

A Fair Ranking and New Model for Panoptic Scene Graph Generation

Julian Lorenz[⊞], Alexander Pest, Daniel Kienzle[⊞], Katja Ludwig[⊞], and Rainer Lienhart[⊞]

University of Augsburg
Augsburg, Germany

{julian.lorenz,alexander.pest,daniel.kienzle,
katja.ludwig,rainer.lienhart}@uni-a.de

Abstract. In panoptic scene graph generation (PSGG), models retrieve interactions between objects in an image which are grounded by panoptic segmentation masks. Previous evaluations on panoptic scene graphs have been subject to an erroneous evaluation protocol where multiple masks for the same object can lead to multiple relation distributions per mask-mask pair. This can be exploited to increase the final score. We correct this flaw and provide a fair ranking over a wide range of existing PSGG models. The observed scores for existing methods increase by up to 7.4 $mR@50$ for all two-stage methods, while dropping by up to 19.3 $mR@50$ for all one-stage methods, highlighting the importance of a correct evaluation. Contrary to recent publications, we show that existing two-stage methods are competitive to one-stage methods. Building on this, we introduce the Decoupled SceneFormer (DSFormer), a novel two-stage model that outperforms all existing scene graph models by a large margin of +11 $mR@50$ and +10 $mNgR@50$ on the corrected evaluation, thus setting a new SOTA. As a core design principle, DSFormer encodes subject and object masks directly into feature space.

Keywords: Panoptic Scene Graph Generation · Fair Benchmark · Vision Transformer

1 Introduction

In scene graph generation (SGG) [10], the goal is to extract a graph that represents a given image. The nodes of the graph are the objects in the image, identified by their respective bounding box and class label. The edges of the graph are relations between the nodes which contain information about the interaction between the two nodes. A relation usually has a single predicate class assigned which has to be classified by a scene graph model. Panoptic scene graph generation (PSGG) [27] is an extension to SGG and replaces the bounding boxes with panoptic segmentation masks¹. A panoptic scene graph model extracts the

¹ Panoptic segmentation classifies and segments every pixel in an image into semantic categories and instance identities. For more information, please refer to [9]

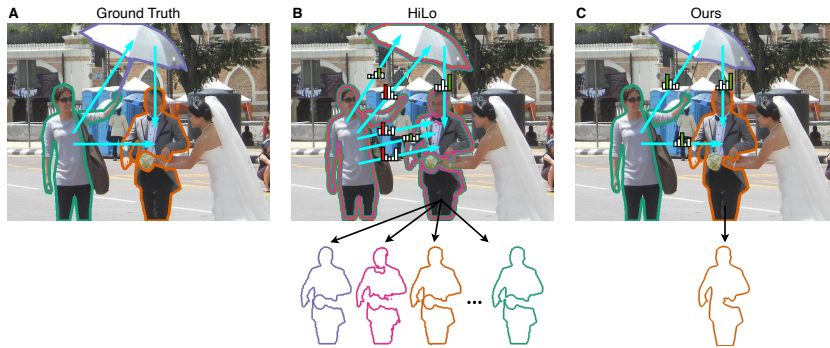


Fig. 1: Schematic comparison of the output from existing one-stage methods (e.g. HiLo, Fig. B) to our proposed two-stage method (Fig. C). One-stage methods often output multiple masks per real world object, visualized with colored masks in Fig. B. This results in one predicate score distribution per mask-mask pair but multiple distributions for pairs that share the same ground truth subject and object. In current evaluation implementations, multiple masks or relations are not aggregated and can therefore be exploited to increase $mR@k$ scores. Our new method does not have this flaw.

segmentation masks, identifies relations between them, and assigns predicate distributions to the relations. Figure 1A shows an example of a panoptic scene graph.

Until now, scene graph models for PSGG have been evaluated using the definition from Yang *et al.* [27] which we will call *Multiple Masks per Object Evaluation Protocol (MultiMPO)*. It shows two problematic peculiarities that can heavily distort the calculated scores. First, masks of a generated scene graph are allowed to contain duplicates. In that case, a 1:1 mapping of the nodes in the graph to the real world is not possible anymore. Second, models are allowed to output multiple predicate distributions for the same subject-object ground truth pair. A model can exploit this to increase the hit chances by predicting the same subject-object pair multiple times with different predicates, which violates the definitions of the applied metrics. We show how to correct these two issues and name the updated and more precise evaluation protocol *Single Mask Per Object Evaluation Protocol (SingleMPO)*. Compared to the old *MultiMPO*, existing PSGG one-stage models achieve much lower $mR@k$ scores than previously anticipated with a decrease of up to 19.3 $mR@k$. Existing two-stage methods are invariant to the choice of the evaluation protocol and can even be boosted by using a suitable state-of-the-art segmentation model upfront.

Recent developments have shifted towards one-stage methods, *i.e.*, inferring the graph and the masks in one pass [24, 27, 31]. However, we show that with ever improving segmentation models, two-stage methods that receive the masks from a SOTA segmentation model are now able to outperform their one-stage counterparts. To demonstrate this, we introduce the *Decoupled SceneFormer (DSFormer)* that is designed from the ground up to process the outputs from a

segmentation model and infer just the scene graph itself. This constraint allows us to use a much simpler network architecture that is easy to train, modify, and significantly outperforms other methods on $mR@20$, $mR@50$, $mNgR@50$, and other metrics by a large margin, setting a new SOTA in PSGG. To summarize, we contribute the following:

1. An analysis of the currently used evaluation protocol for PSGG
2. The new Single Mask Per Object Evaluation Protocol (*SingleMPO*) that corrects the flaws of the currently common evaluation protocol
3. A thorough re-evaluation of existing methods with *SingleMPO*
4. The new Decoupled SceneFormer (DSFormer) two-stage architecture that is easy to train and substantially outperforms current state-of-the-art methods with an increase in $mR@50$ of +11 points

Code for inference, network training, and scripts to evaluate most existing panoptic scene graph models out of the box with *SingleMPO* can be found here: <https://lorjul.github.io/fair-psgg>.

2 Related Work

2.1 Datasets

The most commonly used scene graph dataset is Visual Genome [10], but it lacks segmentation masks in its provided ground truth annotations. In this paper, we are discussing the existing flaws with the *MultiMPO* evaluation protocol for methods in panoptic scene graph generation [27, 28]. Therefore, we focus on the PSG dataset [27]. It contains 48,749 images with panoptic segmentation masks and a total of 273,618 relation annotations. Only about 3% of all relation annotations contain multiple predicates per subject-object pair. Hence, scene graphs are usually evaluated using single-label metrics like *Mean Recall@k* [24, 27, 31]. In the real world, relations often have multiple predicates, *e.g.*, a person can be both holding and drinking a bottle of water. However, panoptic scene graph generation is currently dependent on a single dataset which slows down development in the direction of efficient multi-label training. Recently, the Haystack dataset [18] for PSGG was proposed that tackles some of the multi-label concerns. However, the size of the dataset is only sufficient for scene graph evaluation. Because of these limitations, metrics for PSGG have to be chosen accordingly.

2.2 Metrics

Originally, scene graphs were evaluated using *Recall@k* ($R@k$) [19]: a model selects the top k important relations in an image and assigns a single predicate class to each relation. $R@k$ is calculated as the number of predicates that are correct among the k selected ones, divided by the total number of predicates in the image. However, scene graph datasets have highly imbalanced predicate class frequencies, which is an active field of research [2, 12–14, 30, 31]. In this case, *Mean*

Recall@k ($mR@k$) is a more suitable metric [3]. It is an extension of $R@k$ and is calculated by collecting $R@k$ values per predicate per image and then averaging them at the end. This ensures that frequent predicate classes don't dominate the final score. An extension of $R@k$ is *No Graph Constraint Recall@k* ($ngR@k$) [29]. Instead of only allowing a single predicate class per relation, a model can distribute k predicates over all available relations in an image. Multiple predicates per relation are allowed. Again, the number of correctly assigned predicates is divided by the number of predicates in the ground truth. Analogous to $mR@k$, we use $mNgR@k$ which is calculated by averaging $ngR@k$ over all predicates. The $mNgR@k$ metric can better measure multi-label ground truth and predictions because it allows multiple predicates for the same relation. Recent SOTA methods output multiple relations for the same subject-object pairs. Despite using a single predicate per relation, they effectively assign multiple predicates per subject-object pair, confusing the $mR@k$ metric with something in between $mR@k$ and $mNgR@k$. With our updated evaluation protocol, this confusion is eliminated.

2.3 Existing Methods

In general, scene graph models can be divided into one-stage and two-stage methods. For two-stage methods, subject and object region are given (as a box or as a mask) and the predicate has to be identified. If supported, two-stage methods can also receive the class of subject and object. When publishing the PSG dataset, Yang *et al.* ported four different two-stage architectures to PSGG: IMP [26], MOTIFS [29], GPS-Net [17], and VCTree [23]. We will compare our two-stage approach to these methods. For one-stage methods, a model infers masks for subjects and objects and predicts the predicate for the extracted pairs. The first one-stage methods that were introduced for PSGG are PSGTR [27] and PSGFormer [27] which are based on the DETR [1] architecture and its extension [8] to the Human-Object Interaction task. Pair-Net [24] is another one-stage method that improves PSGFormer by splitting the graph generation process into pair detection, followed by relation classification. The HiLo [31] model tackles the predicate imbalance by efficiently combining rare and common predicate classes during training. Although one-stage methods have been presented with SOTA performance, we will show that there are some caveats when evaluating them. If duplicate masks and relations are prevented, lower evaluation scores are achieved which are surpassed by two-stage models.

Two-stage methods rely on a good performance of the segmentation model in the first stage. For PSGG, we are naturally interested in panoptic segmentation models and will discuss the capabilities of Mask2Former [4], MaskDINO [11], and OneFormer [7]. As shown in Sec. 4.1, choosing a good segmentation model is crucial for two-stage methods and can almost double the performance. Two-stage methods can easily leverage foundation models for image segmentation that have been trained on datasets much larger than the available scene graph datasets. When comparing results with existing two-stage architectures, it is therefore important to use an up-to-date segmentation model for a fair comparison.

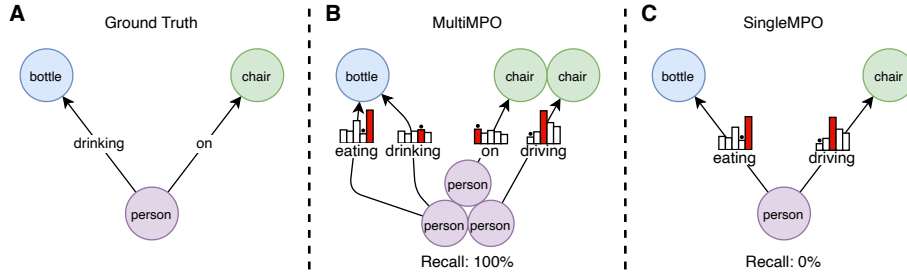


Fig. 2: Schematic comparison of the two considered evaluation protocols. **(A)** The ground truth has a single mask per subject/object. **(B)** There are three different masks for "person" and two for "chair". Keeping them, all ground truth is covered and a recall of 100% is computed by *MultiMPO*, even though the hypothetical model in this example is much more confident with returning *person-eating-bottle* instead of *person-drinking-bottle* and *person-driving-chair* instead of *person-on-chair*. **(C)** Enforcing a single mask per subject/object and a single predicate distribution per subject-object pair reveals the error in predicting the most probable relation.

3 Methods

Following the definition of Yang *et al.* [27], a panoptic scene graph consists of nodes and edges. A node is a segmentation mask with a class label. An edge is a relation between two nodes that usually has a single predicate assigned. Typically, scene graph models predict a predicate class distribution for each relation and select the predicate with the highest confidence when computing a performance metric. We now derive two essential requirements for a correct evaluation of panoptic scene graphs and analyze how they are violated by recent models. We will refer to the currently used evaluation protocol as *Multiple Masks Per Object Evaluation Protocol (MultiMPO)* and to our updated protocol as *Single Mask Per Object Evaluation Protocol (SingleMPO)*.

3.1 Requirements for a Fair Evaluation

Nodes must be unique. We argue that the goal of a potent scene graph model should be to output a connected graph of nodes. Fig. 2B shows that this is not possible with duplicated nodes which arise if a scene graph model outputs multiple masks for the same real world entity. Hence, this must not be allowed. However, *MultiMPO* allows multiple masks to be considered as correct even if they are almost duplicates, as long as the IoU with the ground truth is larger than 50% for each mask. *SingleMPO* on the other hand, only allows a single mask per ground truth subject/object. To merge multiple output masks from a scene graph model, we use a non-maximum suppression-like approach. Given the set of all output masks, we merge them into a set of new masks that do not overlap. At the same time, we keep track of the merge process to reassign

relations to their new masks. As a result, some relations will share the same subject-object combination.

Relations must be unique. Figure 2B shows a model that outputs two predicate distributions per subject-object pair. Even though the model is much more confident with person-eating-bottle than with person-drinking-bottle, it will score a perfect recall with *MultiMPO* because it uses both distributions, which would be incorrect with $mR@k$. Nevertheless, recent scene graph models report this score as $mR@k$, giving them an unfair advantage over models that adhere to the single predicate constraint. With *SingleMPO*, all methods are evaluated equally and the recall for the example in Fig. 2 is correctly calculated as 0%. If multiple predicates are indeed intended, the $mNgR@k$ metric has to be used instead. Nevertheless, even for $mNgR@k$, only a single score per predicate per subject-object pair is allowed and outputting multiple distributions for the same pair would again distort the calculated metric. To aggregate duplicate relations for a specific subject-object pair, *SingleMPO* uses the highest confidence score per predicate and averages the *no-relation* output.

These two postulated requirements ensure that there is no ambiguity when mapping the nodes or relations of a generated scene graph to the real world. In addition, scene graph models cannot gain unfair advantage by outputting multiple predicate distributions per subject-object pair. In contrast to *MultiMPO*, *SingleMPO* ensures that the requirements are always fulfilled.

3.2 Model Overview

Based on the steady improvements on panoptic segmentation methods [4, 7, 11], we decide to use a completely decoupled two-stage approach. The first-stage model is an established segmentation model that outputs segmentation masks and class labels for a given image. Masks and labels are then used as additional inputs to our new Decoupled SceneFormer (DSFormer) architecture. Note that we only have to train the DSFormer scene graph model and not the segmentation model. This approach has four main advantages:

1. Because the segmentation masks have already been inferred, a smaller model can be used to construct the scene graph, lowering hardware requirements and computational cost during training.
2. Two-stage methods directly leverage SOTA foundation models for image segmentation without having to include them in the training pipeline. These models are trained on datasets that are much larger [16, 22, 25] than the available scene graph datasets and will naturally generate superior masks than one-stage scene graph models.
3. Switching to a new segmentation model requires virtually no extra work and no retraining. When comparing new scene graph methods with existing two-stage methods, it is important to use state-of-the-art segmentation models for a fair comparison.
4. Being able to just show selected subject-object pairs to our model during training gives us much more control over sampling strategy and loss weighting.

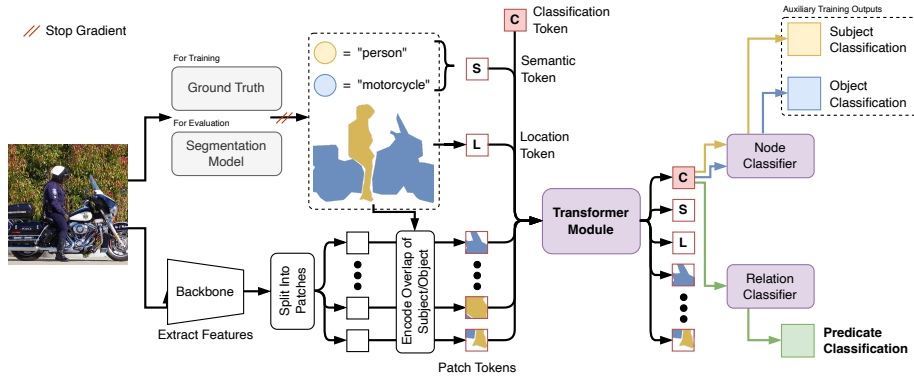


Fig. 3: Our proposed architecture for DSFormer. In a forward pass, the model requires an image, subject and object class, and segmentation masks for subject and object. During training, ground truth data is used. During evaluation, segmentation masks and class labels are inferred from a capable segmentation model. DSFormer outputs a relation prediction as well as an auxiliary subject and object class prediction which are only used during training. Figure 4 shows how the different tokens that enter the transformer module are derived.

An overview of DSFormer’s architecture is depicted in Fig. 3. The model is trained with ground truth segmentation masks and relations. During evaluation, we replace the ground truth masks with inferred masks from a segmentation model that is decoupled from our model. Because DSFormer doesn’t have to construct segmentation masks, its backbone is kept small and we use a ResNet-50 backbone from Faster R-CNN [21] pretrained on object detection. To extract features, we use a feature pyramid network [15] that outputs four different feature tensors with different resolutions, upscale them all to the largest resolution and merge them to one single feature tensor. For an RGB input image of resolution 640×640 , the resulting feature tensor has a shape of $160 \times 160 \times 256$. We split the tensor into non-overlapping patches with a patch size of 8×8 each. Each patch is projected to a token with an embedding dimension of 384. However, before all tokens can be processed by the transformer module, the location of subject and object have to be encoded.

3.3 Subject-Object Encoding

DSFormer has to be prompted with specific subject and object regions for which it will then return a predicate distribution that describes the relation. For that purpose, many two-stage methods (*e.g.* [17,23,26]) utilize the retrieved bounding boxes from the first-stage model and crop the feature tensor using *RoIAlign* [6] or similar methods. Instead of using feature crops to tell DSFormer where the subject and object are located, we keep all the information from the backbone and add a prompt encoding to the patch tokens from the backbone. Therefore,

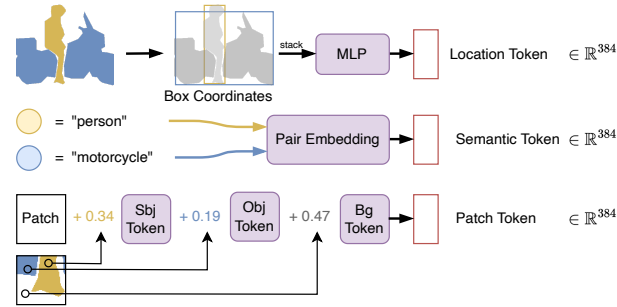


Fig. 4: Most tokens for our proposed model are derived from the segmentation masks. In a patch token, the overlapping ratio of subject and object mask are encoded by adding a weighted sum over learnable subject, object, and background tokens to the initial feature patch. The location token is inferred from the normalized bounding boxes of subject and object using a two-layer MLP. The semantic token is derived directly from subject and object class via a learnable embedding that returns a unique vector for each unique subject-object class combination.

DSFormer can still use global context information that lies outside of the subject or object region for the final decision. For example, an image of a restaurant is more likely to have the predicates "eating" or "drinking".

Figure 4 visualizes how the prompt encoding is computed. It is a weighted sum of three learnable tokens t_{sbj} , t_{obj} , and $t_{bg} \in \mathbb{R}^{384}$ that encodes the presence of subject, object, and background into a specific patch token:

$$token = patch + r_{sbj} \frac{t_{sbj}}{\|t_{sbj}\|} + r_{obj} \frac{t_{obj}}{\|t_{obj}\|} + (1 - r_{sbj} - r_{obj}) \frac{t_{bg}}{\|t_{bg}\|}. \quad (1)$$

r_{sbj} is the ratio that determines how much of the specific patch is covered by the subject mask. r_{obj} is defined respectively. Subject and object mask never overlap, therefore $r_{sbj} + r_{obj} \leq 1$. The learnable tokens are normalized by their magnitude to prevent them from dominating the patch tokens.

3.4 Transformer Module

At the core of DSFormer is a ViT [5] inspired transformer module with 6 layers. It receives the patch tokens and an additional learnable classification token, shown as the filled red box in Fig. 3. The semantic and location tokens shown in Fig. 3 are optional and will be discussed in Sec. 3.7. After the transformer module, the classification token is projected to the desired relation output vector using a two-layer MLP, depicted with green arrows in Fig. 3. The output vector contains a score for each possible predicate plus one additional *no-relation* class. The *no-relation* predicate is a virtual predicate that does not exist in the dataset and is trained to have a high value whenever there is no annotated relation between a subject and an object.

3.5 Relation Loss

During training, we try to penalize false positives less than false negatives, because we cannot reliably evaluate negative ground truth from the PSG dataset. A negative ground truth is present if (a) the predicate does not apply for the relation, (b) the annotator forgot to add the label, or (c) the annotator chose to label the relation with another predicate instead. 97% of all relations in the PSG dataset are annotated with a single label, indicating that option (c) appears frequently. To reduce the impact of incorrect negative ground truth, we choose a loss function that is less sensitive to false positives, shown in Eqs. (2) and (3). Let p be a predicate class out of all P predicate classes and n the sample index in a batch of size N . We denote $y_{n,p}$ as the ground truth, which is either 1 for positive samples or 0 for negative ones. $x_{n,p}$ is the n -th model output for p . Then, $l_{n,p}$ is our used weighted binary cross entropy loss. We weight the positive samples with $w_p = \frac{\# \text{ all neg training samples for } p}{\# \text{ all pos training samples for } p}$. As a consequence, incorrect negative ground truth has a very low impact on the training loss but the model still learns to differentiate between positive and negative because of the abundance of negative ground truth.

$$l_{n,p} = -(w_p y_{n,p} \cdot \log \sigma(x_{n,p}) + (1 - y_{n,p}) \cdot \log(1 - \sigma(x_{n,p}))) \quad (2)$$

$$\mathcal{L}_{rel} = \frac{1}{N \cdot P} \sum_{n=1}^N \sum_{p=1}^P l_{n,p} \quad (3)$$

To provide positive samples for the *no-relation* predicate, we sample additional subject-object pairs without an annotation in the dataset. Each pair is then labelled with a positive *no-relation* ground truth, while the remaining predicates are labelled negative. These pairs are then included in the training set to evenly balance the number of positive and negative *no-relation* samples.

3.6 Auxiliary Node Loss

In addition to the relation loss, we employ an auxiliary node loss that helps the model with better understanding the prompted subject and object regions. Therefore, DSFormer outputs a subject and object classification output that is projected from the transformer module’s classification token by a shared Node Classifier, which is a two-layer MLP. The process is shown in Fig. 3 with yellow and blue lines respectively. For the two outputs, we use cross-entropy loss, weighted with the inverse frequency of each class. The two losses \mathcal{L}_{sbj} and \mathcal{L}_{obj} are then averaged and return the node loss as $\mathcal{L}_{node} = \frac{\mathcal{L}_{sbj} + \mathcal{L}_{obj}}{2}$. The final training loss is a weighted sum of $\mathcal{L} = \lambda_{rel} \cdot \mathcal{L}_{rel} + \lambda_{node} \cdot \mathcal{L}_{node}$ and we choose $\lambda_{rel} = 0.8$ and $\lambda_{node} = 0.2$.

3.7 Additional Input Tokens

Some relations like *on*, *attached to*, *holding* can benefit from additional information about the location of subject and object. Therefore, we derive an addi-

tional location token as shown in Fig. 4 in every forward pass, which is inspired by [20]. We use the inferred subject and object masks of the first-stage model or ground truth to calculate the respective bounding boxes and normalize them to the range of $[-1, +1]$. We concatenate all box coordinates into a single vector $(x_1y_1x_2y_2^{(subj)} x_1y_1x_2y_2^{(obj)})^T \in \mathbb{R}^8$ and project this vector using a two-layer MLP to an additional token for the transformer module.

It has been shown that encoding information about the subject and object classes into scene graph generation increases performance on predicate classification [29]. To encode this semantic information, DSFormer learns a unique token vector for each combination of subject class and object class. During training, the subject and object classes are provided by the ground truth and used to select the correct token. The token is then passed as an additional token to the transformer module. During inference, the subject and object classes are provided by the first-stage model. If DSFormer is intended to work with unknown subject/object classes, the semantic token has to be removed. This slightly decreases performance but enables zero-shot prompting.

3.8 Evaluation

In inference, DSFormer is run on every possible subject-object pair in an image. This can be achieved in reasonable time because the output features from the backbone only have to be inferred once per image. The result is a list of all possible relations with a predicate distribution including the virtual *no-relation* predicate. For *Mean Recall@k* ($mR@k$), a scene graph model has to output a list of k subject-predicate-object triplets per image that have to cover as much ground truth annotation as possible. DSFormer selects the output relations with the k lowest *no-relation* scores. Next, for each selected relation, the argmax over the other predicate scores is used to determine the predicate for the triplet. For *Mean No-Graph-Constraint Recall@k* [29] ($mNgR@k$), a scene graph model again has to output a list of k subject-predicate-object triplets per image, but the same subject-object combination is allowed multiple times as long as the predicates are different. For every predicate p in output relation r , DSFormer combines the estimated predicate score $s_{r,p}$ with the estimated *no-relation* score $s_{r,no}$ of the same relation into a ranking score $x_{r,p}$:

$$x_{r,p} = (1 - \sigma(s_{r,no})) \cdot \sigma(s_{r,p}) \quad \sigma \text{ is the sigmoid function.} \quad (4)$$

Next, DSFormer sorts all $x_{r,p}$ scores within an image and selects the top k ones. The r and p values are used to derive the returned subject-predicate-object triplets.

4 Experiments

In this section, we re-evaluate the different PSGG approaches, including our novel DSFormer. If available, we use published model weights from the authors.

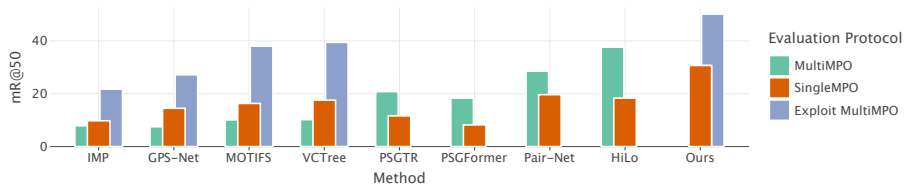


Fig. 5: Comparison of achieved $mR@50$ scores with: (1) originally published unfair *MultiMPO*, (2) our newly introduced fair *SingleMPO*, and (3) a modification of two-stage methods that uses a better mask model and exploits *MultiMPO* similar to some one-stage methods. Even though all methods are evaluated equally, $mR@50$ scores for all one-stage methods decline with a maximum decrease of 19.3 for *SingleMPO*.

Table 1: Performance comparison on the PSG dataset [27] using *SingleMPO* with *Mean Recall@k* ($mR@20$ and $mR@50$) and *Mean No-Graph-Constraint Recall@k* [29] ($mNgR@50$). Higher scores are better. We obtain lower values for PSGTR, PSGFormer, Pair-Net, and HiLo because multiple relation outputs for the same subject-object pair are removed. Missing values indicate that the respective model is a one-stage model and cannot be evaluated on Predicate Classification. The MultiMPO column shows the previously incorrectly calculated $mR@50$ scores. Technically, our DSFormer method inherently uses *SingleMPO*. However, its output can be post-processed to exploit *MultiMPO* (described in the supplementary). This score is shown in parantheses.

Method	Predicate Classification \uparrow			Scene Graph Generation \uparrow			
	mR@20	mR@50	mNgR@50	mR@20	mR@50	mNgR@50	MultiMPO
IMP	11.25	12.72	27.58	8.81	9.78	21.73	7.88
MOTIFS	20.00	21.83	47.98	15.10	16.32	37.96	10.10
GPS-Net	15.46	18.62	33.60	12.35	14.48	27.14	7.49
VCTree	21.19	23.07	50.24	16.29	17.58	39.41	10.20
PSGTR	-	-	-	10.93	11.62	27.57	20.80
PSGFormer	-	-	-	8.20	8.20	21.75	18.30
Pair-Net	-	-	-	18.02	19.64	21.48	28.50
HiLo	-	-	-	17.51	18.33	40.48	37.60
Ours	34.03	40.06	64.05	27.20	30.67	50.08	(50.08)

If not, we train the models as described in the respective publication. We then generate predictions using *SingleMPO*, *i.e.*, by merging segmentation masks that describe the same visual object and by deduplicating relations between the same subject-object pair as described in Sec. 3.1. Next, we evaluate the various approaches by using $mR@k$ and $mNgR@k$ on the usual Predicate Classification (*PredCls*) and Scene Graph Generation (*SGGen*) tasks and show the results in Tab. 1. In *SGGen*, a model has to infer subject/object masks, class labels, and relations on its own. In *PredCls*, ground truth masks and mask labels are provided and just the correct predicate classes have to be retrieved for each relation. One-stage methods cannot be prompted with given subject/object masks and can consequently not be evaluated on *PredCls*.

As can be seen in Tab. 1, we observe about the same *PredCls* scores for existing two-stage methods as reported in [27] where the less rigorous *MultiMPO* was used. The consistent scores are expected because these two-stage methods do

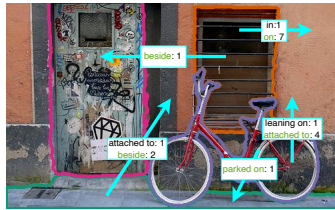


Fig. 6: Example output of DSFormer. The numbers indicate how DSFormer sorts the predicates within a relation. More images are shown in the supplementary.

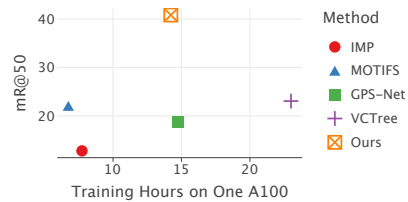


Fig. 7: Comparison of training times of existing two-stage methods on a single A100 GPU. Our method achieves a much higher $mR@50$ while keeping the training time comparably low.

not output overlapping masks and don’t exploit duplicate relation predictions. Thus, they already inherently use *SingleMPO*. On *SGGen*, our reported values differ greatly from the original work. Figure 5 shows how heavily the $mR@50$ scores can be distorted, if the wrong evaluation protocol is selected. Existing one-stage methods output multiple masks per ground truth and duplicate relations which should not be allowed for the final metric. After merging the masks, PSGTR contains on average 4.19 duplicate relations per image that are removed in *SingleMPO*, PSGFormer has 89.36, Pair-Net has 23.44, and HiLo has 36.08. Aggregating these duplicates as described in Sec. 3.1 reveals that $mR@50$ scores for all one-stage models decline with a maximum decrease of 19.3 $mR@50$ lower than previously reported. Existing two-stage methods on the other hand are not affected because they already adhere to *SingleMPO* as discussed before. For a fairer comparison with SOTA models, we replace the first-stage model for every two-stage method with the top-performing MaskDINO segmentation model to obtain better segmentation masks. This increases $mR@50$ of *VCTree* by 7.4 and almost doubles it for *GPS-Net*. We will discuss the choice of first-stage model in more detail in Sec. 4.1.

Contrary to recent developments, we demonstrate that two-stage methods outperform one-stage methods easily in a fair comparison. Our DSFormer model achieves SOTA performance on all reported metrics, with +11 $mR@50$ and +10 $mNgR@50$ compared to the previous state-of-the-art on *SGGen*. Compared to one-stage models, DSFormer increases $mR@50$ by more than 50%. On top of the outstanding performance, training DSFormer is fast as can be seen in Fig. 7.

4.1 Influence of First-Stage Models

When evaluating two-stage methods on PSGG, a good segmentation model is essential for a good overall scene graph performance. Figure 8 shows how $mR@50$ and $mNgR@50$ performance on *SGGen* is directly proportional to $mR@50$ and $mNgR@50$ performance on *PredCls*, regardless of the used segmentation model. Existing two-stage methods are not specifically targeted towards certain segmentation models, which indicates that the first stage can be swapped easily.

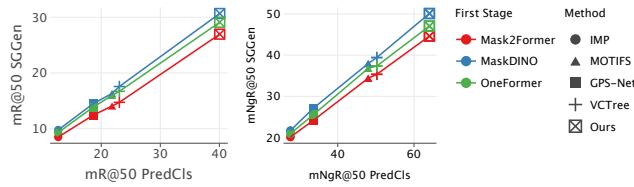


Fig. 8: Performance on *PredCls* (without first-stage model) is directly proportional to *SGen* (with first-stage model) apart from small fluctuations. For all tested two-stage methods, MaskDINO works best.

In Fig. 9, we compare the Mask2Former [4], MaskDINO [11], and OneFormer [7] segmentation models. In addition, we use the segmentation outputs of the PSGTR, PSGFormer, Pair-Net, and HiLo one-stage models. If combined with DSFormer, a segmentation model with a high Panoptic Quality (PQ)² [9] enables a better $mR@50$ in general. HiLo is an outlier and generates segmentation masks with a low PQ which nevertheless help DSFormer to reach a good $mR@50$ of 26.89. To explain this behavior, we design a measure called $mR@inf$. This metric pretends that there is a perfect scene graph model after the segmentation model (or a model that has $k = \infty$ guesses). Given the extracted segmentation masks, $mR@inf$ calculates what the best $mR@k$ for any k would be. A segmentation model with high $mR@inf$ is good at retrieving masks that are relevant to improve the $mR@k$ metric. Again, Fig. 9 shows a correlation between $mR@inf$ and $mR@50$ with DSFormer. In fact, HiLo achieves the best $mR@inf$ compared to all other models. A perfect scene graph model would perform best on segmentation masks from HiLo. However, we suspect that segmentation models with high $mR@inf$ but low PQ would be very good in theory but they make it more difficult to correctly prompt a subsequent scene graph model. To analyze segmentation models, PQ and $mR@inf$ should always be used together. The best segmentation models that enable the highest $mR@50$ for DSFormer are MaskDINO with a score of 30.67, followed by OneFormer (29.10), and Mask2Former (26.97). HiLo is the only one-stage model that gets close with a score of 26.89.

4.2 Ablation Study

The effect of our introduced components can be seen in Tab. 2. As a baseline, we use DSFormer that receives rectangular segmentation masks, derived from the bounding boxes. Adding full mask information, as well as semantic and location information via additional tokens as described in Sec. 3.7, increases performance when applied individually or in combination. In addition, we observe that removing the auxiliary loss from Sec. 3.6, reduces $mR@50$ from 40.06 to 34.18, highlighting the need for additional guidance on subject/object information. A more extensive ablation study can be found in the supplementary.

² PQ is a standard metric for panoptic segmentation that measures segmentation quality and recognition quality.

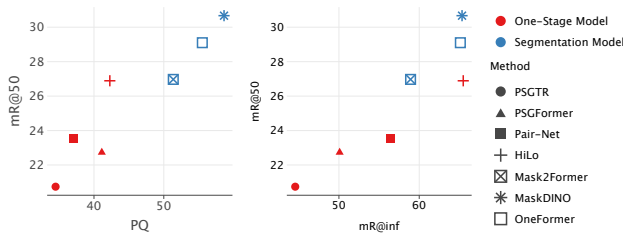


Fig. 9: Performance of the used first-stage models on the PSG dataset. In addition to Panoptic Quality (PQ), we report $mR@inf$ as the highest possible mean recall that a subsequent scene graph model can possibly reach with the inferred masks. For comparison, we interpret the extracted masks from HiLo, Pair-Net, PSGTR, and PSGFormer as a first-stage model. The y-axis shows the best $mR@50$ achieved with DSFormer as the second stage. All shown $mR@50$ scores outperform previous SOTA scores.

Table 2: Improvements caused by the additional semantic and location token. All reported values are calculated for Predicate Classification (PredCls) on the PSG dataset. For the first row, we replace the input segmentation masks for DSFormer with rectangular masks, derived from bounding boxes.

Masks	Semantic Token	Location Token	$mR@50$	$mNgR@50$
×	×	×	34.35	53.48
✓	✓	×	36.64	58.36
✓	×	✓	38.78	61.75
✓	✓	✓	40.06	64.05

5 Conclusion

We have identified unexpected and undesirable effects with the current evaluation protocol of panoptic scene graph generation and discussed the requirements that a good evaluation protocol should fulfil. We believe that our updated *SingleMPO* more accurately captures what makes a good panoptic scene graph model and suggest switching to our protocol instead. Furthermore, we showed that if we correct the current flaws in the evaluation, existing one-stage methods achieve much lower scores than previously reported whereas two-stage methods confirm their reported scores. We introduced DSFormer, a new two-stage architecture that is completely decoupled from the used segmentation model. It can be prompted with subject and object masks from any segmentation system. It uses a specialized patch encoding and outperforms all other scene graph models with a $mR@50$ of 30.67 (+11) and a $mNgR@50$ of 50.08 (+10), thus setting a new SOTA performance. To further improve its performance, DSFormer could be pretrained or extended with external knowledge [12, 14, 32]. Future work should also investigate on how to leverage information between relation pairs.

As panoptic segmentation models progress, we advocate for two-stage scene graph methods as promising candidates for future top-performing PSGG methods. Selecting an up-to-date first-stage segmentation model is crucial for fair comparisons, as *SGGen* scores will improve accordingly.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) *Computer Vision – ECCV 2020*. pp. 213–229. Springer International Publishing, Cham (2020)
2. Chen, L., Song, Y., Cai, Y., Lu, J., Li, Y., Xie, Y., Wang, C., He, G.: Multi-prototype space learning for commonsense-based scene graph generation. *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(2), 1129–1137 (Mar 2024). <https://doi.org/10.1609/aaai.v38i2.27874>, <https://ojs.aaai.org/index.php/AAAI/article/view/27874>
3. Chen, T., Yu, W., Chen, R., Lin, L.: Knowledge-embedded routing network for scene graph generation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 6156–6164 (2019). <https://doi.org/10.1109/CVPR.2019.00632>
4. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. *arXiv* (2021)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houshy, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations* (2021), <https://openreview.net/forum?id=YicbFdNTTy>
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. pp. 2980–2988 (2017). <https://doi.org/10.1109/ICCV.2017.322>
7. Jain, J., Li, J., Chiu, M., Hassani, A., Orlov, N., Shi, H.: Oneformer: One transformer to rule universal image segmentation. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2989–2998 (2023). <https://doi.org/10.1109/CVPR52729.2023.00292>
8. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: HOTR: End-to-end human-object interaction detection with transformers. In: *CVPR. IEEE* (2021)
9. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 9396–9405 (2019). <https://doi.org/10.1109/CVPR.2019.00963>
10. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* **123**(1), 32–73 (2017). <https://doi.org/10.1007/s11263-016-0981-7>, <https://doi.org/10.1007/s11263-016-0981-7>
11. Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y.: Mask DINO: Towards a unified transformer-based framework for object detection and segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 3041–3050 (June 2023)
12. Li, L., Ji, W., Wu, Y., Li, M., Qin, Y., Wei, L., Zimmermann, R.: Panoptic scene graph generation with semantics-prototype learning. *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(4), 3145–3153 (Mar 2024). <https://doi.org/10.1609/aaai.v38i4.28098>, <https://ojs.aaai.org/index.php/AAAI/article/view/28098>
13. Li, L., Qin, Y., Ji, W., Zhou, Y., Zimmermann, R.: Domain-wise invariant learning for panoptic scene graph generation. In: *ICASSP 2024 - 2024 IEEE International*

- Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3165–3169 (2024). <https://doi.org/10.1109/ICASSP48485.2024.10447193>
14. Liang, N., Liu, Y., Sun, W., Xia, Y., Wang, F.: Ckt-rcm: Clip-based knowledge transfer and relational context mining for unbiased panoptic scene graph generation. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3570–3574 (2024). <https://doi.org/10.1109/ICASSP48485.2024.10446810>
 15. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 936–944 (2017). <https://doi.org/10.1109/CVPR.2017.106>
 16. 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: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)
 17. 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) (June 2020)
 18. Lorenz, J., Barthel, F., Kienzle, D., Lienhart, R.: Haystack: A panoptic scene graph dataset to evaluate rare predicate classes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 62–70 (October 2023)
 19. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. pp. 852–869. Lecture Notes in Computer Science, Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46448-0_51
 20. Ludwig, K., Harzig, P., Lienhart, R.: Detecting arbitrary intermediate keypoints for human pose estimation with vision transformers. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW). pp. 663–671 (2022). <https://doi.org/10.1109/WACVW54805.2022.00073>
 21. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(6), 1137–1149 (2017). <https://doi.org/10.1109/TPAMI.2016.2577031>
 22. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8429–8438 (2019). <https://doi.org/10.1109/ICCV.2019.00852>
 23. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: Conference on Computer Vision and Pattern Recognition (2019)
 24. Wang, J., Wen, Z., Li, X., Guo, Z., Yang, J., Liu, Z.: Pair then relation: Pair-Net for panoptic scene graph generation. <https://doi.org/10.48550/arXiv.2307.08699>, <http://arxiv.org/abs/2307.08699>
 25. Wang, W., Shi, M., Li, Q., Wang, W., Huang, Z., Xing, L., Chen, Z., Li, H., Zhu, X., Cao, Z., et al.: The all-seeing project: Towards panoptic visual recognition and understanding of the open world. arXiv preprint arXiv:2308.01907 (2023)
 26. Xu, D., Zhu, Y., Choy, C., Fei-Fei, L.: Scene graph generation by iterative message passing. In: Computer Vision and Pattern Recognition (CVPR) (2017)

27. Yang, J., Ang, Y.Z., Guo, Z., Zhou, K., Zhang, W., Liu, Z.: Panoptic scene graph generation. In: ECCV (2022)
28. Yang, J., CEN, J., PENG, W., Liu, S., Hong, F., Li, X., Zhou, K., Chen, Q., Liu, Z.: 4d panoptic scene graph generation. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) *Advances in Neural Information Processing Systems*. vol. 36, pp. 69692–69705. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/dc6319dde4fb182b22fb902da9418566-Paper-Conference.pdf
29. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: *Conference on Computer Vision and Pattern Recognition* (2018)
30. Zhang, A., Yao, Y., Chen, Q., Ji, W., Liu, Z., Sun, M., Chua, T.S.: Fine-grained scene graph generation with data transfer. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022*. pp. 409–424. Springer Nature Switzerland, Cham (2022)
31. Zhou, Z., Shi, M., Caesar, H.: HiLo: Exploiting high low frequency relations for unbiased panoptic scene graph generation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 21637–21648 (October 2023)
32. Zhou, Z., Shi, M., Caesar, H.: Vlprompt: Vision-language prompting for panoptic scene graph generation. arXiv preprint arXiv:2311.16492 (2023)

A Definition of Mean Recall@k

Calculating the discussed scene graph metrics with *SingleMPO* can be split into two steps. First, a set of subject-predicate-object triplets is retrieved from the model output. Second, the predicted segmentation masks are matched with the ground truth. The set of matched output triplets is used to calculate the recall-based metrics.

$mR@k$ and $mNgR@k$ are defined per image. To calculate the score over the whole dataset, the per-image metric scores are averaged.

For a given image, we define

- M_{gt} : Set of ground truth masks that describes the visual objects in an image.
- M_{out} : Set of predicted masks
- P : Set of all possible predicate classes in the dataset. For example, the PSG dataset contains 56 predicate classes.
- G : Set of ground truth subject-predicate-object triplets. We define such triplets as $(t_{sbj} \in M_{gt}, t_{pred} \in P, t_{obj} \in M_{gt})$.
- X_k : Set of top k subject-predicate-object triplets. Triplets are defined as $(t_{sbj} \in M_{out}, t_{pred} \in P, t_{obj} \in M_{out})$. The model decides what the top k triplets are. For example, DSFormer uses its *no-relation* output.

A.1 Mask Matching

To calculate the metrics, we match the predicted segmentation masks to the ground truth, such that each ground truth mask has at most one predicted mask assigned to. If no predicted segmentation mask overlaps with a ground truth mask with an IoU greater than 0.5, no predicted mask is assigned to the ground truth. In the following, this mapping is called L . See Algorithm 1 for an explanation of the matching process. In the process, some masks from M_{out} cannot be matched. Any predicted relations that are connected to those unassigned masks are discarded.

Algorithm 1 Mask Matching

- 1: **Input:** Predicted masks M_{out} , ground truth masks M_{gt} , minimum IoU threshold t
 - 2: **Output:** Lookup table L that maps masks from M_{out} to masks in M_{gt}
 - 3: **procedure** MATCHING(M_{out}, M_{gt}, t)
 - 4: Initialize lookup table J to $J[x] = null \forall x \in M_{gt}$
 - 5: **for all** m in M_{out} **do**
 - 6: $x \leftarrow \arg \max_{g \in M_{gt}} iou(g, m)$
 - 7: **if** $iou(x, m) > t$ **then**
 - 8: **if** $J[x]$ is *null* **or** $iou(x, m) > iou(x, J[x])$ **then**
 - 9: $J[x] \leftarrow m$
 - 10: $L \leftarrow J^{-1}$ ▷ Use the inverse mapping of J
 - 11: **return** L
-

Algorithm 2 Mean Recall for a Single Image

```

1: Input: Set of all predicate classes  $P$ , ground truth triplets  $G$ ,
2: lookup table  $L: M_{out} \rightarrow M_{gt}$  (Algorithm 1), top  $k$  predicted triplets  $X_k$ 
3: Output: Mean Recall@k
4: procedure MEANRECALL( $P, G, L, X_k$ )
5:   for all  $p$  in  $P$  do
6:      $G^{(p)} \leftarrow \{t \in G \mid t_{predicate} = p\}$ 
7:      $X^{(p)} \leftarrow \{\}$ 
8:     for all  $t$  in  $X_k$  do ▷ Match predicted masks to ground truth
9:       if  $t_{subj} \in L$  and  $t_{obj} \in L$  then
10:         $p \leftarrow t_{predicate}$ 
11:         $t' \leftarrow (L[t_{subj}], p, L[t_{obj}])$ 
12:         $X^{(p)} \leftarrow X^{(p)} \cup t'$ 
13:   return  $\frac{1}{|P|} \sum_{p \in P} \frac{|G^{(p)} \cap X_k^{(p)}|}{|G^{(p)}|}$ 
▷ For the whole dataset, calculate Mean Recall for every image, then average

```

A.2 Metric Definitions for mR@k and mNgR@k

A scene graph model usually returns a set of predicate distributions per subject-object pair and can be sorted and converted to a set of the top k important subject-predicate-object triplets X_k . How the most probable triplets are selected is up to the model and not part of the metrics. Using the matching process described in Appendix A.1, X_k can be matched to ground truth segmentation masks. If a subject/object mask cannot be matched, the whole triplet is removed. After the matching, X'_k is the matched set of triplets with the matched segmentation masks and the predicted predicate classes.

We define $G^{(p)} \subset G$ as the ground truth subset that only contains triplets with predicate p . The model output subset $X_k^{(p)} \subset X'_k$ contains only triplets with predicate p . Therefore $\bigcup_{p \in P} X_k^{(p)} = X'_k$.

For Mean Recall@k ($mR@k$), the matched model output X'_k **must not contain any two triplets that share the same subject and object**. This constraint is not fulfilled with *MultiMPO* and thus leads to incorrect metric scores. $mR@k$ is defined as:

$$mR@k = \frac{1}{|P|} \sum_{p \in P} \frac{|G^{(p)} \cap X_k^{(p)}|}{|G^{(p)}|} \quad (5)$$

Mean No Graph Constraint Recall@k ($mNgR@k$) is calculated like $mR@k$ except that in contrast to $mR@k$, the matched model output X'_k may contain two or more triplets that share the same subject and object as long as the predicates are different.

Algorithm 2 shows how Mean Recall is calculated.

Algorithm 3 Convert SingleMPO Output to MultiMPO Output

```

1: Input: List  $R$  of relation outputs; set of predicate classes  $P$  (excluding the no-
   relation class)
2: Output: Modified list  $R'$  for MultiMPO
3: procedure CONVERT( $R, P$ )
4:   Initialize empty list  $R'$ 
5:   for all relation  $r$  in  $R$  do
6:     for all predicate  $p$  in  $P$  do
7:       Initialize new relation  $r'$ 
8:        $r'[no-rel] \leftarrow (1 - (1 - r[no-rel]) \cdot r[p])$   $\triangleright$  Assign no-graph-constraint score
9:        $r'[p] \leftarrow 1$   $\triangleright$  This ensures that the argmax is  $p$ 
10:      for all predicate  $q$  in  $P \setminus p$  do
11:         $r'[q] \leftarrow 0$ 
12:      Add  $r'$  to  $R'$ 
13:   return  $R'$ 

```

B Evaluating on MultiMPO With DSFormer

DSFormer adheres to the *SingleMPO* evaluation protocol by default. As we have discussed, it does not make sense to use *MultiMPO* for evaluation. However, to prove that *MultiMPO* can be used to gain an unfair advantage over models that already inherently adhere to *SingleMPO*, we can post-process the *SingleMPO* output of DSFormer and convert it to *MultiMPO* as depicted in Algorithm 3. For each predicate p in each relation $r = (r_{no-rel}, r_1, \dots, r_P)$, we create a new relation $r' = (r'_{no-rel}, r'_1, \dots, r'_P)$ that has a score of $r'_p = 1$ for predicate p and a *no-relation* score of $r'_{no-rel} = 1 - ((1 - r_{no-rel}) \cdot r_p)$. DSFormer uses the *no-relation* score to determine the top k triplets for the recall metrics. With the new constructed *no-relation* score, the top k mNgR@k triplets are now the top k mR@k triplets (only when using *MultiMPO*).

C Model Architecture Details

C.1 Parameters

If not specified otherwise, we used the following parameters during training: For the transformer module, we use 6 transformer layers with an embedding dimension of 384. We add a 2D-sine positional encoding in every layer as in [2]. We use a batch size of 32 and train with AdamW [1] with a learning rate of 3.7×10^{-5} and weight decay of 0.04. The subject/object encoding (discussed in Sec. 3.3) is added once to the patch tokens before they enter the transformer module. We resize the input images to a resolution of 640×640 .

C.2 Inference Speed

For n masks, $n^2 - n$ relations must be classified to generate a complete scene graph. Existing one-stage methods circumvent this issue by limiting the number of relations to a fixed number (usually 100 relations per image), resulting

in incomplete scene graphs. With DSFormer on the other hand, we choose to generate the complete scene graph because we expect it to be more useful for downstream tasks. In practice, this approach is feasible as shown in Tab. 3. On a single NVIDIA A100, our implementation can process about 2400 relations in one forward pass, which is sufficiently fast. Additional relations can be processed sequentially without computing the feature tensor again. Only 0.2% of all images in the PSG dataset have to be split.

Table 3: Comparison of number of learnable parameters and required time to run inference on the full test set. Each method was evaluated on a single A100 GPU.

Method	# Parameters	Inference
IMP	78M	2.2 min
MOTIFS	108M	1.3 min
GPS-Net	82M	3.4 min
VCTree	104M	4.4 min
PSGTR	44M	4.8 min
PSGFormer	52M	3.9 min
Pair-Net	54M	3.6 min
HiLo	230M	15.2 min
Ours	50M	8.4 min

C.3 Location Token

Given a segmentation mask for the subject and a segmentation mask for the object, we calculate two bounding boxes. We use x_1, y_1, x_2, y_2 to denote left, top, right, and bottom coordinates of the bounding box. To normalize them, we divide by the image width w or height h and normalize the coordinates to the range of $[-1, +1]$:

$$x'_1 = 2 \cdot \frac{x_1}{w} - 1 \quad (6)$$

$$x'_2 = 2 \cdot \frac{x_2}{w} - 1 \quad (7)$$

$$y'_1 = 2 \cdot \frac{y_1}{h} - 1 \quad (8)$$

$$y'_2 = 2 \cdot \frac{y_2}{h} - 1 \quad (9)$$

The normalized bounding box coordinates for subject and object are stacked to a vector of length 8. This vector is passed to a two-layer MLP which projects the coordinate vector to the desired embedding token. For the size of the hidden layer, we use half the embedding dimension.

C.4 Binary Subject-Object Encoding

Eq. 1 shows how subject and object location are encoded into the patch tokens before they are processed by DSFormer’s transformer module. We use r_{sbj} and r_{obj} to represent the ratio of how much a specific patch is covered by the respective segmentation mask. Alternatively, we can replace the ratios with binary values that are 1 if the patch is covered partially by the segmentation mask and 0 otherwise. However, we did not observe significant performance improvements with this modification.

C.5 Pretraining

For a fair comparison, we did not pretrain DSFormer on precursor tasks. However, DSFormer can be pretrained on segmentation datasets by disabling the relation classifier and just classifying pairs of subject and object using the node classifier.

D Ablation Study

This section contains additional results of our ablation study.

Table 4 shows how the individual components of DSFormer benefit the overall score. In the table, a cross (\times) in the Masks column indicates that the actual segmentation mask was replaced by a rectangular segmentation box of the size of the related bounding box. If these rectangular masks are used, adding the semantic token results in a greater improvement than adding the location token. However, if the actual masks are used, the location token is more important. We assume that rectangular mask and location token encode the same information in two different ways and therefore adding the location token merely improves the model ($+0.08mR@50$). On the other hand, actual segmentation mask and location token encode different information. Consequently, adding the location token drastically improves performance ($+6.3mR@50$).

Table 5 shows the importance of our auxiliary node loss (Sec. 3.6). Without it, the performance degrades. However, adding too much auxiliary loss also degrades performance.

Table 6 shows that with increasing embedding dimension, performance is improved. However, the improvements begin to converge with higher embedding dimension sizes.

Table 6: Embedding dimension size for the transformer module. All experiments were run 3 times. The number behind the \pm sign shows the standard deviation.

Embedding Dimension	mR@50	mNgR@50
192	38.01 \pm 1.27	60.67 \pm 1.47
256	39.24 \pm 1.00	62.88 \pm 0.31
384	40.06 \pm 0.56	64.05 \pm 1.91

Table 4: Components. A cross (×) in the Masks column means that instead of an actual segmentation mask, the enclosing bounding box region was used to create a rectangular segmentation mask. All experiments were run 3 times. The number behind the ± sign shows the standard deviation.

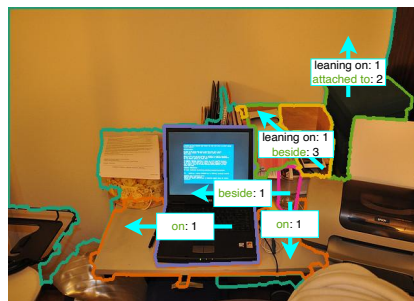
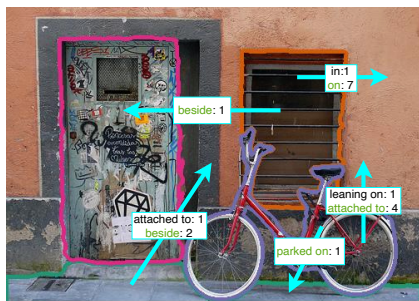
Masks	Location Token	Semantic Token	mR@50	mNgR@50
×	×	×	33.75 ± 0.67	54.47 ± 1.12
×	×	✓	36.52 ± 0.55	56.34 ± 0.96
×	✓	×	33.83 ± 1.15	53.97 ± 1.16
×	✓	✓	38.80 ± 0.79	60.77 ± 1.54
✓	×	×	32.48 ± 0.33	51.80 ± 0.36
✓	×	✓	36.64 ± 0.47	58.36 ± 1.28
✓	✓	×	38.78 ± 0.70	61.75 ± 0.71
✓	✓	✓	40.06 ± 0.56	64.05 ± 1.91

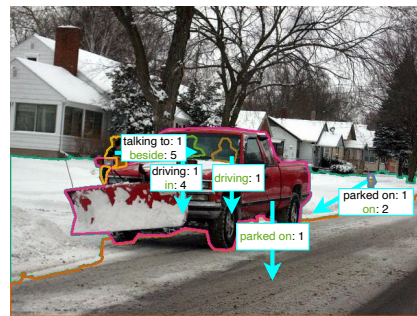
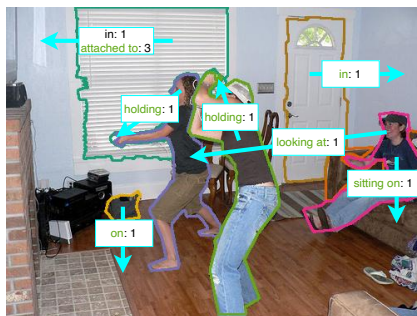
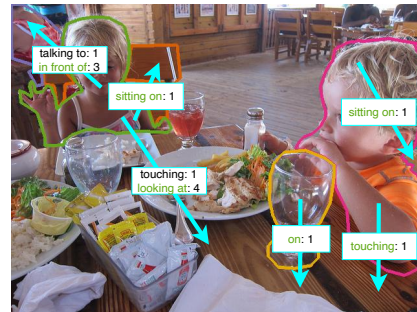
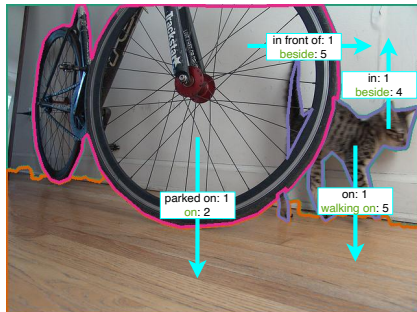
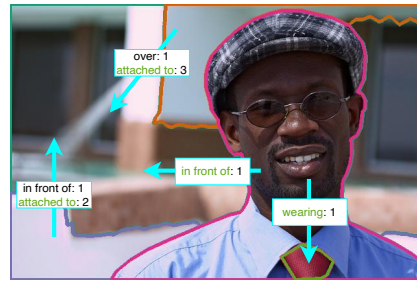
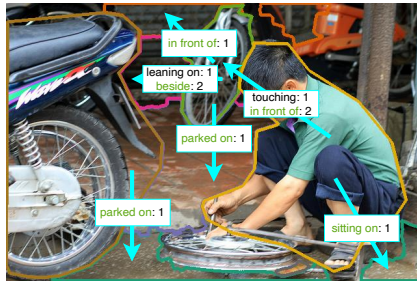
Table 5: Node loss weight. All experiments were run 3 times. The number behind the ± sign shows the standard deviation.

Node Loss Weight	mR@50	mNgR@50
0.00	35.96 ± 2.36	61.04 ± 4.62
0.20	40.06 ± 0.56	64.05 ± 1.91
0.50	39.90 ± 0.76	64.19 ± 1.48

E Example Images

The images below show example outputs from DSFormer. The blue arrows are the ground truth annotations. For every relation, DSFormer assigns a score to each predicate and the predicates can be ranked within a relation. In the shown example images, the ranks are shown as numbers behind the predicate names. A low number means that DSFormer estimates the predicate to be more suitable than other predicates with a higher rank. The highest possible number is 56 (the total number of predicates in the PSG dataset). The text in green is the ground truth predicate label for the relation.





References

1. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)