Vision Relation Transformer for Unbiased Scene Graph Generation

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

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

Abstract

*Recent years have seen a growing interest in Scene Graph Generation (SGG), a comprehensive visual scene understanding task that aims to predict entity relationships using a relation encoder-decoder pipeline stacked on top of an object encoder-decoder backbone. Unfortunately, current SGG methods suffer from an information loss regarding the entities' local-level cues during the relation encoding process. To mitigate this, we introduce the **V**ision **rE**lation **T**ransf**O**rmer (VETO), consisting of a novel local-level entity relation encoder. We further observe that many existing SGG methods claim to be unbiased, but are still biased towards either head or tail classes. To overcome this bias, we introduce a **M**utually **E**xclusive **E**xper**T** (MEET) learning strategy that captures important relation features without bias towards head or tail classes. Experimental results on the VG and GQA datasets demonstrate that VETO + MEET boosts the predictive performance by up to 47% over the state of the art while being $\sim 10\times$ smaller.[1]*

1. Introduction

Visual scene understanding has made great strides in recent years, extending beyond standard object detection and recognition tasks to tackle more complex problems such as visual question answering [1] and image captioning [11]. One powerful tool for scene understanding is Scene Graph Generation (SGG), which aims to identify the relationships between entities in a scene [23]. However, despite recent advancements, SGG models still have significant limitations when it comes to real-world applications.

Conventional SGG approaches, as shown in Fig. 1 (panel 3), generate global-level entity patches for relation encoding. Yet, during the relation encoding process, they lose *local-level* entity information. As illustrated in Fig. 2a, we humans have a tendency to focus on the critical local-level information necessary to construct relations between
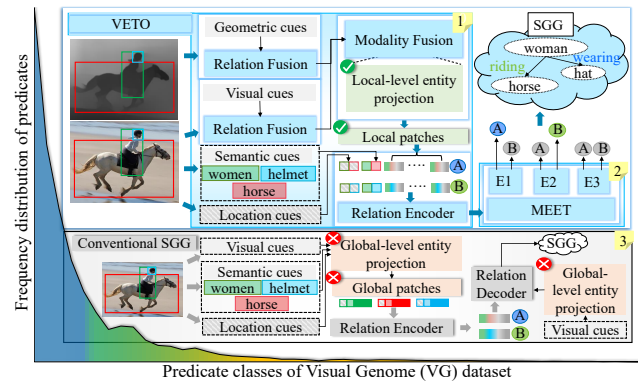

Figure 1. **VETO-MEET *vs*. Conventional SGG.** *(1)* VETO: Enhancing the information flow from entity features to relationship prediction by using a local-level entity relation encoder that conducts relation and modality fusion of local-level entity patches. The local-level components (green ticks) keep the model lightweight while reducing information loss. A (blue) and B (green) denote example relation classes taken from the corresponding colored region in the predicate frequency histogram of the VG dataset [17]. *(2)* MEET: Debiased relation decoder that employs out-of-distribution aware mutually exclusive experts (E1–E3). Grey A and B denote an out-of-distribution prediction discarded by the model. *(3)* Conventional SGG: The projection components (red crosses) yield a computationally expensive model and the global-level entity patches result in a local-level information loss.

things in a scene, which is overlooked by current SGG approaches. Moreover, the major parameter count of current SGG models stems from projections (red-crossed components in Fig. 1) involved in global-level entity patch generation. Another challenge with existing SGG approaches, despite efforts to enhance scene graphs using additional cues like depth maps and knowledge graphs [27, 48], is that they are either resource intensive or limited in exploiting cross-modal information.

Finally but crucially, SGG training setups are challenged by the strong bias of the visual world around us towards a few frequently occurring relationships, leaving a long tail of under-represented relations. This is also the case with

(a) Significance of local-level cues

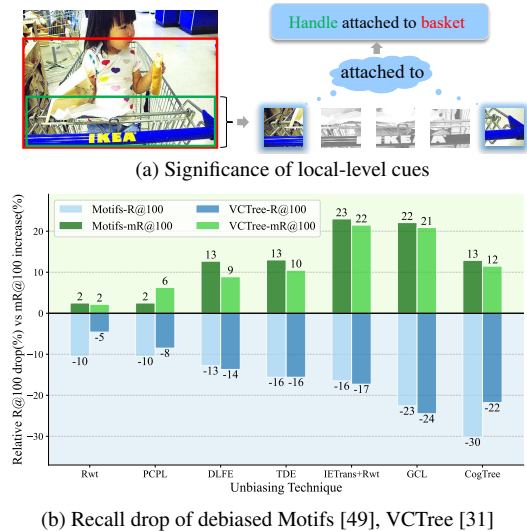(b) Recall drop of debiased Motifs [49], VCTree [31]

Figure 2. **Challenges in SGG.** *(a)* For establishing the *attached to* relation between *Handle* and *Basket*, the attention should be on the corner regions of the object. *(b)* R@100 drop (%) and mR@100 increase (%) of unbiased SGG methods Motifs and VCTree relative to their vanilla versions. The R@100 metric measures the average recall of all predictions, which is higher for models that overfit to the head classes, while mR@100 denotes the per-predicate class mean and is higher for models that overfit to the tail classes.

benchmark SGG datasets, *e.g.*, Visual Genome (VG) [17], as depicted by the predicate[2] class frequency distribution in Fig. 1. Due to the dominance of few head predicates, conventional SGG models [31, 49] are biased towards the head classes. Though several unbiased SGG methods have been proposed [7,30,46] to overcome this issue, they are prone to over-fitting to the tail classes at the expense of head classes (*cf*. Fig. 2b). Despite recent efforts [7] to fix this bias issue using multi-expert learning strategies, we find that they still over-fit to the tail classes ("GCL" in Fig. 2b). Overall, there are two main problems with present unbiased SGG methods: *(1)* Conventional methods, including debiased models, can only learn a limited range of predicates. *(2)* Existing multi-expert SGG models lack adequate exclusivity to enhance both head and tail classes at the same time.

Consequently, we propose the *Vision rElation TransfOrmer (VETO)*. Inspired by Vision Transformers [8] that use image-level patches for classification, VETO generates *local-level* entity patches for the relation prediction task. This improves the information flow from entity features to relationship prediction by channeling the attention towards fused *local* feature cues of subject and object entities (Relation Fusion in Fig. 1) and using a local-level entity relation encoder, which processes entity features at the subregion level. To strengthen the encoder further, we infuse geometric cues into VETO using a Modality Fusion com-

---

[2]We use the terms predicate/relation interchangeably in this paper.

ponent (*cf*. Fig. 1), which unites visual and geometric features to yield local-level entity patches. Finally, to successfully debias VETO, we propose a multi-expert learning strategy termed *Mutually Exclusive ExperTs (MEET)*. After splitting the predicate classes into subgroups, we perform in-distribution and out-of-distribution (OOD) sampling for each subgroup. Then we train each expert on every predicate class but each expert will be responsible for only a subset of predicates with out-of-distribution prediction handling predicates outside its subgroup. In contrast to existing multi-expert methods [7], where expert classifiers are co-dependent to distill knowledge, OOD sampling enables experts to independently interpret every inference sample.

Contributions. Let us summarize: *(1)* We propose a novel SGG method with a local-level entity relation encoding, which is light-weight and reduces the local-level information loss of entities. *(2)* To strengthen the encoder further, we propose an effective strategy to infuse additional geometric cues. *(3)* We devise a mutually exclusive multi-expert learning strategy that effectively exploits our relation network design by learning subgroup-specific diverse feature representations and discriminating from samples outside its subgroups. *(4)* Our extensive experimentation shows the significance of both VETO and MEET.

2. Related Work

Scene Graph Generation (SGG) is a tool for understanding scenes by simplifying the visual relationships into a summary graph. SGG has been receiving increased attention from the research community due to its potential usability in assisting downstream visual reasoning tasks [16, 28, 36]. While SGG aims to provide a comprehensive view of relationships between objects in a visual scene, there is another set of research that represents interactions as relationships between humans and objects called Humanobject Interaction (HOI) [9,13,32,39]. In this work, the focus is on SGG and its associated literature, emphasizing the study of object relationships within visual scenes.

The SGG task was first introduced by Lu *et al*. [23]. Early approaches mainly focused on including additional features from various sources other than the visual context, resulting in sub-optimal performance [5,21,23]. Later work proposed more powerful relation encoders with rich contextual information by employing message passing [40], sequential LSTMs [31, 49], and fully-connected graphs [3, 5, 20, 36, 37, 40, 44, 48]. Recent advancements in attention techniques have also resulted in attention-based SGG methods. Earlier work [43] in this direction used graph attention networks (GAT) [34] to capture object-level visual similarity. Recently, Transformers [33] have also been used for SGG [7,15,22,24] after their successful adoption across computer vision [2, 8, 25]. Current transformer-based SGG methods use attention to capture global context and improve

the visual and semantic modality fusion. Lu *et al.* [24] used sequential decoding to capture context, while Dong *et al.* [7] employed self- and cross-attention to fuse visual and semantic cues. Deviating from this, we use transformers to capture local-level relation cues as well as joint visual and geometric cues.

Scene Graphs with additional knowledge. Due to the long-tail distribution of the relationships, it is difficult to obtain enough training data for every relation. To overcome this, using additional knowledge in the form of knowledge graphs [10, 47, 48], depth maps [27, 42], and data transfer [50] was proposed. Knowledge graph-based works refine features for relation prediction by reasoning using knowledge from large-scale databases. Yang *et al.* [42] and Sharifzahed *et al.* [27] use a monocular depth estimator to infer additional depth cues for relation prediction by fusing with visual features. Zhang *et al.* [50] expanded the dataset by increasing the SGG annotations through internal and external data transfer. Our approach can also use depth maps to provide additional geometric knowledge. However, introducing additional knowledge can increase the parameter count and computation time of the model. To tackle this problem, we strategically prune the parameters, resulting in a light-weight yet powerful SGG model.

Unbiased Scene Graph Generation. The SGG research community started paying attention to the problem of class imbalance only after the introduction of the less biased mean recall metric by Chen *et al.* [3] and Tang *et al.* [31]. Subsequently, various unbiasing strategies [4, 7, 19, 29, 30, 35, 41, 46, 50] were proposed, many of which can be used in a model-agnostic fashion. Tang *et al.* [31] used counterfactuals from causal inference to disentangle unbiased representations from the biased ones. Yu *et al.* [46] utilized tree structures to filter irrelevant predicates. Zareian *et al.* [48] and Yan *et al.* [41] used re-weighting strategies while Li *et al.* [19] employed a re-sampling strategy. Dong *et al.* [7] used a multi-expert learning setup that leverages knowledge distillation. However, while these methods attain high performance on unbiased metrics, they reduce the head class performance significantly as seen in Fig. 2b. Hence, to attain an effective balance between the head and tail classes, we propose a mutually exclusive expert learning setup. Our model not only achieves better head and tail class balance but also sets a new state of the art.

3. Vision rElation TransfOrmer (VETO)

Our goal is to improve the Scene Graph Generation task that parses an input image to generate a structured graphical representation of entities and their relationships. In particular, we focus on enhancing the overall performance of SGG by improving the prediction on both the head and tail relations. To this end, we introduce a relation network that learns richer entity/predicate representations by focusing on

local-level entity features and devise a *multi-expert* learning strategy to achieve a better relation prediction trade-off.

3.1. Problem setting

For a given image $\mathbf{I}$, the goal of SGG is to create a summary graph $\mathcal{G}$ that adequately summarizes the information present in the image. At first, we detect all the entities within image $\mathbf{I}$, denoted as $\mathcal{E} = \{e_i\}_{i=1}^{N}$. Then we predict the predicates $p_{i \to j}$ for each subject-object entity pair (e_i, e_j). Finally, we construct the scene graph $\mathcal{G}$ using the triplet form of the predictions $(e_i, p_{i \to j}, e_j)$ as

$$\mathcal{G} = \big\{ (e_i, p_{i \to j}, e_j) \mid e_i, e_j \in \mathcal{E},\ p_{i \to j} \in \mathcal{P} \big\}. \quad (1)$$

3.2. The VETO backbone

Roughly speaking, as shown in Fig. 3, the VETO model comprises a feature extraction and a proposal network as the backbone, which are fed to the relation network.

Feature extraction. Following previous work, our feature extraction backbone comprises an RGB feature extractor, which is pre-trained and kept frozen [19, 30], and a depth feature extractor, which is trained from scratch during SGG training [27].

Proposal network. We use Faster R-CNN [26] as our object detector. Entity proposals are obtained directly from the output of object detection, which includes their categories and classification scores. We use the entity proposals to extract scaled RGB features $\mathbf{r}_i$ and their corresponding geometric features $\mathbf{g}_i$ from the depth map. We denote the proposal bounding box as $\mathbf{b}_i$ and its detected class as $\mathbf{c}_i$.

Before explaining the proposed VETO local-level entity generator, let us briefly revisit a conventional SGG [19] pipeline that uses global-level entity projection.

Entity global-level patch generator. Given the extracted RGB features $\mathbf{r}_i$, a global-level patch generator in conventional SGG [19, 31, 49] would first densely project the $\mathbf{r}_i$ to a lower-dimensional visual representation $\mathbf{h}_i$ as

$$\mathbf{h}_i = f_{h2}\left(f_{h1}\left(\mathbf{r}_i\right)\right), \quad (2)$$

where f_{h1} and f_{h2} are two fully-connected layers. This global-level projection (Fig. 1, panel 3) of visual features is not only parameter heavy but can also result in a local-level information loss of entities (Fig. 2a).

Given the entity details $(\mathbf{h}_i, \mathbf{b}_i, \mathbf{c}_i)$, conventional SGG then computes a combined entity representation $\mathbf{q}_i$ using another fully-connected network f_q as

$$\mathbf{q}_i = f_q\left(\mathbf{h}_i \oplus \mathbf{l}_i \oplus \mathbf{w}_i\right), \quad (3)$$

where $\mathbf{l}_i$ is a location feature based on the bounding box $\mathbf{b}_i$, $\mathbf{w}_i$ is a semantic feature based on a word embedding of its class $\mathbf{c}_i$, and $\oplus$ is the concatenation operation.

To yield a relationship proposal from entity i to j, conventional SGG [19] has an additional entity-level projection
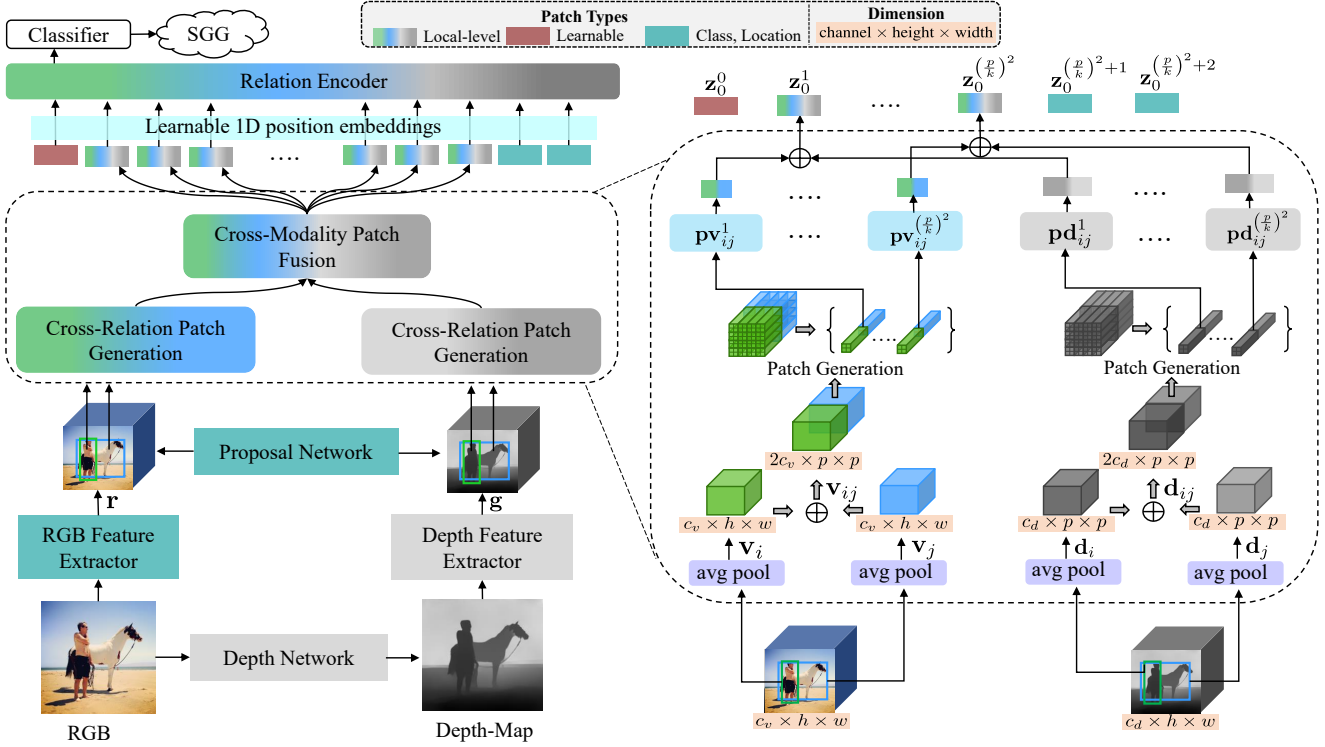
Figure 3. **VETO architecture**. An object detector yields entity proposals and entity features $\mathbf{r}$. Moreover, a depth map is estimated from the RGB input, which is also passed through the feature extractor to obtain geometric features $\mathbf{g}$. Then, for each entity pair, a sequence of local-level patches are generated, which are passed through the transformer-based relation encoder to yield a relation prediction.

(Fig. 1, panel 3), comprising convolutional features of the union region of entity bounding boxes $\mathbf{b}_i$ and $\mathbf{b}_j$, denoted as $\mathbf{u}_{ij}$. The predicate representation $\mathbf{p}_{i \to j}$ is then computed as $\mathbf{p}_{i \to j} = f_u\left(\mathbf{u}_{ij}\right) + f_p\left(\mathbf{q}_i \oplus \mathbf{q}_j\right)$, where $\mathbf{q}_i \oplus \mathbf{q}_j$ denotes the joint representation of entities e_i and e_j, and f_u, f_p are two fully-connected networks.

3.3. The VETO entity local-level patch generator

In contrast to the entity-level patch generator of conventional SGG, the *local-level entity patch generator* of VETO can inculcate *local-level* and geometric cues of entities for relation prediction to learn richer visual representations and reduce crucial information loss during relation prediction. It consists of a two-stage local-level entity patch generator followed by a transformer-based relation encoder and a fully-connected relation decoder. In particular, our network replaces the parameter heavy and computationally expensive fully-connected layers in the conventional relation network resulting from a global entity-level projection (Eq. 2) with less expensive local-level entity projections.

For a given image $\mathbf{I}$ and its depth-map, we extract the RGB features $\mathbf{r}$ with c_v channels of size $w \times h$ and the geometric features $\mathbf{g}$ with c_d channels and the same size. Our relation network starts with the patch generation and fusion modules, which we call Cross-Relation Patch Generation

(CRPG) and Cross-Modality Patch Fusion (CMPF).

Cross-Relation Patch Generation module. In order to strengthen our transformer-based relation encoder with *local-level* entity feature dependencies, we introduce a Cross-Relation Patch Generation module (Fig. 3). It generates combined subject-object local-level patches.

We preserve the local-level entity information by dividing the RGB features $\mathbf{r} \in \mathbb{R}^{c_v \times h \times w}$ and geometric features $\mathbf{g} \in \mathbb{R}^{c_d \times h \times w}$ into $p \times p$ blocks and then average pooling $\mathbf{r}$ and $\mathbf{g}$ features within each block to summarize the average presence of the features that are crucial for relation prediction. Now for a given subject-object entity pair (e_i, e_j) the resultant pooled RGB features $\mathbf{v} \in \mathbb{R}^{c_v \times p \times p}$ from both entities are fused channel-wise (Relation Fusion) as

$$\mathbf{v}_{ij} = C\left(\mathbf{v}_i, \mathbf{v}_j\right) \in \mathbb{R}^{2c_v \times p \times p}, \quad \mathbf{v} = \text{pool}\left(\mathbf{r}\right), \quad (4)$$

where $C\left(\cdot\right)$ denotes concatenation along the channel dimension and pool refers to average pooling. We then split $\mathbf{v}_{ij}$ spatially into sequential patches as

$$\mathbf{v}_{ij}^{\text{pa}} = \left\{\mathbf{v}_{ij}^{\ell} \in \mathbb{R}^{2c_v \times k \times k} \mid \ell = 1, \ldots, (p/k)^2\right\}, \quad (5)$$

where p denotes the pooled width and height of $\mathbf{v}$, $\mathbf{v}^{\ell}$ refers to the ℓth patch, and k is the patch size (in terms of blocks). Thus, the CRPG module produces $(p/k)^2$ patches for $\mathbf{v}_{ij}$.

Similar to $\mathbf{v}_{ij}$, we obtain combined depth features $\mathbf{d}_{ij} \in \mathbb{R}^{2c_d \times p \times p}$ and depth patches $\mathbf{d}_{ij}^{\text{pa}} \in \mathbb{R}^{2c_d \times k \times k}$ with depth patch size $2c_d \times k \times k$ by repeating Eqs. (4) and (5) for the pooled geometric features $\mathbf{d} \in \mathbb{R}^{c_d \times p \times p}$.

Cross-Modality Patch Fusion module. In order to reduce the model parameters and computational expense further and to strengthen the encoder with additional modality cues, we introduce a Cross-Modality Patch Fusion module. It first projects the $\mathbf{v}_{ij}^{\text{pa}}$ and $\mathbf{d}_{ij}^{\text{pa}}$ from the CRPG module to a lower dimensionality:

$$\mathbf{p}\mathbf{x}_{ij}^{\text{pa}} = f_x\left(\mathbf{x}_{ij}^{\text{pa}}, p^{\mathbf{x}}\right), \quad \mathbf{x} \in \{\mathbf{v}, \mathbf{d}\}. \quad (6)$$

The resulting $\mathbf{pv}_{ij}^{\text{pa}}$ and $\mathbf{pd}_{ij}^{\text{pa}}$ denote the locally projected entity patches for RGB and depth features, f_x in Eq. (6) is a fully-connected network, which should be understood as $f_x(\mathbf{y}, M) = \mathbf{W}\mathbf{y}$, where $\mathbf{y} \in \mathbb{R}^N$ and $\mathbf{W} \in \mathbb{R}^{M \times N}$.

Then we fuse the corresponding patches $\mathbf{v}$ and $\mathbf{d}$ of both modalities to capture their dependencies while also further reducing the parameters and computational complexity by reducing the length of the token sequence from $2\left(p/k\right)^2$ to $\left(p/k\right)^2$. This ensures that dependent modality information is closely knit to be efficiently exploited by the subsequent relation encoder:

$$\mathbf{z}_0^{\text{pa}} = \left\{ \left(\mathbf{pv}_{ij}^\ell \oplus \mathbf{pd}_{ij}^\ell\right) \Big| \ell = 1, \ldots, \left(p/k\right)^2 \right\}, \quad (7)$$

where $\mathbf{z}_0^{\text{pa}}$ represents the patch-based input embedding tokens to the first layer of our transformer-based relation encoder and $\oplus$ denotes the concatenation operation.

Overall, the local-level entity projections in VETO enable capturing the crucial local-level cues that global-level entity projection may overlook while simultaneously reducing the overall number of parameters.

Additional cues. Unlike conventional SGG in which location features $\mathbf{l}$ and semantic features $\mathbf{w}$ are fused with RGB features to form the entity representation, *cf*. Eq. (3), we fuse them separately for each subject-object pair and use them as additional input tokens to our relation encoder:

$$\mathbf{z}_0^{(p/k)^2+1} = h\left(f_l\left(\mathbf{l}_i \oplus \mathbf{l}_j, p^v + p^d\right)\right) \quad (8)$$

$$\mathbf{z}_0^{(p/k)^2+2} = h\left(f_w\left(\mathbf{w}_i \oplus \mathbf{w}_j, p^v + p^d\right)\right), \quad (9)$$

where $f_l(\cdot)$ and $f_w(\cdot)$ are fully-connected networks and $h(\cdot)$ is a non-linear function.

Relation encoder. We use a transformer-based relation encoder architecture. The key to successfully capturing the relationship between subject-object pairs in SGG is to adequately express the subject-object joint features that account for their intricate relationship. The Multi-head Self Attention (MSA) component of the transformer can jointly attend to diverse features by enabling each head to focus on a different subspace with distinct semantic or syntactic meanings. Hence, we propose to capture the subject-object inductive bias in SGG with a MSA component

by feeding it embedding tokens $\mathbf{z}_0^{\text{pa}}$ enriched with local-level subject-object information. MSA can be formalized as $\text{MSA}(Q, K, V) = \text{Concat}\left(\text{SA}_1, \text{SA}_2, \ldots, \text{SA}_h\right) W^O$ where $\text{SA}_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}}\right) V_i$ denotes self-attention, Q, K, and V refer to query, key, and value, and W^O are trainable parameters. The input token to the transformer is split into multiple heads to be attended to in parallel.

We prepend the entity local-level information enriched input patches $\mathbf{z}_0^{\text{pa}}$ from the CMPF module with a learnable class token $\mathbf{z}_0^{\text{cl}}$. Additionally, a learnable 1D positional embedding $\mathbf{z}^{\text{pos}}$ is added to each token to preserve the subject-object positional information. The resultant sequence is passed as input to the encoder. The transformer consists of L encoder layers, each with MSA and MLP blocks. Each encoder layer also contains a LayerNorm (LN) before every block and residual connections after every block:

$$\mathbf{z}_0 = \left\{\mathbf{z}_0^{\text{cl}}; \mathbf{z}_0^{\text{pa}}; \mathbf{z}_0^{(p/k)^2+1}; \mathbf{z}_0^{(p/k)^2+2}\right\} + \mathbf{z}^{\text{pos}} \quad (10)$$

$$\mathbf{z}_\ell' = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \ldots L \quad (11)$$

$$\mathbf{z}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{z}_\ell'\right)\right) + \mathbf{z}_\ell', \quad \ell = 1 \ldots L \quad (12)$$

$$\mathbf{y} = \text{LN}\left(\mathbf{z}_L^0\right), \quad (13)$$

where $\mathbf{z}^{\text{pos}} \in \mathbb{R}^{((p/k)^2+3) \times (p^v + p^d)}$, and L denotes the number of total encoder layers. Each encoder layer contains MSA, MLP, and LN blocks as well as residual connections. Finally, $\mathbf{y}$ is linearly projected to the total number of predicate relations, forming the relation classification head.

4. MEET: Mutually Exclusive ExperT learning

As pointed out by Dong *et al.* [7], a single classifier cannot achieve satisfactory performance on all the predicate classes due to the extreme unbalance of SGG datasets. To tackle this issue, they propose to deploy multiple classifiers. However, their classifiers are not mutually exclusive and result in a significant deterioration of head class performance ("GCL" in Fig. 2b), as they distill knowledge from head to tail classifiers. We hypothesise that learning mutually exclusive experts, each responsible for a predicate subgroup, can reduce model bias towards a specific set of predicates.

Therefore, we propose Mutually Exclusive-ExperT learning (MEET). MEET splits the predicate classes into balanced groups based on the predicate frequency in the training set. First, we sort the predicate classes according to their frequency in descending order into $\mathcal{P}_{\text{sort}} = \{p_i\}_{i=1}^M$. The sorted set is split into G predicate groups $\{\mathcal{P}_{\text{sort}}^g\}_{g=1}^G$. We use the same class split as [7], yet in contrast, MEET deploys mutually exclusive classifier experts $\{\mathbf{E}^g\}_{g=1}^G$ responsible for classification within each predicate group $\mathcal{P}_{\text{sort}}^g$.

Unfortunately, training each expert exclusively on a subgroup of predicates $\mathcal{P}_{\text{sort}}^g$ can be challenged during the evaluation stage with samples out of its classification space, resulting in uncertain predictions. To overcome this issue,

Algorithm 1 MEET: Mutually Exclusive ExperT learning

1: Input: predicate classes $\mathcal{P} = \{p_i\}_{i=1}^M$, experts $\mathbf{E} = \{\mathbf{E}\}_{g=1}^G$, where G denotes the total number of experts
2: $\mathcal{P}_{\text{sort}} \leftarrow$ sorted predicate set $\mathcal{P}$
3: $\mathcal{P}_{\text{sort}}^g \leftarrow g^{\text{th}}$ sorted predicate group of $\mathcal{P}_{\text{sort}}$
4: $\mathcal{S}^g \leftarrow$ relation representation sample $\mathbf{z}_L^0$ of gth group
5: $f_s(x, y) = \max\left(\min\left(x/y, 1.0\right), 0.01\right)$
6: **for** $g = 1$ to G **do**
7: $\quad Index \leftarrow$ index of the central predicate in $\mathcal{P}_{\text{sort}}^g$
8: $\quad centre \leftarrow \text{Freq}(\mathcal{P}_{\text{sort}}^g(Index))$
9: $\quad I_{\text{dist}}^g = \{f_s\left(centre, \text{Freq}(p)\right) \mid \forall p \in \mathcal{P}_{\text{sort}}^g\}$
10: $\quad O_{\text{dist}}^g = \{f_s\left(centre, \text{Freq}(p)\right) \mid \forall p \notin \mathcal{P}_{\text{sort}}^g\}$
11: $\quad \mathbf{y}^g(p) = \begin{cases} \text{Index}(p) \text{ in } \mathcal{P}_{\text{sort}}^g, & \text{for } p \in \mathcal{P}_{\text{sort}}^g \\ |\mathcal{P}_{\text{sort}}^g| + 1, & \text{for } p \notin \mathcal{P}_{\text{sort}}^g \end{cases}$
12: $\quad$ Sample from distribution: $\mathcal{S}_{\text{in}}^g \sim I_{\text{dist}}^g, \mathcal{S}_{\text{out}}^g \sim O_{\text{dist}}^g$
13: $\quad \mathbf{w}^g = \mathbf{E}^g\left(\{\mathcal{S}_{\text{in}}^g; \mathcal{S}_{\text{out}}^g\}\right)$ $\qquad \triangleright$ expert output
14: **end for**
15: $\mathcal{L} = \sum_{g=1}^G \mathcal{L}_{CE}\left(\mathbf{w}^g, \mathbf{y}^g\right)$ $\qquad \triangleright$ multi-expert loss
16: **Evaluation stage:**
17: $\hat{\mathbf{w}}^g = \left[\mathbf{w}_i^g\right]_{i=1}^{|\mathcal{P}_{\text{sort}}^g|}$ $\qquad \triangleright$ discard OOD predictions
18: $\mathcal{R}_{\text{conf}} = \{\max_i \hat{\mathbf{w}}_i^g\}_{g=1}^G$
19: $\mathcal{R}_{\text{label}} = \left\{\mathcal{P}_{\text{sort}}\left(\arg\max_i \hat{\mathbf{w}}_i^g + \sum_{j=0}^{g-1} \left|\mathcal{P}_{\text{sort}}^j\right|\right)\right\}_{g=1}^G$

---

we train out-of-distribution aware experts. We summarize MEET in Algorithm 1. **(Lines 9–10)**: During training, we adjust the in-distribution I_{dist}^g and out-of-distribution O_{dist}^g sampling frequency within each expert to prevent the experts from being overwhelmed with OOD samples. $\text{Freq}(\cdot)$ denotes the frequency count of a predicate class. **(Line 11)**: We re-map the relation labels to accommodate an out-of-distribution pseudo-label for every expert group. **(Lines 12, 13, 15)**: For a given image $\mathbf{I}$, each expert $\mathbf{E}^g$ is trained on the in-distribution and out-of-distribution samples $\mathcal{S}_{\text{in}}^g$ and $\mathcal{S}_{\text{out}}^g$, respectively. **(Lines 17–19)**: During the evaluation stage, we discard the OOD predictions from each group and map the prediction labels back to the original labels.

5. Experimental Evaluation

We aim to answer the following questions: **(Q1)** Does VETO + MEET improve the SOTA in unbiased SGG? **(Q2)** What is the impact of MEET on other SGG methods? **(Q3)** Does local-level projection reduce the model size? **(Q4)** Does SGG benefit from local-level patch generation? **(Q5)** Does a depth map improve SGG performance?

5.1. Experimental setup

Dataset. We evaluate our approach on two common SGG datasets: Visual Genome (VG) [17] and GQA [12]. For VG we adopt the popular VG150 split, which consists

of 150 object classes and 50 predicate classes in line with previous SGG work [3, 19, 22, 29–31, 40, 46, 49]. For GQA we adopt the GQA200 split used by Dong *et al.* [7]. For both datasets, depth maps are generated using the monocular depth estimator of Yin *et al.* [45].

Evaluation protocol. We evaluate our model on the most common SGG tasks [40, 49]: *(1)* Predicate Classification (PredCls) predicts the relationships for all object pairs by using both the ground-truth bounding boxes and classes; *(2)* Scene Graph Classification (SGCls) predicts both the object classes and their pairwise relationships by using ground-truth bounding boxes; *(3)* Scene Graph Detection (SGDet) detects, classifies, and predicts the pairwise relationships for all the objects in an image.

Evaluation metrics. Following previous work [7, 19], we use Recall@k (R@k) and mean Recall@k (mR@k) as our evaluation metrics. We also report the Average of recall and mean recall (A@k) to show the combined performance improvement of R@k and mR@k. The A@k metric is relevant because previous models with improved mR@k have lower R@k and vice-versa (*cf*. Fig. 2b).

Implementation details. We implement VETO and MEET in PyTorch on Nvidia A100 GPUs. Following prior work [18, 30], we adopt a ResNeXt-101-FPN [38] backbone and a Faster R-CNN [26] object detector. The parameters of backbone and detector are kept frozen. VETO contains 6 relation encoder layers with 6 attention heads for each MSA [8] component, uses embeddings of size 576, a patch size of 2, and a pooled entity resolution of 8 for entity patch generation. For VETO + Rwt, we use the importance weighting of [48]. We train our model using the Adam optimizer [14], batch size 12, and an initial learning rate of 1.2×10^{-3}. We apply a linear learning rate warmup over the first 3K iterations and train for 125K iterations using a learning rate decay with maximum decay step 3 and patience 2.

5.2. Experimental results

Using this protocol we are now able to address **Qs. 1–5**.

(Q1) Comparison with state of the art. As shown in Tabs. 1 and 2, our VETO + MEET model fulfills the fundamental requirement of unbiased SGG, *i.e.* it improves on both R@k and mR@k metrics, yielding state-of-the-art-performance in terms of the balanced A@k metric across *all* tasks for *both* datasets (except for SGDet in VG where we are comparable). The heat-map pattern reveals that previous models with high mR@k, *e.g.*, SHA + GCL [7], gain performance improvements on the under-represented predicates while losing significantly on the more frequent ones as revealed by the lower R@k. Our VETO model with a simple re-weighting technique (VETO + Rwt, Tab. 1) already outperforms leading baselines without notably reducing the R@k metric. This is exemplified by the A@k met-

Table 1. **Recall (R), mean Recall (mR), and their average (A) on VG** (the higher, the better). Colors in the table vary from blue to green to depict the performance improvement. '+' denotes the combination with a model-agnostic unbiasing strategy. Double citations refer to the original model and its reproduced variant on a ResNeXt-101-FPN backbone. The superscript '†' denotes the method uses Faster-RCNN with VGG-16 as the object detector.

Model	PredCls R@k: 50/100	mR@k: 50/100	A@k: 50/100	SGCls R@k: 50/100	mR@k: 50/100	A@k: 50/100	SGDet R@k: 50/100	mR@k: 50/100	A@k: 50/100
IMP [7,29]	61.1 / 63.1	11.0 / 11.8	36.1 / 37.4	37.4 / 38.3	6.4 / 6.7	21.9 / 22.5	23.6 / 28.7	3.3 / 4.1	13.5 / 16.4
KERN† [3]	65.8 / 67.6	17.7 / 19.2	41.8 / 43.4	36.7 / 37.4	9.4 / 10.0	23.1 / 23.7	27.1 / 29.8	6.4 / 7.3	16.8 / 18.6
GB-Net + Rwt† [48]	66.6 / 68.2	22.1 / 24.0	44.4 / 46.1	37.3 / 38.0	12.7 / 13.4	25.0 / 25.7	26.3 / 29.9	7.1 / 8.5	16.7 / 18.5
DT2-ACBS [6]	23.3 / 25.6	35.9 / 39.7	29.6 / 32.7	16.2 / 17.6	24.8 / 27.5	20.5 / 22.6	15.0 / 16.3	22.0 / 24.0	18.5 / 20.2
PCPL† [41]	50.8 / 52.6	35.2 / 37.8	43.0 / 45.2	27.6 / 28.4	18.6 / 19.6	23.1 / 24.0	14.6 / 18.6	9.5 / 11.7	12.1 / 15.2
GPS-Net [7,22]	65.2 / 67.1	15.2 / 16.6	40.2 / 41.9	37.8 / 39.2	8.5 / 9.1	23.2 / 24.2	31.1 / 35.9	6.7 / 8.6	18.9 / 22.2
SG-CogTree [46]	38.4 / 39.7	28.4 / 31.0	33.4 / 35.3	22.9 / 23.4	15.7 / 16.7	19.3 / 20.1	19.5 / 21.7	11.1 / 12.7	16.8 / 17.2
BGNN [19]	59.2 / 61.3	30.4 / 32.9	44.8 / 47.1	37.4 / 38.5	14.3 / 16.5	25.9 / 27.5	31.0 / 35.8	10.7 / 12.6	20.9 / 24.2
VTransE [30,51]	65.7 / 67.6	14.7 / 15.8	40.2 / 41.7	38.6 / 39.4	8.2 / 8.7	23.4 / 24.1	29.7 / 34.3	5.0 / 6.0	17.3 / 20.2
VTransE + TDE [30]	43.1 / 48.5	24.6 / 28.0	33.9 / 38.3	25.7 / 28.5	12.9 / 14.8	19.3 / 21.7	18.7 / 22.6	8.6 / 10.5	13.7 / 16.7
VTransE + GCL [7]	35.4 / 37.3	34.2 / 36.3	34.8 / 36.8	25.8 / 26.9	20.5 / 21.2	22.8 / 23.7	14.6 / 17.1	13.6 / 15.5	14.1 / 16.3
VTransE + MEET (ours)	58.3 / 64.9	18.3 / 24.9	38.3 / 44.9	35.8 / 39.1	12.8 / 16.7	24.3 / 27.9	22.0 / 27.6	5.8 / 7.6	13.9 / 17.6
Motifs [30,49]	65.2 / 67.0	14.8 / 16.1	40.0 / 41.6	38.9 / 39.8	8.3 / 8.8	23.6 / 24.3	32.8 / 37.2	6.8 / 7.9	19.8 / 22.6
Motifs + Rwt [4]	54.7 / 56.5	17.3 / 18.6	36.0 / 37.6	29.5 / 31.5	11.2 / 11.7	20.4 / 21.6	24.4 / 29.3	9.2 / 10.9	16.8 / 20.1
Motifs + TDE [30]	46.2 / 51.4	25.5 / 29.1	35.9 / 40.3	27.7 / 29.9	13.1 / 14.9	20.4 / 22.4	16.9 / 20.3	8.2 / 9.8	12.5 / 15.1
Motifs + PCPL [7,41]	54.7 / 56.5	17.3 / 18.6	36.0 / 37.6	29.5 / 31.5	11.2 / 11.7	20.4 / 21.6	24.4 / 29.3	9.2 / 10.9	16.8 / 20.1
Motifs + CogTree [46]	35.6 / 36.8	26.4 / 29.0	31.0 / 32.9	21.6 / 22.2	14.9 / 16.1	18.3 / 19.2	20.0 / 22.1	10.4 / 11.8	15.2 / 17.0
Motifs + DLFE [4]	52.5 / 54.2	26.9 / 28.8	39.7 / 41.5	32.3 / 33.1	15.2 / 15.9	23.8 / 24.5	25.4 / 29.4	11.7 / 13.8	18.6 / 21.6
Motifs + EMB [29]	65.2 / 67.3	18.0 / 19.5	41.6 / 43.4	39.2 / 40.0	10.2 / 11.0	24.7 / 25.5	31.7 / 36.3	7.7 / 9.3	19.7 / 22.8
Motifs + GCL [7]	42.7 / 44.4	36.1 / 38.2	39.4 / 41.3	26.1 / 27.1	20.8 / 21.8	23.5 / 24.5	18.4 / 22.0	16.8 / 19.3	17.6 / 20.7
Motifs + IETrans + Rwt [50]	48.6 / 50.5	35.8 / 39.1	42.2 / 44.8	29.4 / 30.2	21.5 / 22.8	25.5 / 26.5	23.5 / 27.2	15.5 / 18.0	19.5 / 22.6
Motifs + **MEET** (ours)	67.4 / 72.7	25.3 / 33.5	46.4 / 53.1	40.5 / 43.2	19.0 / 23.7	29.8 / 33.5	27.9 / 33.3	8.5 / 11.8	18.2 / 22.6
VCTree [30,31]	65.4 / 67.2	16.7 / 18.2	41.1 / 42.7	46.7 / 47.6	11.8 / 12.5	29.3 / 30.1	31.9 / 36.2	7.4 / 8.7	19.7 / 22.5
VCTree + Rwt [4]	60.7 / 62.6	19.4 / 20.4	40.1 / 41.5	42.3 / 43.5	12.5 / 13.1	27.4 / 28.3	27.8 / 32.0	8.7 / 10.1	18.3 / 21.1
VCTree + TDE [30]	47.2 / 51.6	25.4 / 28.7	36.3 / 40.2	25.4 / 27.9	12.2 / 14.0	18.8 / 21.0	19.4 / 23.2	9.3 / 11.1	14.5 / 17.2
VCTree + PCPL [7,41]	56.9 / 58.7	22.8 / 24.5	39.9 / 41.6	40.6 / 41.7	15.2 / 16.1	27.9 / 28.9	26.6 / 30.3	10.8 / 12.6	18.4 / 21.5
VCTree + CogTree [46]	44.0 / 45.4	27.6 / 29.7	35.8 / 37.6	30.9 / 31.7	18.8 / 19.9	24.9 / 25.8	18.2 / 20.4	10.4 / 12.1	14.3 / 16.3
VCTree + DLFE [4]	51.8 / 53.5	25.3 / 27.1	38.6 / 40.3	33.5 / 34.6	18.9 / 20.0	26.2 / 27.3	22.7 / 26.3	11.8 / 13.8	17.5 / 20.1
VCTree + EMB [29]	64.0 / 65.8	18.2 / 19.7	41.1 / 42.8	44.7 / 45.8	12.5 / 13.5	28.6 / 30.0	31.4 / 35.9	7.7 / 9.1	19.5 / 22.5
VCTree + GCL [7]	40.7 / 42.7	37.1 / 39.1	38.9 / 40.1	27.7 / 28.7	22.5 / 23.5	25.1 / 26.1	17.4 / 20.7	15.2 / 17.5	16.3 / 19.1
VCTree + IETrans + Rwt [4]	48.0 / 49.9	37.0 / 39.7	42.5 / 43.5	30.0 / 30.9	19.9 / 21.8	25.0 / 26.4	23.6 / 27.8	12.0 / 14.9	17.8 / 21.4
VCTree + **MEET** (ours)	62.0 / 69.8	25.5 / 34.5	43.8 / 52.2	35.4 / 39.2	14.5 / 18.6	25.0 / 28.9	26.4 / 31.2	8.2 / 11.5	17.3 / 21.4
SHA [7]	64.3 / 66.4	18.8 / 20.5	41.5 / 43.5	38.0 / 39.0	10.9 / 11.6	24.5 / 25.3	30.6 / 34.9	7.8 / 9.1	19.2 / 22.0
SHA + GCL [7]	35.1 / 37.2	41.6 / 44.1	38.4 / 40.7	22.8 / 23.9	23.0 / 24.3	22.9 / 24.1	14.9 / 18.2	17.9 / 20.9	16.4 / 19.6
SHA + **MEET** (ours)	66.3 / 72.4	28.0 / 36.2	47.2 / 54.3	37.9 / 41.2	16.1 / 20.7	27.0 / 31.0	24.2 / 29.7	7.7 / 9.8	16.0 / 19.8
VETO (ours)	64.2 / 66.3	22.8 / 24.7	43.5 / 45.5	35.7 / 36.9	11.1 / 11.9	23.4 / 24.4	27.5 / 31.5	8.1 / 9.5	17.8 / 20.5
VETO (ours) + Rwt	61.9 / 63.9	33.1 / 35.1	47.5 / 49.5	35.1 / 36.3	16.1 / 17.1	25.6 / 26.7	26.2 / 30.4	10.0 / 11.7	18.1 / 21.1
VETO + MEET (ours)	74.0 / 78.9	42.0 / 52.4	58.0 / 65.7	41.1 / 44.0	22.3 / 27.4	31.7 / 35.7	28.6 / 34.0	10.6 / 13.8	19.6 / 23.9

Table 2. **Recall (R), mean Recall (mR), and their average (A) on GQA** (the higher, the better). Conventions as described in Tab. 1.

Model	PredCls R@k: 50/100	mR@k: 50/100	A@k: 50/100	SGCls R@k: 50/100	mR@k: 50/100	A@k: 50/100	SGDet R@k: 50/100	mR@k: 50/100	A@k: 50/100
VTransE [7,51]	55.7 / 57.9	14.0 / 15.0	34.9 / 36.5	33.4 / 34.2	8.1 / 8.7	20.9 / 21.5	27.2 / 30.7	5.8 / 6.6	16.5 / 18.7
VTransE + GCL [7]	35.5 / 37.4	30.4 / 32.3	33.0 / 34.9	22.9 / 23.6	16.6 / 17.4	19.8 / 20.5	15.3 / 18.0	14.7 / 16.4	15.0 / 17.2
VTransE + **MEET** (ours)	55.4 / 60.9	15.2 / 21.7	35.3 / 41.3	28.1 / 30.7	9.2 / 12.0	18.7 / 21.4	24.0 / 27.9	5.6 / 7.8	14.8 / 17.9
Motifs [7,49]	65.3 / 66.8	16.4 / 17.1	40.9 / 42.0	34.2 / 34.9	8.2 / 8.6	21.2 / 21.9	28.9 / 33.1	6.4 / 7.7	17.7 / 20.4
Motifs + GCL [7]	44.5 / 46.2	36.7 / 38.1	40.6 / 42.2	23.2 / 24.0	17.3 / 18.1	20.3 / 21.1	18.5 / 21.3	16.8 / 18.8	17.7 / 20.3
Motifs +**MEET** (ours)	63.6 / 68.4	25.0 / 30.4	44.3 / 49.4	33.7 / 36.1	17.3 / 19.9	25.5 / 28.0	26.8 / 30.8	9.3 / 12.5	18.1 / 21.7
VCTree [7,31]	63.8 / 65.7	16.6 / 17.4	40.2 / 41.6	34.1 / 34.8	7.9 / 8.3	21.0 / 21.6	28.3 / 31.9	6.5 / 7.4	17.4 / 19.7
VCTree + GCL [7]	44.8 / 46.6	35.4 / 36.7	40.1 / 41.7	23.7 / 24.5	17.3 / 18.0	20.5 / 21.3	17.6 / 20.7	15.6 / 17.8	16.6 / 19.3
VCTree + **MEET** (ours)	57.3 / 63.7	26.1 / 31.3	41.7 / 47.5	28.3 / 31.5	12.3 / 14.0	20.3 / 22.8	25.1 / 28.7	7.6 / 10.1	16.4 / 19.4
SHA [7]	63.3 / 65.2	19.5 / 21.1	41.4 / 43.2	32.7 / 33.6	8.5 / 9.0	20.6 / 21.3	25.5 / 29.1	6.6 / 7.8	16.1 / 18.5
SHA + GCL [7]	42.7 / 44.5	41.0 / 42.7	41.9 / 43.6	21.4 / 22.2	20.6 / 21.3	18.1 / 21.9	14.8 / 17.9	17.8 / 20.1	16.3 / 19.0
SHA + **MEET** (ours)	69.7 / 74.4	34.2 / 42.3	52.0 / 58.4	31.1 / 33.7	12.9 / 15.6	22.0 / 24.7	25.3 / 28.9	7.2 / 10.1	16.3 / 19.5
VETO (ours)	64.5 / 66.0	21.2 / 22.1	42.9 / 44.0	30.4 / 31.5	8.6 / 9.1	19.5 / 20.3	26.1 / 29.0	7.0 / 8.1	16.6 / 18.6
VETO + MEET (ours)	73.9 / 78.3	43.3 / 50.5	58.6 / 64.4	34.6 / 37.2	19.7 / 22.5	27.2 / 29.9	26.7 / 31.0	12.1 / 16.0	19.4 / 23.5

ric and the heat-map hue having less within-row variance than the baselines. In addition, our final model VETO + MEET outperforms the previously best Motifs + IETrans + Rwt [50] by a remarkable 47% and 48% relative improvement on A@100 for PredCls for VG and GQA, respectively. To the best of our knowledge, our VETO + MEET model is the first to attain state-of-the-art results on *both* R@k and mR@k metrics for PredCls. It also yields state-of-the-art results on mR@100 for SGCls.

(Q2) MEET with other SGG approaches. Among the models trained with MEET in Tabs. 1 and 2, Motifs + MEET and SHA + MEET show notable improvements on the A@k metric for the PredCls and SGCls tasks. However, there is a significant performance gap in comparison to VETO + MEET on all the metrics and tasks. This shows the unbiasing capabilities of MEET as well as the significance of VETO in reducing information loss by using local-level information, resulting in improved SGG performance.

(Q3) Light-weight VETO. The comparison of SGG models in terms of the number of trainable parameters in Fig. 4 shows how our local-level entity projections significantly reduce the number of parameters compared to global
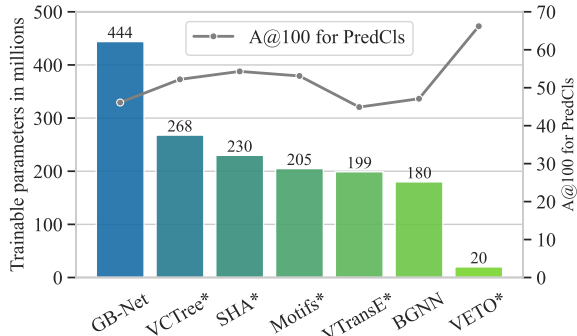
Figure 4. **No. of trainable parameters** (Mio.) *vs.* **performance** (A@100) of leading SGG models (* denotes debiasing with MEET). VETO outperforms prior work and is $\sim 10\times$ leaner.

Table 3. **Ablation study of VETO on VG**. **L**: Local-level Entity Patch Generation; **CR**: Cross-Relation Patch Generation; **CM**: Cross-Modality Patch Fusion; **D**: Depth.

VETO Components				SGCls		
L	CR	CM	D	R@k: 50 / 100	mR@k: 50 / 100	A@k: 50 / 100
✗	✗	✗	✗	32.0 / 33.5	7.6 / 8.3	19.8 / 20.9
✗	✗	✗	✓	33.2 / 34.5	7.0 / 7.6	20.1 / 21.1
✓	✗	✗	✗	34.8 / 36.2	14.1 / 15.1	24.5 / 25.7
✓	✗	✗	✓	35.1 / 36.4	13.0 / 14.1	24.1 / 25.3
✓	✗	✓	✓	35.1 / 36.3	14.4 / 15.4	24.8 / 25.9
✓	✓	✗	✗	35.4 / 36.6	15.2 / 16.1	25.3 / 26.4
✓	✓	✓	✓	35.1 / 36.3	16.1 / 17.1	25.6 / 26.7

entity-level projections (*cf*. Fig. 1). VETO with 20 million parameters is 20 times lighter than GB-Net [48], which uses knowledge graphs as additional modality, and ~10 times lighter than other leading SGG models. Despite this, VETO clearly outperforms previous models in A@k.

(Q4) Benefit of local-level patch generation. Tab. 3 provides an ablation study of the VETO components. The first two rows denote a transformer-based SGG model without local-level patches. We observe that introducing the local-level patch generation (rows 3 & 4) notably improves every metric with a ~23% improvement of A@k, highlighting the significance of local-level patches. We also observe an overall improvement when incrementally adding the VETO components in the subsequent rows. In general, the use of local-level information and the Cross-Relation Patch Generation (2nd to last row) significantly improves performance, with a relative improvement of approximately 28% in A@k compared to the first ablation that does not use local-level patches. We also observe that adding the depth map components to the final model yields an additional small improvement of the mR@k metric.

(Q5) Benefit of depth map. Comparing rows 1 and 2 of Tab. 3 shows that introducing the depth modality to the model without local-level patches yields only a slight R@k improvement while mR@k drops. We observe a similar trend when comparing rows 3 and 4. However, after introducing the Cross-Modality Patch Fusion module, mR@k and A@k improve. We also perform an exten-

Table 4. **Impact of depth map quality** (the higher, the better). For fairer comparison, the Depth-VRD model [27] is reproduced on the ResNeXt-101-FPN backbone. Both models are also debiased using the reweighting strategy of [48].

Model	mR@k	VG-Depth.v1	VG-Depth.v2	Improvement
VETO	20	25.3	27.5	**9%**
	50	31.2	33.1	**6%**
	100	33.5	35.1	**5%**
Depth-VRD [27]	20	17.8	18.2	2%
	50	21.7	21.9	1%
	100	23.1	23.1	0%

Table 5. **Improvement of VETO over Depth-VRD for PredCls.** Conventions as described in Tab. 4.

Depth Map	mR@20	mR@50	mR@100
VG-Depth.v1	42%	44%	45%
VG-Depth.v2	51%	51%	52%

sive depth data analysis to investigate the modality fusion potential of VETO. Fig. 5 shows "noisy" depth-map samples used by Sharifzadeh *et al.* for Depth-VRD [27] (VG-Depth.v1); the bottom row shows the corresponding high-quality depth maps as extracted by the monocular depth estimator of Yin *et al.* [45] (VG-Depth.v2). We use VG-Depth.v1 and VG-Depth.v2 to compare and contrast VETO and Depth-VRD. We analyse the significance of the depth-map quality on the SGG performance and the importance of a careful architectural design to make use of the depth map efficiently. As depicted in Tab. 4, the performance improvement for our model on VG-Depth.v2 generated using the monocular depth estimator of Yin *et al.* [45] over VG-Depth.v1 [27] is around 7%. To the contrary, Depth-VRD shows only a minor improvement of 1% on the high-quality VG-Depth.v2 dataset. Furthermore, the improvement of VETO over Depth-VRD in Tab. 5 shows that, overall, VETO has a significant improvement of more than 40% respectively 50% in mR@k over Depth-VRD for the VG-Depth.v1 respectively VG-Depth.v2 depth maps.

5.2.1 Further results

Fig. 6 shows the predicate-specific improvement of VETO + MEET over SHA + GCL (sorted from frequent to less frequent). Notice that VETO + MEET improves on every part of the distribution (head, body, and tail). As emphasized in Fig. 2a, we find an enormous performance boost over SHA + GCL [7] for relations that can be enhanced with local-level information, *e.g.*, *attached to* (781% improvement) and *part of* (441% improvement). This once again highlights the efficacy of VETO + MEET.

Fig. 7 shows an illustrative example for the challenges of current SGG models, *e.g.*, SHA + GCL [7] overfitting to the tail classes after debiasing (panel 4) or Motifs [49] overfitting to the head classes (panel 5). The generated SG from

Figure 5. Failure cases reported by [27]. The second row shows "noisy" depth maps from [27] (VG-Depth.v1). The bottom row represents the improved depth maps used in VETO (VG-Depth.v2), generated using the monocular depth estimator of [45].
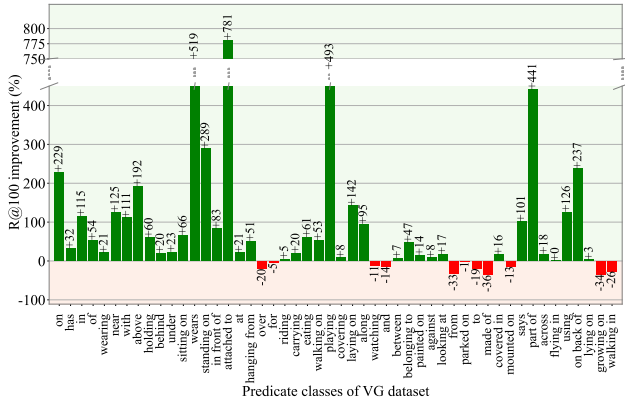

Figure 6. **R@100 improvement on PredCls for VETO + MEET over SHA + GCL [7].** The predicates are sorted based on their frequency in descending order.
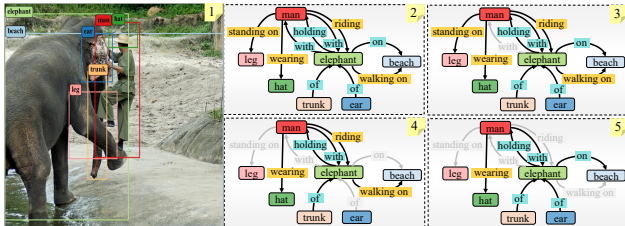

Figure 7. **Qualitative example**. Head and tail relations are highlighted in blue and yellow, respectively. Greyed out relations and arrows denote missed predictions. *(1)* VG sample with ground-truth bounding boxes and labels; *(2)* SGG ground-truth; *(3)* VETO + MEET predicts both head and tail classes; *(4)* SHA + GCL [7] misses head classes such as *on*, *of*; *(5)* Motifs [49] misses many tail classes such as *walking on*, *riding*, or *standing on*.

VETO + MEET (panel 3) attains a better balance between the head and tail predictions.

5.2.2 SGDet sensitivity analysis

The experimental results obtained from the VG dataset in Tab. 1 indicate that VETO's SGDet performance is slightly lower compared to the baselines. To explore whether our

Table 6. **SGDet sensitivity analysis:** Motifs *vs*. VETO.

Detector	mAP drop (%)	A@50 drop (%)		A@100 drop (%)	
		Motifs	VETO	Motifs	VETO
OD1	13	3.3	5.7	3.5	4.2
OD2	32	28.0	29.5	27.0	28.0

model's performance is affected by the object detector's accuracy, we conducted a sensitivity analysis. Tab. 6 displays the results of using weaker Object Detectors (OD1 & 2) with 13% and 32% lower mAP, respectively. The resulting drop in A@k reveals that VETO is indeed slightly more sensitive to the detector accuracy. However, it is worth noting that despite this sensitivity, our lightweight VETO outperforms state-of-the-art (SOTA) methods on the A@k metric in 2 out of 3 tasks on VG and 3 out of 3 tasks on GQA. Furthermore, VETO's performance in SGDet (VG) is comparable to the SOTA methods.

6. Conclusion

We have identified three primary concerns with current SGG models: a loss of local-level information, excessive parameter usage, and biased relation predictions. To address these issues, we introduce the Vision Relation Transformer (VETO) and the Mutually Exclusive Expert Learning (MEET) methods. In most of the cases, our approach achieves superior performance on both biased and unbiased evaluation metrics. Some interesting avenues for future work include improving the contrasting power of multi-experts and reducing the label dependency of experts.

Acknowledgement

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 213–229. Springer, 2020.

[3] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019.

[4] Meng-Jiun Chiou, Henghui Ding, Hanshu Yan, Changhu Wang, Roger Zimmermann, and Jiashi Feng. Recovering the unbiased scene graphs from the biased ones. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1581–1590, 2021.

[5] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3076–3086, 2017.

[6] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15404–15413, 2021.

[7] Xingning Dong, Tian Gan, Xuemeng Song, Jianlong Wu, Yuan Cheng, and Liqiang Nie. Stacked hybrid-attention and group collaborative learning for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19427–19436, 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2021.

[9] Chen Gao, Yuliang Zou, and Jia-Bin Huang. Ican: Instance-centric attention network for human-object interaction detection. In *British Machine Vision Conference*, 2018.

[10] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2019.

[11] MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, 51(6):1–36, 2019.

[12] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019.

[13] ASM Iftekhar, Satish Kumar, R Austin McEver, Suya You, and BS Manjunath. Gtnet: Guided transformer network for detecting human-object interactions. In *Pattern Recognition and Tracking XXXIV*, volume 12527, pages 192–205. SPIE, 2023.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

[15] Rajat Koner, Suprosanna Shit, and Volker Tresp. Relation transformer network. *arXiv preprint arXiv:2004.06193*, 2020.

[16] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6867–6876, 2018.

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, et al. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[18] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[19] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[20] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1261–1270, 2017.

[21] Wentong Liao, Bodo Rosenhahn, Ling Shuai, and Michael Ying Yang. Natural language guided visual relationship detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[22] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. GPS-Net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020.

[23] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 852–869, 2016.

[24] Yichao Lu, Himanshu Rai, Jason Chang, Boris Knyazev, Guangwei Yu, Shashank Shekhar, Graham W. Taylor, and Maksims Volkovs. Context-aware scene graph generation with Seq2Seq transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15931–15941, 2021.

[25] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[27] Sahand Sharifzadeh, Sina Moayed Baharlou, Max Berrendorf, Rajat Koner, and Volker Tresp. Improving visual relation detection using depth maps. In *Proceedings of the 25th International Conference on Pattern Recognition*, pages 3597–3604, 2021.

[28] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8376–8384, 2019.

[29] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13936–13945, 2021.

[30] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3716–3725, 2020.

[31] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 6619–6628, 2019.

[32] Oytun Ulutan, ASM Iftekhar, and Bangalore S Manjunath. Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13617–13626, 2020.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018.

[35] Tzu-Jui Julius Wang, Selen Pehlivan, and Jorma Laaksonen. Tackling the unannotated: Scene graph generation with bias-reduced models. *Proceedings of the British Machine Vision Conference*, 2020.

[36] Wenbin Wang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exploring context and visual pattern of relationship for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2019.

[37] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. LinkNet: Relational embedding for scene graph. *Advances in Neural Information Processing Systems*, 31, 2018.

[38] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[39] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and Mohan S Kankanhalli. Learning to detect human-object interactions with knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[40] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5419, 2017.

[41] Shaotian Yan, Chen Shen, Zhongming Jin, Jianqiang Huang, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. PCPL: Predicate-correlation perception learning for unbiased scene graph generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 265–273, 2020.

[42] Hsuan-Kung Yang, An-Chieh Cheng, Kuan-Wei Ho, Tsu-Jui Fu, and Chun-Yi Lee. Visual relationship prediction via label clustering and incorporation of depth information. In *Proceedings of the European Conference on Computer Vision Workshops*, volume 2, pages 571–581, 2018.

[43] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *Proceedings of the European Conference on Computer Vision*, volume 1, pages 670–685, 2018.

[44] Guojun Yin, Lu Sheng, Bin Liu, Nenghai Yu, Xiaogang Wang, Jing Shao, and Chen Change Loy. Zoom-Net: Mining deep feature interactions for visual relationship recognition. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 322–338, 2018.

[45] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 204–213, 2021.

[46] Jing Yu, Yuan Chai, Yujing Wang, Yue Hu, and Qi Wu. Cogtree: Cognition tree loss for unbiased scene graph generation. *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 2021.

[47] Ruichi Yu, Ang Li, Vlad I. Morariu, and Larry S. Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1974–1982, 2017.

[48] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *Proceedings of the European Conference on Computer Vision*, volume 3, pages 606–623, 2020.

[49] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5831–5840, 2018.

[50] Ao Zhang, Yuan Yao, Qianyu Chen, Wei Ji, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. Fine-grained scene graph generation with data transfer. In *Proceedings of the Euro-*

pean Conference on Computer Vision, volume 2, pages 409–424. Springer, 2022.

[51] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5532–5540, 2017.