Supplementary Material for "Unbiased Scene Graph Generation from Biased Training"

Kaihua Tang[1], Yulei Niu[3], Jianqiang Huang[1,2], Jiaxin Shi[4], Hanwang Zhang[1]
[1]Nanyang Technological University, [2]Damo Academy, Alibaba Group, [3]Renmin University of China, [4]Tsinghua University
kaihua001@e.ntu.edu.sg, niu@ruc.edu.cn, jianqiang.jqh@gmail.com
shijx12@163.com, hanwangzhang@ntu.edu.sg

Abstract

This supplementary document is organized as follows: 1) section A: a comprehensive review of causal effect analysis in causal inference; 2) section B: more details of the simplified network structures in the original paper; 3) section C: more quantitative studies; 4) section D: more qualitative studies.

A. Review of Causal Effect Analysis

In this section, a comprehensive review of causal effect analysis is given in the form of the causal graph we proposed in Section 3, and we still follow the notations from the original paper. More detailed background knowledge about causal inference can be found in [11, 12] while the extension of effect analysis (a.k.a. mediation analysis) is given in [14, 10, 20, 19].

A.1. Mediator

Since the exhaustive introduction of causal inference would beyond the scope of this paper, we simplified or skipped the definitions of several concepts in the original paper without affecting the understanding. One of the skipped concepts is the mediator. In a causal graph, when we care about the effect of a variable X to the output variable Y, the descendant node of X that is located in the path between them is the mediator. For example, in the study of carcinogenesis by smoke (Cigarette $\rightarrow$ Nicotine $\rightarrow$ Cancer), nicotine is the mediator. In our case, object labels Z is the mediator of X to Y, which can be considered as the side effect of X that also affects Y.

A.2. Total, Direct and Indirect Effects

As we discussed in Section 4.2, without further counterfactual intervention on the mediator Z, the overall effect of X towards Y is regarded as the Total Effect (TE) of X on Y, which can be calculated as:

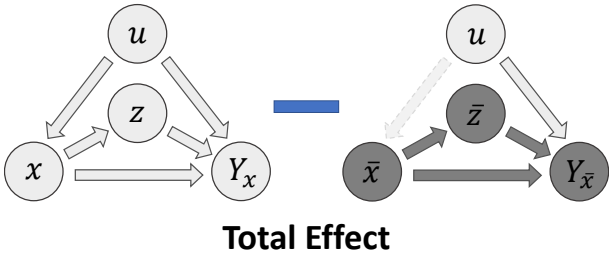


Figure 1. The illustration of Total Effect on causal graph.

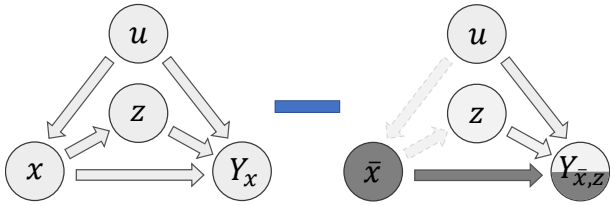
$$TE = Y_x(u) - Y_{\bar{x}}(u). \tag{1}$$

As illustrated in Figure 1, other than the path $I \rightarrow X$ that is cut off by the intervention $X = \bar{x}$, all the other variables will take their values through the links of causal graph. Especially, the mediator Z will get value $\bar{z}$, which is calculated from Eq. (2) given $\bar{x}$ as input.

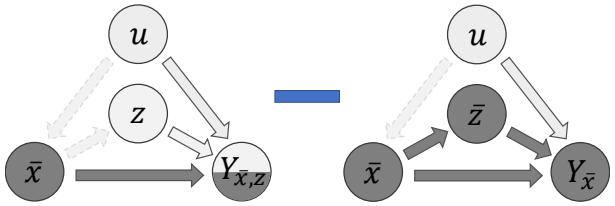
However, by only using the TE, we are still not able to separate the mediator-specific "causal effect" from "side effect", which limits the value of causal effect analysis. Thanks to the development of causal inference, here comes the decomposition of TE [10, 20]. Generally, the TE of X is composed of the Direct Effect (DE) caused by the causal path $X \rightarrow Y$ and Indirect Effect (IE) caused by the side-effect path $X \rightarrow Z \rightarrow Y$. Depending on whose effect we want to obtain, two kinds of decomposition can be applied.
Decomposition 1: The first kind of decomposition is what we used in the Section 4.2, which separates the TE into the Total Direct Effect (TDE) and the Natural/Pure Indirect Effect (NIE/PIE). The former one has already been defined in the original paper as:

$$TDE = Y_x(u) - Y_{\bar{x},z}(u), \tag{2}$$

which can be regarded as the effect of X in the real situation, *i.e.*, Z always takes the value z as if it had seen the real x. Meanwhile, the NIE or PIE is the effect caused by

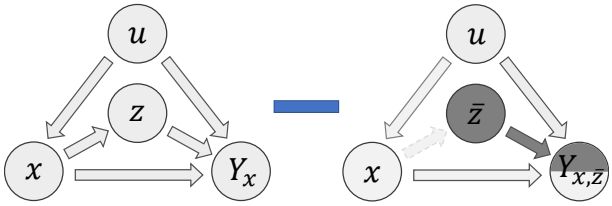


(a) Total Direct Effect

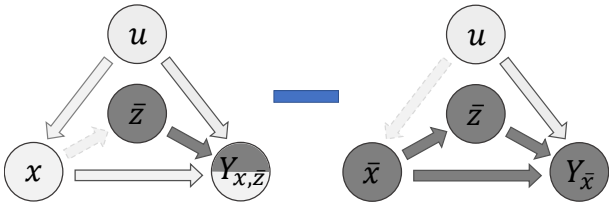


(b) Natural/Pure Indirect Effect

Figure 2. The illustration of Total Direct Effect and Pure/Natural Indirect Effect on causal graph.



(a) Total Indirect Effect



(b) Natural/Pure Direct Effect

Figure 3. The illustration of Total Indirect Effect and Pure/Natural Direct Effect on causal graph.

the mediator Z under a pure/natural situation, *i.e.*, X will not take the value x under the specific case and it's only assigned to the general unactivated value $\bar{x}$. Therefore, the NIE of Z is denoted as:

$$NIE = Y_{\bar{x},z}(u) - Y_{\bar{x}}(u) \tag{3}$$
$$= TE - TDE, \tag{4}$$

where we can easily identify that NIE is the effect of Z when it changes from $\bar{z}$ to z in a pure environment, *i.e.*, $X = \bar{x}$. The illustrations of TDE and NIE are given in Figure 2.

Decomposition 2: The second type of decomposition is opposite to the first one. It's mainly adopted when the indirect effect of the mediator is what we are looking for. For example, in the study of carcinogenesis by smoke (Cigarette $\rightarrow$ Nicotine $\rightarrow$ Cancer), sometimes the side effect of Nicotine is what researchers really care about. In this case, TE can be decomposed into Total Indirect Effect(TIE) and Natural/Pure Direct Effect (NDE/PDE). The definition of the former one is very similar to the NIE except for the environment being the real case $X = x$, which is therefore formulated as:

$$TIE = Y_x(u) - Y_{x,\bar{z}}(u). \tag{5}$$

At the same time, since direct effect is not the target, their pure/natural effect should be removed from the TE. The calculation of NDE/PDE is following:

$$NDE = Y_{x,\bar{z}}(u) - Y_{\bar{x}}(u) \tag{6}$$
$$= TE - TIE, \tag{7}$$

where NDE is the effect of X changing from $\bar{x}$ to x under the pure environment $Z = \bar{z}$. In general, we should put the effect we care under the real environment, *i.e.* TDE or TIE, so we can get the results specific to each cases.

The above two types of decomposition are both commonly used in medical, political or psychological research [13, 4, 2, 8, 6], which depends on which effect we want to obtain, main effect or side effect. Note that, if the system is a pure linear system, both two types of decomposition would be exactly the same.

B. Network Details

B.1. Scene Graph Generation

In the original paper, we simplified the feature extraction module in Link $I \rightarrow X$, the visual context module in Link $I \rightarrow Y$ and skipped the VCTree [18] construction module. Their details will be given in this subsection.

Feature Extraction Module. Since we adopted ResNeXt-101-FPN [7, 21] as the backbone, the extracted $\mathcal{M}$ contains feature maps from 4 scales: $(1/4, 1/8, 1/16, 1/32) \rightarrow (\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3)$. Each bounding box will be assigned to the corresponding $\mathcal{M}_k, (k = 0, 1, 2, 3)$ based on their areas [9]. Given a bounding box b_i with area a_i, the corresponding index k of feature map is calculated as follows:

$$k = \max(2, \min(5, \lfloor 4 + \log_2(a_i/224 + 1 \times 10^{-6}) \rfloor)) - 2. \tag{8}$$

Then ROIAlign [3] will be applied to the selected bounding box b_i on the corresponding $\mathcal{M}_k$ for the feature r_i as we described in Section 3.

Visual Context Module. To extract the visual context feature v'_e for the union box $b_i \cup b_j$, we consider all 4 feature maps will provide complementary contextual information from different levels. Therefore, we extract ROIAlign [3]

Index	Input	Operation	Output
(1)	$(\mathcal{M}_0, b_i \cup b_j)$	ROIAlign	$(7 \times 7 \times 256)$
(2)	$(\mathcal{M}_1, b_i \cup b_j)$	ROIAlign	$(7 \times 7 \times 256)$
(3)	$(\mathcal{M}_2, b_i \cup b_j)$	ROIAlign	$(7 \times 7 \times 256)$
(4)	$(\mathcal{M}_3, b_i \cup b_j)$	ROIAlign	$(7 \times 7 \times 256)$
(5)	(1-4)	Concatenation	$(7 \times 7 \times 1024)$
(6)	(5)	Conv	$(7 \times 7 \times 256)$
(7)	b_i, b_j	dummy mask	$(27 \times 27 \times 2)$
(8)	(7)	Conv+ReLU+BatchNorm	$(14 \times 14 \times 128)$
(9)	(8)	MaxPool	$(7 \times 7 \times 128)$
(10)	(9)	Conv+Relu+BatchNorm	$(7 \times 7 \times 256)$
(11)	(6),(10)	Element-wise Addition	$(7 \times 7 \times 256)$
(12)	(11)	Flatten	12,544
(13)	(12)	FC+ReLU	4,096
(14)	(13)	FC+ReLU	4,096

Table 1. The details of Visual Context Module.

features on all 4 feature maps before we project the visual context feature into a feature space of $\mathbb{R}^{4096}$. The entire module is summarized in the Table 1, where the dummy mask operation in (7) generates two masks for b_i and b_j independently, assigning 1.0 to the pixels inside the bounding box and 0.0 for the rest.

VCTree Construction Module. Unlike VTransE [24] or MOTIFS [23], that doesn't have contextual structures or simply use the position of bounding box to create the fixed left-to-right sequence structures, VCTree requires an additional module to generate the dynamic tree structures before applying TreeLSTM [17] message passing. The construction is based on a pairwise score matrix $S \in \mathbb{R}^{n \times n}$ indicating the probability of two objects having a relationship. It takes the same inputs as Eq. (1) in the original paper, and can be defined as Table 2. where n is the number of bounding boxes for each image, C is the number of object Categories. The output of (12) will be trained by the binary cross entropy loss according to whether a pair of objects has an annotated relationship, while VCTree [18] is the left-child right-sibling version of the maximum spanning tree, which is constructed based on the score of (12).

The Special Treatment for PredCls. In the original paper, we skipped a special case of causal graph, *i.e.*, causal graph for Predicate Classification (PredCls), for simplification. In PredCls, the ground truth object labels are given, which means the link $X \rightarrow Z$ is blocked by assigning ground truth labels. It won't affect TDE calculation, where Z takes the real value z. However, it's involved in the ablation studies of TE and NIE, where Z could be assigned to $\bar{z}$. In this case, $\bar{z}$ will directly use to the mean vector of training set rather than be calculated from Eq.(2). We also need to notice that, for MOTIFS [23], Eq.(3) will take z_e as input too, which is simplified in the original paper, because z_e itself is derived from x_e and it can be considered as the interaction between link $X \rightarrow Y$ and $Z \rightarrow Y$ in the causal graph.

Index	Input	Operation	Output
(1)	$\{[r_i; b_i; l_i]\}$	FC	$(n \times 512)$
(2)	(1)	FC+ReLU	$(n \times 512)$
(3)	(1)	FC+ReLU	$(n \times 512)$
(4)	(2),(3)	Unsqueeze + Element-wise Multi	$(n \times n \times 512)$
(5)	(4)	FC+ReLU+FC+Squeeze	$(n \times n)$
(6)	$\{l_i\}$	Softmax	$(n \times C)$
(7)	(6),(6)	Unsqueeze + Combination	$(n \times n \times C \times C)$
(8)	(7)	FC+Squeeze	$(n \times n)$
(9)	$\{b_i\}$	$(x_1, x_2, y_1, y_2, w, h) \times (b_i, b_j, b_i \cup b_j, b_i \cap b_j)$	$(n \times n \times 24)$
(10)	(9)	FC+Squeeze	$(n \times n)$
(11)	(5),(8),(10)	Element-wise Addition	$(n \times n)$
(12)	(11)	Sigmoid	$(n \times n)$

Table 2. The details of VCTree Construction Module.

B.2. Sentence-to-Graph Retrieval

As we mentioned in the original paper, we treated Sentence-to-Graph Retrieval (S2GR) as the graph-to-graph matching problem, parsing query captions to text-SGs by [16]. Both detected image-SGs and parsed text-SGs are composed of entities $E^k = \{e_i^k\}$ and relationships $R^k = \{r_{ij}^k = (s_i^k, p_{ij}^k, o_j^k)\}$, where $k \in \{text, image\}$, subject and object categories (s_i^k, o_j^k) share the same dictionary with e_i^k for each k, p_{ij}^k denotes the onehot vector of the predicate category.

The image-SGs and text-SGs are equipped with different embedding layers, because they have different dictionaries. The entities and relationships are encoded as:

$$E_{embed}^k = W_e^k E^k, \tag{9}$$

$$R_{embed}^k = [W_s^k S^k; W_p^k P^k; W_o^k O^k], \tag{10}$$

where $E_{embed}^k \in \mathbb{R}^{N_d \times N_e^k}$, $R_{embed}^k \in \mathbb{R}^{3N_d \times N_r^k}$, $N_d = 512$ is the dimension of embedded feature, N_e^k, N_r^k are numbers of entities and relationships for each image.

B.2.1 Bilinear Attention Scene Graph Encoding

Since entities and relationships are both important for SGs, we apply Bilinear Attention Network (BAN) [5] to encode their multimodal interactions into the same representation space. The same BAN model is used for both text-SGs and image-SGs, hence we remove k hereinafter for simplification. The original BAN involves two steps: 1) attention map generation, and 2) bilinear attended feature calculation. Because scene graph has already provides connections between entities and relationships, we skipped the first step and used normalized scene graph connection as attention map $A_{ij} = M_{ij} / \sum_j M_{ij}$, where $A, M \in \mathbb{R}^{N_e \times N_r}$, the scene graph connection M is defined as follows:

$$M_{ij} = \begin{cases} 1, \ if \ E_i \ in \ R_j, \\ 0, \ if \ E_i \ not \ in \ R_j. \end{cases} \tag{11}$$

The bilinear attended scene graph encoding is calculated by Table 3, where steps (4-10) are calculated 2 times, and the final output $E_{graph} \in \mathbb{R}^{1024}$ is a feature vector representing

Index	Input	Loop	Operation	Output
(1)	E_{embed}		Input Shape	$(N_e \times 512)$
(2)	R_{embed}		Input Shape	$(N_r \times 512)$
(3)	A		Input Shape	$(N_e \times N_r)$
(4)	(1)	start	Transpose + Unsqueeze	$(512 \times 1 \times N_e)$
(5)	(2)	↓	Transpose + Unsqueeze	$(512 \times N_r \times 1)$
(6)	(3)	↓	Unsqueeze	$(1 \times N_e \times N_r)$
(7)	(4),(6)	↓	Matrix Multiplication	$(512 \times 1 \times N_r)$
(8)	(5),(7)	↓	Matrix Multiplication	$(512 \times 1 \times 1)$
(9)	(8)	↓	Squeeze + FC	(512)
(10)	(4),(9)	end	Unsqueeze + Element-wise Addition	$(512 \times 1 \times N_e)$
(11)	(10)		Sum Over N_e	512
(12)	(11)		FC + ReLU + FC + ReLU	1024

Table 3. The details of Bilinear Attention Scene Graph Encoding Module.

the whole SG. The same BAN is used for both text-SG or image-SG, *i.e.*, the parameters of the BAN are shared.

The model was trained by the triplet loss [15] with L1 distance. The model was trained in 30 epochs by SGD optimizer and set batch size to be 12. Learning rate was set to be 12×10^{-2}, which was decayed at 10^{th} and 25^{th} epochs by the factor of 10.

C. Quantitative Studies

The full results of Relationship Retrieval, including both conventional Recall@K and the adopted mean Recall@K [18, 1], are given in Table 4. Although a performance drop on conventional Recall@k is observed on TDE, the detailed analysis of the "decreased" predicates in Figure 6 of the original paper implies that it's caused by a more fine-grained predicate classification.

The detailed predicate-level Recall@100 on PredCls of all three models, two fusion functions and baseline *vs.* TDE are given in Figure 5 6 7. Impressively, the distribution of the improved performances is no longer long-tailed while those conventional debiasing methods illustrated in Figure 4 can't surpass the dataset distribution anyway. For TDE, very few decreased predicates are mainly due to the more fine-grained classification and we can observe significant improvements on their subclass predicates. Note that, unlike Reweight, which blindly hurt all frequent predicates, the proposed TDE will even improve some of the top-10 frequent predicates, like `behind` and `above`, which themselves are the subclasses of `near`. It further proves that the improvement of the proposed TDE doesn't come from hacking the distribution.

D. Qualitative Studies

More Relationship Retrieval (RR) and Zero-Shot Relationship Retrieval (ZSRR) results are given in Figure 8, where top 10 relationships under SGCls are selected for each image. As we can see, other than the trivial relationship problem, conventional baseline barely distinguishes different entities. For example, in the left bottom image, the same `sign` is almost `on` every `pole` in the baseline

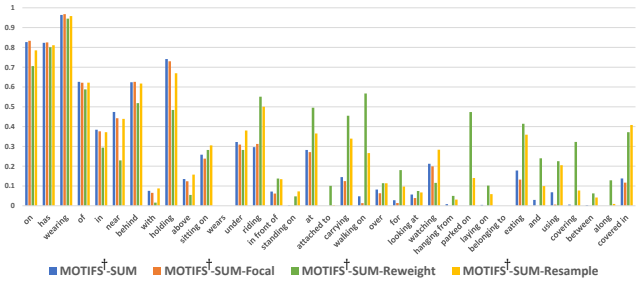


Figure 4. Conventional Debiasing Methods: Recall@100 on Predicate Classification for the most frequent 35 predicates.

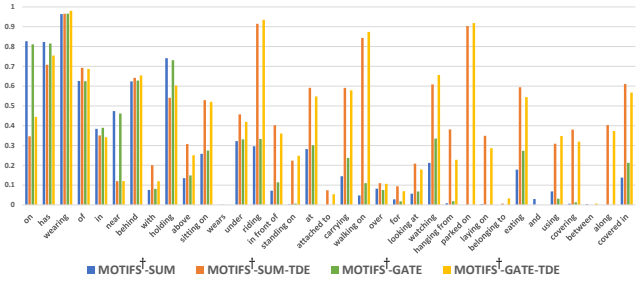


Figure 5. MOTIFS† [23]: Recall@100 on Predicate Classification for the most frequent 35 predicates.

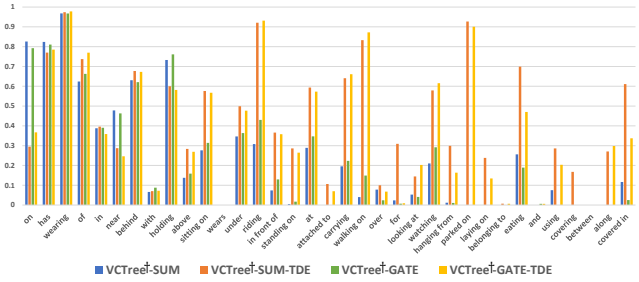


Figure 6. VCTree† [18]: Recall@100 on Predicate Classification for the most frequent 35 predicates.

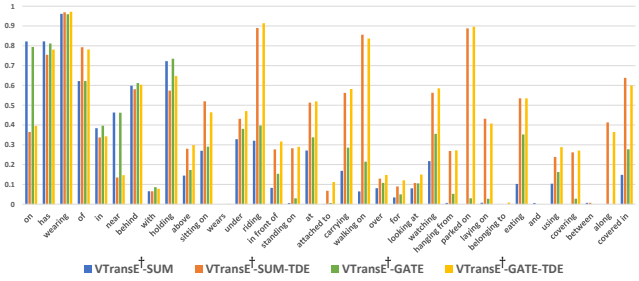


Figure 7. VTransE† [24]: Recall@100 on Predicate Classification for the most frequent 35 predicates.

while the TDE results are more sensitive to different entities. However, one of the problem of TDE is that it over emphasizes the action predicates. It even uses `holding` for `pole` and `sign` while the predicate `on` used by the

			Predicate Classification		Scene Graph Classification		Scene Graph Detection	
Model	Fusion	Method	R@20 / 50 / 100	mR@20 / 50 / 100	R@20 / 50 / 100	mR@20 / 50 / 100	R@20 / 50 / 100	mR@20 / 50 / 100
IMP+ [22, 1]	-	-	52.7 / 59.3 / 61.3	- / 9.8 / 10.5	31.7 / 34.6 / 35.4	- / 5.8 / 6.0	14.6 / 20.7 / 24.5	- / 3.8 / 4.8
FREQ [23, 18]	-	-	53.6 / 60.6 / 62.2	8.3 / 13.0 / 16.0	29.3 / 32.3 / 32.9	5.1 / 7.2 / 8.5	20.1 / 26.2 / 30.1	4.5 / 6.1 / 7.1
MOTIFS [23, 18]	-	-	58.5 / 65.2 / 67.1	10.8 / 14.0 / 15.3	32.9 / 35.8 / 36.5	6.3 / 7.7 / 8.2	21.4 / 27.2 / 30.3	4.2 / 5.7 / 6.6
KERN [1]	-	-	- / 65.8 / 67.6	- / 17.7 / 19.2	- / 36.7 / 37.4	- / 9.4 / 10.0	- / 27.1 / 29.8	- / 6.4 / 7.3
VCTree [18]	-	-	60.1 / 66.4 / 68.1	14.0 / 17.9 / 19.4	35.2 / 38.1 / 38.8	8.2 / 10.1 / 10.8	22.0 / 27.9 / 31.3	5.2 / 6.9 / 8.0
MOTIFS†	SUM	Baseline	59.5 / 66.0 / 67.9	11.5 / 14.6 / 15.8	35.8 / 39.1 / 39.9	6.5 / 8.0 / 8.5	25.1 / 32.1 / 36.9	4.1 / 5.5 / 6.8
		Focal	59.2 / 65.8 / 67.7	10.9 / 13.9 / 15.0	36.0 / 39.3 / 40.1	6.3 / 7.7 / 8.3	24.7 / 31.7 / 36.7	3.9 / 5.3 / 6.6
		Reweight	45.4 / 57.0 / 61.7	16.0 / 20.0 / 21.9	24.2 / 29.5 / 31.5	8.4 / 10.1 / 10.9	18.3 / 24.4 / 29.3	6.5 / 8.4 / 9.8
		Resample	57.6 / 64.6 / 66.7	14.7 / 18.5 / 20.0	34.5 / 37.9 / 38.8	9.1 / 11.0 / 11.8	23.2 / 30.5 / 35.4	5.9 / 8.2 / 9.7
		X2Y	58.3 / 65.0 / 66.9	13.0 / 16.4 / 17.6	35.2 / 38.6 / 39.5	6.9 / 8.6 / 9.2	24.8 / 32.1 / 36.7	5.1 / 6.9 / 8.1
		X2Y-Tr	59.0 / 65.3 / 66.9	11.6 / 14.9 / 16.0	35.5 / 38.9 / 39.7	6.5 / 8.4 / 9.1	25.5 / 32.8 / 37.2	5.0 / 6.9 / 8.1
		TE	34.3 / 46.7 / 51.7	18.2 / 25.3 / 29.0	25.5 / 32.5 / 35.4	8.1 / 12.0 / 14.0	14.8 / 20.1 / 23.9	5.7 / 8.0 / 9.6
		NIE	0.6 / 1.0 / 1.3	0.6 / 1.1 / 1.4	28.6 / 35.0 / 37.4	6.1 / 9.0 / 10.6	17.3 / 22.7 / 26.8	3.8 / 5.1 / 6.0
		TDE	33.6 / 46.2 / 51.4	18.5 / 25.5 / 29.1	21.7 / 27.7 / 29.9	9.8 / 13.1 / 14.9	12.4 / 16.9 / 20.3	5.8 / 8.2 / 9.8
	GATE	Baseline	58.9 / 65.5 / 67.4	12.2 / 15.5 / 16.8	36.2 / 39.4 / 40.1	7.2 / 9.0 / 9.5	25.8 / 33.3 / 37.8	5.2 / 7.2 / 8.5
		TDE	38.7 / 50.8 / 55.8	18.5 / 24.9 / 28.3	21.8 / 27.2 / 29.5	11.1 / 13.9 / 15.2	5.9 / 7.4 / 8.4	6.6 / 8.5 / 9.9
VTransE†	SUM	Baseline	59.0 / 65.7 / 67.6	11.6 / 14.7 / 15.8	35.4 / 38.6 / 39.4	6.7 / 8.2 / 8.7	23.0 / 29.7 / 34.3	3.7 / 5.0 / 6.0
		TDE	36.9 / 48.5 / 53.1	17.3 / 24.6 / 28.0	19.7 / 25.7 / 28.5	9.3 / 12.9 / 14.8	13.5 / 18.7 / 22.6	6.3 / 8.6 / 10.5
	GATE	Baseline	58.7 / 65.3 / 67.1	13.6 / 17.1 / 18.6	34.6 / 38.1 / 38.9	6.6 / 8.2 / 8.7	24.5 / 31.3 / 35.5	5.1 / 6.8 / 8.0
		TDE	40.0 / 50.7 / 54.9	18.9 / 25.3 / 28.4	23.0 / 28.8 / 31.1	9.8 / 13.1 / 14.7	13.7 / 19.0 / 22.9	6.0 / 8.5 / 10.2
VCTree†	SUM	Baseline	59.8 / 66.2 / 68.1	11.7 / 14.9 / 16.1	37.0 / 40.5 / 41.4	6.2 / 7.5 / 7.9	24.7 / 31.5 / 36.2	4.2 / 5.7 / 6.9
		TDE	36.2 / 47.2 / 51.6	18.4 / 25.4 / 28.7	19.9 / 25.4 / 27.9	8.9 / 12.2 / 14.0	14.0 / 19.4 / 23.2	6.9 / 9.3 / 11.1
	GATE	Baseline	59.1 / 65.5 / 67.4	12.4 / 15.4 / 16.6	35.4 / 38.9 / 39.8	6.3 / 7.5 / 8.0	24.8 / 31.8 / 36.1	4.9 / 6.6 / 7.7
		TDE	39.1 / 49.9 / 54.5	17.2 / 23.3 / 26.6	22.8 / 28.8 / 31.2	8.9 / 11.8 / 13.4	14.3 / 19.6 / 23.3	6.3 / 8.6 / 10.3

Table 4. The SGG performances of Relationship Retrieval on both conventional **Recall@K** and **mean Recall@K** [18, 1]. The SGG models reimplemented under our codebase are denoted by the superscript †.

baseline is more natural in this case.

Another example of Sentence-to-Graph Retrieval (S2GR) is illustrated in Figure 9. Although we only reported sub-graphs of the original SGDet results, due to the limited space, we can still find that the conventional baseline model is not able to detect predicate like `eating`, which causes the detected SGs only provide the spatial relationships, missing the most discriminative word `eating` in the query caption.

References

[1] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, 2019.

[2] Graham Dunn, Richard Emsley, Hanhua Liu, Sabine Landau, Jonathan Green, Ian White, and Andrew Pickles. Evaluation and validation of social and psychological markers in randomised trials of complex interventions in mental health: a methodological research programme. *NIHR Journals Library*, 2015.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[4] Luke Keele. The statistics of causal inference: A view from political methodology. *Political Analysis*, 2015.

[5] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, 2018.

[6] Brayden G King. A political mediation model of corporate response to social movement activism. *Administrative Science Quarterly*, 2008.

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[8] David P MacKinnon, Amanda J Fairchild, and Matthew S Fritz. Mediation analysis. *Annu. Rev. Psychol.*, 2007.

[9] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch, 2018.

[10] Judea Pearl. Direct and indirect effects. In *Proceedings of the 17th conference on uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001.

[11] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[12] Judea Pearl and Dana Mackenzie. *THE BOOK OF WHY: THE NEW SCIENCE OF CAUSE AND EFFECT*. Basic Books, 2018.

[13] Lorenzo Richiardi, Rino Bellocco, and Daniela Zugna. Mediation analysis in epidemiology: methods, interpretation and bias. *International journal of epidemiology*, 2013.

[14] James M Robins and Sander Greenland. Identifiability and exchangeability for direct and indirect effects. *Epidemiology*, 1992.

[15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[16] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, 2015.

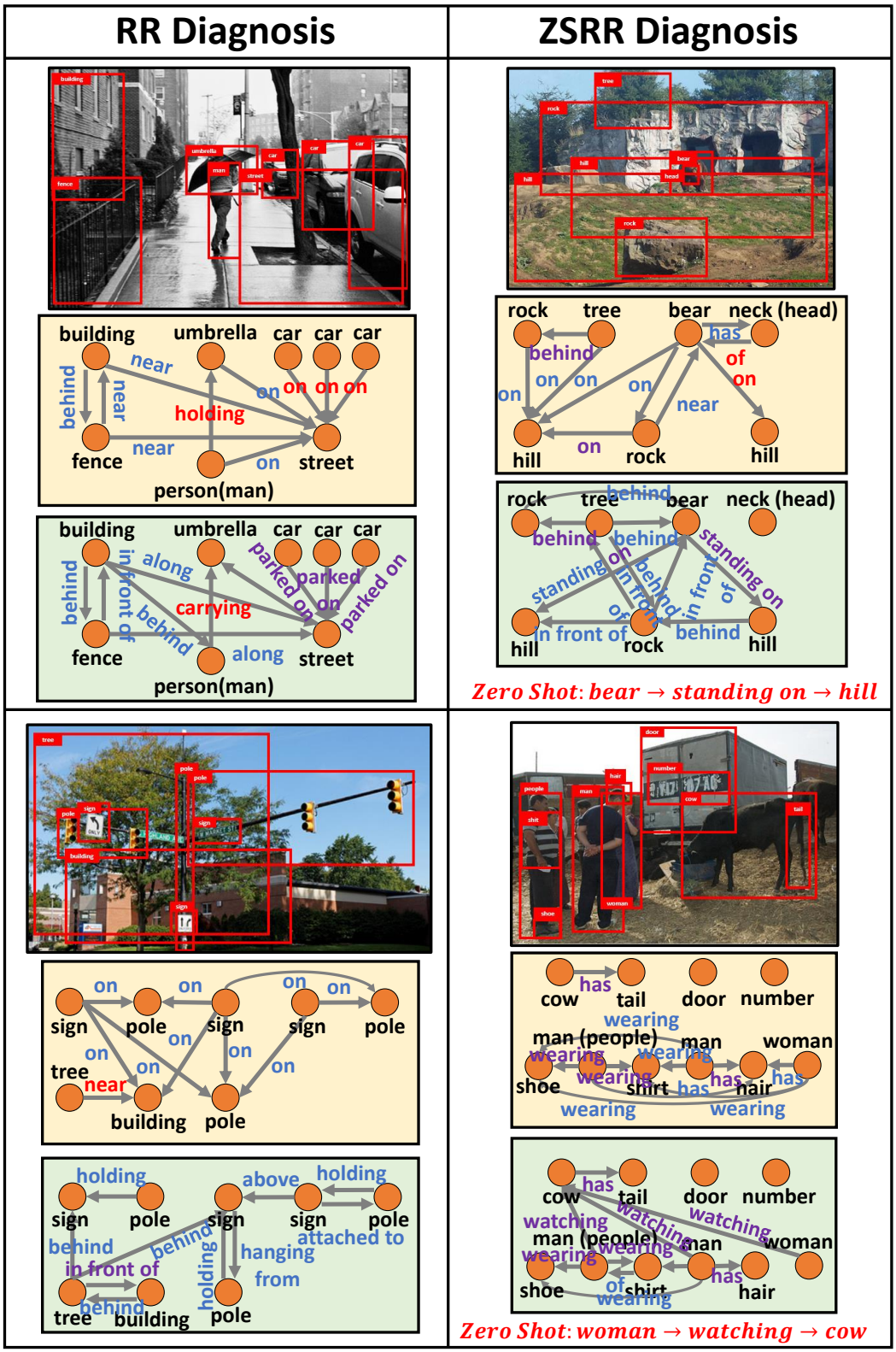


Figure 8. Top 10 Relationship Retrieval (RR) and Zero-Shot Relationship Retrieval (ZSRR) results of SGCls for MOTIFS†+SUM baseline (yellow box) and corresponding TDE (green box). The red predicates indicate misclassified relationships, the purple predicates are those correctly classified relationships (in ground truth), the blue predicates are those not labeled in ground truth.

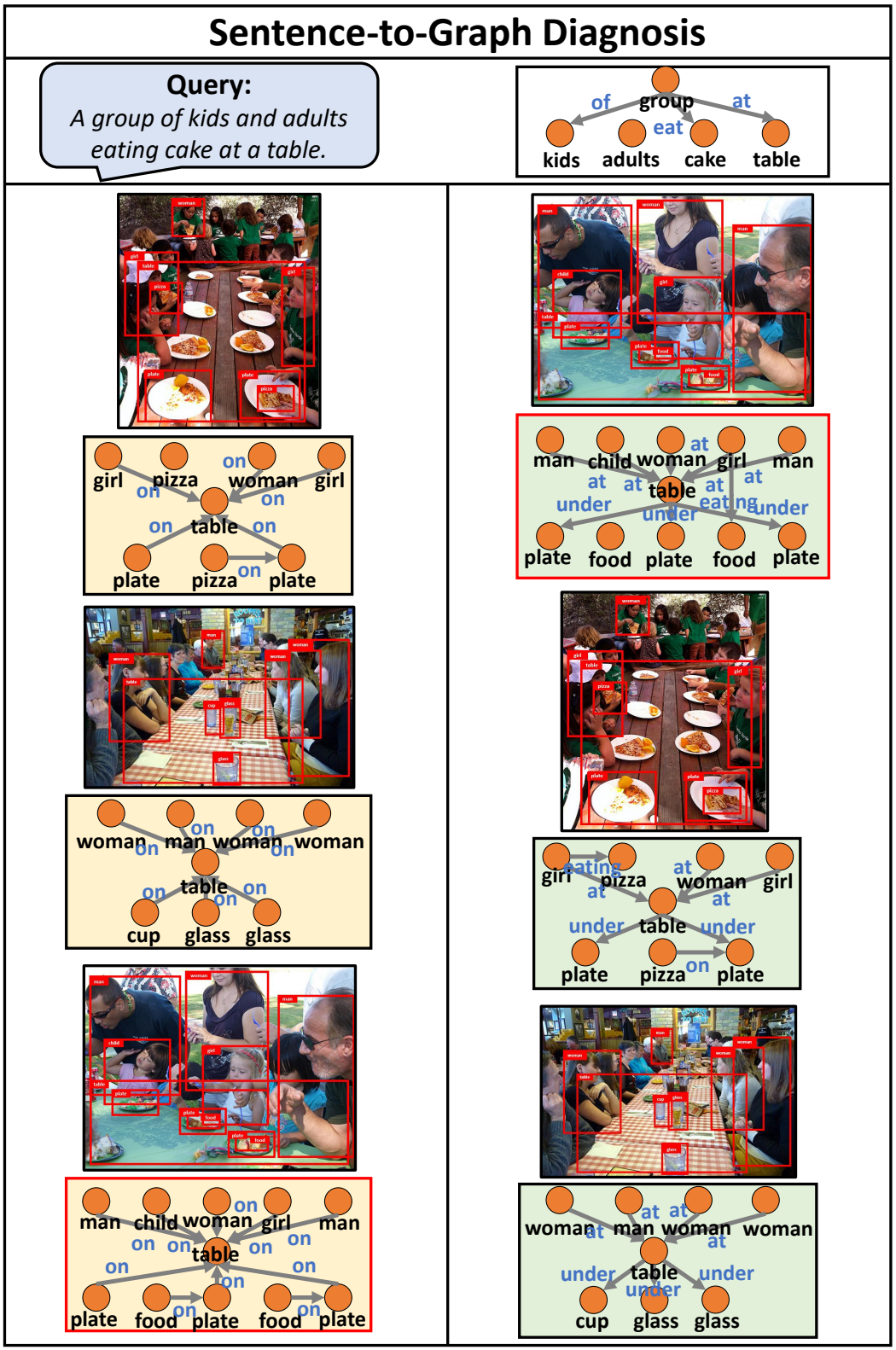


Figure 9. An example of Sentence-to-Graph Retrieval (S2GR) results for MOTIFS†+SUM baseline (yellow box) and corresponding TDE (green box). The red boxes indicate ground truth matching results. Note that we only draw sub-graphs containing important objects and predicates, because the original detected scene graphs from SGDet have too many trivial objects and predicates.

[17] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[18] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019.

[19] Tyler VanderWeele. *Explanation in causal inference: methods for mediation and interaction*. Oxford University Press, 2015.

[20] Tyler J VanderWeele. A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology (Cambridge, Mass.)*, 2013.

[21] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[22] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017.

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

[24] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017.