Abstract

This supplementary material is organized as follows: 1) section A: a more detailed introduction to predicate probability distribution. 2) section B: more quantitative studies.

A. Predicate probability distribution

In this section, predicate probability distribution (PPD) we mentioned in Section 3 will be elaborated with more details.

As we mentioned in Section 3.2.1, we represent each predicate class as a prediction probability distribution with a dimension of predicate categories. In particular, on VG150 dataset with 50 predicate categories, each predicate can be represented as a vector with dimension 50, where each dimension represents the estimated prediction probability over the corresponding predicate class. And the mapping relationship between each dimension of predicate probability distribution and the corresponding predicate category is fixed and corresponds to the dictionary order of each class of predicates. As shown in Fig.7, we illustrate the PPD of the predicate "walking on", which is generated by the VCTree+PPDL method on the VG150 dataset.

Just as we discussed in Section 3.2.2, in order to estimate the real probability distribution of each predicate class, we propose a dynamic updating strategy for the PPD during training time. And here, to further elaborate the PPD dynamic updating process in more detail, we demonstrate the dynamic updating process in Fig.6. As shown in Fig.6, a vector with a value of 1 in the corresponding dimension of the predicate "walking on" is used to initialize the PPD of "walking on", and then the average PPD of each mini-batch is used to update previous estimated PPD by the hyperparameter α. Finally, the estimated predicate probability distribution of the predicate "walking on" can be obtained at the end of training.

B. Quantitative studies

The full results of SGG, including both conventional Recall@K and the adopted mean Recall@K, are given in Table 6. Although our proposed PPDL method shows some performance degradation in the Recall@K metric, our method achieves comparable or less performance loss on Recall@K than most other debiasing methods (e.g., TDE [28], CogTree [38]), and at the same time, it achieves a more substantial mean Recall@K performance improvement.

To quantitatively illustrate the outstanding performance of our proposed PPDL method in eliminating the long-tailed bias, we show the mean Recall@100 of each 10 predicate classes among the top 50 frequent predicate categories in Table 5. Intuitively, we can see that the performance of tail predicate classes improves significantly on both the biased VCTree method and the unbiased TDE method after applying our PPDL method, and this result indicates that our method can significantly improve the performance of low-frequency predicates and thus effectively eliminate the long-tailed bias of SGG.

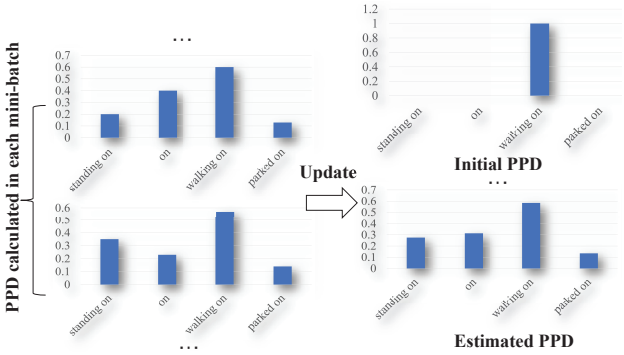

Figure 6. A brief illustration about the dynamic updating strategy for the predicate probability distribution of predicate "walking on".

Method	Average R@100				
	1-10	11-20	21-30	31-40	41-50
VCTree‡ [29]	**60.8**	17.8	16.1	3.1	1.9
VCTree‡+PPDL	39.1	**27.5**	**42.9**	**35.6**	**36.8**
Unbiased‡ [28]	**56.6**	38.6	44.4	18.6	2.4
Unbiased‡+PPDL	53.5	**39.5**	**50.3**	**34.8**	**17.5**

Table 5. Mean Recall@100 metrics of each 10 predicate classes among the top-50 frequent predicate categories in the PredCls task on VG150. The Unbiased [28] method means the TDE method applying on the VCTree [29] baseline, and it is equivalent to the previously mentioned VCTree+TDE method. † and ‡ are with the same meaning as in Table 1 of the main paper.

Furthermore, the detailed predicate-level Recall@100 on PredCls sub-task of all four models, VCTree baseline and TDE *vs.* VCTree+PPDL and VCTree+PPDL&TDE, are illustrated in Fig.8. It is clear that our method performs better on tail predicate classes compared to the VCTree baseline, and impressively we can see that applying both PPDL and TDE to the VCTree baseline can further improve the performance on tail predicate classes with limited performance loss on head classes. The TDE method [28] eliminates the visual bias by masking the visual features in the bounding box region and further applies counterfactual inference to generate an unbiased scene graph. Thus, we speculate that it is the annotation error of the fixed ground truth bounding box that undermines the effectiveness of the TDE method. In SGCls task, the performance improvement of VCTree+TDE+PPDL combination is poor due to the fixed bounding box and the lack of ground truth object labels.

Method	Predicate Classification		Scene Graph Classification		Scene Graph Generation	
	mR@20/50/100	R@20/50/100	mR@20/50/100	R@20/50/100	mR@20/50/100	R@20/50/100
IMP† [12]	- /9.8/10.5	52.7/59.3/61.3	- /5.8/6	31.7/34.6/35.4	- /3.8/4.8	14.6/20.7/24.5
MOTIFS† [41]	10.8/14.0/15.3	58.5/65.2/67.1	6.3/7.7/8.2	32.9/35.8/36.5	4.2/5.7/6.6	21.4/27.2/30.3
VCTree† [29]	14.0/17.9/19.4	60.1/66.4/68.1	8.2/10.1/10.8	35.2/38.1/38.8	5.2/6.9/8.0	22.0/27.9/31.3
PCPL† [34]	- /35.2/37.8	- /50.8/52.6	- /18.6/19.6	- /27.6/28.4	- /9.5/11.7	- /14.6/18.6
IMP+EBML† [26]	9.43/11.8/12.8	- / - / -	5.7/6.8/7.2	- / - / -	2.8/4.2/5.4	- / - / -
IMP‡+PPDL	22.3/24.8/25.3	38.4/39.5/39.7	13.0/14.2/15.9	24.5/25.8/26.7	7.8/9.8/10.4	14.4/18.5/19.4
MOTIFS+TDE† [28]	18.5/25.5/29.1	33.6/46.2/51.4	9.8/13.1/14.9	21.7/27.7/29.9	5.8/8.2/9.8	12.4/16.9/20.3
MOTIFS+CogTree† [38]	20.9/26.4/29.0	31.1/35.6/36.8	12.1/14.9/16.1	19.4/21.6/22.2	7.9/10.4/11.8	15.7/20.0/22.1
MOTIFS+EBML† [26]	14.2/18.0/19.5	- / - / -	8.2/10.2/11.0	- / - / -	5.6/7.7/9.1	- / - / -
MOTIFS‡+PPDL	27.9/32.2/33.3	44.4/47.2/47.6	15.8/17.5/18.2	26.6/28.4/29.3	9.2/11.4/13.5	17.7/21.2/23.9
VCTree+TDE† [28]	18.4/25.4/28.7	36.2/47.2/51.6	8.9/12.2/14.0	19.9/25.4/27.9	6.9/9.3/11.1	14.0/19.4/23.2
VCTree+CogTree† [38]	22.0/27.6/29.7	39/44.0/45.4	15.4/18.8/19.9	27.8/30.9/31.7	7.8/10.4/12.1	14.0/18.2/20.4
VCTree+EBML† [26]	14.2/18.2/19.7	- / - / -	10.4/12.5/13.5	- / - / -	5.7/7.7/9.1	- / - / -
VCTree+TDE&EBML† [26]	19.9/26.7/30.0	- / - / -	13.9/18.2/20.5	- / - / -	7.1/9.7/11.6	- / - / -
VCTree‡+PPDL	29.7/33.3/33.8	45.1/47.6/48.0	20.3/21.8/22.4	20.4/32.1/33.0	9.1/11.3/13.3	16.7/20.1/22.9
VCTree+TDE‡&PPDL	25.3/33.0/36.2	34.7/41.6/43.6	16.2/20.2/22.0	20.4/24.8/26.2	9.4/12.2/14.4	9.8/13.6/16.5

Table 6. Comparison of Recall@K and mean Recall@K for PredCls, SGCls and SGGen tasks on VG150. † and ‡ are with the same meaning as in Table 1 of the main paper.

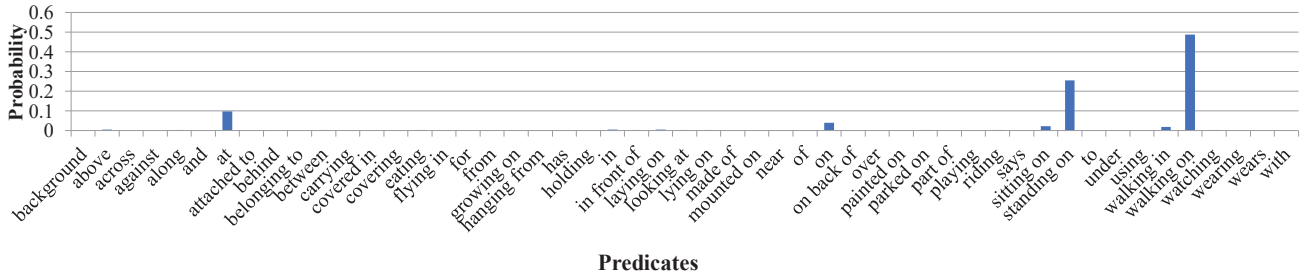

Figure 7. The estimated probability distribution of predicate "walking on" generated by VCTree+PPDL method.

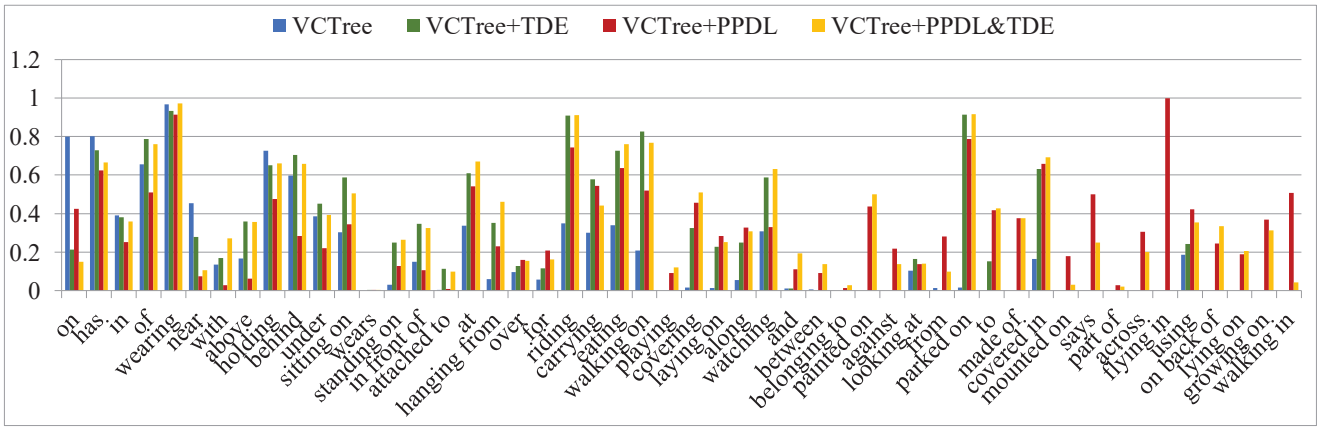

Figure 8. Performance comparison among several methods on VG150 dataset. The constrained R@100 for the top 50 predicate classes on the PredCls task is presented. Best viewed in color.