Probabilistic Modeling of Semantic Ambiguity for Scene Graph Generation

Gengcong Yang¹‡, Jingyi Zhang²⁺, Yong Zhang³‡, Baoyuan Wu⁴⁻, 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, jggjingyizhang@foxmail.com, zhangyong201303@gmail.com
wubaoyuan@cuhk.edu.cn, yang.yujiu@sz.tsinghua.edu.cn

Abstract

To generate “accurate” scene graphs, almost all existing methods predict pairwise relationships in a deterministic manner. However, we argue that visual relationships are often semantically ambiguous. Specifically, inspired by linguistic knowledge, we classify the ambiguity into three types: Synonymy Ambiguity, Hyponymy Ambiguity, and Multi-view Ambiguity. The ambiguity naturally leads to the issue of implicit multi-label, motivating the need for diverse predictions. In this work, we propose a novel plug-and-play Probabilistic Uncertainty Modeling (PUM) module. It models each union region as a Gaussian distribution, whose variance measures the uncertainty of the corresponding visual content. Compared to the conventional deterministic methods, such uncertainty modeling brings stochasticity of feature representation, which naturally enables diverse predictions. As a byproduct, PUM also manages to cover more fine-grained relationships and thus alleviates the issue of bias towards frequent relationships. Extensive experiments on the large-scale Visual Genome benchmark show that combining PUM with newly proposed ResCAGCN can achieve state-of-the-art performances, especially under the mean recall metric. Furthermore, we show the universal effectiveness of PUM by plugging it into some existing models and provide insightful analysis of its ability to generate diverse yet plausible visual relationships.

*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. Introduction

Scene graph generation (SGG) has been an important task in computer vision, serving as an intermediate task to bridge the gap between upstream object detection [21] and downstream high-level visual understanding tasks, such as image captioning [33, 42] and visual question answering [36]. Intuitively, the latter would get greater benefit from more human-like scene graphs.

Almost all existing works [30, 35, 4, 24] view SGG as an objective task and predict pairwise relationships in a de-
terministic manner. Namely, given a pair of objects, they always generate an identical predicate. However, compared with humans, such methods pursue “accurate” scene graphs but overlook the intrinsic semantic ambiguity of visual relationships. Specifically, the collaborative annotations from human annotators tend to be diverse, covering different descriptions of relationships for similar visual scenes.

We observe that there exist multiple types of semantic ambiguity in the large-scale Visual Genome dataset. Inspired by linguistic knowledge, we classify the ambiguity into three types. The first type is **Synonymy Ambiguity**, where multiple synonymous predicates that share overlapping definitions are suitable to describe similar visual scenes. For example, in Figure 1 (a), carrying and holding are interchangeable to describe the relationship between a man and an umbrella. If we visualize these two words in the semantic space, they should point to the same position where the visual relationship lies. The second one is **Hyponymy Ambiguity**, indicating that different humans tend to use predicates across adjacent abstract levels. One would simply use on to describe the visual scene where a cat is on top of a bench, while others may choose to use more fine-grained laying on, as shown in Figure 1 (b). In this case, laying on is a hyponym of on. Namely, the semantic range of the former is included by that of the latter. As for the third type, we notice that different human annotators often focus on different types of visual relationships, which originate from different points of view. Therefore, we refer to this phenomenon as **Multi-view Ambiguity**. An example is illustrated in Figure 1 (c), where both using (actional) and in front of (spatial) are plausible to describe the relationship between a working man and a laptop. If we consider the visual scene in three-dimension space, it can be a multicolor sphere that reflects different colors from different views. Although most predicates have single labels in the dataset, due to the ubiquitous semantic ambiguity mentioned above, we argue that many of them should have multiple labels, since similar visual scenes are annotated as different predicates. We refer to the issue as an implicit multi-label problem, which motivates the need to generate diverse predictions for visual relationships.

In this work, we focus on modeling the semantic ambiguity of visual relationships and propose a novel plug-and-play Probabilistic Uncertainty Modeling (PUM) module, which can be easily deployed in any existing SGG model. Specifically, we utilize a probability distribution to represent each union region, rather than a deterministic feature vector as in previous methods. From a geometric perspective, the probabilistic representation allows us to map each visual relationship to a soft region in space, instead of merely a single point [26]. For ease of modeling, we adopt Gaussian distributions to represent them. Namely, each union region is now parametrized by a mean and variance. The former acts like the normal feature vector as in the conventional model, whereas the latter measures the feature uncertainty. To some extent, in this way, the feature instance of each union region can be viewed as a random variable drawn from a Gaussian distribution. Thanks to this uncertainty modeling, ambiguous union regions will be assigned to Gaussian distributions with large variances, which generate diverse samples and result in diverse predictions. As a byproduct, we find that PUM also manages to cover more fine-grained relationships and thus well alleviates the infamous issue of bias towards frequent relationships [4, 24].

We firstly demonstrate the effectiveness of PUM on the Visual Genome benchmark. Combining with the newly proposed Residual Cross-attention Graph Convolutional Network (ResCAGCN) in concurrent work [39], we achieve state-of-the-art performances under the existing evaluation metrics, especially the mean recall. Note that our PUM can serve as a plug-and-play component. Therefore, we plug PUM into state-of-the-art models and observe obvious universal improvement over these baselines, which mainly lies in the mean recall again. We owe the performance gain in the mean recall to the ability to generate diverse relationships, which improves the chances to hit the ground-truth with infrequent predicate labels. We further propose oracle recall as an indirect evaluation metric to measure the diversity of multiple inferences, which takes results of multiple consecutive predictions as an ensemble and computes recall. The oracle recall of the proposed model increases with the number of predictions, indicating that the model generates plausible diverse relationships and thus gradually covers the ground-truth more and more.

Overall, our contributions can be summarized as follows:

- We identify the semantic ambiguity of visual relationships and propose a novel plug-and-play Probabilistic Uncertainty Modeling (PUM) module, which utilizes a probability distribution to represent each union region instead of a deterministic feature vector.
- Combining PUM with ResCAGCN, we achieve state-of-the-art performances on the large-scale Visual Genome benchmark under the existing evaluation metrics, especially the mean recall.
- Extensive evaluations demonstrate the superiority of PUM to alleviate the bias towards frequent categories when plugged into the existing SGG models, reflected in the improvement on the mean recall.
- To the best of our knowledge, we are the first to explore diverse predictions for SGG. We conduct experiments both qualitatively and quantitatively to demonstrate that the proposed PUM module can generate diverse yet plausible relationships.
2. Related Work

**Scene Graph Generation.** Visual relationships have raised wide concern in the computer vision community since Lu et al. [18] formalized Visual Relationship Detection as a visual task. In early works [18, 44, 43, 38, 37], objects and pairwise relationships were detected independently. Such models overlooked the rich visual context and led to suboptimal performance. To make full use of visual context, later methods consider the whole image and utilized various kinds of message passing mechanisms [30, 15, 16, 32, 35, 24, 8, 27, 17]. For example, Xu et al. [30] were the first to formally define the problem of SGG and addressed it with iterative refinement via message passing. Afterward, Zellers et al. [35] represented the global context via LSTM, a recurrent sequential architecture. More recently, Chen et al. [4] incorporated statistical correlations into the graph propagation network. Meanwhile, Tang et al. [24] composed dynamic tree structures to allow content-specific message passing. While all these methods overlook the semantic ambiguity of visual relationships and make inferences in a deterministic manner, we propose to address the ambiguity via probabilistic modeling.

**Uncertainty Modeling.** Conventionally, the high-level representation of an input instance, e.g., an image or a word, is modeled as a fixed-length feature vector, namely, a single point in $\mathbb{R}^D$. However, such a point estimate is not sufficient to express uncertainty. In recent years, Gaussian embedding has been getting more attention in deep learning since [26] utilized it to represent words instead of the conventional word2vec [19], where the covariance naturally measures the ambiguity of the words. In the computer vision community, there exist prior works on modeling images as Gaussian distributions [20, 34, 3]. However, all the existing SGG methods represent each union region as a deterministic vector. In this work, we are the first to focus on the intrinsic semantic ambiguity of visual relationships and model each union region as a Gaussian distribution.

**Diverse Predictions.** Generally, there are two types of approaches to generate multiple diverse predictions. One is to train multiple models and aggregate their predictions. To better obtain diversity in the union of predictions, Multiple Choice Learning (MCL) [9] and other variants [14, 13, 25] were proposed to establish cooperation among all the models and train each to specialize on one particular subset of the data distribution. Another is to infer diverse predictions from a single model. Before deep learning, this type of methods mainly focused on probabilistic graphical models [2]. Afterward, existing single-model methods can be roughly categorized into two types: 1) via random noise added to Generative Adversarial Networks (GANs), applied in image captioning [5], image annotation [28], text generation [31] and so on; 2) mapping an instance to a probability distribution in latent space and sampling from it [6, 41, 7]. Our model can be regarded as the second category. While the other methods in this category usually utilize Variational Autoencoders (VAEs) in a generative way, we simply make
use of Gaussian distributions to generate stochastic representations without a reconstruction loss and achieve diversity via this stochasticity in the inference stage.

3. Approach

Generally, a scene graph is a structured representation that describes the contents of a visual scene, which encodes object instances via nodes and relationships between objects via edges [10]. As defined by Xu et al. [30], the task of Scene Graph Generation is to generate an accurate visually-grounded scene graph associated with an image.

Mathematically, a scene graph can be defined as $G = \{B, O, R\}$, where $B$ is a set of bounding boxes, $O$ is object labels, and $R$ is relationship labels. Conventionally, given an image $I$, the probability distribution of a scene graph $P(G|I)$ is decomposed into three factors [35, 4]:

$$P(G|I) = P(B|I)P(O|B, I)P(R|O, B, I). \tag{1}$$

Firstly, the widely used Faster R-CNN [21] is utilized to model $P(B|I)$ and generates a set of object proposals. Next, conditioning on the candidate bounding boxes, the object model $P(O|B, I)$ predicts the class label regarding each box. Finally, based on the result of object detection, the relationship model $P(R|O, B, I)$ infers the relationship of each object pair, leading to the whole scene graph for the current image. Existing works treat $P(R|O, B, I)$ as a deterministic model, which always generates an identical label for the same object pair. This framework is illustrated in Figure 2 (a)(b)(c). However, such a method overlooks the intrinsic semantic ambiguity of visual relationships and is likely to get stuck in the issue of biased predictions [4, 24], with the tendency to generate frequent and “safe” labels.

In this work, we propose a plug-and-play module for the relationship model, named Probabilistic Uncertainty Modeling (PUM), which addresses the semantic ambiguity mentioned above in a probabilistic manner. We replace the conventional deterministic modeling with PUM, as illustrated in Figure 2 (d). To better demonstrate the effectiveness of PUM, we adopt the newly proposed ResCAGCN [39] as our object model, which is introduced in Section 3.1. However, note that any object model from the existing SGG methods will be compatible with PUM theoretically. Then, we describe our PUM module in detail in Section 3.2.

3.1. Object Model

In our approach, we take Residual Cross-attention Graph Convolutional Network (ResCAGCN) from [39] as our object model to fuse object features and predict object labels.

The core of ResCAGCN is the cross-attention module ($CA$), which is designed to capture the semantic relevance among the object features and the pairwise union region features. The module is formulated as:

$$CA(x_i, x_j) = \left(W_i x_i \odot \sigma(W'_j x_j) \oplus W_i x_i\right) \odot \left(W_j x_j \odot \sigma(W'_i x_i) \oplus W_j x_j\right), \tag{2}$$

where $\odot$ and $\oplus$ denote element-wise product and sum, respectively. $\sigma$ is the sigmoid function to normalize the attention scores. All $W_s$ denote linear transformations to embed features into the same dimension, both here and below.

Given two object features $x_i$ and $x_j$, and their union region feature $u_{ij}$, to model the contextual information, ResCAGCN utilizes the cross-attention module to compute the contextual coefficient $c_{ij}$, which is formulated as:

$$c_{ij} = \sigma(W^T \left(CA(CA(x_i, x_j), u_{ij})\right)). \tag{3}$$

Instead of directly using the aggregated features as the output features, ResCAGCN uses a residual connection to add them back to the original features:

$$\hat{x}_i = x_i + ReLU\left(W_1LN\left(W_2 \sum_{j \in N_i} c_{ij} \otimes W_3x_j\right)\right), \tag{4}$$

where $\otimes$ denotes Kronecker product, $N_i$ denotes the $i$-th node’s neighborhood, and $LN$ denotes layer normalization [1]. The refined object features $\hat{x}_j$ are then fed into a classifier to predict the object labels.

3.2. Probabilistic Uncertainty Modeling

Conventionally, the union of two proposals is represented as a single point in space, namely, point embedding [20]. As [26] observed, however, such point estimate does not naturally express the uncertainty induced by the input. In the case of visual relationships, this could be caused by ambiguous annotations, e.g., holding and looking at may be both plausible to describe a scene containing a man and a cell phone.

As shown in Figure 2 (d), in order to capture the intrinsic uncertainty of visual relationships, we propose to explicitly model the feature distribution of each union region as Gaussian. That is, we represent each union region as a stochastic embedding. From a stochastic perspective, the final representation of each union region is no longer a deterministic vector but randomly drawn from a Gaussian distribution. As a result, we can generate predicates diversely for the same object pair, leading to diversity in scene graph generation.

Stochastic Representation. For each object pair, following ResCAGCN, we first fuse their contextual object features $\hat{x}_i$ and $\hat{x}_j$, as described in Section 3.1, and their visual union region feature $u_{ij}$ to obtain the relationship feature:

$$e_{ij} = \hat{x}_i \odot \hat{x}_j \odot u_{ij}, \tag{5}$$
where $\phi$ denotes the fusion function defined in [40, 24], $x \circ y = \text{ReLU}(W_x x + W_y y) - (W_x x - W_y y)^2$. Based on each fused relationship feature, we define the associated representation $z_{ij}$ in latent space as a Gaussian distribution,

$$p(z_{ij}|e_{ij}) = \mathcal{N}(z_{ij}; \mu_{ij}, \sigma^2_{ij}),$$

where $\mu_{ij}$ and $\sigma^2_{ij}$ refer to mean vector and diagonal covariance matrix respectively. They are formulated as:

$$\mu_{ij} = W \mu e_{ij},$$

$$\sigma^2_{ij} = W \sigma e_{ij}.$$  

At test time, we sample multiple $z_{ij}$s, feed them into the relationship classifier $\phi_r$ respectively and compute the average posterior probability distribution:

$$p_{ij} = \frac{1}{K} \sum_{k=1}^K \phi_r(z^{(k)}_{ij}),$$

where $z^{(k)}_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2_{ij})$, and $K$ is the number of samples drawn from the Gaussian. Then we simply take the argmax of $p_{ij}$ as the predicted relationship label.

**Uncertainty-aware Loss.** $\mu_{ij}$ can be viewed as the original deterministic representation of the union box, while the random variable $z_{ij}$ serves as a stochastic representation sample. Here, we consider both representations and feed them into $\phi_r$ respectively. Then, we train the relationship model with cross-entropy loss,

$$L = (1-\lambda)\text{CE}(\phi_r(\mu), y) + \lambda E[E_{z \sim p(z|e)}\text{CE}(\phi_r(z), y)],$$

where $\lambda$ is the weight to trade off between deterministic prediction and stochastic predictions, and $\text{CE}$ means cross-entropy loss. Note that we omit the subscripts $ij$ for clarity.

In practice, we approximate the expectation term via Monte-Carlo sampling from $z^{(k)} \sim p(z|e)$:

$$L_{cc} \approx (1-\lambda)\text{CE}(\phi_r(\mu), y) + \lambda \frac{1}{N} \sum_{k=1}^N \text{CE}(\phi_r(z^{(k)}), y),$$

where $N$ is the number of samples drawn from the Gaussian. It is clear that conventional deterministic training can be seen as a special case of Eq. 11 where $\lambda$ is set to 0.

Inspired by [34], as training progresses, the variance $\sigma^2$ always decreases with $L_{cc}$ alone and reverts our stochastic representation back to deterministic model. This problem could be alleviated by the following regularization term:

$$L_{reg} = \max(0, \gamma - h(N(\mu, \sigma^2))),$$

where $\gamma$ is a margin to bound the uncertainty level, and $h(N(\mu, \sigma^2))$ is the differential entropy of a multivariate Gaussian distribution which is actually only related to $\sigma$:

$$h(N(\mu, \sigma^2)) = \frac{1}{2} \log(\det(2\pi e \sigma^2)).$$

It is obvious that $L_{reg}$ will maintain the uncertainty level of the learned stochastic representations.

In conclusion, our final uncertainty-aware loss for the relationship model is expressed as:

$$L_{rel} = L_{cc} + \alpha L_{reg},$$

where $\alpha$ is the weight of regularization term.

**Reparameterization Trick.** Sampling $z$ from $\mathcal{N}(\mu, \sigma^2)$ directly will prevent gradients from propagating back to the preceding layers. Thus, we use the reparameterization trick [11] to bypass the problem. Specifically, we first sample a random noise $\epsilon$ from the standard Gaussian and generate $z$ as the equivalent sampling representation,

$$z = \mu + \epsilon \sigma, \epsilon \sim \mathcal{N}(0, I).$$

4. Experiments

4.1. Experiment Setting

**Dataset.** We evaluate the proposed method on the popular large-scale Visual Genome (VG) benchmark [12], which originally contains 108,077 images with average annotations of 38 objects and 22 relationships per image. Since the majority of the annotations are noisy, following previous works [35, 4, 24], we adopt the most popular split from [30], which selects top-150 object categories and top-50 predicate categories by frequency.

**Evaluation.** We follow three conventional tasks to evaluate the proposed SGG model: (1) **Predicate Classification** (PredCls): given the bounding boxes and their object labels in an image, predict the predicates of all pairwise relationships. (2) **Scene Graph Classification** (SGCls): given the ground-truth bounding boxes in an image, predict the predicate as well as the object labels in every pairwise relationship. (3) **Scene Graph Detection** (SGDet): given merely an image, simultaneously detect a set of objects and predict the predicate between each pair of the detected objects.

Since the distribution of relationships in the VG dataset is highly imbalanced, we follow [4, 24] to utilize mean Recall@K (short as mR@K) to evaluate each relationship in a balanced way. For reference, all the methods are also evaluated by the conventional Recall@K (short as R@K) metric.

**Implementation Details.** Following the prior works [35, 4, 24], we adopt the same Faster-RCNN [21] to detect object bounding boxes and extract RoI features. For the hyperparameters in PUM, we set $K$ to 8, $N$ to 8, $\lambda$ to 0.1, and $\gamma$ to 200. We optimize the proposed model by the Adam optimizer with a batch size of 8, and momentums of 0.9 and 0.999. Our method is implemented by Pytorch and MindSpore. Intuitively, our uncertainty modeling would cause variance of performances. However, in practice, under the
Table 1. Comparisons among various methods on mR@K (%). † denotes the re-implemented version from [35]. † and ↓ indicate the performance change before and after plugging in PUM.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SGDet mR@50</th>
<th>SGDet mR@100</th>
<th>SGCls mR@50</th>
<th>SGCls mR@100</th>
<th>PredCls mR@50</th>
<th>PredCls mR@100</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMP† [30, 35]</td>
<td>3.8</td>
<td>4.8</td>
<td>5.8</td>
<td>6.0</td>
<td>9.8</td>
<td>10.5</td>
<td>6.8</td>
</tr>
<tr>
<td>FREQ [35]</td>
<td>4.3</td>
<td>5.6</td>
<td>6.8</td>
<td>7.8</td>
<td>13.3</td>
<td>15.8</td>
<td>8.9</td>
</tr>
<tr>
<td>SMN [35]</td>
<td>5.3</td>
<td>6.1</td>
<td>7.1</td>
<td>7.6</td>
<td>13.3</td>
<td>14.4</td>
<td>9.0</td>
</tr>
<tr>
<td>KERN [4]</td>
<td>6.4</td>
<td>7.3</td>
<td>9.4</td>
<td>10.0</td>
<td>17.7</td>
<td>19.2</td>
<td>11.7</td>
</tr>
<tr>
<td>VCTREE-SL [24]</td>
<td>6.7</td>
<td>7.7</td>
<td>9.8</td>
<td>10.5</td>
<td>17.0</td>
<td>18.5</td>
<td>11.7</td>
</tr>
<tr>
<td>VCTREE-HL [24]</td>
<td>6.9</td>
<td>8.0</td>
<td>10.1</td>
<td>10.8</td>
<td>17.9</td>
<td>19.4</td>
<td>12.2</td>
</tr>
<tr>
<td>IMP† + PUM</td>
<td>4.5↑0.7</td>
<td>5.5↑0.7</td>
<td>6.4↑0.6</td>
<td>6.8↑0.8</td>
<td>11.3↑1.5</td>
<td>12.3↑1.8</td>
<td>7.8↑1.0</td>
</tr>
<tr>
<td>SMN + PUM</td>
<td>7.5↑2.2</td>
<td>8.6↑2.5</td>
<td>9.4↑2.3</td>
<td>10.1↑2.5</td>
<td>16.4↑3.1</td>
<td>18.1↑3.7</td>
<td>11.7↑2.7</td>
</tr>
<tr>
<td>KERN + PUM</td>
<td>6.5↑0.1</td>
<td>7.4↑0.1</td>
<td>9.9↑0.5</td>
<td>10.6↑0.6</td>
<td>18.7↑1.0</td>
<td>20.4↑1.2</td>
<td>12.3↑0.6</td>
</tr>
<tr>
<td>VCTREE-SL + PUM</td>
<td>7.1↑0.4</td>
<td>8.2↑0.5</td>
<td>11.0↑1.2</td>
<td>11.9↑1.4</td>
<td>19.0↑2.0</td>
<td>20.9↑2.4</td>
<td>13.0↑1.3</td>
</tr>
<tr>
<td>ResCAGCN [39]</td>
<td>7.9</td>
<td>8.8</td>
<td>10.2</td>
<td>11.1</td>
<td>18.3</td>
<td>19.9</td>
<td>12.7</td>
</tr>
<tr>
<td>ResCAGCN + PUM</td>
<td>7.7↓0.2</td>
<td>8.9↑0.1</td>
<td>11.9↑1.7</td>
<td>12.8↑1.7</td>
<td>20.2↑1.9</td>
<td>22.0↑2.1</td>
<td>13.9↑1.2</td>
</tr>
</tbody>
</table>

Table 2. Comparisons among various methods on R@100 (%).

<table>
<thead>
<tr>
<th>Methods</th>
<th>SGDet R@100</th>
<th>SGCls R@100</th>
<th>PredCls R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMP† [30, 35]</td>
<td>24.5</td>
<td>35.4</td>
<td>61.3</td>
</tr>
<tr>
<td>FREQ [35]</td>
<td>30.1</td>
<td>32.9</td>
<td>62.2</td>
</tr>
<tr>
<td>SMN [35]</td>
<td>30.3</td>
<td>36.5</td>
<td>67.1</td>
</tr>
<tr>
<td>KERN [4]</td>
<td>29.8</td>
<td>37.4</td>
<td>67.6</td>
</tr>
<tr>
<td>VCTREE-SL [24]</td>
<td>31.1</td>
<td>38.6</td>
<td>67.9</td>
</tr>
<tr>
<td>VCTREE-HL [24]</td>
<td>31.3</td>
<td>38.8</td>
<td>68.1</td>
</tr>
<tr>
<td>IMP† + PUM</td>
<td>25.0↑0.5</td>
<td>35.7↑0.3</td>
<td>61.8↑0.5</td>
</tr>
<tr>
<td>SMN + PUM</td>
<td>30.6↑0.3</td>
<td>37.4↑0.9</td>
<td>67.5↑0.4</td>
</tr>
<tr>
<td>KERN + PUM</td>
<td>29.8</td>
<td>37.1↑0.3</td>
<td>67.5↑0.1</td>
</tr>
<tr>
<td>VCTREE-SL + PUM</td>
<td>30.9↓0.2</td>
<td>38.1↑0.5</td>
<td>67.6↑0.3</td>
</tr>
<tr>
<td>ResCAGCN [39]</td>
<td>30.9</td>
<td>38.7</td>
<td>67.9</td>
</tr>
<tr>
<td>ResCAGCN + PUM</td>
<td>31.3↑0.4</td>
<td>39.0↑0.3</td>
<td>68.3↑0.4</td>
</tr>
</tbody>
</table>

hyperparameter setting mentioned above, we observe that the variance is always negligible enough\(^1\) to be ignored.

4.2. Comparisons with State-of-the-Art Methods

Comparing Methods. In this part, we compare our model with existing state-of-the-art methods: (1) designed to improve the recall, including Iterative Message Passing (IMP) [30], frequency baseline without using visual contexts (FREQ) [35] and Stacked Motif Network (SMN) [35]; (2) intended for more balanced prediction on relationships, including Knowledge-Embedded Routing Network (KERN) [4] and Visual Context Tree model (VCTREE-SL, trained by supervised learning, and VCTREE-HL, trained by hybrid learning) [24]. Although [17] also reported new state-of-the-art performances recently, we argue that their results are not directly comparable to ours. Please refer to the supplementary material for details.

Quantitative Results. From Table 1, compared with the previous state-of-the-art methods, the proposed model (ResCAGCN + PUM) shows the best performances on the mR@K metric, with a relative improvement of 13.9% compared with VCTREE-HL according to the mean. This indicates that the proposed model achieves notable improvement on infrequent categories. Meanwhile, it does not sacrifice frequent categories a lot, since its performances on R@100 also reach state-of-the-art, as shown in Table 2.

To gain a more comprehensive understanding of this

\(^1\)0.03% at most, with respect to mR/R@K.
phenomenon, as depicted in Figure 3, we further present the R@100 improvement of each predicate category over VCTREE-HL in the PredCls setting. Note that the x-axis labels are in descending order according to their number of samples in the VG dataset and categories with the same performances are filtered out. It is obvious that the proposed model achieves significant improvement in most categories. Importantly, the improvement is much larger for those infrequent categories in the long tail. We mainly owe this phenomenon to the byproduct of PUM, which endows the model with more chances to cover infrequent categories and thus alleviates the issue of biased predictions.

4.3. Ablation Study

To better prove the effectiveness of PUM, in this part, we explore the gain of PUM as a plug-and-play module. Firstly, as illustrated in the third part of Table 1, PUM brings about a significant increase over the vanilla ResCAGCN [39] by a margin of 1.2% on mR@K, according to the mean. Meanwhile, from Table 2, PUM also improves the model moderately on R@K. The results suggest that PUM indeed plays a critical role in the proposed model, which especially lies in more balanced predictions.

Then, we apply PUM to existing state-of-the-art methods, which conventionally utilize deterministic representations for visual relationships. Specifically, by regarding the original relationship features in each model as the $e_{ij}$ in Eq. 5, we adopt the subsequent operations in Section 3.2 to model the uncertainty of relationships in a probabilistic manner. We present comparisons on mR@K between the existing state-of-the-art methods (IMP [30], SMN [35], KERN [5] and VCTREE-SL,2 [24]) and the counterparts plugged in PUM in the second part of Table 1. We find that PUM improves the performances of all models by a significant amount, with a relative improvement of 14.7%, 30.0%, 5.1% and 11.1% compared with the baselines respectively. The results show the universal superiority of PUM to the deterministic modeling, which mainly lies in the effectiveness to alleviate the issue of biased predictions towards frequent relationships. We also present comparisons on R@K in Table 2. Note that PUM does not necessarily improve all baselines under this metric. However, according to the discussions by Tang et al. [23], R@K is not a “qualified” metric for SGG, since simply catering to frequent categories while neglecting the infrequent ones would unfairly obtain a good performance. Meanwhile, similar to [23], we also observe that the performance drops caused by PUM mainly originate from classifying trivial “head” predicates into more fine-grained “tail” classes, e.g. from on to parked on.

4.4. Understand Uncertainty Modeling

We observe that the proposed model can generate diverse relationships, which helps to address the implicit multi-label issue caused by the semantic ambiguity. In this part, we qualitatively and quantitatively analyze this characteristic to gain more insights about our uncertainty modeling. Qualitative Results. From Eq. 9, the relationship features fed into the classifier are randomly drawn from Gaussian distributions, resulting in varied predicted confidences, even for the same union region. Therefore, given a pair of objects, the proposed model can produce different plausible predicate at each inference. In other words, it is able to describe the same visual scene in different ways, leading to more human-like diverse predictions. This diversity well matches the three types of ambiguity illustrated in Figure 1. We show qualitative examples from two consecutive predictions of the proposed model in the PredCls setting in Figure 4. From the first row, the proposed model generates semantically-similar predicates consecutively, i.e. at vs. near and holding vs. carrying. Although the ground-truth only considers a single label, we argue that there exists such Synonymy Ambiguity, where multiple synonyms are plausible at the same time. Hyponymy Ambiguity is also a common phenomenon, where predicates across adjacent abstract levels are interchangeable. In the second row, the ground truth can be fine-grained (walking on) or coarse-grained (on). Thanks to our uncertainty modeling, the proposed model covers both levels of granularity and thus increases the chance of hitting the ground-truth. In Figure 1 (c), different human annotators tend to describe similar visual scenes from different points of view, resulting in Multi-view Ambiguity. We observe that the proposed model also simulates this phenomenon well. From Figure 4 (c), for the relationship between person and sheep, the proposed model focuses on either the spatial position (behind) or the person’s action (looking at). On the other hand, for the scene on the right, the predictions could be a possessive verb (has) or an actional verb (holding). In a word, these qualitative examples suggest that semantic ambiguity is quite common when describing visual relationships, while the proposed model manages to generate diverse yet plausible predictions.

Oracle Evaluation. Inspired by the oracle error rate in Multiple Choice Learning [9, 14, 13, 25], we propose to use oracle Recall (short as oR) to measure the diversity of predictions indirectly, which counts a hit if one of the multiple consecutive predictions matches the ground-truth:

$$oR = \frac{\sum_{i=1}^{R} \left(\sum_{m=1}^{M} 1(y_m = y_i > 0)\right)}{R},$$

(16)

$$1(x) = \begin{cases} 1 & x = \text{True}, \\ 0 & x = \text{False}. \end{cases}$$

(17)
In this work, we considered the semantic ambiguity of visual relationships, which could be classified into Synonymy Ambiguity, Hyponymy Ambiguity and Multi-view Ambiguity. To address the implicit multi-label issue caused by the ambiguity, we proposed a novel plug-and-play module dubbed PUM. Although we aimed for diverse predictions, thanks to the byproduct of PUM, we achieved state-of-the-art performances under the existing evaluation metrics when combining it with ResCAGCN. Furthermore, we showed the universal effectiveness of PUM and explored its ability to generate diverse yet plausible relationships both qualitatively and quantitatively. A possible future direction would be to apply this kind of uncertainty modeling in down-stream tasks that also emphasize diversity, such as diverse visual captioning [5, 22, 29].
References


