Supplementary Materials

In the supplementary materials, we investigate in detail how different settings of subgraph-based clustering influence the performance of our proposed model.

Experiments on different Subgraph clustering thresholds

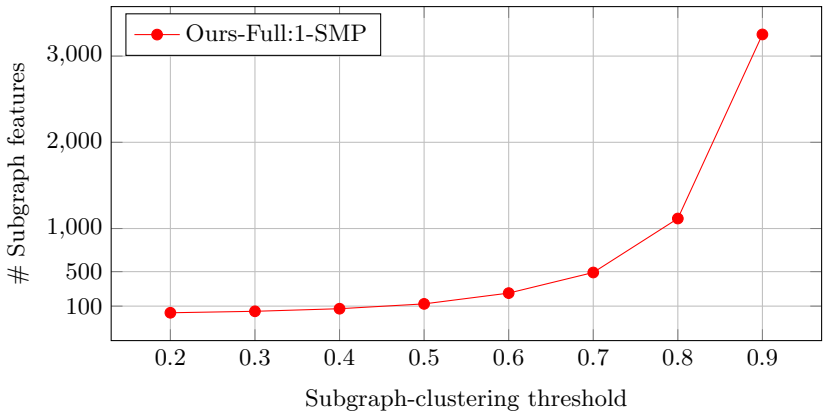

Fig. 4: Relationship between the number of subgraph features and the subgraph-clustering threshold (NMS thresholds in Sec. 3.3). Model is evaluated on VRD [37]. 200 object proposals are adopted.

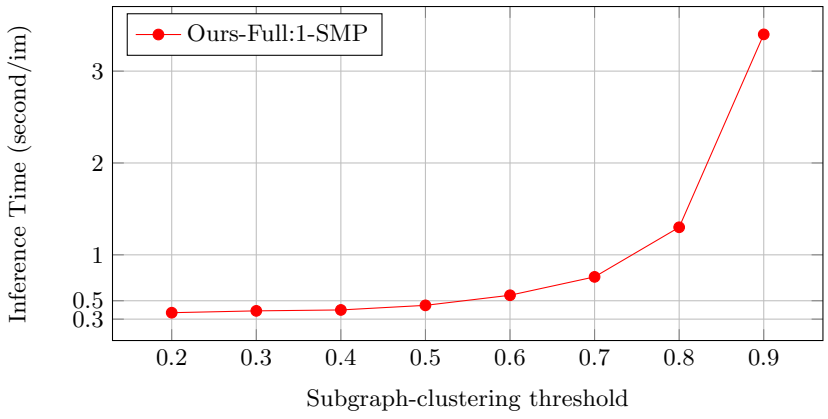

Fig. 5: Relationship between the model inference time and the subgraph-clustering threshold (NMS thresholds in Sec. 3.3). Model is evaluated on VRD [37]. 200 object proposals are adopted. Model 5 in Tab. 2 is used.

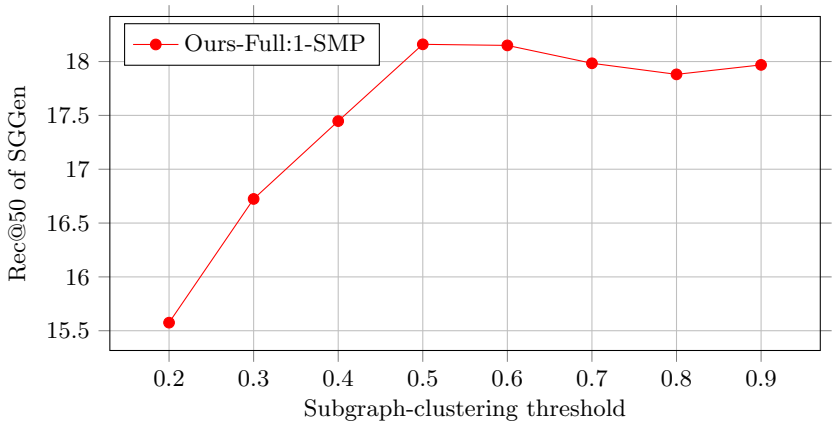

Fig. 6: Top-50 Recalls (Rec@50) of Scene graph generation (SGGen) with different the subgraph-clustering thresholds (NMS thresholds in Sec. 3.3) on VRD [37]. 200 object proposals are adopted. Model 5 in Tab. 2 is used.

From Fig. 4 (the relationship between the subgraph number and the threshold) and Fig. 5 (the relationship between the inference time and the threshold), we can see that, as the subgraph clustering threshold grows, more subgraph features are retained and the inference time grows correspondingly. However, Fig. 6 shows that higher clustering threshold (more subgraph features) does not always mean better performance, where the higher Rec@50 on Scene Graph Generation is achieved at 0.5∼0.6. It is mainly because more subgraph features introduce more complicated connections to the object features (*i.e.* each object feature should connect to more subgraph features), which deteriorates the object feature learning.

To summarize, higher threshold retains more subgraph features and usually improves the model performance. But too many subgraph features deteriorate the object feature learning and the inference speed. From Fig. 6 we can see that the highest Rec@50 is achieved at 0.5∼0.6. Thus, we select 0.5 as the subgraph-clustering threshold to balance the inference speed and the model accuracy.

Experiments on different numbers of object proposals

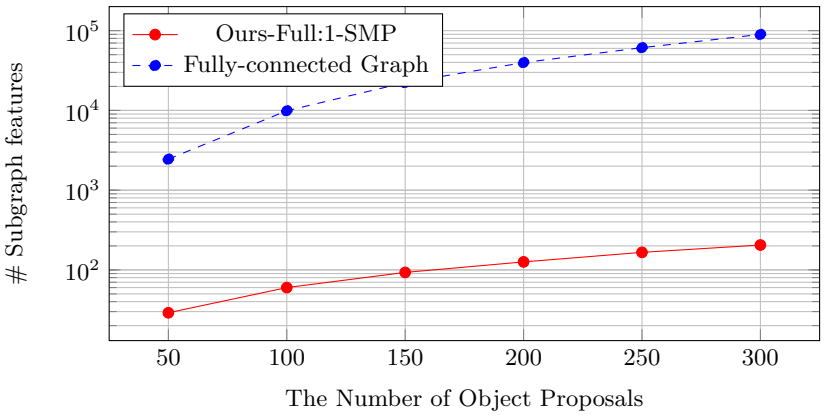

Fig. 7: Relationship between the number of subgraph features and the number of object proposals on VRD [37]. The dashed blue lines denotes the number of edges in the fully-connected graph. Subgraph clustering threshold is set to 0.5.

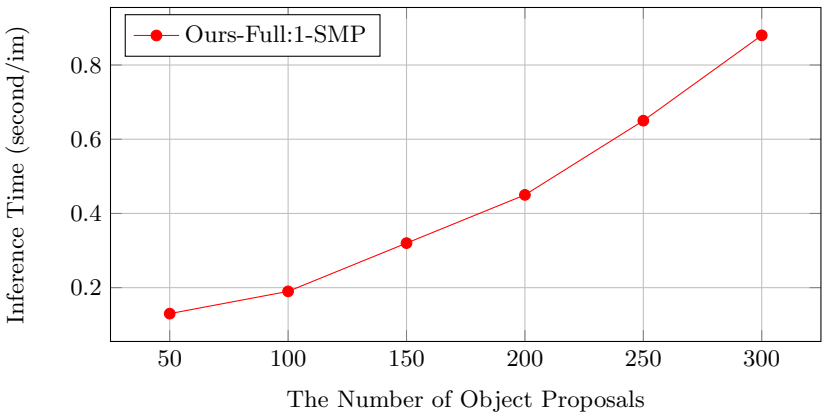

Fig. 8: Relationship between the inference time and the number of object proposals on VRD [37]. Subgraph clustering threshold is set to 0.5. Model 5 in Tab. 2 is used.

From Fig. 7 (the relationship between the number of subgraph features and that of object proposals), we can see that, as the number of object proposals grows, we have more subgraph features. Compared to the number of edges in the fully-connected graph (blue dashed line in Fig. 7), our proposed subgraph clustering method significantly reduces the representations in the intermediate

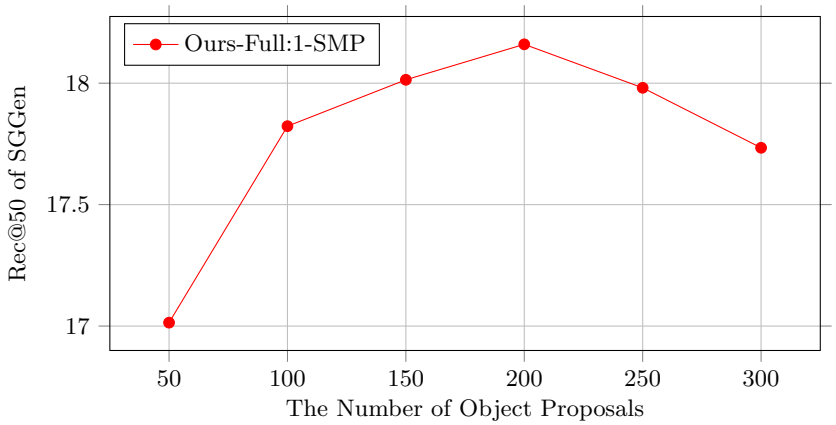

Fig. 9: Top-50 Recalls (Rec@50) of Scene graph generation (SGGen) with different numbers of object proposals on VRD [37]. Subgraph clustering threshold is set to 0.5. Model 5 in Tab. 2 is used.

level stage. Moreover, Fig. 8 (the relationship between the inference time and the number of object proposals) shows that the model speed gets slower when more object proposals are used. Fig. 9 shows that more object proposals do not always bring higher recall. It is mainly because that, too many object proposals will lead to the situation where many phrase features are merged into the subgraph representations, which deteriorates the subgraph feature learning.

To summarize, more object proposals can improve the model performance, but too many object proposals will deteriorate the model accuracy and the inference speed. Thus, we select 200 object proposals in our work to balance the inference speed and the model accuracy.

Qualitative Results

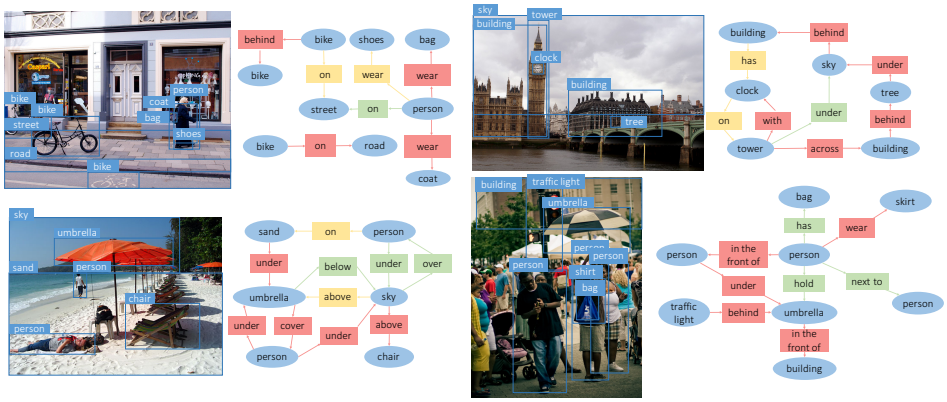

Fig. 10: Qualitative Results. Green denotes *hit by top-50 detection*. Yellow denotes *hit by Top-100 but missed by Top-50*. Red denotes *missed by top-100*.

Table 4: Object detection results on VG-MSDN [28, 35]. FRCNN-**K** denotes Faster R-CNN with **K** proposals adopted during the testing stage. MSDN [35] adopts 64 proposals, while ours uses 200 proposals. **Ours-w/o-Rel** denotes training our F-Net without relationship annotations (just object annotations).

Model	FRCNN-64 [48]	FRCNN-300 [48]	MSDN [35]	Ours-w/o-Rel	Ours
mean AP(%)	6.72	10.21	7.43	13.02	**15.70**

Further Investigation on Object Detection

To investigate the improvement on detecting objects, we evaluate our proposed F-Net with baseline Faster R-CNN [48]. In addition, we also compare our method with MSDN [35], which has reported the object detection result. Detailed results are shown in Tab. 4.

By comparing Faster R-CNN-300 with our F-Net without relationship annotations denoted by Ours-w/o-Rel in Tab. 4, 2.81% absolute mean AP increase shows that our proposed F-Net itself can learn to model the connections between objects and their contextual regions without relationship annotations. Furthermore, when the relationship annotations are used, more gains are observed. It is mainly because the relationship annotations can help to learn a better subgraph representations, and a better subgraph feature can further help the object feature learning with message passing. In comparison, MSDN performs worse than our method because their framework can only use fewer (64) RPN proposals, restricted by the GPU memory size. Comparison between Faster R-CNN-64 and MSDN validates our claim and shows that MSDN could improve the object detection given the same number of proposals.