Location-Free Scene Graph Generation

Supplementary Material

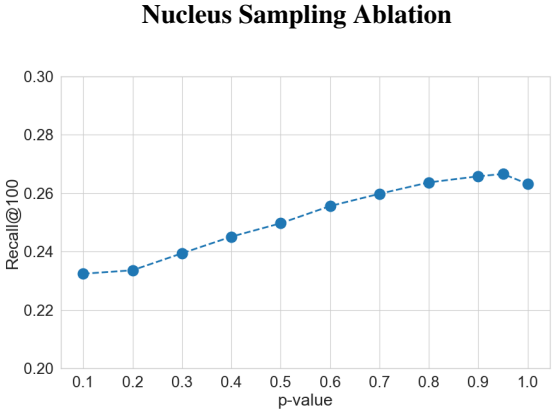

Figure 8. Effect of the p-value for nucleus sampling on R@100 on Visual Genome in the validation set.

Table 7. Inference speed (FPS) on Visual Genome. For Pix2SG, we show FPS and R@20 for predicting 300, 100 and 20 relations.

Method	Location-free	FPS	R@20
IMP [40]		3.00	21.66
MOTIFS [46]		2.55	**29.02**
Transformer [34]		3.10	28.79
VCTree [33]		1.50	27.06
SS-R-CNN [35]		**5.59**	22.09
RelTR [6]		4.90	25.86
SGTR [19]		4.98	23.62
Pix2SG 300	✓	0.41	**21.51**
Pix2SG 100	✓	3.90	21.31
Pix2SG 20	✓	**17.7**	21.11

6. Nucleus sampling

We investigate the effect of different p-values for nucleus sampling [13] and show the effect on the recall in Fig. 8. The p-value for nucleus sampling determines the classes that are sampled for the next token in the prediction sequence. The classes with the highest probabilities are sampled up to the cumulative probability of p. Higher p-values increase the likelihood of sampling more tokens with lower output probabilities, allowing the model to make more diverse predictions. For lower p-values, only the most probable classes are considered. In our experiments on Visual Genome [16], we found that the highest recall is achieved for a p-value of 0.95. The improved performance with high p-values indicates that, forcing more diverse predictions is beneficial.

7. Inference Speed

In Tab. 7, we present the inference speed of Pix2SG and compare it to other methods on Visual Genome. One interesting aspect of our autoregressive method is that it can be configured to predict any number of relations, offering a speed-accuracy trade off. When predicting only 100 or 20 relations, our model loses little performance, but gains a significant speed boost, which can be valuable in time critical domains and tasks.

8. Sequence Decoding

We provide further details and examples of our encoding and decoding of scene graphs into a sequence of tokens for our Pix2SG autoregressive architecture. During training, we randomly order the quintuples into a sequence, which is inspired by Pix2Seq [4] following a similar procedure for permutationally invariant bounding boxes. E.g. for a ground truth with two sets of quintuples [A,B,C,D,E], [V,W,X,Y,Z], both [A,B,C,D,E;V,W,X,Y,Z] and [V,W,X,Y,Z;A,B,C,D,E] would be valid sequences. This ambiguity is limited through teacher-forcing, where during training, the model is guided by ground truth labels. I.e., the prediction of "X" would be conditioned on the preceding token sequence e.g. [A,B,C,D,E,V,W]. This reduces the uncertainty during training, especially for the latter tokens in the sequence. Empirically, we find that this partially noisy training method works well. For evaluation, we decode the predicted sequence into a scene graph and compare it directly with the ground truth scene graph. Since our evaluation method is entirely based on graphs, it is unaffected by the ordering of the quintuples or the resulting ambiguity, which is specific to our Pix2SG model architecture. This also makes our evaluation algorithm agnostic to the SGG method used.

9. More Attention Visualizations

In Fig. 9, we provide more examples of attention maps to show our model's location awareness. Generally, the attention seems to be focused on the location of the corresponding entity. When predicting the subject index or object index, the model's attention seems to be very focused on the corresponding object category. For the prediction of the predicate, the model attentions seems focused on the contact points between the two entities that were predicted before. These examples highlight the benefits of our autore-

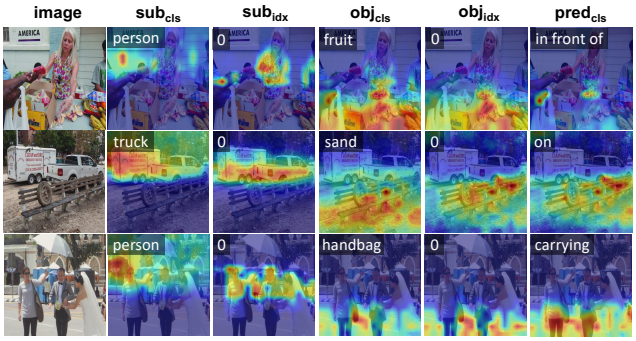

Figure 9. Visualization of the attention maps of Pix2SG on three images. While the subject attention seems to focus on a few entities, the object and predicate attentions tend to focus on the surroundings as well.

gressive approach, where the step-by-step prediction of the quintuples allows our model to focus its attention to a single relevant component at a time.

10. Visualization of the Tree-Based Graph Matching Algorithm

To provide further insight into our tree-based graph-matching algorithm, we show an exemplary illustration for a branching factor $K = 2$ in Fig. 10.

11. More Qualitative Results on Visual Genome

In Fig. 11, we provide more examples for the predictions of our model and the corresponding ground truth labels on Visual Genome, and highlight the sparsity and consistency of the annotations. Interestingly, we observe that most predicted entities and predicates seem plausible, even though they are not included in the Visual Genome annotations. In A, the model's predictions describe the visual scene in more detail than the sparse annotations. In B, we observe that the model repeats many predictions linked to the entities of *man* and *boy*. In the dataset, similar images can have the entity label *man* or *boy* without a clear distinctive signal. We suspect the model is optimizing for this ambiguity by predicting two separate scene graphs and sets of entities for each of the semantically similar classes. In C, our model predicts one instance of $< banana, on, bike >$, which can be seen in the scene even though it is not included in the annotations. Interestingly, there is no example of the triplet $< banana, on, bike >$ in Visual Genome, showing our model can generalize to previously unseen semantic connections.

12. More Qualitative Results on Panoptic Scene Graph Dataset

In Fig. 12, we provide more examples for the predictions of our model and the corresponding ground truth labels on the panoptic scene graph dataset. Compared to visual genome, the annotations are more comprehensive and consistent, yet there are still many predictions of our model that seem plausible but are not part of the ground truth annotations. We do not observe the same behaviour as in Fig. 11 B) where the model is unsure of the correct classification ("boy" vs. "man") and therefore predicts multiple scene graphs, as these classes are combined into the class "person" in PSG. The model can, however, still struggle to predict the correct number of entities, as seen in Fig. 12 A) ("bottle" and "cup").

13. Qualitative Image Generation Results

In Fig. 13, we include qualitative examples for a third downstream task, image generation. We generate an image from an location-free scene graph, by using GPT-4 Vision, a model not trained for this task. We prompt it with the following prompt "A realistic real-world photo that matches the following scene graph. The photo should not have any details not mentioned here: <SG>", where <SG> is a location-free scene graph represented as a list of quintuplets. These first results indicate that location-free scene graphs could also be a potent representation for image generation tasks.

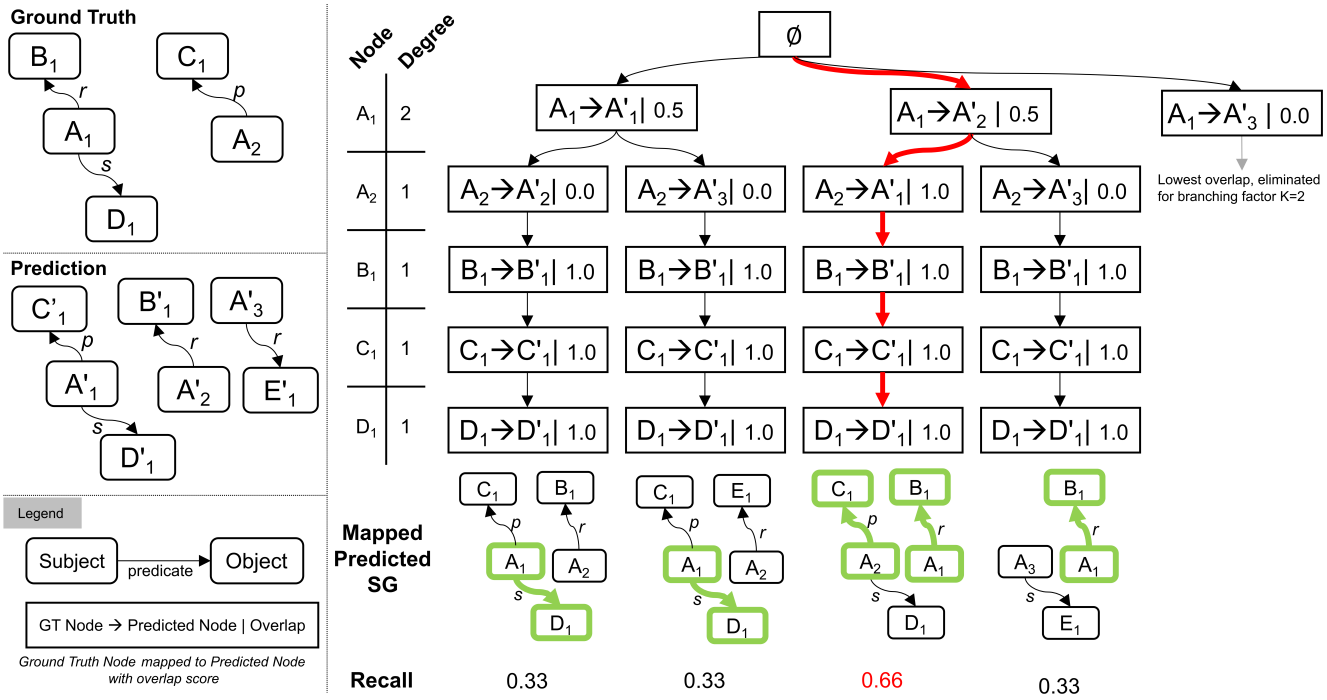

Figure 10. Illustration of our tree-based graph-matching algorithm for a branching factor $K = 2$. Given a ground truth scene graph and a predicted scene graph, our algorithm iterates over the ground truth nodes from the highest degree to the lowest in each step and explores K different branches. Afterward, the different pathways are used to calculate the instance match proposals, and the pathway with the highest recall is used to map the predicted scene graph to the ground truth scene graph. For nodes, capital letters denote their class and indices their instance ID, i.e. "A_1" and "A_2" are two instances of class "A".

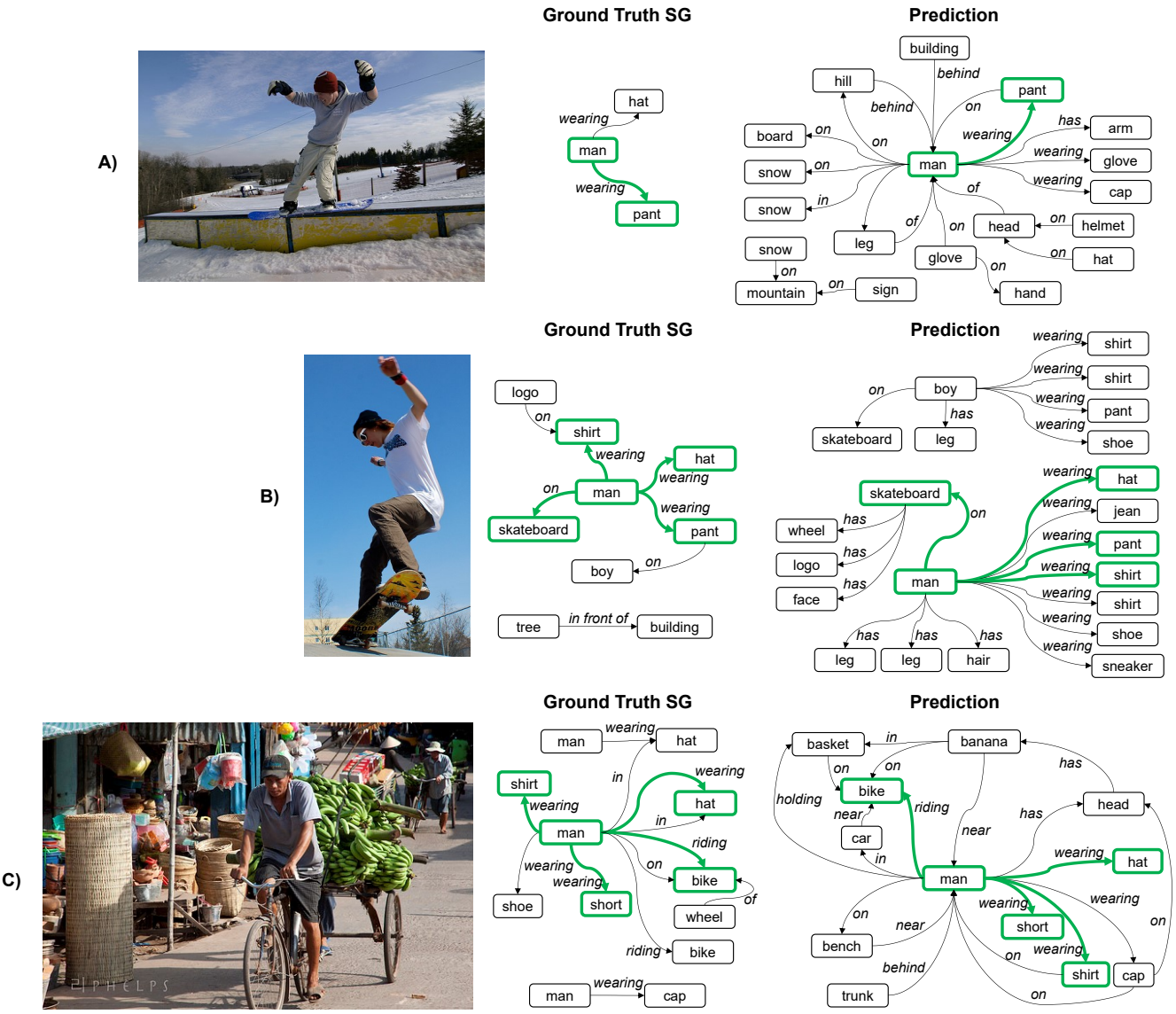

Figure 11. Extended Qualitative Analysis of Pix2SG on Visual Genome. Image, Ground Truth Scene Graph and Predicted Scene Graph are shown. The Predicted Scene Graph is constructed only from the 20 most probable predictions of the model (as in Recall@20). Matching triplets from Ground Truth and Prediction are highlighted in green.

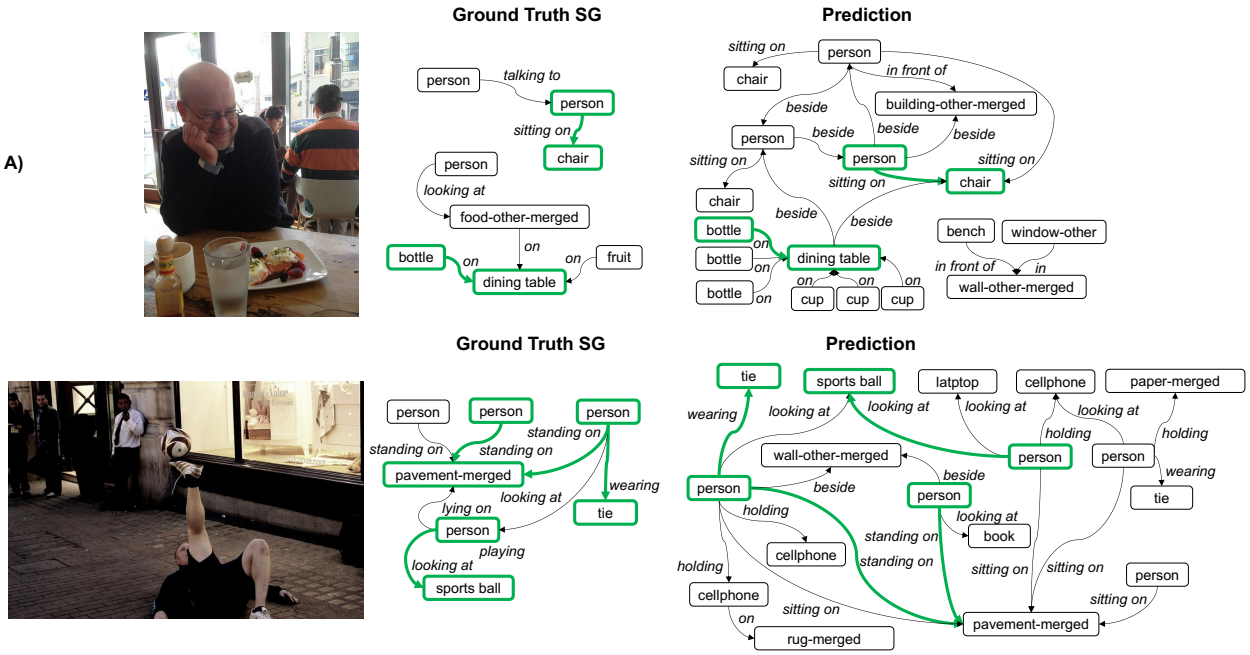

Figure 12. Extended Qualitative Analysis of Pix2SG on Panoptic Scene Graph Dataset. Image, Ground Truth Scene Graph and Predicted Scene Graph are shown. The Predicted Scene Graph is constructed only from the 20 most probable predictions of the model (as in Recall@20). Matching triplets from Ground Truth and Prediction are highlighted in green.

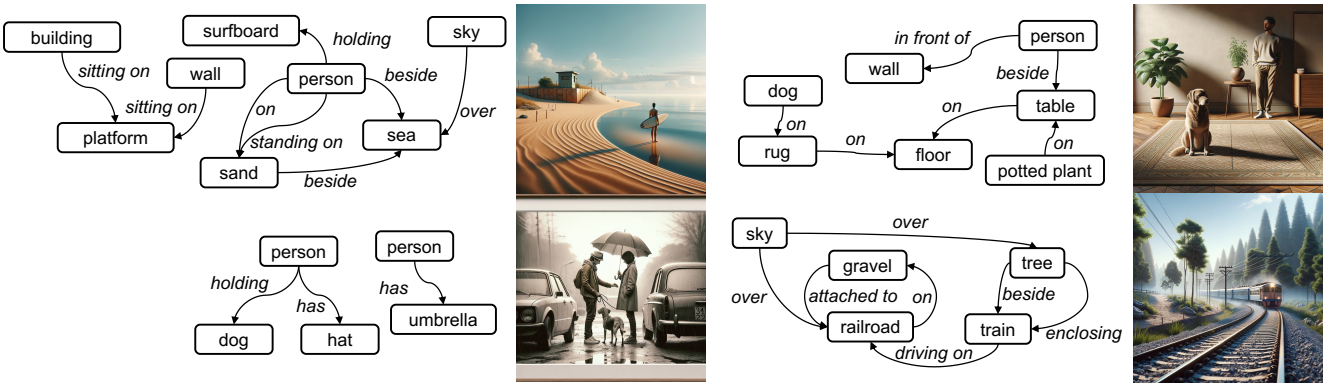

Figure 13. Qualitative examples of image generation from location-free scene graphs.