Unbiased Scene Graph Generation from Biased Training

Kaihua Tang
Nanyang Technological University
kaihua001@e.ntu.edu.sg

Yulei Niu
Renmin University of China
niu@ruc.edu.cn

Jianqiang Huang
Damo Academy, Alibaba Group
jianqiang.jqh@gmail.com

Jiaxin Shi
Tsinghua University
shijx12@163.com

Hanwang Zhang
Nanyang Technological University
hanwangzhang@ntu.edu.sg

Abstract

Today's scene graph generation (SGG) task is still far from practical, mainly due to the severe training bias, e.g., collapsing diverse `human walk on/ sit on/lay on beach` *into* `human on beach`*. Given such SGG, the down-stream tasks such as VQA can hardly infer better scene structures than merely a bag of objects. However, debiasing in SGG is not trivial because traditional debiasing methods cannot distinguish between the good and bad bias, e.g., good context prior (e.g.,* `person read book` *rather than* `eat`*) and bad long-tailed bias (e.g.,* `behind/in front of` *collapsed to* `near`*). In this paper, we present a novel SGG framework based on* ***causal inference*** *but not the conventional likelihood. We first build a causal graph for SGG, and perform traditional biased training with the graph. Then, we propose to draw the* ***counterfactual causality*** *from the trained graph to infer the effect from the bad bias, which should be removed. In particular, we use* ***Total Direct Effect*** *as the proposed final predicate score for unbiased SGG. Note that our framework is agnostic to any SGG model and thus can be widely applied in the community who seeks unbiased predictions. By using the proposed* ***Scene Graph Diagnosis*** *toolkit*[1] *on the SGG benchmark Visual Genome and several prevailing models, we observed significant improvements over the previous state-of-the-art methods.*

1. Introduction

Scene graph generation (SGG) [31, 64] — a visual detection task of objects and their relationships in an image — seems to have never fulfilled its promise: a comprehensive visual scene representation that supports *graph reasoning* for high-level tasks such as visual captioning [67, 66] and VQA [56, 14]. Once equipped with SGG, these high-level tasks have to abandon the ambiguous visual relationships — yet on which are our core efforts made [69, 55, 6], then pretend that there is a graph — nothing but a sparse object layout with binary links, and finally shroud it into graph neural networks [65, 64] for merely more contextual object representations [69, 25, 55]. Although this is partly due to the research gap in graph reasoning [2, 51, 15], the crux lies in the *biased* relationship prediction.

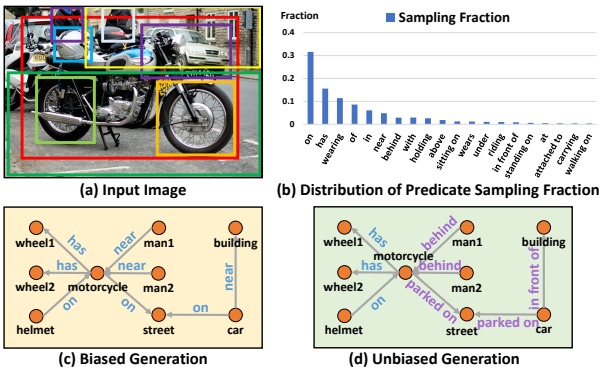


Figure 1. An example of scene graph generation (SGG). (a) An input image with bounding boxes. (b) The distribution of sample fraction for the most frequent 20 predicates in Visual Genome [22]. (c) SGG from MOTIFS [69]. (d) SGG by the proposed unbiased prediction from the same model.

Figure 1 visualizes the SGG results from a state-of-the-art model [69]. We can see a frustrating scene: among almost perfectly detected objects, most of their visual relationships are trivial and less informative. For example in Figure 1(c), except the trivial 2D spatial layouts, we know little about the image from `near`, `on`, and `has`. Such heavily biased generation comes from the *biased training* data, more specifically, as shown in Figure 1(b), the highly-skewed long-tailed relationship annotations. For example, if a model is trained for predicting `on` 1000 times more than `standing on`, during test, it is likely to predict

[1]Code is publicly available on GitHub: https://github.com/KaihuaTang/Scene-Graph-Benchmark.pytorch

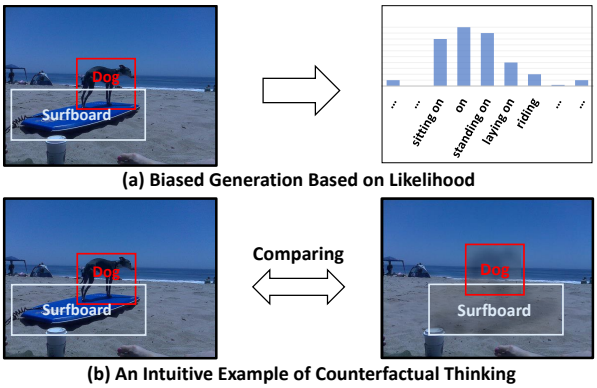


Figure 2. (a) The biased generation that directly predicts labels from likelihood. (b) An intuitive example of the proposed total direct effect, which calculates the difference between the real case and the counterfactual one. Note that the "wipe-out" is only for the illustration purpose but not considered as visual processing.

the former even if it has also considered the latter as correct. Therefore, to perform a sensible graph reasoning, we need to distinguish more fine-grained relationships in Figure 1(d), such as using `behind/in front of` for `near`, and `parking on/driving on` for `on`.

However, we should not blame the biased training because both our visual world *per se* and the way we describe it are biased: there are indeed more `person carry bag` than `dog carry bag` (*i.e.*, the long-tail theory); it is easier for us to label `person beside table` rather than `eating on` (*i.e.*, bounded rationality [52]); and we prefer to say `person on bike` rather than `person ride on bike` (*i.e.*, language or reporting bias [35]). In fact, most of the biased annotations can help the model learn good contextual prior [31, 69] to filter out the unnecessary search space such as `apple park on table` or `apple wear hat`. A promising but embarrassing finding [69] is that: by only using the statistical prior of detected object class in the Visual Genome benchmark [22], we can already achieved 30.1% on Recall@100 for Scene Graph Detection — rendering all the much more complex SGG models almost useless — that is only 1.1-1.5% lower than the state-of-the-art [5, 55, 72]. Not surprisingly, as we will show in Section 5, conventional debiasing methods who do not respect the "good bias" during training, *e.g.*, re-sampling [11] and re-weighting [29], fail to generalize to unseen relationships, *i.e.*, zero-shot SGG [31].

For both machines and humans, decision making is a collaboration of *content* (endogenous reasons) and *context* (exogenous reasons) [58]. Take SGG as an example, in most SGG models [69, 5, 72], the content is the visual features of the subject and object, and the context is the visual features of the subject-object union regions and the pairwise object classes. We humans — born and raised in the biased nature — are ambidextrous in embracing the good while avoiding the bad context, and making unbiased decisions together with the content. The underlying mechanism is *causality-based*: the decision is made by pursuing the main causal effect caused by the content but not the side-effect by context. However, on the other hand, machines are usually *likelihood-based*: the prediction is analogous to look-up where the content and its context are in a huge likelihood table, interpolated by population training. We believe that the key is to teach machines how to distinguish between the "main effect" and "side-effect".

In this paper, we propose to empower machines the ability of *counterfactual causality* [41] to pursue the "main effect" in unbiased prediction:

> *If I **had not** seen the content, would I still make the same prediction?*

The counterfactual lies between the fact that "I see" and the imagination "I had not", and the comparison between the factual and counterfactual will naturally *remove* the effect from the context bias, because the context is the only thing unchanged between the two alternates.

To better illustrate the profound yet subtle difference between likelihood and counterfactual causality, we present a `dog standing on surfboard` example in Figure 2(a). Due to the biased training, the model will eventually predict the winner `on`. Note that even though the rest choices are not all correct, thanks to the bias, they still help to filter out a large amount of unreasonable ones. To take a closer look at what relationship it is in the context, we are essentially comparing the original scene with a counterfactual scene (Figure 2(b)): only the visual features of the `dog` and `surfboard` are wiped out, while keeping the rest — the scene and the object classes — untouched, as if they had existed. By doing this, we can focus on the main visual effects of the relationship without losing the context.

We propose a novel unbiased SGG method based on the Total Direct Effect (TDE) analysis framework in causal inference [59, 39, 60]. Figure 3(a) shows the underlying causal graphs [40, 41] of the two alternate scenes: factual and counterfactual. Although a formal introduction of them is given in Section 3-4, now you can simply understand the nodes as data features and the directed links as data flows. For example, $X \rightarrow Y$, $Z \rightarrow Y$, and $I \rightarrow Y$ indicate that the relationship Y is a combined effect caused by the content: the pair of object visual features X, and the context: their classes Z, and image scene I; the faded links denote that the wiped-out $\bar{X}$ is no longer caused by I or affects Z. These graphs offer an algorithmic formulation to calculate the TDE, which exactly realizes the counterfactual thinking in Figure 2. As shown in Figure 3(b), the proposed TDE significantly improves most of the predicates, and impressively, the distribution of the improved performances is no longer long-tailed, indicating the fact that our improvement is indeed from the proposed method, but NOT from the better exploitation of the context bias. A closer anal-

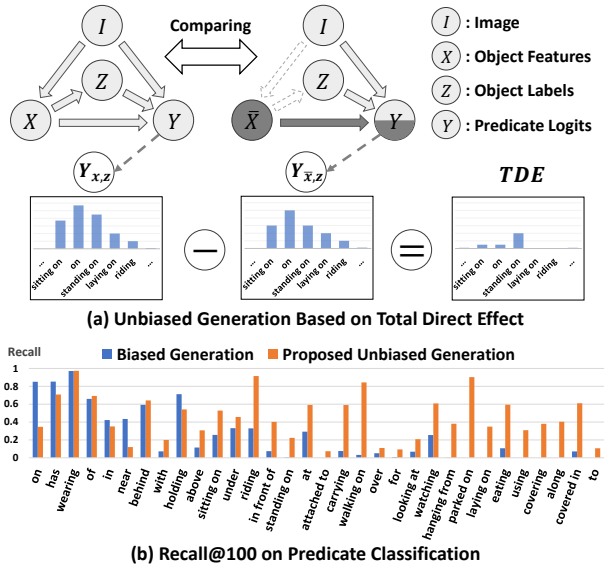


Figure 3. (a) The example of total direct effect calculation and corresponding operations on the causal graph, where $\bar{X}$ represents wiped-out X. (b) Recall@100 of Predicate Classification for selected predicates ranking by sampling fraction. The biased generation refers to MOTIFS [69, 55] and the proposed unbiased generation is the result from the same model using TDE.

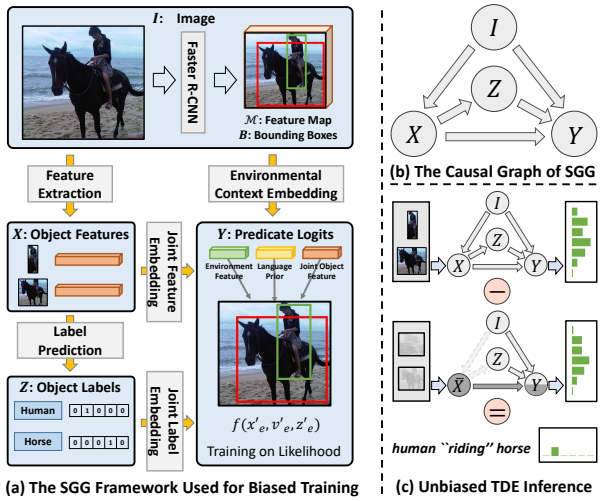


Figure 4. (a) The framework used in our biased training. (b) The causal graph of the SGG framework. (c) An illustration of the proposed TDE inference.

ysis in Figure 6 further shows that the worse predictions like `on` — though very few — are due to the more fine-grained classification such as `stand on` and `park on`. We highlight that TDE is a model-agnostic prediction strategy and thus applicable for a variety of models and fusion tricks [71, 69, 55].

Last but not least, we propose a new standard of SGG diagnosis toolkit (cf. Section 5.2) for more comprehensive SGG evaluations. Besides traditional evaluation tasks, it consists of the bias-sensitive metric: mean Recall [55, 6] and a new Sentence-to-Graph Retrieval for a more comprehensive graph-level metric. By using this toolkit on SGG benchmark Visual Genome [22] and several prevailing baselines, we verify the severe bias in existing models and demonstrate the effectiveness of the proposed unbiased prediction over other debiasing strategies.

2. Related Work

Scene Graph Generation. SGG [64, 69] has received increasing attention in computer vision community, due to the potential revolution that would be brought to down-stream visual reasoning tasks [51, 66, 21, 16]. Most of the existing methods [64, 62, 7, 25, 68, 55, 65, 10, 43, 61] struggle for better feature extraction networks. Zellers *et al*. [69] firstly brought the bias problem of SGG into attention and the followers [55, 6] proposed the unbiased metric (mean Recall), yet, their approaches are still restricted to the feature extraction networks, leaving the biased SGG problem unsolved. The most related work [27] just prunes those dominant and easy-to-predict relationships in the training set.

Unbiased Training. The bias problem has long been investigated in machine learning [57]. Existing debiasing methods can be roughly categorized into three types: 1) data augmentation or re-sampling [9, 24, 26, 11, 3], 2) unbiased learning through elaborately designed training curriculums or learning losses [70, 29], 3) disentangling biased representations from the unbiased [35, 4]. The proposed TDE analysis can be regarded as the third category, but the main difference is that TDE doesn't require to train additional layers like [35, 4] to model the bias, it directly separates the bias from existing models through counterfactual surgeries on causal graphs.

Mediation Analysis. It is also known as effect analysis [59, 41], which is widely adopted in medical, political or psychological research [45, 18, 8, 32, 20] as the tool of studying the effect of certain treatments or policies. However, it has been neglected in the community of computer vision for years. There are very few recent works [36, 23, 37, 42, 54] trying to endow the model with the capability of causal reasoning. More detailed background knowledge can be found in [40, 41, 59].

3. Biased Training Models in Causal Graph

As illustrated in Figure 4, we summarize the SGG framework in the form of *Causal Graph* (*a.k.a.*, structural causal model) [41, 38, 40]. It is a directed acyclic graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, indicating how a set of variables $\mathcal{N}$ interact with each other through the causal links $\mathcal{E}$. It provides a sketch of the causal relations behind the data and how variables obtain their values, *e.g.*, $(I, X, Z) \rightarrow Y$. Before we conduct counterfactual analysis that deliberately manipulates the values of nodes and prunes the causal graph, we first

revisit the conventional biased SGG model training in the graphical view.

The causal graph in Figure 4(b) is applicable to a variety of SGG methods, since it is highly general, imposing no constraints on the detailed implementations. We case-study three representative model formulations: the classic VTransE [71], the state-of-the-art MOTIFS [69] and VCTree [55], using the language of nodes and links.

Node I (Input Image&Backbone). A Faster R-CNN [44] is pre-trained and frozen in this node, It outputs a set of bounding boxes $B = \{b_i | i = 1...n\}$ and the feature map $\mathcal{M}$ from image I.

Link $I \rightarrow X$ (Object Feature Extractor). It firstly extracts RoIAlign features [12] $R = \{r_i\}$ and tentative object labels $L = \{l_i\}$ by the object classifier on Faster R-CNN. Then, like MOTIFS [69] or VCTree [55], we can use the following module to encode visual contexts for each object:

$$\textit{Input} : \{(r_i, b_i, l_i)\} \Longrightarrow \textit{Output} : \{x_i\}, \tag{1}$$

where MOTIFS implements it as bidirectional LSTMs (Bi-LSTMs) and VCTree [55] adopts bidirectional TreeLSTMs (Bi-TreeLSTMs) [53], early works like VTransE [71] simply use fully connected layers.

Node X (Object Feature). The pairwise object feature X takes value from $\{(x_i, x_j) | i \neq j; i, j = 1...n\}$. We slightly abuse the notation hereinafter, denoting the combination of representations from i and j as subscript e: $x_e = (x_i, x_j)$.

Link $X \rightarrow Z$ (Object Classification). The fine-tuned label of each object is decoded from the corresponding x_i by:

$$\textit{Input} : \{x_i\} \Longrightarrow \textit{Output} : \{z_i\}, \tag{2}$$

where MOTIFS [69] and VCTree [55] utilizes LSTM and TreeLSTM as decoders to capture the co-occurrence among object labels, respectively. The input of each LSTM/ TreeLSTM cell is the concatenation of feature and the previous label $[x_i; z_{i-1}]$. VTransE [71] uses the conventional fully connected layer as the classifier.

Node Z (Object Class). It contains a pair of one-hot vectors for object labels $z_e = (z_i, z_j)$.

Link $X \rightarrow Y$ (Object Feature Input for SGG). For relationship classification, pairwise feature X are merged into a joint representation by the module:

$$\textit{Input} : \{x_e\} \Longrightarrow \textit{Output} : \{x'_e\}, \tag{3}$$

where another Bi-LSTMs and Bi-TreeLSTMs layers are applied in MOTIFS [69] and VCTree [55], respectively, before concatenating the pair of object features. VTransE [71] uses fully connected layers and element-wise subtraction for feature merging.

Link $Z \rightarrow Y$ (Object Class Input for SGG). The language prior is calculated in this link through a joint embedding layer $z'_e = W_z[z_i \otimes z_j]$, where $\otimes$ generates the one-hot unique vector $\mathbb{R}^{N \times N}$ for the pair of N-way object labels.

Link $I \rightarrow Y$ (Visual Context Input for SGG). This link extracts the contextual union region features $v'_e = \text{Convs}(\text{RoIAlign}(\mathcal{M}, b_i \cup b_j))$ where $b_i \cup b_j$ indicates the union box of two RoIs.

Node Y (Predicate Classification). The final predicate logits Y that takes inputs from the three branches is then generated by a fusion function. In Section 5, we test two general fusion functions: 1) SUM: $y_e = W_x x'_e + W_v v'_e + z'_e$, 2) GATE: $y_e = W_r x'_e \cdot \sigma(W_x x'_e + W_v v'_e + z'_e)$, where $\cdot$ is element-wise product, $\sigma(\cdot)$ is the sigmoid function.

Training Loss. All models are trained by using the conventional cross-entropy losses of object labels and predicate labels. To avoid any single link dominating the generation of logits y_e, especially $Z \rightarrow Y$, we further add auxiliary cross-entropy losses that individually predict y_e from each branch.

4. Unbiased Prediction by Causal Effects

Once the above training has been done, the causal dependencies among the variables are learned, in terms of the model parameters. The conventional biased prediction can only see the output of the entire graph given an image $I = u$ without any idea about how a specific pair of objects affect their predicate. However, causal inference [41] encourages us to think out of the box. From the graphical point of view, we are no longer required to run the entire graph as a whole. We can directly manipulate the values of several nodes and see what would be going on. For example, we can cut off the link $I \rightarrow X$ and assign a dummy value to X, then guess what the predicate would be. The above operation is termed intervention in causal inference [40]. Next, we will make unbiased predictions by intervention and counterfactuals.

4.1. Notations

Intervention. Its formulation can be denoted as $\boldsymbol{do(\cdot)}$. It wipes out all the in-coming links of a variable and directly demands the variable to take certain value, *e.g.* $do(X = \bar{x})$ in Figure 5(b).

Counterfactual. It means *"counter to the facts"* [47], and takes one step further that assigns the "clash of worlds" combination of values to variables. Take Figure 5(c) as an example, if the intervention $do(X = \bar{x})$ is conducted on X, the variable Z still takes original z as if x had existed.

Causal Effect. Throughout this section, we will use the pairwise object feature X as our control variable, *i.e.*, on which the intervention is conducted, aiming to assess its effects, due to the fact that there won't be any valid relationship if the pair of objects do not exist. The observed X is denoted as x while the intervened unseen value is $\bar{x}$, which is set to either the mean feature of the training set or zero vector. The object label z on Figure 5(c) is calculated from Eq. (2), taking x as input. We denote the output logits Y

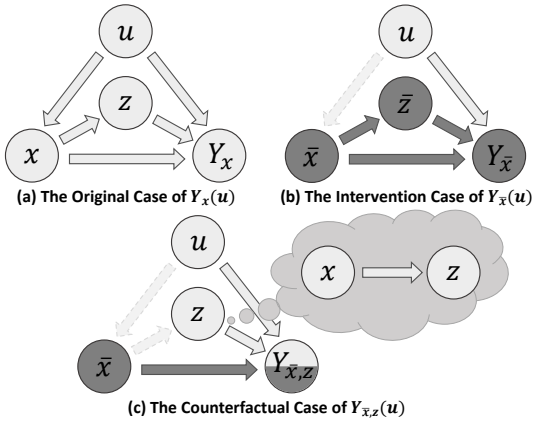


Figure 5. The original causal graph of SGG together with two interventional and counterfactual alternates.

after the intervention $X = \bar{x}$ as follows:

$$Y_{\bar{x}}(u) = Y(do(X = \bar{x})|u), \tag{4}$$

where u is the input image in SGG (or situation). Following the above notation, the original and counterfactual Y, as shown in Figure 5, can be re-written as $Y_x(u)$ and $Y_{\bar{x},z}(u)$.

4.2. Total Direct Effect

As we discussed in Section 1, instead of the static likelihood that tends to be biased, the unbiased prediction lies in the difference between the observed outcome $Y_x(u)$ and its counterfactual alternate $Y_{\bar{x},z}(u)$. The later one is a context-specific bias that we want to remove from prediction. Intuitively, the unbiased prediction that we seek is the visual stimuli from blank to the observed real objects with specific attributes, states, and behaviors, but not merely from the surroundings and language priors. Those specific visual cues of objects are the key to the more fine-grained and informative unbiased predictions, because even if the overall prediction is biased towards the relationship like `dog on surfboard`, the "straight legs" would cause more effect on `standing on` rather than `sitting on`. In causal inference [59, 60], the above prediction process can be calculated as Total Direct Effect (TDE):

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

where the former corresponds to the original graph and the later is the counterfactual case, as illustrated in Figure 5.

Note that there is another type of effect [59], Total Effect (TE), which is easy to be mixed up with TDE. Instead of deriving counterfactual bias $Y_{\bar{x},z}(u)$, TE lets all the descendant nodes of X change with intervention $do(X = \bar{x})$ as shown in Figure 5(b). TE is therefore formulated as:

$$TE = Y_x(u) - Y_{\bar{x}}(u). \tag{6}$$

The main difference lies in the fact that $Y_{\bar{x}}(u)$ is not conditioned on real object labels, so TE removes the general bias between any two types of objects, rather than the two types we care. The marginal difference between TE and TDE is further defined as Natural Indirect Effect (NIE) [59] or Pure Indirect Effect (PIE) [60]. More experimental analyses among these three types of effect are given in Section 5.

Overall SGG. At last, the proposed unbiased prediction $y_e^\dagger$ is obtained by replacing the conventional one-time prediction with the TDE, which literally thinks twice: one for observational $Y_{x_e}(u) = y_e$, the other for imaginary $Y_{\bar{x},z_e}(u) = y_e(\bar{x}, z_e)$. The unbiased logits of Y is therefore defined as follows:

$$y_e^\dagger = y_e - y_e(\bar{x}, z_e). \tag{7}$$

It is also worth mentioning that the proposed TDE doesn't introduce any additional parameters and is widely applicable to a variety of models.

5. Experiments

5.1. Settings and Models

Dataset. For SGG, we used Visual Genome (VG) [22] dataset to train and evaluate our models, which is composed of 108k images with 75k object categories and 37k predicate categories. However, 92% of the predicates have no more than 10 instances, so we adopted the widely accepted VG split [64, 69, 55, 5] containing the most frequent 150 object categories and 50 predicate categories. The original split only has training set (70%) and test set (30%). We followed [69] to sample a 5k validation set from training set for parameter tuning. In Sentence-to-Graph Retrieval, we selected the overlapped 41,859 images between VG and MS-COCO Caption dataset [30] and divided them into train/test-1k/test-5k (35,859/1,000/5,000) sets. The later two only contain images from VG test set in case of exposing to ground truth SGs. Each image has at least 5 captions serving as human queries, the same as how we use searching engines.

Model Zoo. Three models (VTransE [71], MOTIFS [69], VTree [55]) and two fusion functions (SUM, GATE), re-implemented by the same codebase as we proposed, were evaluated. All models shared the same hyper-parameters and the pre-trained detector with different feature extraction layers and object prediction layers. In GATE fusion, the additional cross-entropy constraint is only applied to $X \rightarrow Y$ branch.

5.2. Scene Graph Generation Diagnosis

The evaluation standards we used for SGG diagnosis are composed of three folds: 1) Relationship Retrieval, 2) Zero-Shot Relationship Retrieval, and 3) Sentence-to-Graph Retrieval. For the first two, three standard protocols are adopted: (1) Predicate Classification **(PredCls)**: taking ground truth bounding boxes and labels as inputs, (2) Scene

Model	Fusion	Method	Predicate Classification			Scene Graph Classification			Scene Graph Detection		
			mR@20	mR@50	mR100	mR@20	mR50	mR100	mR@20	mR50	mR100
IMP+ [64, 6]	-	-	-	9.8	10.5	-	5.8	6.0	-	3.8	4.8
FREQ [69, 55]	-	-	8.3	13.0	16.0	5.1	7.2	8.5	4.5	6.1	7.1
MOTIFS [69, 55]	-	-	10.8	14.0	15.3	6.3	7.7	8.2	4.2	5.7	6.6
KERN [6]	-	-	-	17.7	19.2	-	9.4	10.0	-	6.4	7.3
VCTree [55]	-	-	14.0	17.9	19.4	8.2	10.1	10.8	5.2	6.9	8.0
MOTIFS†	SUM	Baseline	11.5	14.6	15.8	6.5	8.0	8.5	4.1	5.5	6.8
		Focal	10.9	13.9	15.0	6.3	7.7	8.3	3.9	5.3	6.6
		Reweight	16.0	20.0	21.9	8.4	10.1	10.9	**6.5**	**8.4**	**9.8**
		Resample	14.7	18.5	20.0	9.1	11.0	11.8	5.9	8.2	9.7
		X2Y	13.0	16.4	17.6	6.9	8.6	9.2	5.1	6.9	8.1
		X2Y-Tr	11.6	14.9	16.0	6.5	8.4	9.1	5.0	6.9	8.1
		TE	18.2	25.3	29.0	8.1	12.0	14.0	5.7	8.0	9.6
		NIE	0.6	1.1	1.4	6.1	9.0	10.6	3.8	5.1	6.0
		TDE	**18.5**	**25.5**	**29.1**	**9.8**	**13.1**	**14.9**	5.8	8.2	**9.8**
	GATE	Baseline	12.2	15.5	16.8	7.2	9.0	9.5	5.2	7.2	8.5
		TDE	**18.5**	**24.9**	**28.3**	**11.1**	**13.9**	**15.2**	**6.6**	**8.5**	**9.9**
VTransE†	SUM	Baseline	11.6	14.7	15.8	6.7	8.2	8.7	3.7	5.0	6.0
		TDE	**17.3**	**24.6**	**28.0**	**9.3**	**12.9**	**14.8**	**6.3**	**8.6**	**10.5**
	GATE	Baseline	13.6	17.1	18.6	6.6	8.2	8.7	5.1	6.8	8.0
		TDE	**18.9**	**25.3**	**28.4**	**9.8**	**13.1**	**14.7**	**6.0**	**8.5**	**10.2**
VCTree†	SUM	Baseline	11.7	14.9	16.1	6.2	7.5	7.9	4.2	5.7	6.9
		TDE	**18.4**	**25.4**	**28.7**	**8.9**	**12.2**	**14.0**	**6.9**	**9.3**	**11.1**
	GATE	Baseline	12.4	15.4	16.6	6.3	7.5	8.0	4.9	6.6	7.7
		TDE	**17.2**	**23.3**	**26.6**	**8.9**	**11.8**	**13.4**	**6.3**	**8.6**	**10.3**

Table 1. The SGG performances of Relationship Retrieval on mean Recall@K [55, 6]. The SGG models reimplemented under our codebase are denoted by the superscript †.

Graph Classification **(SGCls)**: using ground truth bounding boxes without labels, (3) Scene Graph Detection **(SGDet)**: detecting SGs from scratch. For Sentence-to-Graph Retrieval, the results of SGDet are generated as image representations. PredCls and SGCls are not qualified for both of them require ground truth annotations.

Relationship Retrieval (RR). The conventional metric of RR is **Recall@K (R@K)**, which was abandoned in this paper due to the reporting bias [35]. As illustrated in Figure 3(b), previous method [69] with good performance on R@K gave semantically informative predicates extremely low Recall, if not completely unable to be detected, *e.g.*, predicates like `parked on`, `laying on` have embarrassingly 0.0 Recall@100. To speak for all predicates rather than very few trivial ones, we adopted a recent replacement, **mean Recall@K (mR@K)**, proposed by Tang *et al*. [55] and Chen *et al*. [6]. mR@K retrieves each predicate separately then averages R@K for all predicates.

Zero-Shot Relationship Retrieval (ZSRR). ZSRR was introduced by Lu *et al*. [31] as **Zero-Shot Recall@K** and was firstly evaluated on VG in this paper, which only reports the R@K of those subject-predicate-object triplets that had never been observed in the training set.

Sentence-to-Graph Retrieval (S2GR). Both RR and ZSRR are triplet-level evaluations, ignoring the graph-level comprehensiveness. Therefore, we design S2GR, using human descriptions to retrieve detected SGs. We didn't use proxy vision-language tasks like captioning [66, 67] and VQA [56, 14] as the diagnosis, because their implementations have too many components unrelated to SGG and their datasets are challenged by their own biases [1, 13, 33]. In S2GR, the detected SGs were regarded as the only representations of images, cutting off all the reliances on black-box visual features, so any bias on SGG would directly violate the comprehensiveness of SGs, resulting in worse retrieval results. For example, if `walking on` was detected as the biased alternate `on`, it would be mixed up with `sitting on` or `laying on`. Meanwhile, if the rare triplet `person standing on table` failed to be detected, it would never be retrieved from a bunch of `person at table`. Note that previous image retrieval with scene graph [17, 50] takes SGs as queries rather than image representations, which is different to the proposed S2GR. **Recall@20/100 (R@20/100)** and median ranking indexes of retrieved results **(Med)** on the gallery size of 1,000 and 5,000 were evaluated in Table 3.

5.3. Implementation Details

Object Detector. Following the previous works [64, 69, 55], we pre-trained a Faster R-CNN [44] and froze it to be the underlying detector of our SGG models. We equipped the Faster R-CNN with a ResNeXt-101-FPN [28, 63] backbone and scaled the longer side of input images to be 1k pixels. The detector was trained on the training set of VG using SGD as optimizer. We set the batch size to 8 and the initial learning rate to 8×10^{-3}, which was decayed by the factor of 10 on 30k$^{\text{th}}$ and 40k$^{\text{th}}$ steps. The final detector achieved 29.6mAP on VG (using 0.5 IoU threshold). 4

Zero-Shot Relationship Retrieval			PredCls	SGCls	SGDet
Model	Fusion	Method	R@50/100	R@50/100	R@50/100
MOTIFS†	SUM	Baseline	10.9 / 14.5	2.2 / 3.0	0.1 / 0.2
		Focal	10.9 / 14.4	2.2 / 3.1	0.1 / 0.3
		Reweight	0.7 / 0.9	0.1 /0.1	0.0 / 0.0
		Resample	11.1 / 14.3	2.3 / 3.1	0.1 / 0.3
		X2Y	11.8 / 17.6	2.3 / 3.7	1.6 / 2.7
		X2Y-Tr	13.7 / 17.6	3.1 / 4.2	1.8 / 2.8
		TE	14.2 / 18.1	1.4 / 2.0	1.4 / 1.8
		NIE	2.4 / 3.2	0.2 / 0.4	0.3 / 0.6
		TDE	**14.4 / 18.2**	**3.4 / 4.5**	**2.3 / 2.9**
	GATE	Baseline	7.4 / 10.6	0.9 / 1.3	0.2 / 0.4
		TDE	**7.7 / 11.0**	**1.9 / 2.6**	**1.9 / 2.5**
VTransE†	SUM	Baseline	11.3 / 14.7	2.5 / 3.3	0.8 / 1.5
		TDE	**13.3 / 17.6**	**2.9 / 3.8**	**2.0 / 2.7**
	GATE	Baseline	4.2 / 5.9	1.9 / 2.6	**1.9** / 2.6
		TDE	**5.3 / 7.9**	**2.1 / 3.0**	**1.9 / 2.7**
VCTree†	SUM	Baseline	10.8 / 14.3	1.9 / 2.6	0.2 / 0.7
		TDE	**14.3 / 17.6**	**3.2 / 4.0**	**2.6 / 3.2**
	GATE	Baseline	4.4 / 6.8	2.5 / 3.3	1.8 / 2.7
		TDE	**5.9 / 8.1**	**3.0 / 3.7**	**2.2 / 2.8**

Table 2. The results of Zero-Shot Relationship Retrieval.

Sentence-to-Graph Retrieval								
Gallery Size			1000			5000		
Model	Fusion	Method	R@20	R@100	Med	R@20	R@100	Med
MOTIFS†	SUM	Baseline	11.6	39.9	155	3.1	12.1	708
		Focal	10.9	39.0	163	2.9	11.1	737
		Reweight	9.7	36.8	159	3.0	11.4	725
		Resample	13.1	43.6	124	2.5	13.4	593
		X2Y	14.3	44.8	125	3.5	14.6	556
		X2Y-Tr	14.5	45.6	114	3.9	16.8	525
		TE	15.9	49.9	100	4.4	16.9	469
		NIE	6.7	29.2	202	1.6	8.6	1050
		TDE	**17.0**	**53.6**	**91**	**5.2**	**18.9**	**425**
	GATE	Baseline	13.7	45.6	143	4.4	16.2	618
		TDE	**20.8**	**59.2**	**72**	**5.2**	**21.3**	**325**
VTransE†	SUM	Baseline	12.3	42.3	129	**3.6**	15.0	596
		TDE	**14.7**	**48.4**	**106**	**3.6**	**16.3**	**483**
	GATE	Baseline	12.9	41.8	136	3.8	14.3	634
		TDE	**18.5**	**50.4**	**110**	**4.5**	**19.1**	**486**
VCTree†	SUM	Baseline	9.9	37.4	150	3.1	11.5	745
		TDE	**19.0**	**57.0**	**82**	**5.0**	**20.0**	**385**
	GATE	Baseline	13.4	44.1	121	3.7	13.6	583
		TDE	**19.1**	**55.5**	**87**	**5.1**	**20.3**	**395**

Table 3. The results of Sentence-to-Graph Retrieval.

RTX 2080ti GPUs were required for the pre-training.

Scene Graph Generation. On top of the frozen detector, we trained SGG models using SGD as optimizer. Batch size and initial learning rate were set to be 12 and 12×10^{-2} for PredCls and SGCls; 8 and 8×10^{-2} for SGDet. The learning rate would be decayed by 10 two times after the validation performance plateaus. For SGDet, 80 RoIs were sampled for each image and Per-Class NMS [48, 69] with 0.5 IoU was applied in object prediction. We sampled up to 1,024 subject-object pairs containing 75% background pairs during training. Different from previous works [69, 55, 5], we didn't assume subject-object pairs without overlapping to be invalid in SGDet, making our models more general.

Sentence-to-Graph Retrieval. We handled S2GR as a graph-to-graph matching problem. The query captions of each image were merged together and parsed to a text-SG by [50]. We set all the text entities and predicates that appear less than 5 times to "UNKNOWN" token, obtaining text entity and predicate dictionaries of size 4,459 and 645 respectively. The original image-SG generated from SGDet detected a fixed number of RoIs and forced all valid subject-object pairs to predict foreground relationships, which is inappropriate for a real-world SG. Therefore, we used a threshold of 0.1 to filter RoIs by the probabilities of their labels and removed all background predicates from the graph. The sizes of entity and predicate dictionaries are 150 and 50 as we mentioned before. The output dimensions of all vocabulary embedding layers were set to 512. We further adopted a 2 layer Bilinear Attention Network (BAN) [19] to encode the interaction among entities and triplet relationships using the detected SG as a hard attention map. The same BAN is shared by both text-SG and image-SG. The encoded representations were projected into the dimension of 1,024 and trained by the triplet loss [49] 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.

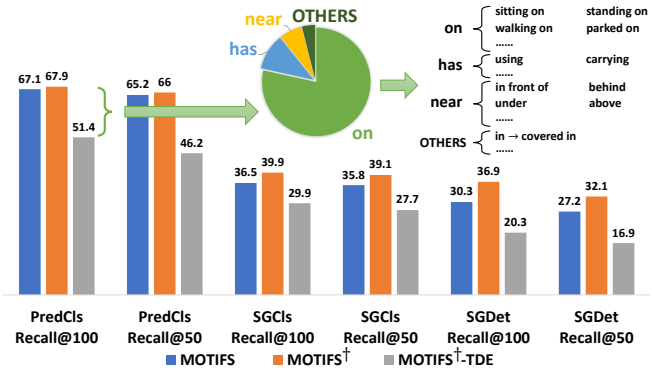


Figure 6. To investigate the decrease of Recall@K after applying TDE, we use the pie chart to summarize all the relationships, that are correctly detected by the baseline model but assigned "incorrect" predicates by TDE. The right side of the pie chart shows the corresponding labels given by the TDE, which imply that the relationships predicted by TDE are even more fine-grained than some ground truth annotations. Combining with our qualitative examples, we believe that the decrease is caused by two reasons: 1) the annotators preference towards simple annotations caused by bounded rationality, 2) TDE tends to predict more action-like relationships rather than vague prepositions such as on/near.

5.4. Ablation Studies

Except for the models and fusion functions that we've discussed before, we also investigated three conventional debiasing methods, two intuitive causal graph surgeries, and other two types of causal effects: 1) **Focal**: focal loss [29] automatically penalizes well-learned samples and focuses on the hard. We followed the best hyper-parameters ($\gamma = 2.0, \alpha = 0.25$) in [29]. 2) **Reweight**: weighted cross-entropy is widely accepted in the industry for biased data. The inversed sample fractions were assigned to each predicate category as weights. 3) **Resample** [3]: rare categories were up-sampled by the inversed sample fraction during training. 4) **X2Y**: since we argued that the unbiased effect was under the effect of object features X, it directly

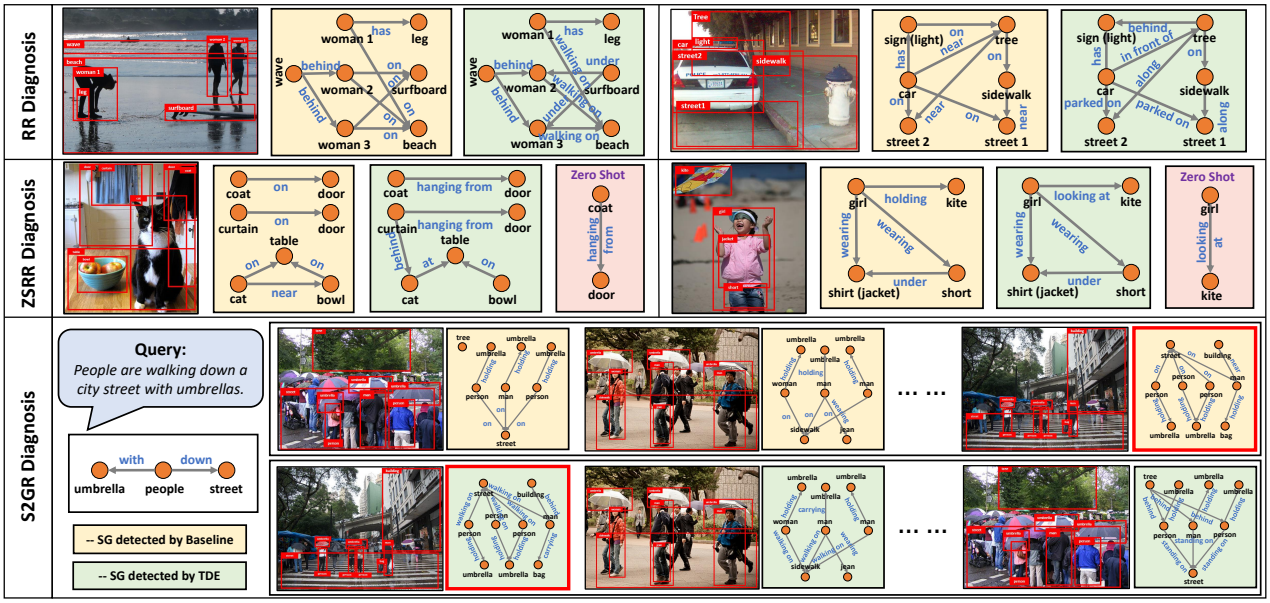


Figure 7. Results of scene graphs generated from MOTIF†-SUM baseline (yellow) and corresponding TDE (green). Top: relationship retrieval results. Mid: zero shot relationship retrieval results. Red boxes indicate the zero shot triplets. Bottom: results of S2GR. Red boxes mean the correctly retrieved SGs. Part of the trivial detected objects are removed from the graphs, due to space limitation.

generated SG by the outputs of $X \rightarrow Y$ branch after biased training. 5) **X2Y-Tr**: it even cut off other branches, using $X \rightarrow Y$ for both training and testing. 6) **TE**: as we introduced in Section 4, TE is the debiasing method that not conditioned on the language contexts. 7) **NIE**: it is the marginal difference between TDE and TE, *i.e.*, NIE = TE-TDE, which can be considered as the pure effect caused by introducing language bias $Z \rightarrow Y$. **NOTE**: although zero vector can also be used as the wiped-out input $\bar{x}$, we chose the mean feature of training set for tiny improvements.

5.5. Quantitative Studies

RR & ZSRR. Despite the conventional debiasing methods, Reweight and Resample, directly hack the mR@K metric, they only gained limited advantages on RR but not on ZSRR. Contrast to the high mR@K Reweight had on RR SGDet, it got embarrassingly $0.0/0.0$ in ZSRR SGDet, indicating that such debiased training methods hurt the useful prior. Focal loss [29] barely worked for both RR and ZSRR. Causal graph surgeries, X2Y and X2Y-Tr, both improved RR and ZSRR from the baseline, yet their increments were limited. TE had very similar performance with TDE, but as we discussed, it removed the general bias rather than subject-object specific bias. NIE is the marginal improvements from TE to TDE, which was even worse than baseline. Although R@K is not a qualified metric for RR as we discussed, we reported the R@50/100 performance of MOTIFS†-SUM in Figure 6 anyway. Not surprisingly, we observed a performance drop from baseline to TDE, but a further analysis shows that those considered as correct in baseline and "incorrect" in TDE were mainly several superclass predicates and just classified into their subclass alternates as we listed on the right of Figure 6. Among all three models and two fusion functions, even the worst TDE performance outperforms previous state-of-the-art methods [55, 6] by large margin on RR mR@K.

S2GR. In S2GR, Focal and Reweight were even worse than the baseline. Among all three conventional debiasing methods, Resample is the most stable one based on our experiments. X2Y and X2Y-Tr had little advantages over baseline, yet they were still much worse than effect based methods. TE took second place and was only a little bit worse than TDE. NIE is the worst as we expected. It is worth highlighting that all three models and two fusion functions had significant improvements after we applied TDE.

5.6. Qualitative Studies

We visualized several SGCls examples that generated from MOTIFS†-SUM baseline and TDE in the top and mid rows of Figure 7, scene graphs generated by TDE are much more discriminative compared to the baseline model which prefers trivial predicates like `on`. The right half of the mid row shows that baseline model would even generate `holding` due to the long-tail bias when the girl is not touching the kite, implying that the biased predictions are easy to get "blind", while TDE successfully predicted `looking at`. The bottom of Figure 7 is an example of S2GR, where the SGs detected by baseline model lost the detailed actions of people, considering both `person walking on street` and `person standing on street` as `person on street`, which caused worse retrieval results. All the qualitative examples show a clear

tendency that TDE was much more sensitive to those semantically informative relationships instead of biased trivial ones.

6. Conclusions

We presented a general framework for unbiased SGG from biased training, and this is the first work addressing the serious bias issue in SGG. With the power of *counterfactual causality*, we can remove the harmful bias from the context, which cannot be easily identified by traditional debiasing methods such as the data augmentation [9, 11] and unbiased learning [29]. We implemented this by calculating the Total Direct Effect (TDE) with the help of a causal graph, which is a roadmap for training any SGG model. By using the proposed Scene Graph Diagnosis toolkit, our unbiased SGG results are considerably better than their biased counterparts. Our future plans are two-fold: 1) develop more SGG diagnostic tools such as the graph reasoning in GQA [14], and 2) as the framework is model-agnostic, we will bring more models in the zoo and provide a comprehensive bias study for the community.

References

[1] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018.

[2] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[3] E. Burnaev, P. Erofeev, and A. Papanov. Influence of resampling on accuracy of imbalanced classification. In *ICMV*, 2015.

[4] R. Cadene, C. Dancette, H. Ben-younes, M. Cord, and D. Parikh. Rubi: Reducing unimodal biases in visual question answering. *arXiv preprint arXiv:1906.10169*, 2019.

[5] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang. Counterfactual critic multi-agent training for scene graph generation. In *ICCV*, 2019.

[6] T. Chen, W. Yu, R. Chen, and L. Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, 2019.

[7] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *CVPR*, 2017.

[8] G. Dunn, R. Emsley, H. Liu, S. Landau, J. Green, I. White, and A. Pickles. Evaluation and validation of social and psychological markers in randomised trials of complex interventions in mental health: a methodological research programme. 2015.

[9] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[10] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, 2019.

[11] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 2009.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[13] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*. Springer, 2018.

[14] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[15] D. A. Hudson and C. D. Manning. Learning by abstraction: The neural state machine. *NeurIPS*, 2019.

[16] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018.

[17] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015.

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

[19] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, 2018.

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

[21] R. Krishna, I. Chami, M. Bernstein, and L. Fei-Fei. Referring relationships. In *CVPR*, 2018.

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 2017.

[23] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, 2017.

[24] Y. Li, Y. Li, and N. Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, 2018.

[25] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and caption regions. In *ICCV*, 2017.

[26] Y. Li and N. Vasconcelos. Repair: Removing representation bias by dataset resampling. In *CVPR*, 2019.

[27] Y. Liang, Y. Bai, W. Zhang, X. Qian, L. Zhu, and T. Mei. Vrr-vg: Refocusing visually-relevant relationships. In *ICCV*, pages 10403–10412, 2019.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014.

[31] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016.

[32] D. P. MacKinnon, A. J. Fairchild, and M. S. Fritz. Mediation analysis. *Annu. Rev. Psychol.*, 2007.

[33] V. Manjunatha, N. Saini, and L. S. Davis. Explicit bias discovery in visual question answering models. In *CVPR*, 2019.

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

[35] I. Misra, C. Lawrence Zitnick, M. Mitchell, and R. Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *CVPR*, 2016.

[36] S. Nair, Y. Zhu, S. Savarese, and L. Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019.

[37] Y. Niu, K. Tang, H. Zhang, Z. Lu, X. Hua, and J.-R. Wen. Counterfactual vqa: A cause-effect look at language bias. *arXiv*, 2020.

[38] J. Pearl. *Causality: models, reasoning and inference*. Springer, 2000.

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

[40] J. Pearl, M. Glymour, and N. P. Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[41] J. Pearl and D. Mackenzie. *THE BOOK OF WHY: THE NEW SCIENCE OF CAUSE AND EFFECT*. Basic Books, 2018.

[42] J. Qi, Y. Niu, J. Huang, and H. Zhang. Two causal principles for improving visual dialog. *arXiv preprint arXiv:1911.10496*, 2019.

[43] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo. Attentive relational networks for mapping images to scene graphs. In *CVPR*, 2019.

[44] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015.

[45] L. Richiardi, R. Bellocco, and D. Zugna. Mediation analysis in epidemiology: methods, interpretation and bias. *International journal of epidemiology*, 2013.

[46] J. M. Robins and S. Greenland. Identifiability and exchangeability for direct and indirect effects. *Epidemiology*, 1992.

[47] N. J. Roese. Counterfactual thinking. *Psychological bulletin*, 1997.

[48] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 1971.

[49] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[50] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. 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.

[51] J. Shi, H. Zhang, and J. Li. Explainable and explicit visual reasoning over scene graphs. In *CVPR*, 2019.

[52] H. A. Simon. Bounded rationality. In *Utility and probability*. Springer, 1990.

[53] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[54] H. Z. Q. S. Tan Wang, Jianqiang Huang. Visual commonsense r-cnn. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[55] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, 2019.

[56] D. Teney, L. Liu, and A. van den Hengel. Graph-structured representations for visual question answering. In *CVPR*, 2017.

[57] A. Torralba, A. A. Efros, et al. Unbiased look at dataset bias. In *CVPR*, 2011.

[58] N. Van Hoeck, P. D. Watson, and A. K. Barbey. Cognitive neuroscience of human counterfactual reasoning. *Frontiers in human neuroscience*, 2015.

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

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

[61] W. Wang, R. Wang, S. Shan, and X. Chen. Exploring context and visual pattern of relationship for scene graph generation. In *CVPR*, 2019.

[62] S. Woo, D. Kim, D. Cho, and I. S. Kweon. Linknet: Relational embedding for scene graph. In *Advances in Neural Information Processing Systems*, 2018.

[63] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[64] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, 2017.

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

[66] X. Yang, K. Tang, H. Zhang, and J. Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019.

[67] T. Yao, Y. Pan, Y. Li, and T. Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018.

[68] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. Change Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. In *ECCV*, 2018.

[69] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, 2018.

[70] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *ICML*, 2013.

[71] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *CVPR*, 2017.

[72] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, 2019.

Abstract

This supplementary document is organized as follows:

- *Section A: A comprehensive review of causal effect analysis in causal inference.*
- *Section B: More details of the simplified network structures in the original paper.*
- *Section C: More quantitative studies.*
- *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 [40, 41] while the extension of effect analysis (a.k.a. mediation analysis) is given in [46, 39, 60, 59].

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 between the path of two 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:

$$TE = Y_x(u) - Y_{\bar{x}}(u). \tag{8}$$

As illustrated in Figure 8, 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.

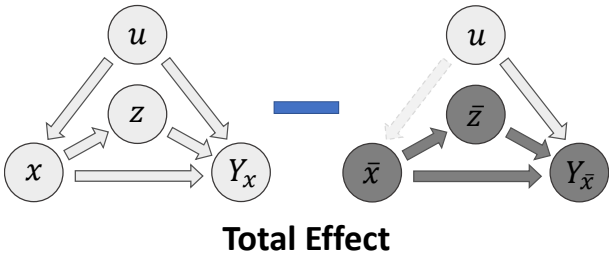


Figure 8. The illustration of Total Effect on causal graph.

However, by only using the TE, we are still not able to separate the "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 [39, 60]. 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:

$$TDE = Y_x(u) - Y_{\bar{x},z}(u), \tag{9}$$

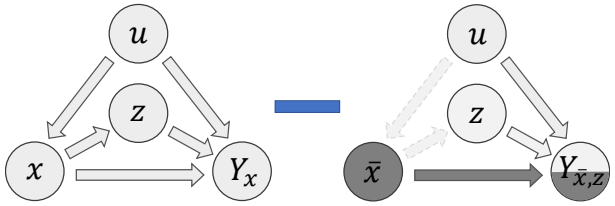
which can be regarded as the effect of X in the real situation, *i.e.*, Z always takes the value z as if it had seen the real x. Meanwhile, the NIE or PIE is the effect caused by the mediator Z under a pure/natural situation, *i.e.*, X will not take the value x under the specific case and it's only assigned to the general unactivated value $\bar{x}$. Therefore, the NIE of Z is denoted as:

$$\begin{aligned} NIE &= Y_{\bar{x},z}(u) - Y_{\bar{x}}(u) && (10)\\ &= TE - TDE, && (11) \end{aligned}$$

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 9.

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:

$$TIE = Y_x(u) - Y_{x,\bar{z}}(u). \tag{12}$$



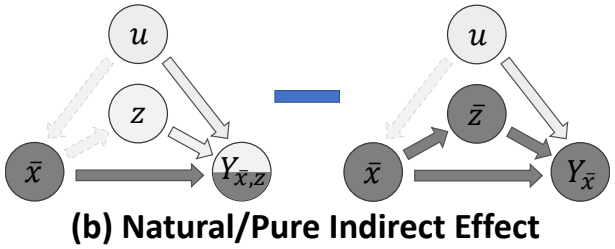
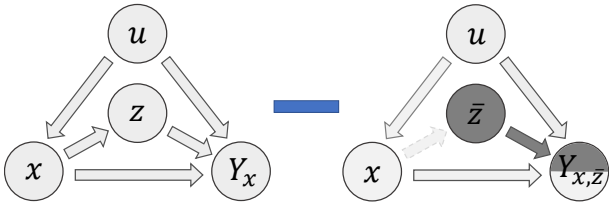


Figure 9. The illustration of Total Direct Effect and Pure/Natural Indirect Effect on causal graph.



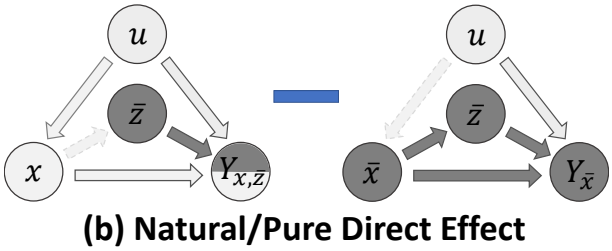


Figure 10. The illustration of Total Indirect Effect and Pure/Natural Direct Effect on causal graph.

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:

$$NDE = Y_{x,\bar{z}}(u) - Y_{\bar{x}}(u) \quad (13)$$
$$= TE - TIE, \quad (14)$$

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 [45, 18, 8, 32, 20], 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 [55] construction module. Their details will be given in this subsection.

Feature Extraction Module. Since we adopted ResNeXt-101-FPN [28, 63] 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 [34]. Given a bounding box b_i with area a_i, the corresponding index k of feature map is calculated as follows:

$$k = \max(2, \min(5, \lfloor 4 + \log_2(a_i/224 + 1 \times 10^{-6}) \rfloor)) - 2. \quad (15)$$

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 4. The details of Visual Context Module.

Then ROIAlign [12] 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 [12] 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 4, 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 [71] or MOTIFS [69], 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 [53] message passing. The construc-

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 5. The details of VCTree Construction Module.

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)	$\downarrow$	Transpose + Unsqueeze	$(512 \times N_r \times 1)$
(6)	(3)	$\downarrow$	Unsqueeze	$(1 \times N_e \times N_r)$
(7)	(4),(6)	$\downarrow$	Matrix Multiplication	$(512 \times 1 \times N_r)$
(8)	(5),(7)	$\downarrow$	Matrix Multiplication	$(512 \times 1 \times 1)$
(9)	(8)	$\downarrow$	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 6. The details of Bilinear Attention Scene Graph Encoding Module.

tion 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 5. 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 [55] 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 [69], 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.

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 [50]. 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:

$$E_{embed}^k = W_e^k E^k, \tag{16}$$

$$R_{embed}^k = [W_s^k S^k; W_p^k P^k; W_o^k O^k], \tag{17}$$

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) [19] 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:

$$M_{ij} = \begin{cases} 1, \ if \ E_i \ in \ R_j, \\ 0, \ if \ E_i \ not \ in \ R_j. \end{cases} \tag{18}$$

The bilinear attended scene graph encoding is calculated by Table 6, where steps (4-10) are calculated 2 times, and the final output $E_{graph} \in \mathbb{R}^{1024}$ is a feature vector representing the whole SG. The same BAN is used for both text-SG or image-SG, *i.e.*, the parameters of the BAN are shared.

C. Quantitative Studies

The full results of Relationship Retrieval, including both conventional Recall@K and the adopted mean Recall@K [55, 6], are given in Table 7. 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 12 13 14. Impressively, the distribution of the improved performances is no longer long-tailed while those conventional debiasing methods illustrated in

			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+ [64, 6]	-	-	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 [69, 55]	-	-	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 [69, 55]	-	-	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 [6]	-	-	- / 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 [55]	-	-	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 7. The SGG performances of Relationship Retrieval on both conventional **Recall@K** and **mean Recall@K** [55, 6]. The SGG models reimplemented under our codebase are denoted by the superscript †.

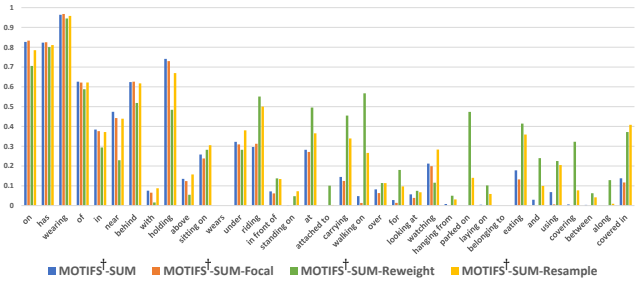


Figure 11. Conventional Debiasing Methods: Recall@100 on Predicate Classification for the most frequent 35 predicates.

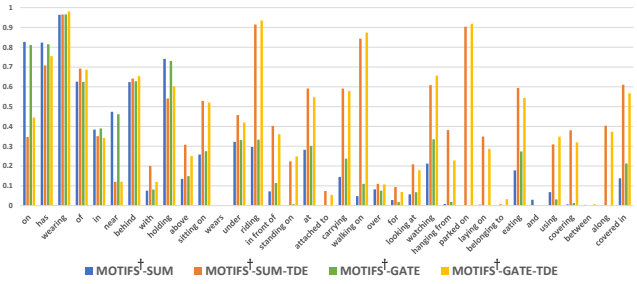


Figure 12. MOTIFS† [69]: Recall@100 on Predicate Classification for the most frequent 35 predicates.

Figure 11 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.

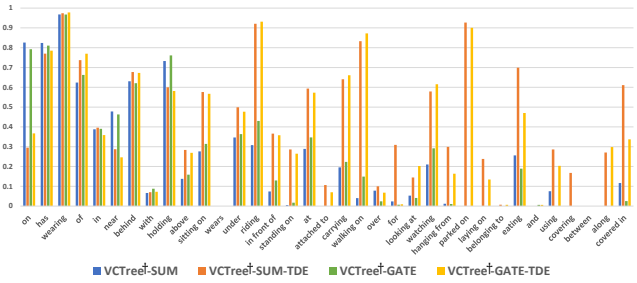


Figure 13. VCTree† [55]: Recall@100 on Predicate Classification for the most frequent 35 predicates.

D. Qualitative Studies

More Relationship Retrieval (RR) and Zero-Shot Relationship Retrieval (ZSRR) results are given in Figure 15, 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 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

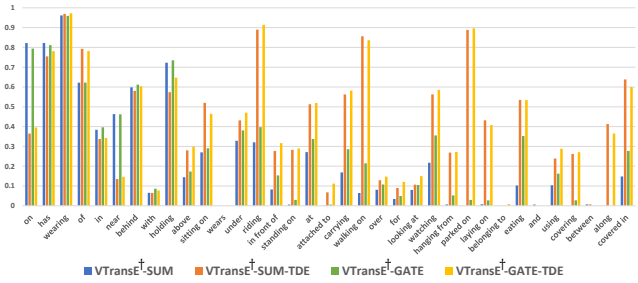


Figure 14. VTransE† [71]: Recall@100 on Predicate Classification for the most frequent 35 predicates.

baseline is more natural in this case.

Another example of Sentence-to-Graph Retrieval (S2GR) is illustrated in Figure 16. 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.

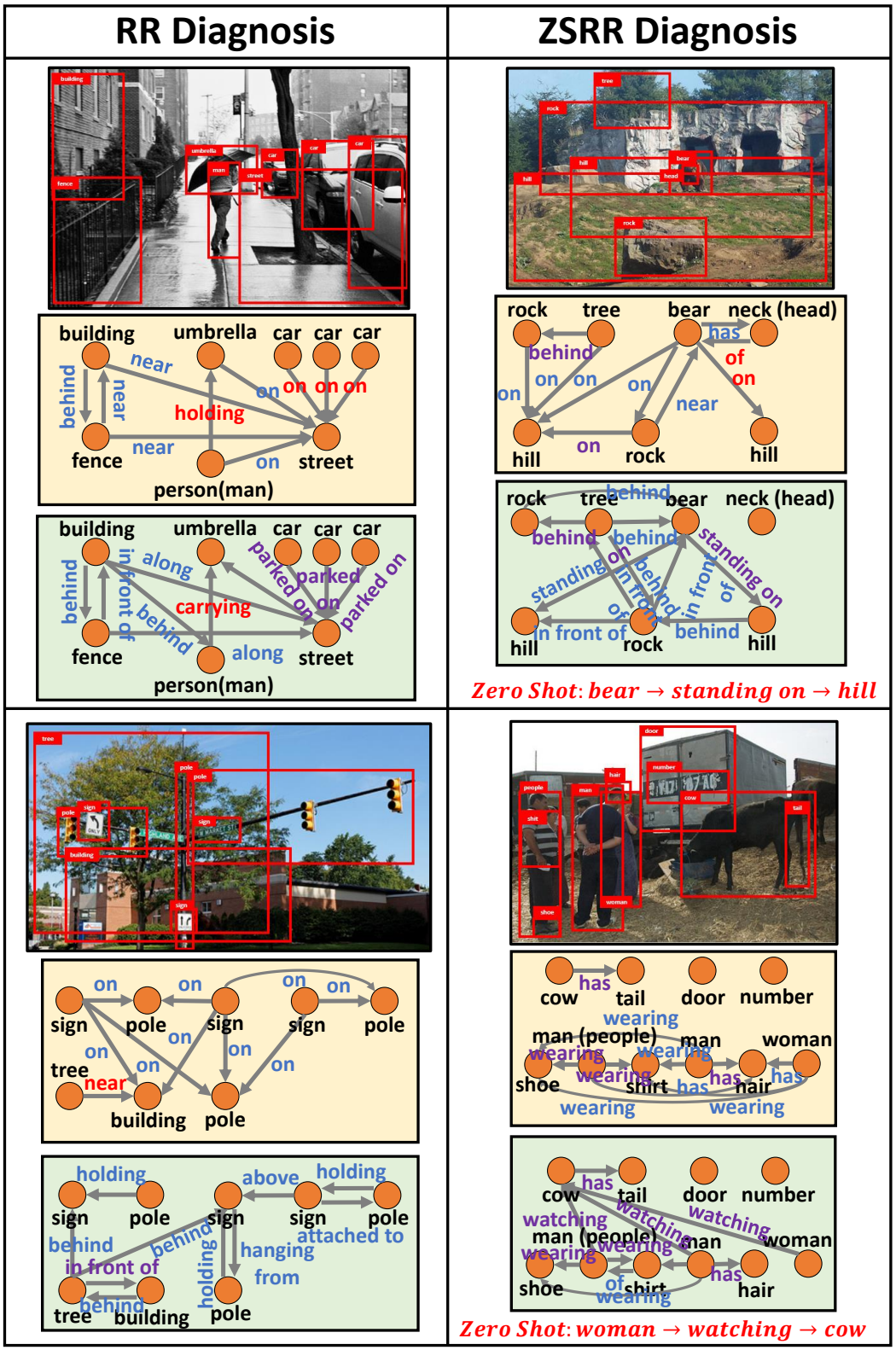


Figure 15. Top 10 Relationship Retrieval (RR) and Zero-Shot Relationship Retrieval (ZSRR) results of SGCls for MOTIFS†+SUM baseline (yellow box) and corresponding TDE (green box). The red predicates indicate misclassified relationships, the purple predicates are those correctly classified relationships (in ground truth), the blue predicates are those not labeled in ground truth.

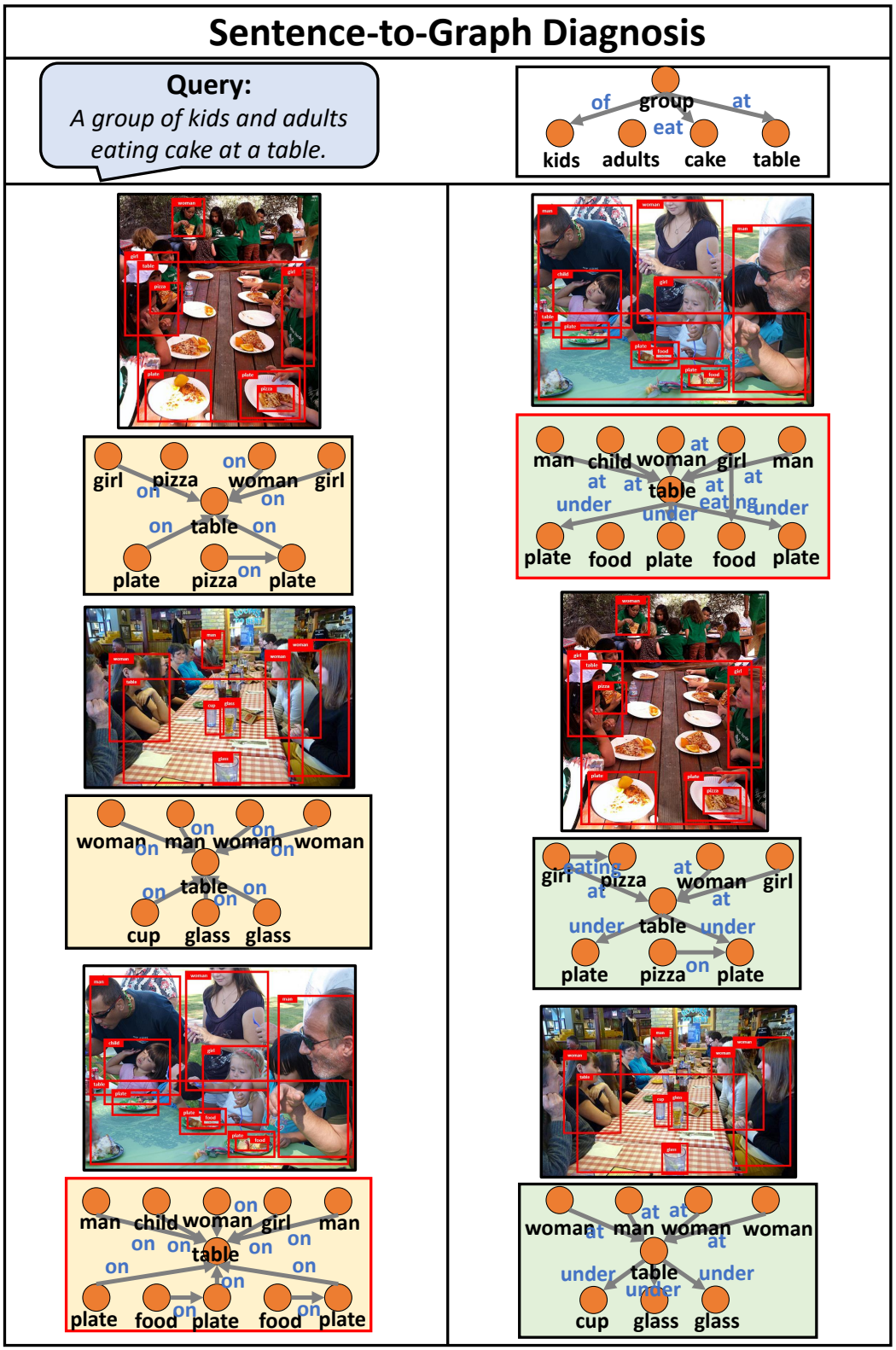


Figure 16. An example of Sentence-to-Graph Retrieval (S2GR) results for MOTIFS†+SUM baseline (yellow box) and corresponding TDE (green box). The red boxes indicate ground truth matching results. Note that we only draw sub-graphs containing important objects and predicates, because the original detected scene graphs from SGDet have too many trivial objects and predicates.