Supplementary Material for "Prototype-based Embedding Network for Scene Graph Generation"

A. Calculation of IV and IIVR

In this paper, we introduce the Intra-class Variance (**IV**) to measure the intra-class compactness of the entity's and predicate's representations:

$$\sigma^2_{within} = \frac{1}{Mn}\sum_{i=0}^{M}\sum_{j=1}^{n}\|\boldsymbol{\phi_{i,j}} - \boldsymbol{\mu_i}\|_2^2, \tag{1}$$

where $\boldsymbol{\phi_{i,j}}$ is a feature vector in class i, $\boldsymbol{\mu_i}$ is the mean of feature vectors in class i, n is the number of data points per class, and M is the number of classes. Furthermore, we introduce the Intra-class to Inter-class Variance Ratio (**IIVR**) to measure the inter-class distinctiveness of the representations:

$$\frac{\sigma^2_{within}}{\sigma^2_{between}} = \frac{1}{n}\frac{\sum_{i=0}^{M}\sum_{j=1}^{n}\|\boldsymbol{\phi_{i,j}} - \boldsymbol{\mu_i}\|_2^2}{\sum_{i=0}^{M}\|\boldsymbol{\mu_i} - \boldsymbol{\mu}\|_2^2}, \tag{2}$$

where $\boldsymbol{\mu}$ is the mean across all feature vectors.

B. Additional Ablation Studies

In this part, we construct additional ablation studies from another perspective to explore the effect of cosine similarity (CS) and Euclidean distance (ED) metrology in Prototype-guided Learning (PL) and Prototype Regularization (PR), respectively. The results are summarized in Tab. 2. Exp 1, PE-Net is the baseline model, which is trained without PL and PR, and only uses a linear classifier to classify the relation representation. Exp 2, PE-Net is trained with only cosine similarity metrology in PL and PR, *i.e.*, training with $\mathcal{L}_{e\_sim}$ and $\mathcal{L}_{r\_sim}$. Exp 3, PE-Net is trained with both cosine similarity and Euclidean distance metrology in PL and PR. When trained with only cosine similarity metrology in Exp 2, the model significantly outperforms the baseline (*i.e.*, Exp 1), *e.g.*, 70.2% *vs*. 68.2% at R@100, and 26.2% *vs*. 20.0% on PredCls tasks. This verifies that explicitly establishing entity-predicate matching empowers the SGG model with more powerful relation recognition ability than learning decision boundaries with trainable classifiers. In Exp 3, we further integrate the Euclidean distance metrology into PL and PR, and our model obtains significant gains on mR@K, *e.g.*, 33.8% *vs*. 26.2% at mR@100 on PredCls. It illustrates that using only angle-based cosine similarity metrology in PL and PR is not sufficient to learn accurate entity-predicate matching, mainly because some predicates are not distinctive enough against others. Therefore, it is necessary to use the Euclidean distance metrology as a supplement for further distinction. Also, we try to use only Euclidean metrology in PL and PR, but find that it does not work at all.

For an intuitive illustration of the performance improvement brought by the Euclidean distance metrology, we provide Recall@100 on each predicate of the models trained in Exp 2 and Exp 3, as shown in Fig. 1. Obviously, with the integration of Euclidean distance metrology, the performance of fine-grained predicates has been significantly improved, *e.g.*, "parked on", "walking on", "looking at", and "standing on". This shows that Euclidean distance metrology effectively help the model distinguish fine-grained predicates from coarse-grained ones.

γ_1	γ_2	PredCls					
		R@50	R@100	mR@50	mR@100	M@50	M@100
1.0	1.0	**68.1**	**70.1**	24.7	26.9	46.4	48.5
1.0	3.0	67.6	69.6	27.1	29.3	47.4	49.5
1.0	5.0	66.6	68.6	29.3	31.6	48.0	50.1
1.0	7.0	64.9	67.2	**31.5**	**33.8**	**48.2**	**50.5**
1.0	9.0	63.3	65.6	32.1	34.3	47.7	50.0
1.0	7.0	64.9	67.2	**31.5**	**33.8**	**48.2**	**50.5**
3.0	7.0	67.3	69.3	27.5	29.7	47.4	49.5
5.0	7.0	67.6	69.5	26.0	28.2	46.8	48.9
7.0	7.0	67.7	69.6	24.7	27.0	46.2	48.3
9.0	7.0	**67.8**	**69.8**	24.2	26.5	46.0	48.2

Table 1. Hyper-parameters analysis of the distance margin γ_1 and γ_2.

C. Hyper-parameters analysis of γ_1 and γ_2

To investigate the impact of distance margin hyper-parameters γ_1 and γ_2 in PL and PR, we construct experiments for them, and the results are summarized in Tab. 1. In Tab. 1, when γ_1 is fixed, we observe that with the increase of γ_2, the performance of mR@K gradually improves. It indicates that enlarging the distinctions between predicate prototypes makes fine-grained predicates distinctive from

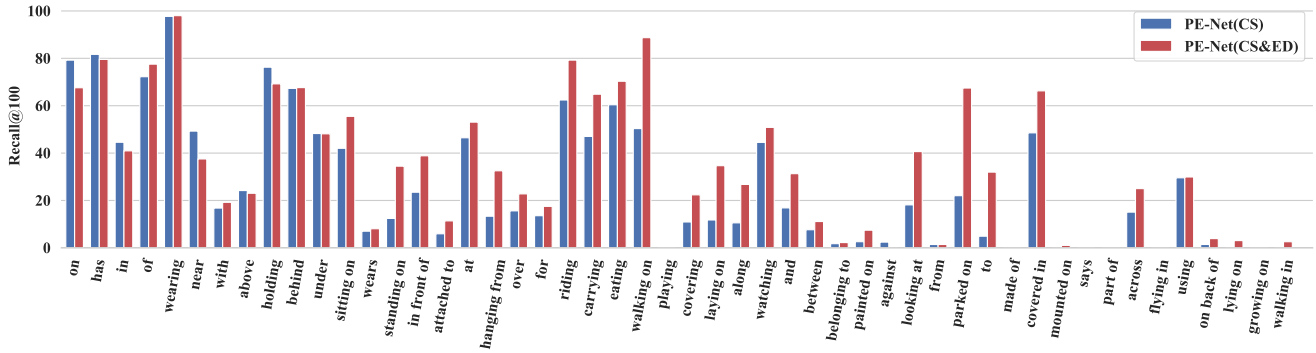

Figure 1. Recall@100 of all predicate classes of PE-Net(CS) and PE-Net(CS&ED) on the PredCls task. Predicates are sorted in decreasing order of sample frequency.

Exp	Metrology		PredCls			SGCls			SGDet		
	CS	ED	R@50/100	mR@50/100	M@50/100	R@50/100	mR@50/100	M@50/100	R@50/100	mR@50/100	M@50/100
1	✗	✗	66.5 / 68.2	18.5 / 20.0	42.5 / 44.1	39.5 / 40.4	9.9 / 10.5	24.7 / 25.5	32.3 / 36.8	7.8 / 9.3	20.1 / 23.1
2	✓	✗	**68.3 / 70.2**	24.0 / 26.2	46.2 / 48.2	**41.5 / 42.4**	13.6 / 14.7	27.6 / 28.6	**32.5 / 37.1**	9.3 / 11.1	20.9 / 24.1
3	✓	✓	64.9 / 67.2	**31.5 / 33.8**	**48.2 / 50.5**	39.4 / 40.7	**17.8 / 18.9**	**28.6 / 29.8**	30.7 / 35.2	**12.4 / 14.5**	**21.6 / 24.9**

Table 2. Additional ablation study of the effectiveness of each metrology in PE-Net. CS and ED denote the cosine similarity and Euclidean distance metrology.

k_1	k_2	PredCls					
		R@50	R@100	mR@50	mR@100	M@50	M@100
1	1	**67.6**	**69.6**	25.5	27.6	46.6	48.6
5	1	66.6	68.6	29.1	31.4	47.9	50.0
10	1	64.9	67.2	**31.5**	**33.8**	**48.2**	**50.5**
10	1	64.9	67.2	**31.5**	**33.8**	**48.2**	**50.5**
10	5	66.5	68.6	29.0	31.1	47.8	49.9
10	10	**67.2**	**69.2**	27.8	30.0	47.5	49.6

Table 3. Hyper-parameters analysis of k_1 and k_2.

Method	Predcls					
	R@50	R@100	mR@50	mR@100	M@50	M@100
Random	**67.8**	**70.0**	23.9	26.1	45.9	48.1
CLIP [3]	57.7	60.0	**32.5**	**34.4**	45.1	47.2
BERT [1]	47.7	50.1	28.0	30.1	37.9	40.1
GloVe [2]	64.9	67.2	31.5	33.8	**48.2**	**50.5**

Table 4. The ablation of the different initialization methods for prototypes.

the coarse-grained predicates, alleviating the semantic overlap and improving the accuracy of entity-predicate matching. The model achieves the best overall performance when $\gamma_2 = 7.0$. Furthermore, when γ_2 is fixed, with the increase of γ_1, we observe that the mR@K of PE-Net is decreasing while the R@K is increasing. Finally, when $\gamma_1 = 1.0$ and $\gamma_2 = 7.0$, the model achieves the best comprehensive performance on M@K.

D. Hyper-parameters analysis of k_1 and k_2

In this section, we investigate the impact of hyper-parameters k_1 and k_2 in PL and PR, and the results are shown in Tab. 3. In Tab. 3, when $k_2 = 1$ is fixed, we observe that with the increase of k_1, the performance of mR@K gradually improves, and the performance of R@K decreases. Furthermore, when $k_1 = 10$ is fixed, with the increase of k_2, we observe that the mR@K is decreasing while the R@K is increasing. With $k_1 = 10, k_2 = 1$ the model achieves the best performance on mR@100 and M@100.

E. Analysis of Prototype Initialization Methods

The semantic prototype is the core of our method. In this section, we construct an ablation study to explore the impact of different prototype initialization methods on the model, including Random weights, CLIP [3], BERT [1], and GloVe [2], and the results are summarized in Tab. 4. From Tab. 4, we have the following observations: (1) Using random weights initialization, the model achieves the best R@K performance, but its mR@K result is low. (2) Using the initialization of the word embedding obtained from Clip's text encoder, the model achieves the best mR@K performance. (3) Using the GloVe, the model achieves the best comprehensive performance M@K and relatively good R@K and mR@K performance.

F. Feature Fusion Function Selection

In the Sec. 3.1 of our manuscript, we use the $\mathcal{F}(\boldsymbol{s}, \boldsymbol{o})$ to fuse the features of subject and object. In order to explore the impact of different fusion functions, we test the following functions: 1) Add function, *i.e.*, $\mathcal{F}(\boldsymbol{s}, \boldsymbol{o}) = \boldsymbol{s} + \boldsymbol{o}$.

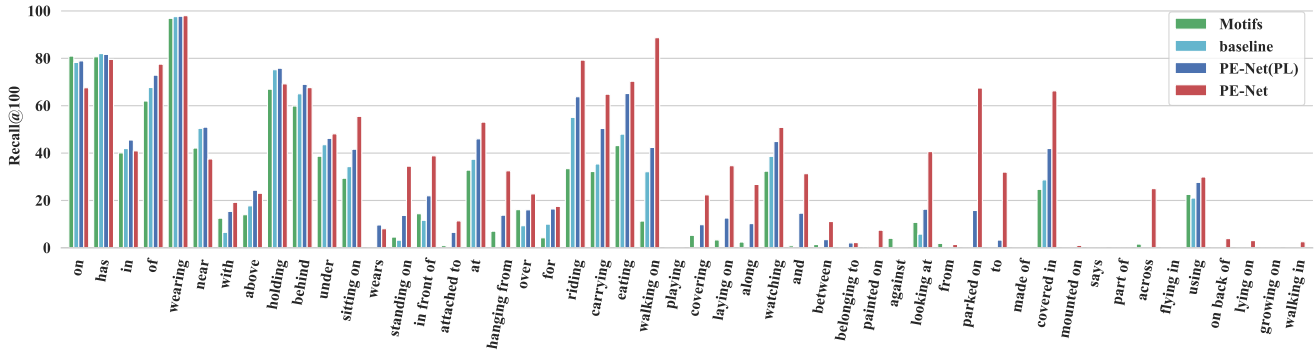

Figure 2. Recall@100 of all predicate classes of Motifs, baseline, PE-Net(PL), and PE-Net on the PredCls task. Predicates are sorted in decreasing order of sample frequency.

2) Sub function, *i.e.*, $\mathcal{F}(\boldsymbol{s}, \boldsymbol{o}) = \boldsymbol{s} - \boldsymbol{o}$. 3) Multi function, *i.e.*, $\mathcal{F}(\boldsymbol{s}, \boldsymbol{o}) = \boldsymbol{s} * \boldsymbol{o}$. 4) Learnable linear layer, *i.e.*, $\mathcal{F}(\boldsymbol{s}, \boldsymbol{o}) = f(\boldsymbol{s} \oplus \boldsymbol{o})$, where $f(\cdot)$ is a fully connected linear layer, $\oplus$ is the concatenation operation. 5) Hybrid function, *i.e.*, $\mathcal{F}(\boldsymbol{s}, \boldsymbol{o}) = \text{ReLU}(\boldsymbol{s} + \boldsymbol{o}) - (\boldsymbol{s} - \boldsymbol{o})^2$. The experimental results are shown in Tab. 5. In Tab. 5, we observe that similar results are obtained for the different functions, and Hybrid function achieves the best mR@K and M@K performance.

Function	Predcls		
	R@50/100	mR@50/100	M@50/100
Add	65.3 / 67.6	30.6 / 32.9	48.0 / 50.3
Sub	**65.6 / 67.7**	30.2 / 32.5	47.9 / 50.1
Multi	65.4 / 67.4	30.8 / 33.2	48.1 / 50.3
Learnable	65.4 / 67.6	30.8 / 32.9	48.1 / 50.3
Hybrid	64.9 / 67.2	**31.5 / 33.8**	**48.2 / 50.5**

Table 5. The ablation for different feature fusion functions.

G. Necessity of Instance-varied Semantic Contents

For Prototype-based Modeling in PE-Net, we use the instance-varied semantic content to model the diversity of each instance. To verify its necessity, we remove the semantic contents $\boldsymbol{v_s}$, $\boldsymbol{v_o}$, and $\boldsymbol{u_p}$ from the representations of subject ($\boldsymbol{s}$), object ($\boldsymbol{o}$), and predicate ($\boldsymbol{o}$), and model them only with class-specific semantic prototypes:

$$\begin{aligned} \boldsymbol{s} &= \mathbf{W}_s \boldsymbol{t_s}, \\ \boldsymbol{o} &= \mathbf{W}_o \boldsymbol{t_o}, \\ \boldsymbol{p} &= \mathbf{W}_p \boldsymbol{t_p}. \end{aligned} \tag{3}$$

The experimental results are shown in Tab. 6. In Tab. 6, we observe that the model's performance decreases substantially when the semantic contents are removed, *e.g.*, 61.6% *vs*. 67.2% at R@100 and 26.1% *vs*. 33.8% at mR@100 on PredCls task. It demonstrates that semantic contents are crucial for entity-predicate matching, otherwise the model only learns fixed matching between entity pairs and predicates, lacking diversity.

Model	Predcls		
	R@50/100	mR@50/100	M@50/100
w/o sc	57.3 / 61.6	22.9 / 26.1	40.1 / 43.9
PE-Net	**64.9 / 67.2**	**31.5 / 33.8**	**48.2 / 50.5**

Table 6. The ablation of the necessity of instance-varied semantic content in Prototype-based Modeling. w/o sc denotes remove semantic contents from Prototype-based Modeling.

H. Qualitative Results of Predicate Recall

For a more intuitive illustration of PE-Net's relation recognition ability, we provide Recall@100 on each predicate among Motifs [4], baseline, PE-Net(PL), and PE-Net, as shown in Fig. 2. From the results, we have several observations as follows: 1) Our baseline model outperforms Motifs on most predicates, illustrating the effectiveness of our Prototype-based Modeling in PE-Net. Intuitively, Prototype-based Modeling provides compact and distinctive entity representations, which greatly contributes to relation recognition in SGG. 2) When constrained by PL, PE-Net(PL) significantly outperforms the baseline on almost all predicates. It suggests that PL effectively establishes the matching between entity pairs and predicates, which is superior to learning decision boundaries with trainable classifiers by itself. 3) After being integrated with PR, PE-Net's performance on fine-grained predicates is significantly improved. It powerfully demonstrates that PR significantly enhances the discrimination between predicate prototypes, enabling the model to distinguish fine-grained predicates from coarse-grained ones, and thus achieves accurate entity-predicate matching.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018. 2

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *ACL*, 2014. 2

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[4] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 3