Vision Relation Transformer for Unbiased Scene Graph Generation
– Supplemental Material –

Gopika Sudhakaran[1,3] Devendra Singh Dhami[1,3] Kristian Kersting[1,2,3] Stefan Roth[1,2,3]

[1]Department of Computer Science, Technical University of Darmstadt, Germany
[2]Centre for Cognitive Science, TU Darmstadt [3]Hessian Center for AI (hessian.AI)

A. Introduction

This document contains additional results and analyses that were excluded from the main document due to space constraints. We examine the effectiveness of VETO + MEET over other MEET debiased models and perform a comparison of the computational cost advantages achieved by VETO when contrasted with its baseline methods.

B. Additional Network Details

Feature extractor. The extracted feature maps $\mathbf{r}$ from the ResNeXt-101-FPN [27, 34, 54] backbone consist of 4 spatial scales: $(1/4, 1/8, 1/16, 1/32) \rightarrow (\mathbf{r}^0, \mathbf{r}^1, \mathbf{r}^2, \mathbf{r}^3)$; the extracted geometric features $\mathbf{g}$ from ResNet-50 [24, 53] consist of a single spatial scale: $(1/8) \rightarrow (\mathbf{g})$. Each bounding box $\mathbf{b}_i$ is mapped to the corresponding scale to extract entity RGB features from $\mathbf{r}$ and to the fixed scale to extract the entity geometric features from $\mathbf{g}$. The extracted features from both the modalities are ROIAligned [52] and average pooled to obtain the visual features $\mathbf{v}$ and depth features $\mathbf{d}$.

Relation network. The feature projection dimensions of the local-level entity visual and depth features $\mathbf{v}$ and $\mathbf{d}$ are chosen as $p^v = 64$ and $p^d = 512$, respectively. The transformer input consists of 19 tokens, which comprise 16 local-level entity tokens, 2 tokens from location features $\mathbf{l}$ and semantic features $\mathbf{w}$, and a learnable (class) token. For the transformer feed-forward network, we use a hidden dimensionality that is double the token dimensionality, with a dropout of 0.35. We use the predicate class split with $G = 5$ groups [6] for MEET training.

C. Additional Experimental Results

C.1. MEET analysis

In this section, we compare VETO + MEET to its baselines to analyse the effectiveness of MEET training.

Fig. 8 reveals that the combination of VETO with MEET has the highest gain in terms of both R@k and mR@k after MEET debiasing, followed by the combination of MEET with Motifs [43] and SHA [6]. It can also be seen that for VCTree [28] and VTransE [45], there is a recall drop after debiasing. The pattern reveals that the strength of the underlying relation network has a crucial influence on the efficacy of MEET. We further examine the predicate class-level improvement of VETO + MEET over its strong counterparts Motifs + MEET in Fig. 9 and SHA + MEET in Fig. 10. Fig. 11 shows that debiasing VETO using MEET notably improves prediction. The improvement pattern sets VETO + MEET apart from debiased models like SHA + GCL [6], which exhibits a significant drop in head class performance. In contrast, VETO + MEET enhances prediction across head, body, and tail classes. For all the comparisons, the major boost for VETO + MEET comes from the body and tail classes. However, even a slight improvement in head classes is remarkable because those predicate classes are present in the majority of the samples. Thus, VETO + MEET shows a consistent performance gain over its baselines across the entire predicate class frequency distribution.

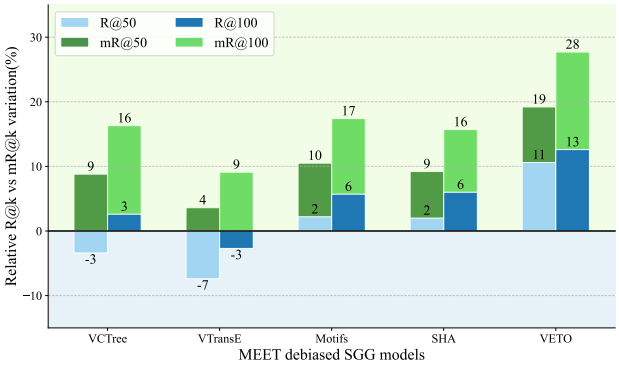

Figure 8. **Effect of debiasing with MEET on Recall (R) and mean Recall (mR)**. R@k and mR@k improvement/drop of MEET-debiased models relative to their vanilla versions.

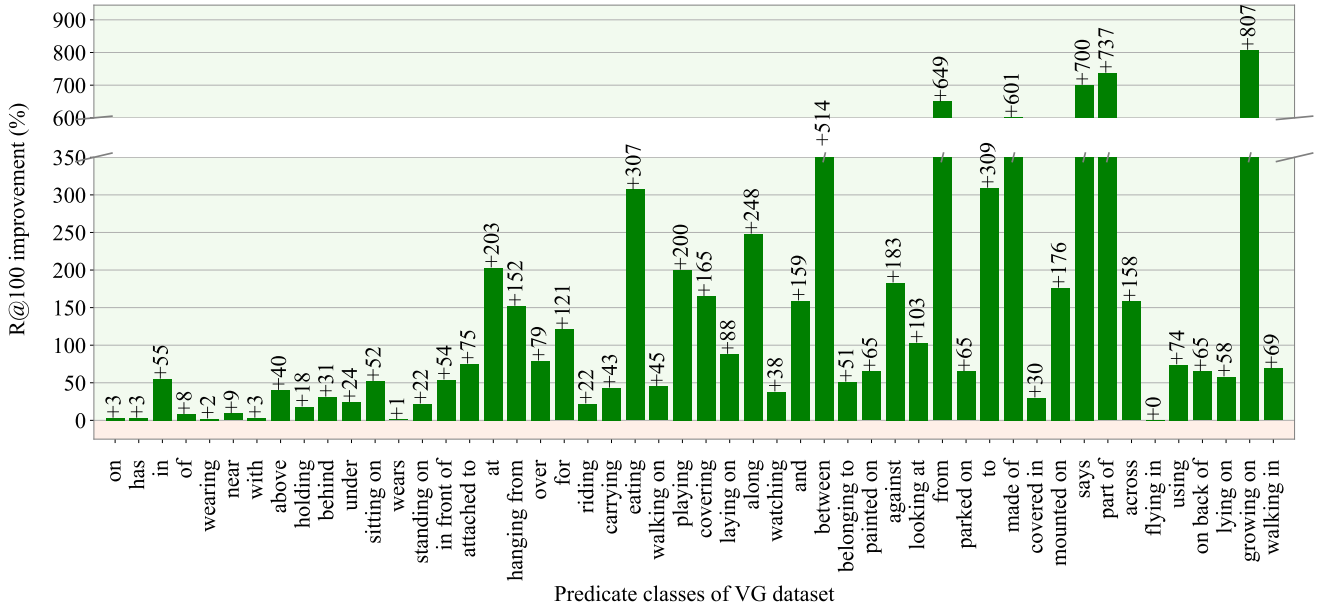

Figure 9. **R@100 improvement on PredCls for VETO + MEET over Motifs + MEET.** The predicates are sorted based on their frequency in descending order.

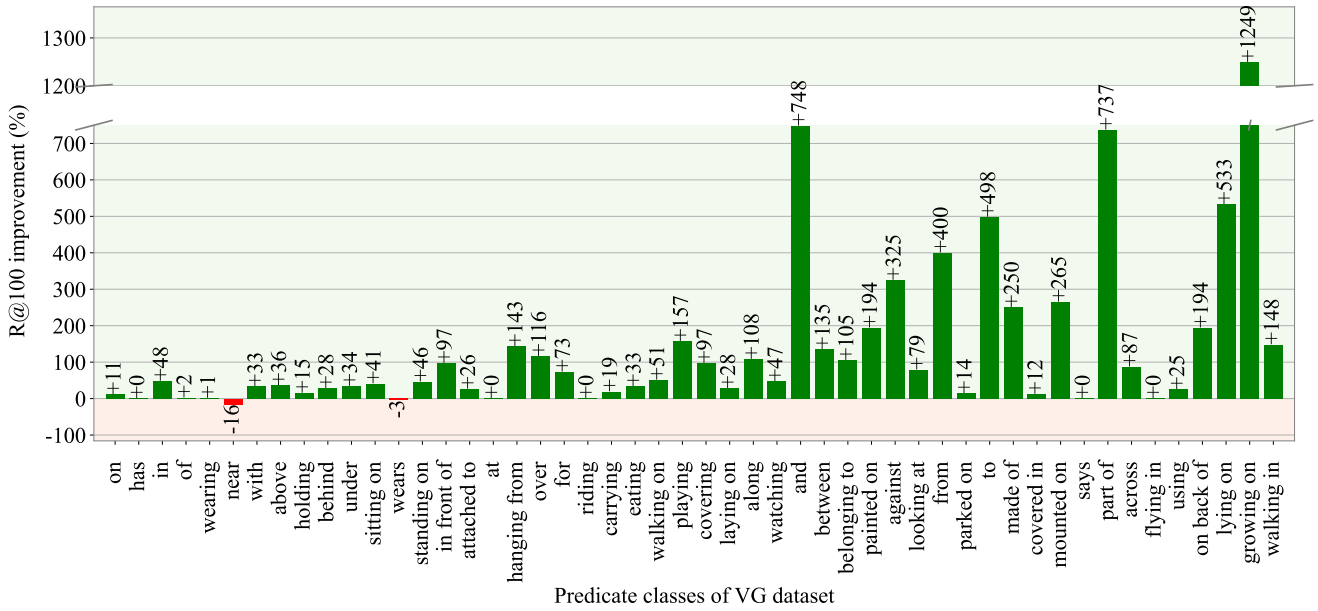

Figure 10. **R@100 improvement on PredCls for VETO + MEET over SHA + MEET.** The predicates are sorted based on their frequency in descending order.

C.2. Computation cost comparisons

Tab. 7 compares the training time in seconds/iteration (Sec./Itr), inference time per image in milliseconds (Inf.Time (ms)/Img), number of trainable parameters and total parameters in millions (Train. par. (M) and Total par. (M)), and maximum memory consumption in Gigabytes (GB) for batch size 8. The results clearly show that VETO outperforms the comparison models in several aspects. It possesses significantly fewer total parameters, leading to lower GPU memory consumption while exhibiting competitive training and inference times, making it a more efficient choice overall.

References

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE Inter-*

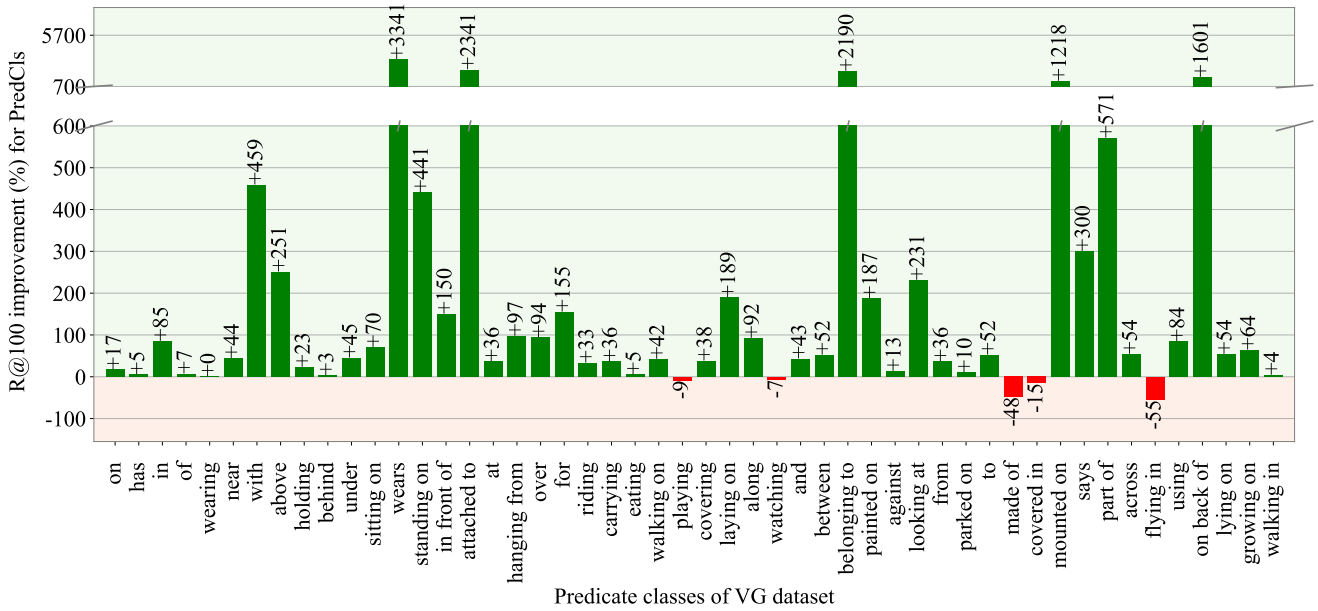

Figure 11. **R@100 improvement of VETO + MEET over VETO.** The predicates are sorted based on their frequency in descending order.

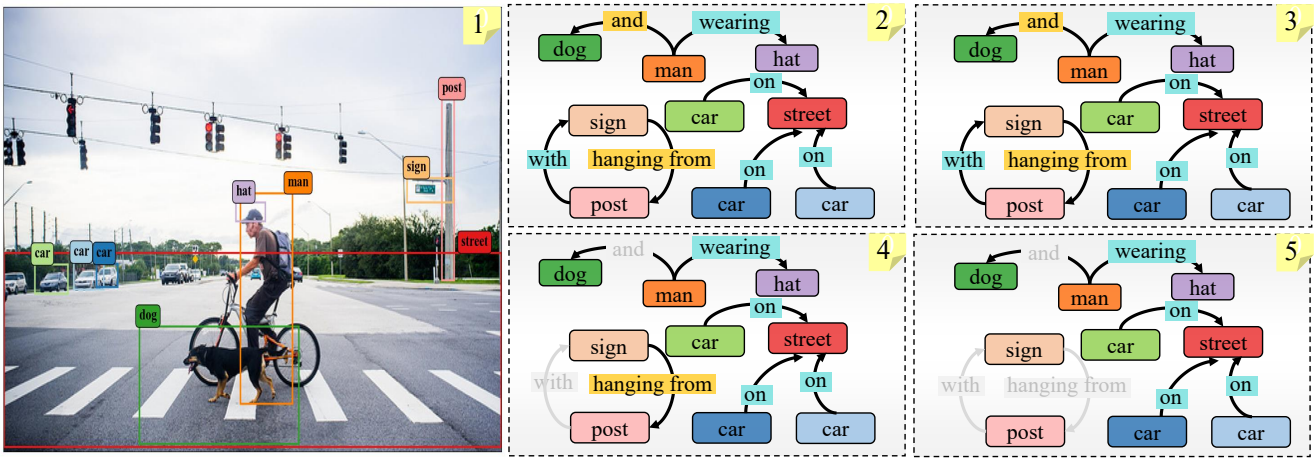

Figure 12. **Qualitative example**. Head and tail relations are highlighted in blue and yellow, respectively. Greyed out relations and arrows denote missed predictions. *(1)* Illustrative VG sample with ground-truth bounding boxes and labels; *(2)* SGG ground-truth; *(3)* VETO + MEET predicts all the head and tail classes for this example; *(4)* Motifs + MEET misses head class *with* and tail class *and*; *(5)* SHA + MEET also misses head class *with* and tail classes *and*, *hanging from*.

Table 7. **Computational cost comparison of MEET debiased SGG models on PredCls.** Colors in the table vary from blue to green to depict the cost improvement.

Model	Sec./ Itr.	Inf.Time (ms)/Img	Train. par.(M)	Total par.(M)	Max. mem.(GB)
VCTree	1.73	142	268	430	42
SHA	1.48	100	230	392	39
VTransE	0.84	75	199	361	37
Motifs	0.98	70	205	367	37
VETO	0.8	70	20	182	24

national Conference on Computer Vision, pages 2961–2969, 2017.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[54] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.