Probabilistic Modeling of Semantic Ambiguity for Scene Graph Generation

Gengcong Yang1*, Jingyi Zhang2*, Yong Zhang3†, Baoyuan Wu4,5‡, Yujiu Yang1‡
1Tsinghua Shenzhen International Graduate School, Tsinghua University, 2Ascend Lab, Huawei Technologies.
3Tencent AI Lab, 4School of Data Science, The Chinese University of Hong Kong, Shenzhen,
5Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data

ygcl9@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 \( O(2d^2 + Kd) \) 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

![Feature Space](https://example.com/feature_space.png)

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.
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).

<table>
<thead>
<tr>
<th>Methods</th>
<th>SGDet R@100</th>
<th>SGDet mR@100</th>
<th>SGCls R@100</th>
<th>SGCls mR@100</th>
<th>PredCls R@100</th>
<th>PredCls mR@100</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours w/o dl</td>
<td>31.0</td>
<td>8.0</td>
<td>38.5</td>
<td>11.4</td>
<td>68.0</td>
<td>19.9</td>
<td>29.5</td>
</tr>
<tr>
<td>Ours w/o rt</td>
<td>31.2</td>
<td>8.3</td>
<td>38.5</td>
<td>12.2</td>
<td>68.3</td>
<td>21.6</td>
<td>30.0</td>
</tr>
<tr>
<td>Ours</td>
<td>31.3</td>
<td>8.9</td>
<td>39.0</td>
<td>12.8</td>
<td>68.3</td>
<td>22.0</td>
<td>30.4</td>
</tr>
</tbody>
</table>