Probabilistic Modeling of Semantic Ambiguity for Scene Graph Generation

Gengcong Yang[1*†], Jingyi Zhang[2*], Yong Zhang[3‡], Baoyuan Wu[4,5‡], Yujiu Yang[1‡]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, [2]Ascend Lab, Huawei Technologies.
[3]Tencent AI Lab, [4]School of Data Science, The Chinese University of Hong Kong, Shenzhen,
[5]Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data
ygc19@mails.tsinghua.edu.cn, jgg_jingyizhang@foxmail.com, zhangyong201303@gmail.com
wubaoyuan@cuhk.edu.cn, yang.yujiu@sz.tsinghua.edu.cn

A. Discussion of the Complexity

In the inference stage, our PUM module only adds an extra $O(2d^2 + Kd)$ [1] computational complexity from Eq. 7, 8, and 9. In practice, the extra cost is trivial enough so that it hardly takes longer to train than a non-PUM model.

B. How PUM works

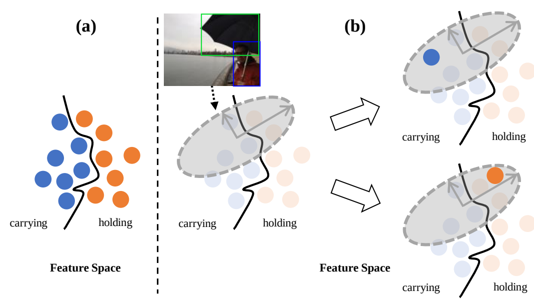

Figure 1. Illustration of semantic ambiguity in feature space. Blue and orange denote two different classes, curves denote the decision boundary, and ellipses indicate Gaussian embeddings.

We would like to clarify that, when we discuss about *solving* the problems of semantic ambiguity, we mean to *simulate* the behavior where different people may describe the same visual content in different ways. When inspected in feature space, the three types of semantic ambiguity are essentially the same. They are all caused by a situation where instances are classified into different classes even though they share similar visual features. As illustrated in Figure 1, we take `carrying` *vs.* `holding` as an example. Our method may map a union region (*e.g.* a man and an umbrella) into a Gaussian distribution rather than a deterministic point. The stochasticity enables the feature to pass across the decision boundary, leading to different plausible predictions, either `carrying` or `holding`, as shown in Figure 1 (b). By focusing on such stochastic feature representation, which is independent of the classifier, we implement diverse predictions and also simulate the semantic ambiguity.

C. Examples of Generated Scene Graph

We present some complete generated scene graphs in Figure 2.

D. More Ablation Studies

We present more ablation studies in Table 1. The results indicate that the training process would reach a local optimum without the deterministic loss in Eq. 11. We also encounter a performance drop when removing the regularization term of Eq. 12.

---

*Equal contribution.

†Work done in part during an internship at Tencent AI Lab.

‡Yong Zhang, Baoyuan Wu and Yujiu Yang are the corresponding authors. This research was partially supported by the Key Program of National Natural Science Foundation of China under Grant No. U1903213 and the Guangdong Basic and Applied Basic Research Foundation (No. 2019A1515011387). Baoyuan Wu is supported by the Natural Science Foundation of China under grant No. 62076213, the university development fund of the Chinese University of Hong Kong, Shenzhen under grant No. 01001810, and the special project fund of Shenzhen Research Institute of Big Data under grant No. T00120210003.

[1] d and K denote the dimension of features fed into the classifier and the number of sampling in Eq. 9, respectively.

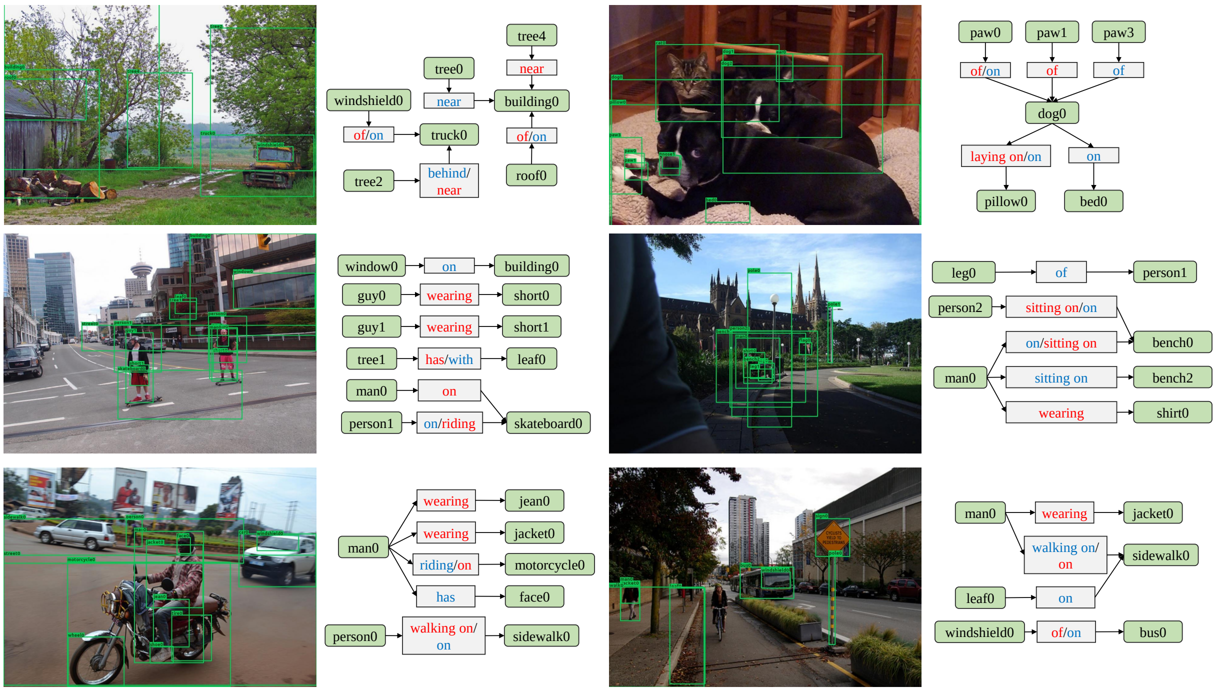

Figure 2. Examples of generated scene graph by ensembling two consecutive inferences in the PredCls setting. Blue indicates correctly classified predicates compared to the ground truth; red indicates the misclassified ones.

Table 1. Comparisons of the R@100 and mR@100 in % of our full model, our model without the conventional deterministic loss in Eq. 11 (w/o dl), and our model without the regularization term of Eq. 12 (w/o rt).

	SGDet		SGCls		PredCls		
Methods	R@100	mR@100	R@100	mR@100	R@100	mR@100	Mean
Ours w/o dl	31.0	8.0	38.5	11.4	68.0	19.9	29.5
Ours w/o rt	31.2	8.3	38.5	12.2	68.3	21.6	30.0
Ours	**31.3**	**8.9**	**39.0**	**12.8**	**68.3**	**22.0**	**30.4**