	5.7	✗	BBox	Clean VG
VtR-VG [42]	59K	1,600	117	3.4	✗	BBox	Clean VG
GQA [24]	85K	1,703	310	50.6	✓	BBox	Re-annotate VG
<b>PSG</b>	49K	133	56	5.7	✓	Seg	Annotate COCO

excessive visually-irrelevant predicates [42], and inaccurate localization of bounding boxes [29]. In particular, a very recent study [29] shows that training an SGG model to simultaneously generate scene graphs and predict semantic segmentation masks can bring about improvements, which inspires our research. In our work, we study panoptic segmentation-based scene graph generation in a more systematic way by formulating a new problem and building a new benchmark. We also notice that a closely-related topic human-object interaction (HOI) [19] shares a similar goal with SGG, *i.e.*, to detect prominent relations from the image. However, the HOI task restricts the model to only detect human-related relations while ignoring the valuable information between objects that is often critical to comprehensive scene understanding. Nevertheless, many HOI methods are applicable to SGG tasks [17, 28, 6, 60, 40, 78, 61, 22, 39, 15, 30, 45, 55, 23, 72, 59, 73], and some of them have inspired our PSG baselines [31, 79].

**Scene Graph Datasets** While early SGG works have constructed several smaller-size datasets such as RW-SGD [27], VRD [46], and UnRel-D [48], the large-scale Visual Genome (VG) [35] quickly became the standard SGG dataset as soon as it was released in 2017, prompting subsequent work to research in a more realistic setting. However, several critical drawbacks of VG were raised by the community, and therefore, some VG variants were gradually introduced to address some problems. Notice that VG contains an impractically large number of 33,877 object classes and 40,480 predicate classes, leading VG-150 [64] to only keep the most frequent 150 object classes and 50 predicate classes for a more realistic setting. Later, VrR-VG [42] argued that many predicates in VG-150 can be easily estimated by statistical priors, hence deciding to re-filter the original VG categories to only keep visually relevant predicate classes. However, by scrutinizing VrR-VG, we find many predicates are redundant (*e.g.*, `alongside`, `beside`, `besides`) and ambiguous (*e.g.*, `beyond`, `down`). Similar drawbacks appeared in another large-scale dataset with scene graph annotations called GQA [24]. In summary, although relations play a pivotal role in SGG tasks, a systematic definition of relations is unfortunately overlooked across all the existing SGG datasets. Therefore, in our proposed PSG dataset, we consider both comprehensive coverage and non-overlap between words, and carefully define a predicate



**Fig. 2:** Word Cloud for PSG Predicates.

dictionary with 56 classes to better formulate the scene graph problem. Fig. 2 shows the word cloud of the predicate classes, where font size indicates frequency.

Apart from the problem of predicate definition, another critical issue of SGG datasets is that they all adopt bounding box-based object grounding, which inevitably causes a number of issues such as coarse localization (bounding boxes cannot reach pixel-level accuracy), inability to ground comprehensively (bounding boxes cannot ground backgrounds), tendency to provide trivial information (current datasets usually capture objects like `head` to form the trivial relation of `person-has-head`), and duplicate groundings (the same object could be grounded by multiple separate bounding boxes). These issues together have caused the low-quality of current SGG datasets, which impede the progress of the field. Therefore, the proposed PSG dataset tries to address all the above problems by grounding the images using accurate and comprehensive panoptic segmentations with COCO’s appropriate granularity of object categories. Table 1 compares the statistics of the PSG dataset with classic SGG datasets.

**Panoptic Segmentation** The panoptic segmentation task unifies semantic segmentation and instance segmentation [33] for comprehensive scene understanding, and the first approach is a simple combination of a semantic segmentation model and an instance segmentation model to produce stuff masks and thing masks respectively [33]. Follow-up work, such as Panoptic FPN [32] and UPSNet [63], aim to unify the two tasks in a single model through multi-task learning to achieve gains in compute efficiency and segmentation performance. Recent approaches (e.g., MaskFormer [10], Panoptic Segformer [41] and K-Net [76]) have turned to more efficient architectures based on Transformers like DETR [4], which simplifies the detection pipeline by casting the detection task as a set prediction problem.

### 3 Problem and Dataset

**Recap: Scene Graph Generation** We first briefly review the goal of the classic scene graph generation (SGG) task, which aims to model the distribution:

$$\Pr(\mathbf{G} | \mathbf{I}) = \Pr(\mathbf{B}, \mathbf{O}, \mathbf{R} | \mathbf{I}), \quad (1)$$

where  $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$  is the input image, and  $\mathbf{G}$  is the desired scene graph which comprises the bounding boxes  $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$  and labels  $\mathbf{O} = \{o_1, \dots, o_n\}$  that correspond to each of the  $n$  objects in the image, and their relations in the set  $\mathbf{R} = \{r_1, \dots, r_l\}$ . More specifically,  $\mathbf{b}_i \in \mathbb{R}^4$  represents the box coordinates,  $o_i \in \mathbb{C}^O$  and  $r_i \in \mathbb{C}^R$  belong to the set of all object and relation classes.

**Panoptic Scene Graph Generation** Instead of localizing each object by its bounding box coordinates, the new task of panoptic scene graph generation (PSG task) grounds each object with the more fine-grained panoptic segmentation. For conciseness, we refer to both objects and background as objects.

Formally, with panoptic segmentation, an image is segmented into a set of masks  $\mathbf{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_n\}$ , where  $\mathbf{m}_i \in \{0, 1\}^{H \times W}$ . Each mask is associated

with an object with class label  $o_i \in \mathcal{C}^O$ . A set of relations  $\mathbf{R}$  between objects are also predicted. The masks do not overlap, *i.e.*,  $\sum_{i=1}^n \mathbf{m}_i \leq \mathbf{1}^{H \times W}$ . Hence, PSG task models the following distribution:

$$\Pr(\mathbf{G} | \mathbf{I}) = \Pr(\mathbf{M}, \mathbf{O}, \mathbf{R} | \mathbf{I}). \quad (2)$$

**PSG Dataset** To address the PSG task, we build our PSG dataset following these three major steps. Readers can find more details in the Appendix.

**Step 1: A Coarse COCO & VG Fusion:** To create a dataset with both panoptic segmentation and relation annotations, we use the 48,749 images in the intersection of the COCO and VG datasets with an automatic but coarse dataset fusion process. Specifically, we use an object category matching procedure to match COCO’s segmentations with VG’s bounding boxes, so that part of VG’s predicates are applied to COCO’s segmentation pairs. Due to the inherent mismatch between the label systems and localization annotations of VG and COCO, the auto-generated dataset is very noisy and requires further cleaning.

**Step 2: A Concise Predicate Dictionary:** Inspired by the appropriate granularity of COCO categories [43], we carefully identify 56 salient relations by taking reference from common predicates in the initial noisy PSG dataset and all VG-based datasets including VG-150 [64], VrR-VG [42] and GQA [24]. The selected 56 predicates are maximally independent (*e.g.*, we only keep “over/on” and do not have “under”) and cover most common cases in the dataset.

**Step 3: A Rigorous Annotation Process:** Building upon the noisy PSG dataset, we require the annotators to 1) *filter* out incorrect triplets, and 2) *supplement* more relations between not only object-object, but also object-background and background-background pairs, using the predefined 56 predicates. To prevent ambiguity between predicates, we ask the annotators strictly not to annotate using general relations like `on`, `in` when a more precise predicate like `parked on` is applicable. With this rule, the PSG dataset allows the model to understand the scene more precisely and saliently.

**Quality Control:** The PSG dataset goes through a professional dataset construction process. The main authors first annotated 1000 images to construct a detailed documentation (available in project page), and then employed a professional annotation company (sponsored by SenseTime) to annotate the training set within a month (US\$11K spent). Each image is annotated by two workers and examined by one head worker. All the test images are annotated by the authors.

**Summary:** Several merits worth highlighting by virtue of the novel and effective procedure of PSG dataset creation: 1) *Good grounding annotation* from the pixel-wise panoptic segmentation from COCO dataset [43], 2) *Clear category system* with 133 objects (*i.e.*, things plus stuff) and 56 predicates with appropriate granularity and minimal overlaps, and 3) *Accurate and comprehensive relation annotations* from a rigorous annotation process that pays special attention to salient relations between object-object, object-background and background-background. These merits address the notorious shortcomings [37] of classic scene graph datasets discussed in Sec. 2.

**Evaluation and Metrics** This section introduces the evaluation protocol for the PSG task. Following the settings of the classic SGG task [65,5], our PSG task comprises two sub-tasks: *predicate classification* (when applicable) and *scene graph generation* (main task) to evaluate the PSG models. *Predicate classification (PredCls)* aims to generate a scene graph given the ground-truth object labels and localization. The goal is to study the relation prediction performance without the interference of the segmentation performance. Notice that this metric is only applicable to two-stage PSG models in Sec. 4.1, since one-stage models cannot leverage the given segmentations to predict scene graph. *Scene graph generation (SGDet)* aims to generate scene graphs from scratch, which is the main result for the PSG task.

We also notice that classic SGG tasks contain another sub-task of scene graph classification (SGCls), which provide the ground-truth object groundings to simplify the scene graph generation process. We find SGCLS is not applicable for PSG baselines. Unlike SGG tasks where object detectors such as Faster-RCNN [51] can utilize ground-truth object bounding boxes to replace predictions from the Region Proposal Network (RPN), panoptic segmentation models are unable to directly use the ground-truth segmentations for classification, so the SGCLS task is inapplicable even for two-stage PSG methods.

The classic metrics of  $R@K$  and  $mR@K$  are used to evaluate the previous two sub-tasks, which calculates the triplet recall and mean recall for every predicate category, given the top K triplets from the PSG model. Notice that PSG grounds objects with segmentation, a successful recall requires both subject and object to have mask-based IOU larger than 0.5 compared to their ground-truth counterparts, with the correct classification on every position in the S-V-O triplet.

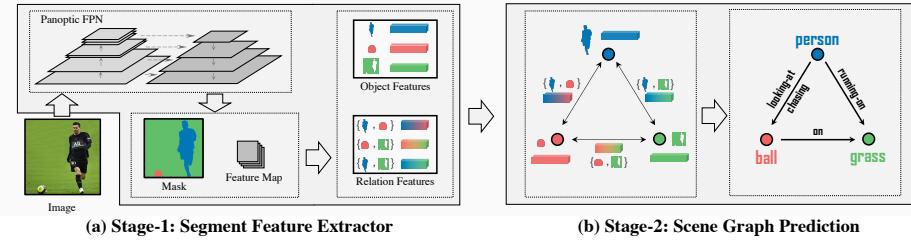
While the triplet recall rates that mentioned above are the main metric for PSG task, since objects are required to be grounded by segmentation masks, panoptic segmentation metrics [33] such as PQ [32] can be used for model diagnosis. However, it is not considered as the core evaluation metric of PSG task.

## 4 PSG Baselines

To build a comprehensive PSG benchmark, we refer to frameworks employed in the classic SGG task [5,65] and prepare two-stage and one-stage baselines.

### 4.1 Two-Stage PSG Baselines

Most prior SGG approaches tackle the problem in two stages: first performing object detection using off-the-shelf detectors like Faster-RCNN [51], then pairwise relationship prediction between these predicted objects. As shown in Figure 3, we follow a similar approach in establishing two-stage baselines for the PSG task: 1) using pretrained panoptic segmentation models of classic Panoptic FPN [32] to extract initial object features, masks and class predictions, and then 2) processing them using a relation prediction module from classic scene graph generation methods like IMP [64], MOTIFS [71], VCTree [58], and GPSNet [44]



**Fig. 3: Two-stage PSG baselines using Panoptic FPN.** **a)** In stage one, for each thing/stuff object, Panoptic FPN [32] produces a segmentation mask with its tightest bounding box to crop out the object feature. The union of relevant objects can produce relation features. **b)** In the second stage, the extracted object and relation features are fed into by any existing SGG relation model to predict the relation triplets.

to obtain the final scene graph predictions. In this way, classic SGG methods can be adapted to solve the PSG task with minimal modification. Formally, the two-stage PSGG baselines decompose formulation from Eq. 1 to Eq. 3.

$$\Pr(\mathbf{G} \mid \mathbf{I}) = \Pr(\mathbf{M} \mid \mathbf{I}) \cdot \Pr(\mathbf{O} \mid \mathbf{M}, \mathbf{I}) \cdot \Pr(\mathbf{R} \mid \mathbf{O}, \mathbf{M}, \mathbf{I}). \quad (3)$$

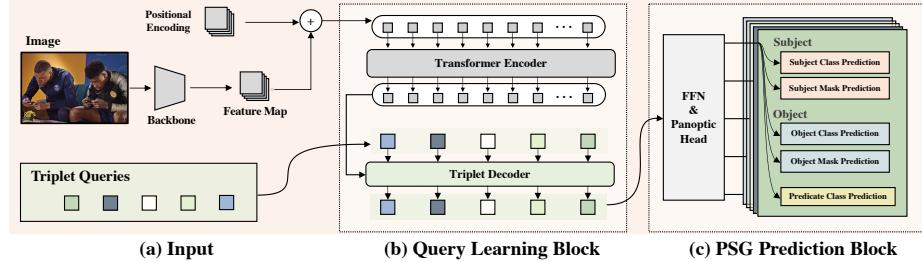
## 4.2 A One-Stage PSG Baseline - PSGTR

Unlike classic dense prediction models (*e.g.*, Faster-RCNN [51]) with sophisticated design, the transformer-based architectures support flexible input and output specifications. Based on the end-to-end DETR [4] and its extension to the HOI task [79], we naturally design a one-stage PSG method named PSGTR to predict triples and localizations simultaneously, which can be directly modeled as Eq. 2 without decomposition.

**Triplet Query Learning Block** As shown in Fig. 4, PSGTR first extracts image features from a CNN backbone and then feeds the features along with queries and position encoding into a transformer encoder-decoder. Here we expect the queries to learn the representation of scene graph triplets, so that for each triplet query, the subject/predicate/object predictions can be extracted by three individual Feed Forward Networks (FFNs), and the segmentation task can be completed by two panoptic heads for subject and object, respectively.

**PSG Prediction Block** To train the model, we extend the DETR’s Hungarian matcher [36] into a triplet Hungarian matcher. To match the triplet query  $\mathcal{T}_i \in \mathbb{Q}^T$  with ground truth triplets  $\mathcal{G}$ , all contents in the triplet (*i.e.*, all outputs that are predicted from  $\mathcal{T}_i$ ) are used, including the class of subject  $\tilde{\mathcal{T}}_i^S$ , relation  $\tilde{\mathcal{T}}_i^R$ , and object  $\tilde{\mathcal{T}}_i^O$ , and localization of subjects  $\tilde{\mathcal{T}}_i^S$  and objects  $\tilde{\mathcal{T}}_i^O$ . Therefore, the triplet matching (tm) cost  $\mathbf{C}_{\text{tm}}$  is designed with the combination of class matching  $\mathbf{C}_{\text{cls}}$  and segments matching  $\mathbf{C}_{\text{seg}}$ :

$$\mathbf{C}_{\text{tm}}(\mathcal{T}_i, \mathcal{G}_{\sigma(i)}) = \sum_{k \in \{\text{S,O}\}} \mathbf{C}_{\text{seg}}(\tilde{\mathcal{T}}_i^k, \tilde{\mathcal{G}}_{\sigma(i)}^k) + \sum_{k \in \{\text{S,R,O}\}} \mathbf{C}_{\text{cls}}(\tilde{\mathcal{T}}_i^k, \tilde{\mathcal{G}}_{\sigma(i)}^k), \quad (4)$$



**Fig. 4: PSGTR: One-stage PSG baseline.** The one-stage model takes in **a)** features extracted by CNNs with positional encoding, and a set of queries aiming to represent triplets. **b)** Query learning block processes image features with Transformer encoder-decoder and use queries to represent triplet information. Then, **c)** the PSG prediction head concretes the triplet predictions by producing subject/object/predicate classes using simple FFNs, and uses panoptic heads for panoptic segmentation.

where  $\sigma$  is the mapping function to correspond each triplet query  $\mathcal{T}_i \in \mathbb{Q}^T$  to the closest ground truth triplet. The triplet query set  $\mathbb{Q}^T$  collects all the  $|\mathbb{Q}^T|$  triplet queries. The optimization objective is thus:

$$\hat{\sigma} = \arg \max_{\sigma} \sum_{i=1}^{|\mathbb{Q}^T|} \mathbf{C}_{\text{tm}}(\mathcal{T}_i, \mathcal{G}_{\sigma(i)}). \quad (5)$$

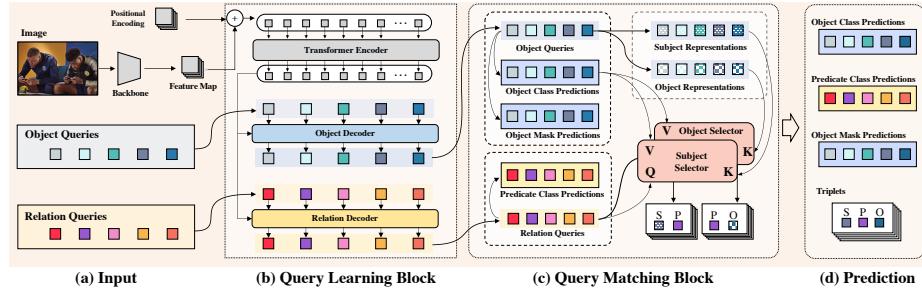
Once the matching is done, the total loss  $\mathcal{L}_{\text{total}}$  can be calculated by applying cross-entropy loss  $\mathcal{L}_{\text{cls}}$  for labels and DICE/F-1 loss [47] for segmentation  $\mathcal{L}_{\text{cls}}$ :

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{|\mathbb{Q}^T|} \left( \sum_{k \in \{\text{S}, \text{O}\}} \mathcal{L}_{\text{seg}}(\tilde{\mathcal{T}}_i^k, \tilde{\mathcal{G}}_{\hat{\sigma}(i)}^k) + \sum_{k \in \{\text{S}, \text{R}, \text{O}\}} \mathcal{L}_{\text{cls}}(\tilde{\mathcal{T}}_i^k, \tilde{\mathcal{G}}_{\hat{\sigma}(i)}^k) \right). \quad (6)$$

### 4.3 Alternative One-Stage PSG Baseline - PSGFormer

Based on the PSGTR baseline that explained in Section 4.2, we extend another end-to-end HOI method [31] and further propose the alternative one-stage PSG baseline named PSGFormer, featured by an explicit relation modeling with a prompting-like matching mechanism. The model diagram is illustrated in Figure 5 and will be elaborated as follows.

**Object & Relation Query Learning Block** Compared to the classic object-oriented tasks such as object detection and segmentation, the most significant uniqueness of PSG task as well as SGG task is their extra requirements on the predictions of relations. Notice that the relation modeling in our two-stage baselines depends on features from object-pairs, while PSGTR implicitly models the objects and relations altogether within the triplets, the important relation modeling has not been given a serious treatment. Therefore, in the exploration of



**Fig. 5: PSGFormer: The proposed one-stage PSG method.** **a)** Two types of queries, *i.e.*, object queries and relation queries, are fed into transformer block with CNN features and positional encoding. **b)** Query Learning Block processes image features with one encoder and output object or relation queries with the corresponding decoder. **c)** Object queries and relation queries interact with each other in the prompting-like query matching block, so that the triplets are formed and proceed to **d)** PSG prediction block to output results or compute loss as PSGTR behaves.

PSGFormer, we explicitly model the relation query  $\mathcal{R}_i \in \mathbb{Q}^R$ , as well as object query  $\mathcal{O}_i \in \mathbb{Q}^O$  separately, in hope that object queries to specially pay attentions to objects (*e.g.*, person and phones), and relation queries to focus on the area where the relationship takes place in the picture (*e.g.*, person looking-at phone). Similar to PSGTR in Figure 4, both object and relation queries with CNN features and position encoding are fed into a transformer encoder, but being decoded with their corresponding decoder, *i.e.*, object or relation decoder, so that the queries can learn the corresponding representations.

**Object & Relation Query Matching Block** In PSGFormer, each object query yields an object prediction with FFN and a mask prediction with a panoptic head, and each relation query yields a relation prediction. However, due to the parallel process of object queries and relation queries, the missing interdependence between different query types makes the triplet still not formed. To connect object and relation queries for compositing triplets, we are inspired by the design in HOTR [31] and implement a prompting-like query matching block.

Query matching block models the triplet composition task as a fill-in-the-blank question with prompts, *i.e.*, by prompting a relation, we expect a pair of suitable objects provided by their corresponding object queries can be selected, so that a complete **subject-predicate-object** triplet, can be generated. Therefore, two selectors, *i.e.*, subject selector and object selector, are required.

Given a relation query  $\mathcal{R}_i \in \mathbb{Q}^R$  as prompt, both subject selector and object selector should return the most suitable candidate to form a complete triplet. We use the most standard cosine similarity between object queries and the provided relation query and pick the highest similarity to determine the subject and object candidates. It should also be noticed that subject and object selectors should rely on the level of association between objects and relation queries rather than the semantic similarity. Besides, object queries are regarded as different roles

(*i.e.*, subject or object) in different selectors. Therefore, the object queries are expected to pass another two FFNs to extract some specific information for subject (with FFN denoted as  $f_S$ ) and object (with FFN denoted as  $f_O$ ), so that the distinguishable subject and object representations are obtained from the object queries. With the idea above, a set of subjects  $\mathbb{S}$  are generated in Eq. 7, with the  $i$ -th subject corresponding to the  $i$ -th relation query  $\mathcal{R}_i$ . With a similar process, the object set  $\mathbb{O}$  is also generated.

$$\mathbb{S} = \left\{ \mathcal{S}_i \mid \mathcal{S}_i = \arg \max_{\mathcal{O}} (f_S(\mathcal{O}_j) \cdot \mathcal{R}_i), \mathcal{O}_j \in \mathbb{Q}^O, \mathcal{R}_i \in \mathbb{Q}^R \right\}. \quad (7)$$

Till now, the subject set  $\mathbb{S}$  and the object set  $\mathbb{O}$  are well-prepared by subject and object selectors, with the  $i$ -th subject query  $\mathcal{S}_i$  and the  $i$ -th object  $\mathcal{O}_i$  corresponding to the  $i$ -th relation query  $\mathcal{R}_i$ . Finally, it is straightforward to obtain all the matched triplet  $\mathbb{T}$ , which is shown in Eq. 8.

$$\mathbb{T} = \{(\mathcal{S}_i, \mathcal{R}_i, \mathcal{O}_i) \mid \mathcal{S}_i \in \mathbb{S}, \mathcal{R}_i \in \mathbb{R}, \mathcal{O}_i \in \mathbb{O}\}. \quad (8)$$

Apart from interpreting the query matching as a prompt-like process, it can also be considered as a cross-attention mechanism. For a relation query ( $Q$ ), the goal is to find the high-attention relations among all subject / object representations, which are considered as keys ( $K$ ). The subject / object labels predicted by the corresponding representations are regarded as values ( $V$ ), so that the QKV attention model outputs the labels of selected keys. Fig. 5–c is generally depicted following this interpretation.

**PSG Prediction Block** Similar to PSGTR, with the predicted triplets prepared, the prediction block can finally train the model using  $\mathcal{L}_{\text{total}}$  from Eq. 6. In addition, with object labels and masks predicted by object queries, a standard training loss introduced in panoptic segmentation DETR [4] is used to enhance the object decoder and avoid duplicate object groundings.

## 5 Experiments

In this section, we first report the results of all PSG methods introduced in the paper. Implementation details are available in the appendix, and all codes are integrated in the [OpenPSG](#) codebase, which is developed based on MMDection [7]. Most of the two-stage SGG implementations refer to MMScene-Graph [62] and Scene-Graph-Benchmark.pytorch [56].

### 5.1 Main Results

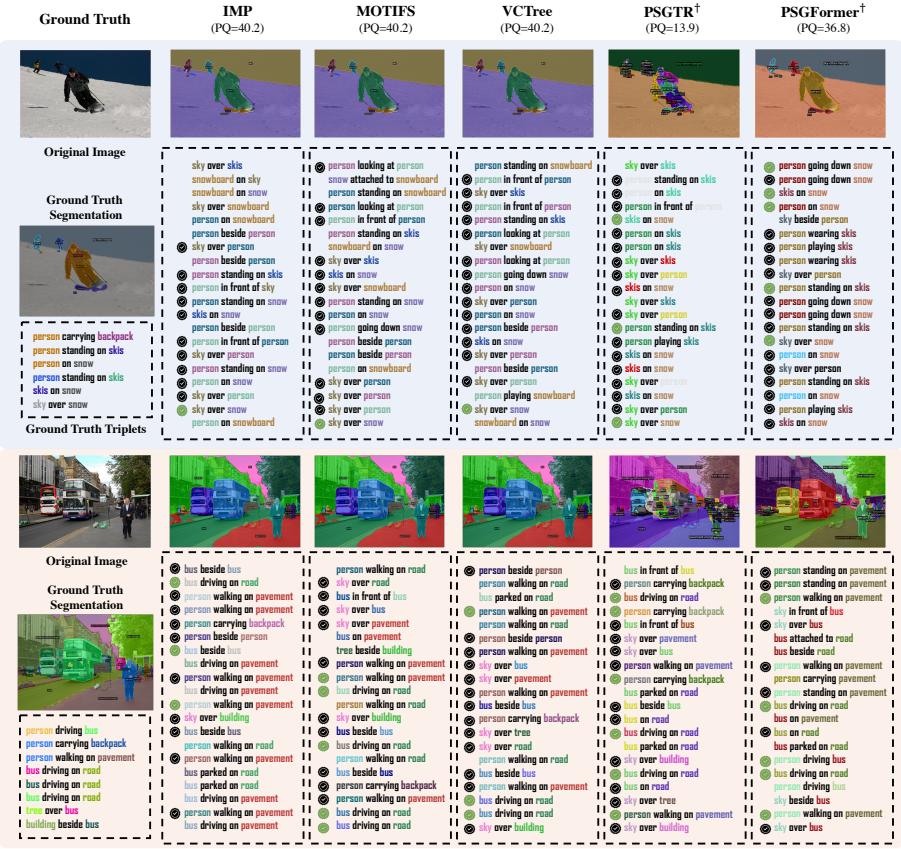
Table 2 reports the scene graph generation performance of all the methods mentioned in Sec. 4.1, Sec. 4.2, and Sec. 4.3 under the PSG dataset. Fig. 6 reports the panoptic segmentation result using PQ and visualizes the segmentation results of two examples as well as the predicted scene graph in the form of triplet lists.

**Table 2: Comparison between all baselines and PSGFormer.** Recall (R) and mean recall (mR) are reported. IMP [64] (CVPR’17), MOTIFS [71] (CVPR’18), VC-Tree [58] (CVPR’19), and GPSNet [44] (CVPR’20) all originate from the SGG task and are adapted for the PSG task. Different backbones of ResNet-50 [20] and ResNet-101 [20] are used. Notice that predicate classification task is not applicable to one-stage PSG models, so the corresponding results are marked as ‘–’. Models are trained using 12 epochs by default.  $\dagger$  denotes that the model is trained using 60 epochs.

Backbone	Method	Predicate Classification			Scene Graph Generation		
		R/mR@20	R/mR@50	R/mR@100	R/mR@20	R/mR@50	R/mR@100
ResNet-50	IMP	31.9 / 9.55	36.8 / 10.9	38.9 / 11.6	16.5 / 6.52	18.2 / 7.05	18.6 / 7.23
	MOTIFS	44.9 / 20.2	50.4 / 22.1	52.4 / 22.9	20.0 / 9.10	21.7 / 9.57	22.0 / 9.69
	VCTree	<b>45.3 / 20.5</b>	<b>50.8 / 22.6</b>	<b>52.7 / 23.3</b>	<b>20.6 / 9.70</b>	<b>22.1 / 10.2</b>	<b>22.5 / 10.2</b>
	GPSNet	31.5 / 13.2	39.9 / 16.4	44.7 / 18.3	17.8 / 7.03	19.6 / 7.49	20.1 / 7.67
	PSGTR	–	–	–	3.82 / 1.29	4.16 / 1.54	4.27 / 1.57
	PSGFormer	–	–	–	<b>16.8 / 14.5</b>	<b>19.2 / 17.4</b>	<b>20.2 / 18.7</b>
	PSGTR $\dagger$	–	–	–	<b>28.4 / 16.6</b>	<b>34.4 / 20.8</b>	<b>36.3 / 22.1</b>
	PSGFormer $\dagger$	–	–	–	18.0 / 14.8	19.6 / 17.0	20.1 / 17.6
ResNet-101	IMP	30.5 / 8.97	35.9 / 10.5	38.3 / 11.3	17.9 / 7.35	19.5 / 7.88	20.1 / 8.02
	MOTIFS	45.1 / 19.9	50.5 / 21.5	52.5 / 22.2	20.9 / 9.60	22.5 / 10.1	23.1 / 10.3
	VCTree	<b>45.9 / 21.4</b>	<b>51.2 / 23.1</b>	<b>53.1 / 23.8</b>	<b>21.7 / 9.68</b>	<b>23.3 / 10.2</b>	<b>23.7 / 10.3</b>
	GPSNet	38.8 / 17.1	46.6 / 20.2	50.0 / 21.3	18.4 / 6.52	20.0 / 6.97	20.6 / 7.17
	PSGTR	–	–	–	3.47 / 1.18	3.88 / 1.56	4.00 / 1.64
	PSGFormer	–	–	–	<b>18.0 / 14.2</b>	<b>20.1 / 18.3</b>	<b>21.0 / 19.8</b>
	PSGTR $\dagger$	–	–	–	<b>28.2 / 15.4</b>	<b>32.1 / 20.3</b>	<b>35.3 / 21.5</b>
	PSGFormer $\dagger$	–	–	–	18.6 / 16.7	20.4 / 19.3	20.7 / 19.7

**Two-Stage Baselines Rely on First-Stage Performance** For predicate classification task (PredCls) that is only applicable to two-stage models, the provided ground-truth segmentation can significantly improve the triplet prediction performance. For example, even the most classic method IMP can reach over 30% R@20, which already exceeds all the available R@20 under the scene graph generation (SGDet) task (*cf.* 28.4% by PSGTR). This phenomenon indicates that a good panoptic segmentation performance could naturally benefit the PSG task. Further evidence where the performance of IMP on the SGDet task is almost halved (from 32% to 17% on R@20) strengthens the above conjecture.

**Some SGG Techniques for VG are not Effective for PSG** Table 2 shows that the results of some two-stage baselines (*i.e.*, IMP, MOTIFS, and VCTree) are generally proportional to their performance on SGG tasks, indicating that the advantages of the two-stage models (*i.e.*, MOTIFS and VCTree) are transferable to PSG task. However, we notice that another two-stage baseline, GPSNet, does not seem to exceed its SGG baselines of MOTIFS and VCTree in the PSG task. The key advantage of GPSNet over MOTIFS and VCTree is that it explicitly models the direction of predicates. While the design can be effective in the VG dataset where many predicates are trivial with obvious direction of predicates (*e.g.*, `of` in `hair-of-man`, `has` in `man-has-head`), PSG dataset gets rid of these predicates, so the model may not be effective as expected.



**Fig. 6: Visualization of segmentations and the top 20 predicted triplets of 5 PSG methods.** The panoptic segmentation metric PQ is also reported. The colors of the subject and object in the triplet corresponds to the mask in the segmentation result. Reasonable triplets are marked by ticks. Triplets that match the ground-truth are marked by green ticks. One-stage models can provide more reasonable and diverse triplet predictions, but they are unable to achieve a good panoptic segmentation result.

**PSGFormer is an Unbiased PSG Model** When the training schedule is limited to 12 epochs, the end-to-end baseline PSGFormer outperforms the best two-stage model VCTree by significant 4.8% on mR@20 and 8.5% on mR@100. Although PSGFormer still cannot exceed two-stage methods on the metrics of R@20/50/100, its huge advantage in mean recall indicates that the model is unbiased in predicting relations. As Fig. 6 shows, PSGFormer can predict unusual but accurate relations such as `person-going down-snow` in the upper example, and the imperceptible relation `person-driving-bus` in the lower example. Also, in the upper example, PSGFormer predicts an interesting and exclusive triplet `person-wearing-skis`. This unique prediction should come from the design of

the separate object / relation decoders, so that relation queries can independently capture the meaning of the predicates.

**PSGTR Obtains SOTA Results with Long Training Time** In PSGTR, every triplet is expected to be captured by a query, which needs to predict everything in the triplet simultaneously, so the model is required to better focus on the connections between objects. Besides, the cross-attention mechanism in the transformer encoder and triplet decoder enable each triplet query access to the information of the entire image. Therefore, PSGTR is considered as the most straightforward and simplest one-stage PSG baseline with minimal constraints or prior knowledge. As a result, although PSGTR only achieves one-digit recall scores in 12 epochs, it surprisingly achieves SOTA results with a prolonged training time of 60 epochs.

## 6 Challenges and Outlook

**Challenges** While some *prior knowledge* introduced by some two-stage SGG methods might not be effective in the PSG task, we expect that more creative knowledge-aided models can be developed for the PSG task in the era of multimodality, so that more interesting triplets can be extracted with extra priors. However, it should be noted that although priors can be useful to enhance performance, the PSG prediction should heavily rely on *visual clues*. For example, in Fig. 6, **person-walking on-pavement** should be identified if the model can perceive and understand the subtle visual differences between **walking** and **standing**. Also, PSG models are expected to *predict more meaningful and diverse relations*, such as rare relations like **feeding** and **kissing**, rather than only being content with statistically common or positional relations.

**Relation between PSG and Panoptic Segmentation** Fig. 6 visualizes the panoptic segmentation results of PSG methods, where PSGTR only obtains a miserable PQ result even with good PSG performance. The reason is that triplet queries in PSGTR produce object groundings independently, so that one object might be referred and segmented by several triplets, and the deduplication or the re-identification (Re-ID) process is non-trivial. Although the performance of Re-ID does not affect PSG metrics, it might still be critical to form an accurate and logical scene understanding for real-world applications.

**Outlook** Apart from attracting more research on the learning of relations (either closed-set or open-set) and pushing the development of scene understanding, we also expect the PSG models to empower more exciting downstream tasks such as visual reasoning and segmentation-based scene graph-to-image generation.

## Acknowledgements

This work is supported by NTU NAP, MOE AcRF Tier 2 (T2EP20221-0033), and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

## References

1. Aditya, S., Yang, Y., Baral, C., Aloimonos, Y., Fermüller, C.: Image understanding using vision and reasoning through scene description graph. Computer Vision and Image Understanding (2018)
2. Amiri, S., Chandan, K., Zhang, S.: Reasoning with scene graphs for robot planning under partial observability. IEEE Robotics and Automation Letters (2022)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. arXiv preprint arXiv:2005.12872 (2020)
5. Chang, X., Ren, P., Xu, P., Li, Z., Chen, X., Hauptmann, A.: Scene graphs: A survey of generations and applications. arXiv preprint arXiv:2104.01111 (2021)
6. Chao, Y.W., Liu, Y., Liu, X., Zeng, H., Deng, J.: Learning to detect human-object interactions. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2018)
7. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al.: Mmdetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
8. Chen, S., Jin, Q., Wang, P., Wu, Q.: Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
9. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
10. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. vol. abs/2107.06278 (2021)
11. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
12. Desai, A., Wu, T.Y., Tripathi, S., Vasconcelos, N.: Learning of visual relations: The devil is in the tails. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
13. Dhamo, H., Farshad, A., Laina, I., Navab, N., Hager, G.D., Tombari, F., Ruprecht, C.: Semantic image manipulation using scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
14. Gadre, S.Y., Ehsani, K., Song, S., Mottaghi, R.: Continuous scene representations for embodied ai. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
15. Gao, C., Xu, J., Zou, Y., Huang, J.B.: Drg: Dual relation graph for human-object interaction detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
16. Gao, L., Wang, B., Wang, W.: Image captioning with scene-graph based semantic concepts. In: Proceedings of the International Conference on Machine Learning and Computing (2018)
17. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object interactions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

18. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
19. Gupta, S., Malik, J.: Visual semantic role labeling. arXiv preprint arXiv:1505.04474 (2015)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
21. Hildebrandt, M., Li, H., Koner, R., Tresp, V., Günnemann, S.: Scene graph reasoning for visual question answering. ICML Workshop Graph Representation Learning and Beyond (GRL+) (2020)
22. Hou, Z., Peng, X., Qiao, Y., Tao, D.: Visual compositional learning for human-object interaction detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
23. Hou, Z., Yu, B., Qiao, Y., Peng, X., Tao, D.: Affordance transfer learning for human-object interaction detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
24. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
25. Hung, Z.S., Mallya, A., Lazebnik, S.: Contextual translation embedding for visual relationship detection and scene graph generation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
26. Johnson, J., Gupta, A., Fei-Fei, L.: Image generation from scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
27. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
28. Kato, K., Li, Y., Gupta, A.: Compositional learning for human object interaction. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
29. Khandelwal, S., Suhail, M., Sigal, L.: Segmentation-grounded scene graph generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
30. Kim, B., Choi, T., Kang, J., Kim, H.J.: Uniondet: Union-level detector towards real-time human-object interaction detection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
31. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: Hotr: End-to-end human-object interaction detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
32. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
33. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
34. Kolesnikov, A., Kuznetsova, A., Lampert, C., Ferrari, V.: Detecting visual relationships using box attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (CVPR-W) (2019)

35. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)* (2017)
36. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval research logistics quarterly* (1955)
37. Li, L., Chen, L., Huang, Y., Zhang, Z., Zhang, S., Xiao, J.: The devil is in the labels: Noisy label correction for robust scene graph generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
38. Li, Y., Ouyang, W., Zhou, B., Shi, J., Zhang, C., Wang, X.: Factorizable net: an efficient subgraph-based framework for scene graph generation. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2018)
39. Li, Y.L., Liu, X., Lu, H., Wang, S., Liu, J., Li, J., Lu, C.: Detailed 2d-3d joint representation for human-object interaction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
40. Li, Y.L., Zhou, S., Huang, X., Xu, L., Ma, Z., Fang, H.S., Wang, Y., Lu, C.: Transferable interactiveness knowledge for human-object interaction detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
41. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Lu, T., Luo, P.: Panoptic segformer. *arXiv preprint arXiv:2109.03814* (2021)
42. Liang, Y., Bai, Y., Zhang, W., Qian, X., Zhu, L., Mei, T.: Vrr-vg: Refocusing visually-relevant relationships. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
43. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2014)
44. Lin, X., Ding, C., Zeng, J., Tao, D.: Gps-net: Graph property sensing network for scene graph generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
45. Liu, Y., Chen, Q., Zisserman, A.: Amplifying key cues for human-object-interaction detection. In: *Proceedings of the European Conference on Computer Vision (ECCV)* (2020)
46. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 852–869 (2016)
47. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: *International Conference on 3D Vision (3DV)* (2016)
48. Peyre, J., Sivic, J., Laptev, I., Schmid, C.: Weakly-supervised learning of visual relations. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2017)
49. Qi, M., Li, W., Yang, Z., Wang, Y., Luo, J.: Attentive relational networks for mapping images to scene graphs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
50. Qi, M., Wang, Y., Li, A.: Online cross-modal scene retrieval by binary representation and semantic graph. In: *Proceedings of the ACM International Conference on Multimedia (ACM MM)* (2017)

51. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* (2015)
52. Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., Manning, C.D.: Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In: *Proceedings of the fourth workshop on vision and language* (2015)
53. Shi, J., Zhang, H., Li, J.: Explainable and explicit visual reasoning over scene graphs. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
54. Suhaeil, M., Mittal, A., Siddiquie, B., Broaddus, C., Eledath, J., Medioni, G., Sigal, L.: Energy-based learning for scene graph generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
55. Tamura, M., Ohashi, H., Yoshinaga, T.: 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 (CVPR)* (2021)
56. Tang, K.: A scene graph generation codebase in pytorch (2020), <https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch>
57. Tang, K., Niu, Y., Huang, J., Shi, J., Zhang, H.: Unbiased scene graph generation from biased training. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
58. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
59. Wang, S., Duan, Y., Ding, H., Tan, Y.P., Yap, K.H., Yuan, J.: Learning transferable human-object interaction detector with natural language supervision. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)
60. Wang, T., Anwer, R.M., Khan, M.H., Khan, F.S., Pang, Y., Shao, L., Laaksonen, J.: Deep contextual attention for human-object interaction detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
61. Wang, T., Yang, T., Danelljan, M., Khan, F.S., Zhang, X., Sun, J.: Learning human-object interaction detection using interaction points. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
62. Wang, W.: Mmscenegraph (2021), <https://github.com/Kenneth-Wong/MMSceneGraph>
63. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019)
64. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017)
65. Xu, P., Chang, X., Guo, L., Huang, P.Y., Chen, X., Hauptmann, A.G.: A survey of scene graph: Generation and application. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)* (2020)
66. Yang, C.A., Tan, C.Y., Fan, W.C., Yang, C.F., Wu, M.L., Wang, Y.C.F.: Scene graph expansion for semantics-guided image outpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022)

67. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
68. Ye, K., Kovashka, A.: Linguistic structures as weak supervision for visual scene graph generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
69. Zareian, A., Karaman, S., Chang, S.F.: Bridging knowledge graphs to generate scene graphs. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
70. Zareian, A., Wang, Z., You, H., Chang, S.: Learning visual commonsense for robust scene graph generation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
71. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
72. Zhang, A., Liao, Y., Liu, S., Lu, M., Wang, Y., Gao, C., Li, X.: Mining the benefits of two-stage and one-stage hoi detection. Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2021)
73. Zhang, F.Z., Campbell, D., Gould, S.: Efficient two-stage detection of human-object interactions with a novel unary-pairwise transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
74. Zhang, H., Kyaw, Z., Chang, S.F., Chua, T.S.: Visual translation embedding network for visual relation detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
75. Zhang, J., Elhoseiny, M., Cohen, S., Chang, W., Elgammal, A.: Relationship proposal networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
76. Zhang, W., Pang, J., Chen, K., Loy, C.C.: K-Net: Towards unified image segmentation. In: Proceedings of Advances in Neural Information Processing Systems (NeurIPS) (2021)
77. Zhong, Y., Shi, J., Yang, J., Xu, C., Li, Y.: Learning to generate scene graph from natural language supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
78. Zhou, T., Wang, W., Qi, S., Ling, H., Shen, J.: Cascaded human-object interaction recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
79. Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., Li, B., Zhang, C., Zhang, C., Wei, Y., 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 (CVPR) (2021)

## A PSG Dataset Details

### A.1 More comparisons between VG and PSG

More comparisons between VG-150 and PSG examples are shown in Fig. A1. In particular, we would like to highlight some specific advances in the PSG dataset from sub-figure (a), which readers can confirm from other sub-figures.

In (a), VG-150 does not contain key information of ‘woman flying kite’, and also has ambiguous relations like ‘at’. Fortunately, PSG addresses the key information but with a more general predicate ‘playing’. It is because in PSG, the predicate definition follows the rule of ‘being representative with proper granularity, not too specific’. Refer to Sec. A.4 for more information. Also, PSG gathers far more comprehensive and accurate triplets.

For object groundings, it is noticeable that in VG, the grounding of ‘beach’ is inaccurate, which only covers half of the actual beach. It can be problematic since a successful recall of a triplet requires a correct matching (big IOU) between predicted groundings and ground truth, in addition to a correct classification of triplets. Therefore, an incorrect annotation on object grounding can cause an inaccurate evaluation on scene graph generation too. Apparently, object grounding of PSG is far more accurate than VG.

### A.2 PSG Dataset Statistics

The PSG dataset has a total of 48,749 annotated images with 56 predicate classes, and 80 thing and 53 stuff classes (same as the COCO dataset [43]).

Here is a list of average statistics for each image:

- 11.0 instances per image
- 5.6 relations per image
- 1.9 (34%) thing-thing relations per image
- 1.2 (21%) stuff-stuff relations per image
- 2.5 (45%) thing-stuff relations per image

### A.3 PSG Dataset Construction Details

**Built on COCO and Visual Genome** In order to create a dataset with both panoptic segmentation *and* relationship annotations, we took advantage of the overlap between the COCO [43] and Visual Genome datasets [35], where they share 48,749 images. Namely, for a given image, it has panoptic segmentation annotations from COCO, as well as scene graph annotations from Visual Genome. However, we cannot merge the two datasets directly as not only do they have different object annotations, they also define different object categories. Therefore, we attempted to conduct the following dataset merging process.



**Fig. A1:** More comparisons between VG-150 and PSG examples. Every sub-figure has VG triplets and their groundings on the left and PSG on the right. Apart from precise pixel-wise grounding, PSG dataset gets rid of trivial relations (*e.g.*, head-on-dog in (d), person-has-hair in (e)), and keep salient ones (*e.g.*, person-holding-remote in (e)). Relations with background are also included (*e.g.*, elephant-walking-on-grass).

**Merging COCO and Visual Genome Annotations** Dataset merging requires solving two intermediate tasks, saying 1) *Object Category Matching*: to figure out a mapping between the object categories in COCO to the object categories in Visual Genome, and 2) *Object Instance Matching*: for each image, to find out which object annotations in COCO correspond to which object annotations in Visual Genome.



**Fig. A2: PSG Dataset Construction Process.** The object categories and annotations in (a) VG-150 are different from that in COCO (image in (b) shows the COCO panoptic segmentation). After matching COCO objects to VG objects using the object matching process, any corresponding relationship annotations can also be transferred over from (a) to (b). These preliminary automatic but noisy relationship annotations are then sent to be processed by a final round of cleaning and annotation to produce the final PSG dataset in (c).

The goal of dataset merging is that, once the matching is achieved, we can transfer over the relationship annotations directly. However, since the matching process will not be completely perfect, we planned to bring these preliminary annotations to experienced and trained annotators for a final round of cleaning and annotation.

With a clear goal of dataset matching, we introduce the details of two intermediate tasks of object category matching and object instance matching. After the matching process, we can get a noisy preliminary PSG dataset (ref. Fig.A5(b)) that awaits the final cleaning.

**Object Category Matching:** For a given COCO object category, what object categories in Visual Genome does it correspond to? For example, “car” in COCO may be matched to both “car” and “vehicle” in Visual Genome.

We encode the text of each object category into a feature vector using fastText [3] word embeddings, and compute a similarity score for each (COCO, Visual Genome) object category pair using their cosine similarity.

This similarity score will be useful for matching object instances in COCO to that in Visual Genome, as described in the next section.

**Object Instance Matching:** The relationship annotations for each image in Visual Genome are tied to Visual Genome object annotations (object categories + bounding boxes), which are different from its corresponding COCO object annotations (object categories + panoptic segmentations). In order to transfer over the relationships to COCO object annotations, we can attempt to match each object instance as annotated in COCO, to an object instance as annotated in Visual Genome. Intuitively, if an object as annotated in COCO has a high overlap with an object as annotated Visual Genome, and their object categories are similar, they are likely to be referring to the same object. A sketch of the algorithm used to perform the matching is as follows. For each image, we:

1. compute the bounding box IoU of each object in COCO to each object in Visual Genome (using the tightest bounding box of the segmentation).
2. we then perform a greedy approach by always considering the instance pair with the highest IoU:
  - (a) If their categories match, i.e. if the similarity score between the word embeddings of their category names are above a certain threshold, we'll **match** the pair together and remove them from the candidate pool.
  - (b) If the categories don't match, we **don't match** them and regard this pair as invalid.
  - (c) Move on to the next object pair with the highest IoU (start from 2. again). Repeat until there are no object pair candidates left, or if the remaining pairs have an IoU of 0.

After the matching, the relationship annotations in Visual Genome can be transferred over to the COCO object annotations. This process is repeated for all the variants VG-150 [64], GQA [24] and VrR-VG [42]. This helps to maximize the recall of potentially correct scene graph relationships and alleviates the difficulty of the final annotation task for the annotators.

**Annotation Process** Building upon the preliminary (noisy) PSG dataset (shown in Fig. A5), we patiently trained our annotators to 1) filter out incorrect triplets, and 2) supplement more relations between not only object-object, but also object-background and background-background pairs, using the predefined 56 predicates. The definition of 56 predicates will be explained in the next section. The noisy triplets (for later filtering) are shown to be a good practice for annotators, prompting them providing both salient and detailed information.

#### A.4 Predicate Dictionary

The design of predicate dictionary is inspired by COCO's practice on object categories selection. According to COCO, the selected categories must be representative, be relevant to practical applications, and be common with high occurrence. Also, a proper level of granularity should also be considered. With these principles in mind, we refer to all the predicates left in the preliminary PSG dataset, sorting them according to their occurrence, and carefully select the predicates.

**Practice for the Principles** To meet the principles mentioned above, several processes are designed:

- **For Representative:** we deduplicate the predicates and try to make the remaining predicates orthogonal. For example, we shrink a list of similar predicates of ‘parked along’, ‘parked alongside’, ‘parked at’, ‘parked behind’, ‘parked beside’, ‘parked by’, ‘parked in’, ‘parked in front of’, ‘parked near’, ‘parked next to’, ‘parked on’ (existed in VG and GQA) to only keep one predicate as ‘parked on’. Also, for the bidirected relation pairs such as ‘in front of’ and ‘behind’, we only keep one direction, *i.e.*, ‘in-front-of’. Similarly, only ‘over’ is included while ‘beneath’ is excluded. With this process, we make our vocabulary very concise and thus representative.
- **For Practicality:** Since the goal of the PSG dataset is to facilitate the development of scene understanding tasks, inspired by VrR-VG [42], we get rid of many positional relations that fill the GQA dataset [24], such as ‘on the left of’ and ‘on the right of’, and especially focus on the visual-related predicates during our dictionary building.
- **For Coverage:** After several iterations, we finally decided to include 56 predicates with property and can well cover almost all the existing critical relations in the PSG dataset.

## 56 Predicates in the Dictionary

- **Positional Relations (6):** over, in front of, beside, on, in, attached to.
- **Common Object-Object Relations (5):** hanging from, on the back of, falling off, going down, painted on.
- **Common Actions (31):** walking on, running on, crossing, standing on, lying on, sitting on, leaning on, flying over, jumping over, jumping from, wearing, holding, carrying, looking at, guiding, kissing, eating, drinking, feeding, biting, catching, picking (grabbing), playing with, chasing, climbing, cleaning (washing, brushing), playing, touching, pushing, pulling, opening.
- **Human Actions (4):** cooking, talking to, throwing (tossing), slicing.
- **Actions in Traffic Scene (4):** driving, riding, parked on, driving on.
- **Actions in Sports Scene (3):** about to hit, kicking, swinging.
- **Interaction between Background (3):** entering, exiting, enclosing (surrounding, warping in).

**Detailed Predicate Definitions** We provided a detailed explanation on each predicate with image examples to ensure the consistent performance from annotators. We will provide the handbook in our PSG website.

## B Implementation Details

All experiments are performed in a single unified codebase using the `MMDetection` framework to facilitate reproducibility.

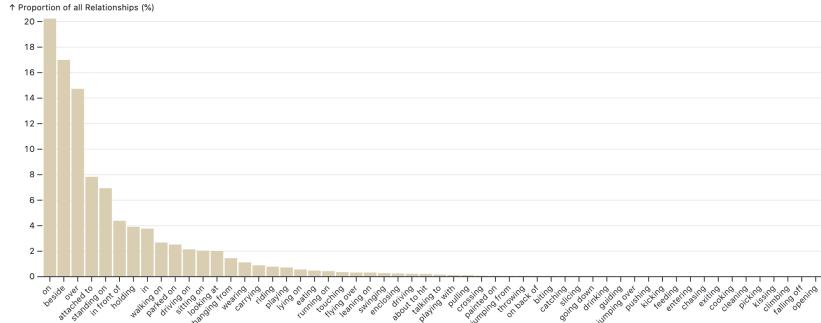
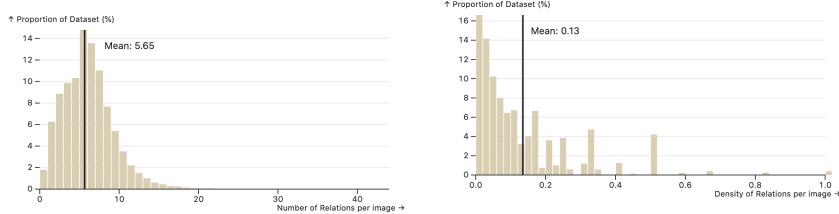


Fig. A3: Proportion of Predicate Classes (Sum=100%).



**Fig. A4: Distribution of the Number of Relations per image in the PSG Dataset.** The bulk of the images have around 5 - 10 relationship annotations, and ranges from 1 - 43 annotations.

**Fig. A5: Distribution of the Density of Relations per image in the PSG Dataset.** The density of relations is defined as the number of annotated relations divided by the total number of possible relations in an image.

### B.1 Two-Stage PSG Baseline

**Fine-tuning the Base Model** We first fine-tune the Panoptic FPN base model on the panoptic segmentation annotations from our PSG dataset. The model is initialized from the best performing pretrained weights provided by MMDetection, and then trained using a batch size of 8. The SGD optimizer is used with a learning rate of 0.02, momentum of 0.9, weight decay of 0.0001, and gradient clipping with a max L2 norm of 35. Training runs for 12 epochs, with a learning rate schedule that linearly warms up from 0.02 / 3 to 0.02 over 500 iterations, and decays by a factor of 0.1 at the 8th and 11th epochs.

**Training the Scene Graph Prediction Head** Using the fine-tuned Panoptic FPN above, we freeze its weights and only train the scene graph prediction head. This essentially treats the Panoptic FPN as a black-box feature extractor and panoptic segmentation predictor. For each predicted object, we extract a feature vector using RoIAlign (like in MaskRCNN), making use of the tightest bounding box around its segmentation mask. With grid features, and class predictions and bounding box localizations for each object at hand, we can feed

these into any scene graph prediction head for training and prediction. We use a batch size of 16, and the SGD optimizer with a learning rate of 0.03, momentum of 0.9, and weight decay of 0.0001, and gradient clipping with a max L2 norm of 35. Training runs for 12 epochs, with a learning rate schedule that linearly warms up from  $0.03 / 3$  to 0.03 over 500 iterations, and decays by a factor of 0.1 at the 7th and 10th epochs. The hyperparameters for the MOTIFS, VCTree and GPSNet models all follow the same settings in their respective papers.

## B.2 PSGTR

As it is described in Section 4.2, our PSGTR model extends DETR to PSG task with new heads and a triplet Hungarian matcher. In detail, we implement each of those Feed Forward Networks (FFNs) by a 3-layer MLP, and each panoptic head, following DETR segmentation, consists of a multi-head attention layer and a 6-layer FPN-like CNN. Besides, the number of queries is set as 100 which indicates that 100 possible relations are predicted.

**Training settings** In general, we follow most of the training strategies of DETR. We adopt the same AdamW optimizer with  $10^{-4}$  learning rate and  $10^{-4}$  weight decay for PSGTR except for the backbone which is trained with learning rate of  $10^{-5}$ . For initialization, we directly use COCO pretrained DETR to initialize the weights of our backbone and transformer. Besides, we also generally follow DETR’s data augmentation which does cropping and resizing operations with settings such that the shortest side is at least 480 and at most 800 pixels while the longest at most 1333. However, it should be noted that when cropping images, we also filter the ground truth of bounding boxes and relations pairs that might be cropped. We train our model for 60 epochs with a step scheduler at epoch 40, and it finally takes us around 2 days to train on eight V100 GPUs with batch size 1.

## B.3 PSGFormer

PSGFormer is built on the baseline of PSGTR so that most of the training details are shared. In detail, PSGFormer also implements each FFN by a 3-layer MLP, and each panoptic head by a multi-head attention layer and a 6-layer FPN-like CNN as DETR does. Besides, the number of object queries and relation queries are set as 100. We follow the training hyperparameters of PSGTR including optimizer, learning rate, data augmentation, etc. Notice that PSGFormer also has an auxiliary task of pure panoptic segmentation with object decoder, the ratio between the main task on triplet supervision and the auxiliary panoptic segmentation supervision is 5 to 1. We train our model for 60 epochs, taking around 2.5 days to train on eight V100 GPUs with batch size 1.

## C Visualization of PSGTR Result Triplet-by-Triplet

Fig. A6 shows PSGTR’s predict results in a triplet-by-triplet fashion, as a complementary to the lower example in Fig. 6. Notice that panoptic segmentation



**Fig. A6: Visualization of PSGTR Result Triplet-by-Triplet.** PSGTR uses triplet queries to directly predict subject / object masks in the triplets, and the subject / object across triplets are not dependent, so the panoptic segmentation visualization is chaos in Fig. 6. However, if we visualize PSGTR result triplet-by-triplet, the result looks good.

visualization is chaos in Fig. 6. It is because PSGTR uses triplet queries to directly predict subject / object masks in the triplets, and the subject / object across triplets are not dependent, and the re-identification of each subject / object is no-trivial, and we use a simple post-processing method of pixel-wise argmax function to merge the segments, but it will still split one object into parts. However, it does not mean that PSGTR cannot segment objects well when predicting triplets. As we visualize the PSGTR result triplet-by-triplet in Fig. A6, the result looks good.