Appendix

This supplementary document is organized as follows:

- Section A shows more details about the smoothed ℓ_p-norm distance metric.
- Section B provides the proof that the ℓ_p-based graph regularization is upper bounded by Eq. (12) in the main paper.
- Section C describes the extension and implementation details of U-MP.
- Section D first visualizes the attention maps of different GMP modules in Figure 1. Then, the qualitative comparisons between RU-Net and MOTIFS [4] are illustrated in Figure 2.

A. Smoothed ℓ_p-norm Distance Metric

To improve the robustness against spurious correlations between nodes, we utilize a smoothed ℓ_p-norm distance metric [2] as follows:

$$\kappa_p^\epsilon(x) \triangleq \begin{cases} \epsilon^{p-2}|x|^2, & |x| \le \epsilon \\ \frac{2}{p}|x|^p - \frac{2-p}{p}\epsilon^p, & |x| > \epsilon \end{cases}, \tag{1}$$

where $\epsilon > 0$ and $0 < p \le 2$. Here, the generalized ℓ_p function $\kappa_p^\epsilon(x)$ is a smoothed approximation of $\kappa_p(x) \triangleq |x|^p$ with a regional quadratic envelope around its nondifferentiable point $x = 0$ for $0 < p < 2$. Such a smoothed extension can also be viewed as a generalization of the Huber penalty [1] which is a special case of $\kappa_p^\epsilon(x)$ with $p = 1$. Furthermore, the relationship between $\kappa_p^\epsilon(x)$ and $\kappa_p(x)$ is summarized as follows [1]:

$$0 \le \kappa_p(x) - \kappa_p^\epsilon(x) \le \frac{(2-p)\epsilon^p}{2}, x \in \mathbb{R}. \tag{2}$$

The above inequality Eq. (2) indicates that $\kappa_p^\epsilon(x)$ is always smaller than $\kappa_p(x)$, and the distance between them increases for large ϵ or small p and vanishes at $\epsilon = 0$ or $p = 2$. For this reason, $\kappa_p^\epsilon(x)$ places much less emphasis for large $|x|$ and is more robust against large outliers than $\kappa_p(x)$. Besides this merit, the cutoff parameter ϵ decides how much we retain the quadratic behaviour. Therefore, a lot of flexibility can be obtained by adjusting the parameters ϵ and p, leading to a better robustness for accommodating various kinds of outlier.

B. Proof of Eq. (12) in the Main Paper

Proposition 1 *Suppose that $\epsilon > 0$, $\tau > 0$ and $0 < p < 2$. Then, $\mathcal{G}_p(\boldsymbol{Y}, \boldsymbol{L})$ can be upper-bounded by*

$$\mathcal{G}_p(\boldsymbol{Y}, \boldsymbol{L}) \le \sum\nolimits_{(i,j)\in\mathcal{E}} [\boldsymbol{A}]_{ij}[\Omega]_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 + \text{const}. \tag{3}$$

where const. *is a constant term irrelevant to the optimization variable and*

$$[\Omega]_{ij} \triangleq \begin{cases} \epsilon^{p-2}, & \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2 \le \epsilon \\ \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^{p-2}, & \text{otherwise}. \end{cases} \tag{4}$$

Proof: Our proof is equivalent to showing that

$$\mathcal{K}_p^\epsilon(x) = \min_{\omega \ge \epsilon} g(\omega) \triangleq |\omega|^{p-2}|x|^2 + \frac{2-p}{p}\left(|\omega|^p - \epsilon^p\right) \tag{5}$$

and the unique minimizer is given by

$$\omega^\star = \begin{cases} \epsilon, & |x| \le \epsilon \\ |x|, & |x| > \epsilon \end{cases}. \tag{6}$$

[1] If $x = 0$ or $|x| > \epsilon$, the inequality (2) holds trivially. If $0 < |x| < \epsilon$, it can be readily derived by checking that $\theta(x) \triangleq \kappa_p(x) - \kappa_p^\epsilon(x)$ is nondecreasing resp. over both $[-\epsilon, 0)$ and $(0, \epsilon]$, eventually resulting in $\theta\left(0^+\right) = 0 \le \theta(x) \le \theta(\pm\epsilon) = \frac{(2-p)\epsilon^p}{2p}$, as desired

The above fact can be verified under two cases: (i) $|x| \leq \epsilon$ and (ii) $|x| > \epsilon$. First, for $\omega \geq |x|$, we have

$$g'(\omega) = (2-p)\omega|\omega|^{p-2}\left(1 - \left|\frac{x}{\omega}\right|^2\right) \geq 0,$$

which implies that $g(\omega)$ is a non-decreasing function when $\omega \geq |x|$ and hence the minimizer of $g(\omega)$ over $\omega \geq \epsilon \geq |x|$, i.e., Case (i), is $\omega^\star = \epsilon$. Similarly, the minimizer of $g(\omega)$ over $\omega \geq |x| > \epsilon$ is $\omega^\star = |x|$. In addition, it can be shown that $g(\omega)$ is strictly convex within $\epsilon \leq \omega < |x|$ by noting that

$$\begin{aligned} g''(\omega) &= (2-p)\omega|\omega|^{p-3}\left[(3-p)\left|\frac{x}{\omega}\right|^2 + (p-1)\right] \\ &\geq (2-p)\omega|\omega|^{p-3} > 0. \end{aligned}$$

This admits an optimal solution $\omega^\star = |x|$ of minimizing $g(\omega)$ over $\epsilon \leq \omega \leq |x|$ by checking $g'(\omega) = 0$. Therefore, the minimizer of Eq. (5) under Case (ii) is $\omega^\star = |x|$. The above two cases result in Eq. (6). Then, substituting ω^* back into $g(\omega)$ directly leads to the left-hand side of Eq. (5). Then, we can derive that

$$\begin{aligned} \mathcal{G}_p &= \sum\nolimits_{(i,j)\in\mathcal{E}_s} [\boldsymbol{A}]_{ij}\kappa_p^\epsilon\left(\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2\right) \\ &\leq \sum\nolimits_{(i,j)\in\mathcal{E}_s} [\boldsymbol{A}]_{ij}[\Omega]_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2 + \text{ const.}, \end{aligned} \tag{7}$$

as desired.

C. Extension and Implementation Details of U-MP

We can define the layer-wise GLD problem to refine the node representation for each U-MP layer. Specifically, the layer-wise GLD problem for the $k+1$-th U-MP layer can be formulated as follows:

$$\mathcal{L}_{\text{GLD}}(\boldsymbol{Y}^{k+1}, \boldsymbol{L}^{k+1}) \triangleq \left\|\boldsymbol{Y}^{k+1} - \mathcal{F}(\boldsymbol{Y}^k; \boldsymbol{W}^k)\right\|_F^2 + \mathcal{G}_{\text{GLR}}(\boldsymbol{Y}^{k+1}, \boldsymbol{L}^{k+1}), \tag{8}$$

Here, $\mathcal{F}(\boldsymbol{Y}^k; \boldsymbol{W}^k)$ denotes a function (parameterized by $\boldsymbol{W}^k$) of $\boldsymbol{Y}^k$. For example, we can have $\mathcal{F}(\boldsymbol{Y}^k; \boldsymbol{W}^k) = \text{MLP}[\boldsymbol{Y}^k; \boldsymbol{W}^k]$, *i.e.*, a multi-layer perceptron with weights $\boldsymbol{W}^k$. In this way, a transformation weight matrix $\boldsymbol{W}^k$ can be integrated into the $k+1$-th U-MP layer as follows:

$$\begin{cases} \tilde{\boldsymbol{A}}^{(k+1)} = \text{Normalize}(\mathcal{H}(\hat{\boldsymbol{Y}}^k)) \\ \boldsymbol{Y}^{(k+1)} = \text{ReLU}(\frac{1}{3}(\hat{\boldsymbol{Y}}^{(k)} + \tilde{\boldsymbol{A}}^{(k+1)}\hat{\boldsymbol{Y}}^{(k)} + \hat{\boldsymbol{Y}}^{(k)})) \end{cases}, \tag{9}$$

where

$$\hat{\boldsymbol{Y}}^k = \mathcal{F}(\boldsymbol{Y}^k; \boldsymbol{W}^k). \tag{10}$$

In addition, to choose an appropriate value of ϵ and efficiently reweight the attention map, we set the size of the output node representation in each U-MP layer to O and utilize the softmax function to normalize it.

D. Qualitative Results

As shown in the first row of Figure 1, Graph-RCNN [3] produces similar attention weights for each node; therefore, elements in each column of its attention map are similar. Our baseline model tends to produce high attention weight for two spatially nearby nodes, *e.g.*, *chair* and *umbrella1*, which results in ambiguous object representations. Compared with Graph-RCNN [3] and the baseline model, RU-Net produces a more reasonable attention map that has medium weight for two disparate nodes, *e.g.*, *chair* and *umbrella1* and has high weight for two nodes with the same object categories, *e.g.*, *umbrella1* and *umbrella2*. In the first row of Figure 2, it is shown that RU-Net achieves more diverse and accurate relationship predictions than MOTIFS [4] for the four triplets with the same object categories, *i.e.*, *man* and *bike*.

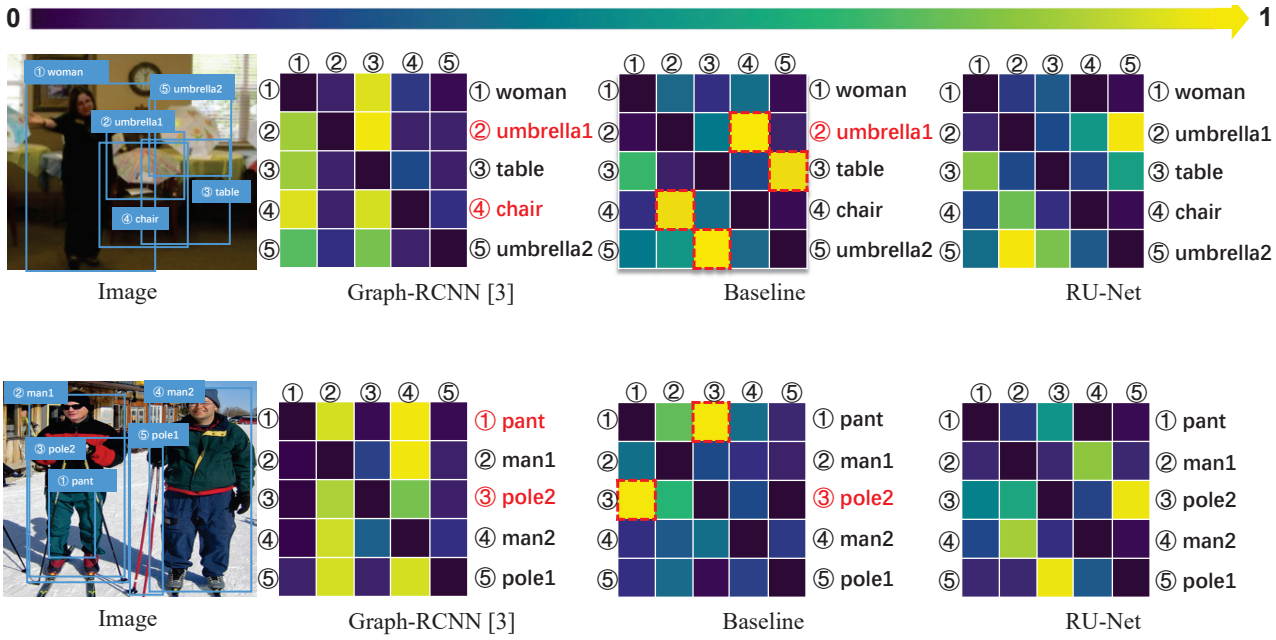

Figure 1. Qualitative comparisons between RU-Net, Graph-RCNN [3], and our baseline model in the SGCLS setting. Specifically, we show the attention maps and node prediction results by different models. The red color indicates misclassified objects. The red dotted box denotes the spurious correlations between nodes. Best viewed in color.

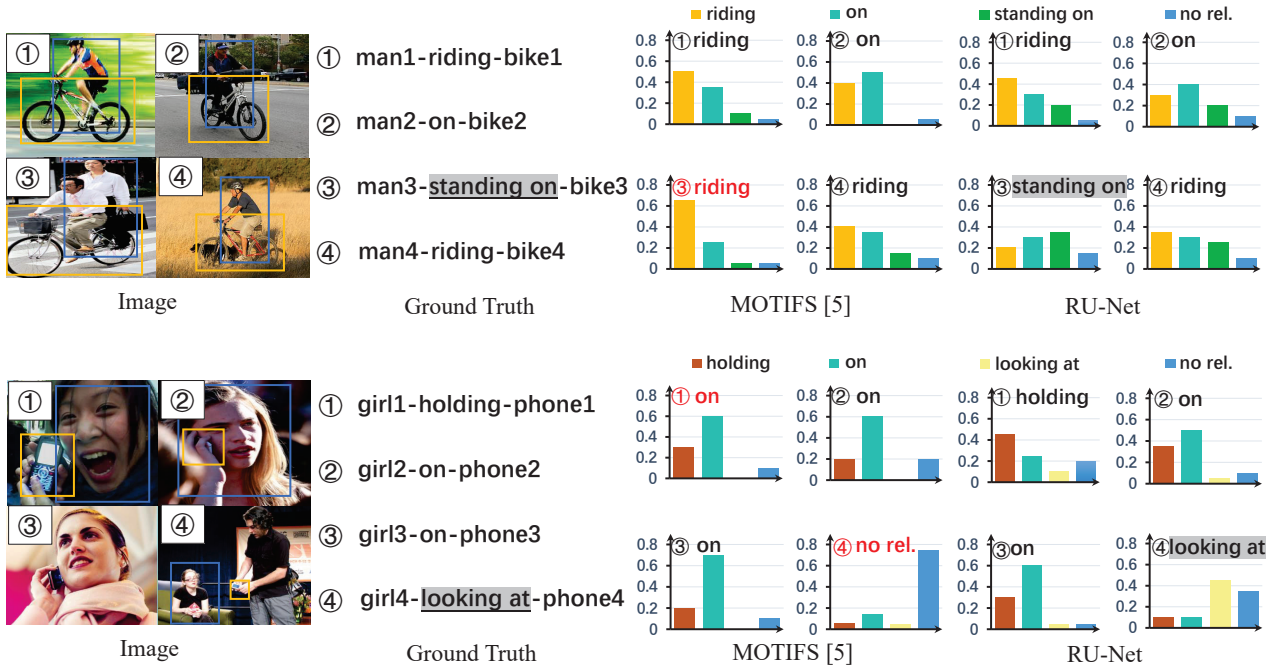

Figure 2. Qualitative comparisons between RU-Net and MOTIFS [4] in the PREDCLS setting. Specifically, we compare the relationship predictions between MOTIFS [4] and RU-Net. The red color indicates misclassified relationships. The gray color highlights missing relationship annotations. Best viewed in color.

References

[1] P. Huber. *Robust statistics*. 2004. 1

[2] J. Song, P. Babu, and D. Palomar. Sparse generalized eigenvalue problem via smooth optimization. *TSP*, 2015. 1

[3] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, 2018. 2, 3

[4] R. Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018. 1, 2, 3