Generative Compositional Augmentations for Scene Graph Prediction

Boris Knyazev[*,1,2] Harm de Vries[3] Cătălina Cangea[4]
Graham W. Taylor[1,2] Aaron Courville[5,6] Eugene Belilovsky[5,7]
[1]School of Engineering, University of Guelph [2]Vector Institute for Artificial Intelligence
[3]Element AI [4]University of Cambridge [5]Mila [6]Université de Montréal [7]Concordia University

Abstract

Inferring objects and their relationships from an image in the form of a scene graph is useful in many applications at the intersection of vision and language. We consider a challenging problem of compositional generalization that emerges in this task due to a long tail data distribution. Current scene graph generation models are trained on a tiny fraction of the distribution corresponding to the most frequent compositions, e.g. *<cup, on, table>. However, test images might contain zero- and few-shot compositions of objects and relationships,* e.g. *<cup, on, surfboard>. Despite each of the object categories and the predicate (*e.g. *'on') being frequent in the training data, the models often fail to properly understand such unseen or rare compositions. To improve generalization, it is natural to attempt increasing the diversity of the training distribution. However, in the graph domain this is non-trivial. To that end, we propose a method to synthesize rare yet plausible scene graphs by perturbing real ones. We then propose and empirically study a model based on conditional generative adversarial networks (GANs) that allows us to generate visual features of perturbed scene graphs and learn from them in a joint fashion. When evaluated on the Visual Genome dataset, our approach yields marginal, but consistent improvements in zero- and few-shot metrics. We analyze the limitations of our approach indicating promising directions for future research.*

1. Introduction

Reasoning about the world in terms of objects and relationships between them is an important aspect of human and machine cognition [21]. In our environment, we can often observe frequent compositions such as "person on a surfboard" or "person next to a dog". When we are faced with a rare or previously unseen composition such as "dog on a surfboard", to understand the scene we need to understand the concepts of 'person', 'dog', 'surfboard' and 'on'. While such unbiased reasoning about concepts is easy for humans, for machines this task has remained extremely challenging [3, 32, 4, 35, 40]. Learning-based models tend to capture spurious statistical correlations in the training data [2, 52], *e.g.* 'person' rather than 'dog' has always occurred on a surfboard. When the evaluation is explicitly focused on *compositional generalization* – ability to recognize novel or rare combinations of objects and relationships – such models then can fail remarkably [3, 45, 67, 36].

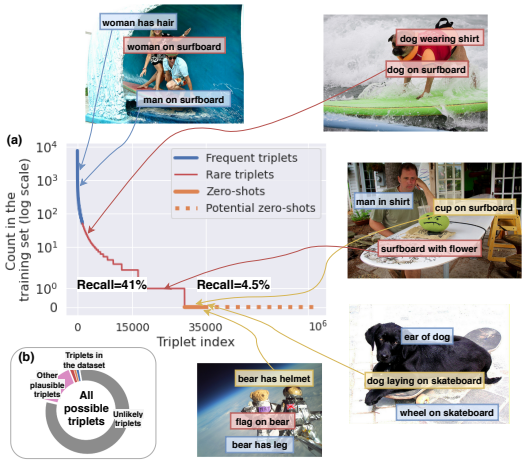

Figure 1. (**a**) The triplet distribution in Visual Genome [38] is extremely long-tailed, with numerous few- and zero-shot compositions (highlighted in red and yellow respectively). (**b**) The training set contains a tiny fraction (3%) of all possible triplets, while many other plausible triplets exist. We aim to "hallucinate" such compositions using GANs to increase the diversity of training samples and improve generalization. Recall results are from [67].

Predicting compositions of objects and the relationships between them from images is part of the scene graph generation (SGG) task. SGG is important, because accurately inferred scene graphs can improve downstream results in tasks, such as VQA [83, 29, 7, 41, 63, 26, 12], image captioning [78, 22, 42, 72, 48], retrieval [33, 5, 67, 69, 62] and others [1, 75]. However, inferring scene graphs accurately is

[*]This work was partially done while the author was an intern at Mila. Correspondence to: bknyazev@uoguelph.ca

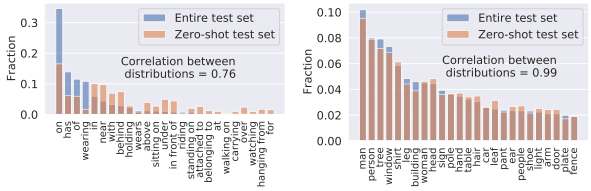

Figure 2. The distributions of top-25 predicate (**left**) and object (**right**) categories in Visual Genome [38] (split of [74]).

challenging due to a long tail data distribution and inevitable appearance of zero-shot (ZS) compositions (triplets) of objects and relationships at test time, *e.g.* "cup on surfboard" (Figure 1). The SGG results using the recent Total Direct Effect (TDE) method [67] show a severe drop in ZS recall highlighting the extreme challenge of compositional generalization. This might appear surprising given that the marginal distributions in the entire scene graph dataset (*e.g.* Visual Genome [38]) and the ZS subset are very similar (Fig. 2). More specifically, the predicate and object categories that are frequent in the entire dataset, such as 'on', 'has' and 'man', 'person' *also dominate* among the ZS triplets. For example, both "cup on surfboard" and "bear has helmet" consist of frequent entities, but represent extremely rare compositions (Fig. 1). This strongly suggests that the challenging nature of correctly predicting ZS triplets does not directly stem from the imbalance of predicates (or objects), as commonly viewed in the previous SGG works, where the models attempt to improve mean (or predicate-normalized) recall metrics [9, 18, 68, 85, 67, 10, 80, 44, 81, 76]. Therefore, we focus on compositional generalization and associated zero- and few-shot metrics.

Despite recent improvements in compositional generalization within the SGG task [67, 36, 65], the state-of-the-art result in zero-shot recall is still 4.5% compared to 41% for all-shot recall (Figure 3). To address compositional generalization, we consider exposing the model to a large diversity of training examples that can lead to emergent generalization [27, 58]. To avoid expensive labeling of additional data, we propose a compositional augmentation approach based on conditional generative adversarial networks (GANs) [19, 49]. Our general idea is augmenting the dataset by perturbing scene graphs and corresponding visual features of images, such that together they represent a novel or rare situation.

Overall, we make the following **contributions**:

- We propose scene graph perturbation methods (§ 3.1.1) as part of a GAN-based model (§ 3.1), to augment the training set with underrepresented compositions;
- We propose natural language- and dataset-based metrics to evaluate the quality of (perturbed) scene graphs (§ 3.2);
- We extensively evaluate our model and outperform a strong baseline in zero-, few- and all-shot recall (§ 4).

Our code is available at https://github.com/bknyaz/sgg.

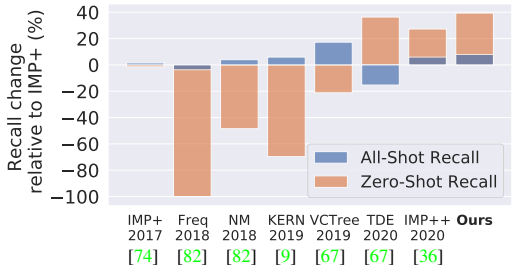

Figure 3. In this work, the compositional augmentations we propose improve on zero-shot (ZS) as well as all-shot recall.

2. Related Work

Scene Graph Generation. SGG [74] extended an earlier visual relationship detection (VRD) task [45, 60], enabling generation of a complete scene graph (SG) for an image. This spurred more research at the intersection of vision and language, where a SG can facilitate high-level visual reasoning tasks such as VQA [83, 29, 63] and others [1, 75, 55]. Follow-up SGG works [43, 77, 82, 85, 23, 68, 46, 47] have significantly improved the performance in terms of all-shot recall (Fig. 3). While the problem of zero-shot (ZS) generalization was already actively explored in the VRD task [84, 79, 73], in a more challenging SGG task and on a realistic dataset, such as Visual Genome [38], this problem has been addressed only recently in [67] by proposing Total Direct Effect (TDE), in [36] by normalizing the graph loss, and in [65] by the energy-based loss. Previous SGG works have not addressed the compositional generalization issue by synthesizing rare SGs. The closest work that also considers a generative approach is [73] solving the VRD task. Compared to it, our model follows a standard SGG pipeline and evaluation [74, 82] including object and predicate classification, instead of classifying only the predicate. We also condition a GAN on SGs rather than triplets, which combinatorially increases the number of possible augmentations. To improve SG's likelihood, we leverage both the language model and dataset statistics as opposed to random compositions as in [73].

Predicate imbalance and mean recall. Recent SGG works have focused on the predicate imbalance problem [9, 18, 68, 85, 67, 10, 80, 44, 81, 76] and mean (over predicates) recall as a metric not sensitive to the dominance of frequent predicates. However, as we discussed in § 1, the challenge of compositional generalization does not directly stem from the imbalance of predicates, since frequent predicates (*e.g.* 'on') still dominate in unseen/rare triplets (Fig. 2). Moreover, [67] showed mean recall is relatively easy to improve by standard Reweight/Resample methods, while ZS recall is not.

Data augmentation with GANs. Data augmentation is a standard method for improving machine learning models [57]. Typically these methods rely on domain specific knowledge such as applying known geometric transformations to

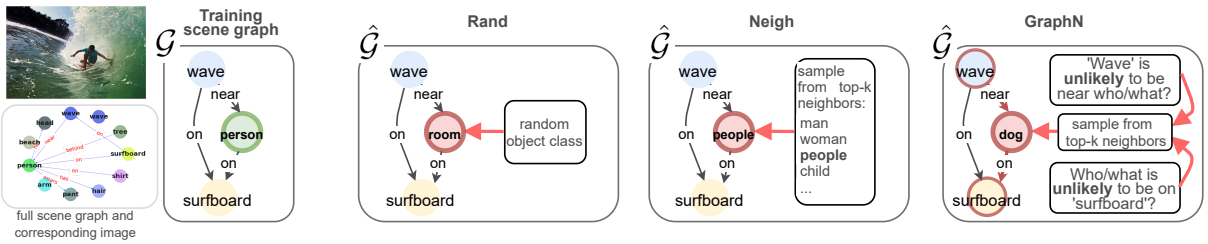

Figure 4. Illustrative examples of different perturbation schemes we consider. Only the subgraph is shown for clarity.

images [17, 11]. In the case of SGG we require more general augmentation methods, so here we explore a GAN-based approach as one of them. GANs [19] have been significantly improved w.r.t. stability of training and the quality of generated samples [6, 34], with recent works considering their usage for data augmentation [58, 64, 61]. Furthermore, recent work has shown that it is possible to produce plausible out-of-distribution (OOD) examples conditioned on unseen label combinations, by intervening on the underlying graph [37, 8, 66, 13, 20]. In this work, we have direct access to the underlying graphs of images in the form of SGs, which allows us to condition on OOD compositions as in [8, 13].

3. Methods

We consider a dataset of N tuples $\mathcal{D} = \{(I, \mathcal{G}, B)\}^N$, where I is an image with a corresponding *scene graph* $\mathcal{G}$ [33] and bounding boxes B. A scene graph $\mathcal{G} = (O, R)$ consists of n objects $O = \{o_1, ..., o_n\}$, and m relationships between them $R = \{r_1, ..., r_m\}$. For each object o_i there is an associated bounding box $b_i \in \mathbb{R}^4, B = \{b_1, ..., b_n\}$. Each object o_i is labeled with a particular category $o_i \in \mathcal{C}$, while each relationship $r_k = (i, e_k, j)$ is a triplet with a subject (start node) i, an object (end node) j and a predicate $e_k \in \mathcal{R}$, where $\mathcal{R}$ is a set of all predicate classes. For further convenience, we define a categorical triplet (*composition*) $\tilde{r}_k = (o_i, e_k, o_j)$ consisting of object and predicate categories, $\tilde{R} = \{\tilde{r}_1, ..., \tilde{r}_m\}$. An example of a scene graph is presented in Figure 4 with objects $O = \{person, surfboard, wave\}$ and relationships $R = \{(3, near, 1), (1, on, 2)\}$ and categorical relationships $\tilde{R} = \{(wave, near, person), (person, on, surfboard)\}$.

3.1. Generative Compositional Augmentations

In a given dataset $\mathcal{D}$, such as Visual Genome [38], the distribution of triplets is extremely long-tailed with a small fraction of dominating triplets (Fig. 1). To address the long-tail issue, we consider a GAN-based approach to augment $\mathcal{D}$ and artificially upsample rare compositions. Our model is based on the high-level idea of generating an additional set $\hat{\mathcal{D}} = \{(\hat{I}, \hat{\mathcal{G}}, \hat{B})\}^{\hat{N}}$. A typical scene-graph-to-image generation pipeline is [31] $\hat{\mathcal{G}} \rightarrow \hat{B} \rightarrow \hat{I}$. We describe our model accordingly by beginning with constructing $\hat{\mathcal{G}}$ and $\hat{B}$ (§ 3.1.1) followed by the generation of $\hat{I}$ (in our case, features) (§ 3.1.2). See Figure 5 for the overall pipeline.

3.1.1 Scene Graph Perturbations

We propose three methods to synthetically upsample underrepresented triplets in the dataset (Fig. 4). Our goal is to construct diverse compositions avoiding both very likely (already abundant in the dataset) and very unlikely ("implausible") combinations of objects and predicates, so that the distribution of synthetic $\hat{\mathcal{G}}$ will resemble the tail of the real distribution of $\mathcal{G}$. To construct $\hat{\mathcal{G}}$, we perturb existing $\mathcal{G}$ available in $\mathcal{D}$, since constructing graphs from scratch is more difficult: $\mathcal{G} \rightarrow \hat{\mathcal{G}}$. We focus on perturbing nodes only as it allows the creation of highly diverse compositions, so $\hat{\mathcal{G}} = (\hat{O}, R)$, where $\hat{O} = \{\hat{o}_1, ..., \hat{o}_n\}$ are the replacement object categories. We perturb only $L \cdot n$ nodes, where $L \in \mathbb{R}^{[0,1]}$, so $\hat{o}_i = o_i$ for $n(1-L)$ nodes. We sample $L \cdot n$ nodes for perturbation based on their sum of in and out degrees. Each scene graph typically has a few "hub" nodes densely connected to other nodes. So, by perturbing the hubs, we introduce more novel compositions with fewer perturbations.

RAND (random) is the simplest strategy, where for a node i we uniformly sample a category $\hat{o}$ from $\mathcal{C}$, so that $o_i = \hat{o}$.

NEIGH (semantic neighbors) leverages pretrained GloVe word embeddings [54] available for each of the object categories $\mathcal{C}$. Thus, given node i of category o_i we retrieve the top-k neighbors of o_i in the embedding space using cosine similarity. We then uniformly sample $\hat{o}$ from the top-k neighbors replacing o_i with $\hat{o}$.

GRAPHN (graph-structured semantic neighbors). RAND and NEIGH do not take into account the graph structure or dataset statistics leading to unlikely or not diverse enough compositions. To alleviate that, we propose the GRAPHN method. Given node i of category o_i in the graph $\mathcal{G}$, we consider all triplets $\tilde{R}_i = \{\tilde{r}_{k,i}\}$ in $\mathcal{G}$ that contain i as the start or end node, i.e. $\tilde{r}_{k,i} = (o_i, e_k, o_j)$ or (o_j, e_k, o_i). For example in Figure 4, if o_i is 'person', then $\tilde{R}_i = \{(person, on, surfboard), (wave, near, person)\}$. For each $\tilde{r}_{k,i}$ we find all triplets $\tilde{R}_c$ in the dataset $\mathcal{D}$ matching (o_c, e_k, o_j) or (o_j, e_k, o_c), where $o_c \neq o_i$ is a candidate replacement for o_i. For each candidate o_c, we count matched triplets $n_c = |\tilde{R}_c|$ and define unnormalized probabilities $\hat{p}_c$ based on the inverse of n_c, namely $\hat{p}_c = 1/n_c$. This way we define a set of possible replacements $\{o_c, \hat{p}_c\}$ for node i.

One of our key observations is that depending on the evaluation metric and amount of noise in the dataset, we might want to avoid sampling candidates with very high $\hat{p}_c$ (low n_c).

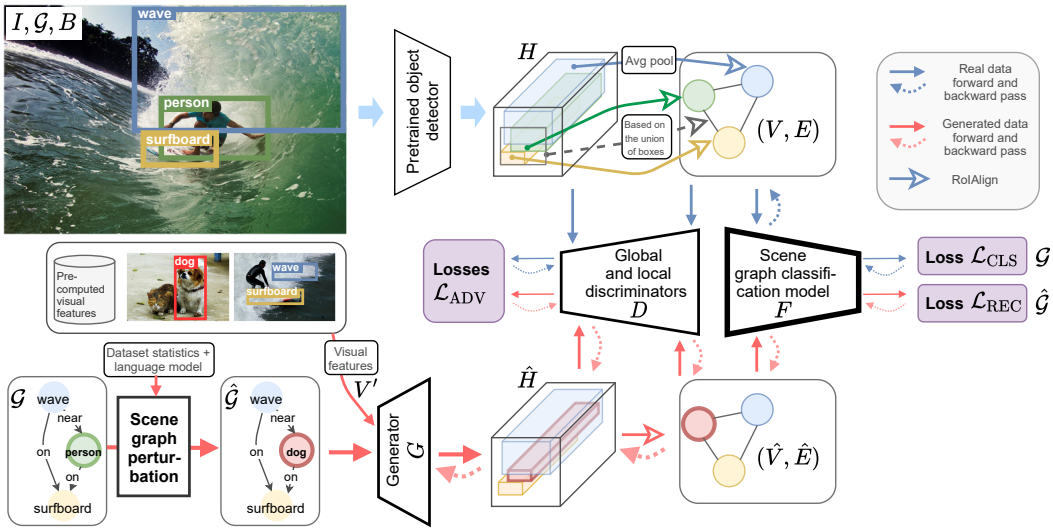

Figure 5. Our generative scene graph augmentation pipeline with its main components: discriminators D, a generator G and a scene graph classification model F. See § 3 and § A.1 in *Appendix* for a detailed description of our pipeline and model architectures.

Therefore, to control for that, we introduce an additional hyperparameter α that allows to filter out candidates with $n_c < \alpha$ by setting their $\hat{p}_c$ to 0. This way we can trade-off between upsampling rare and frequent triplets. We then normalize p_c to ensure $\sum p_c = 1$ and sample $o' \sim p_c$. To further increase the diversity, the final $\hat{o}$ is chosen from the top-k semantic neighbors of o' as in NEIGH, including o' itself. GRAPHN is a sequential perturbation procedure, where for each node the perturbation is conditioned on the current graph state. In contrast, RAND and NEIGH perturb all $L \cdot n$ nodes in parallel.

Bounding boxes. Since we perturb only a few nodes, for simplicity we assume that the perturbed graph has the same bounding boxes B: $\hat{B} = B$. While one can reasonably argue that object sizes and positions vary a lot depending on the category, i.e. "elephant" is much larger than "dog", we can often find instances disproving that, *e.g.* if a toy "elephant" or a drawing of an elephant is present. Empirically we found this approach to work well. Please see § B.3 in *Appendix* for the experiments with predicting $\hat{B}$ conditioned on $\hat{\mathcal{G}}$.

3.1.2 Scene Graph to Visual Features

Given perturbed $(\hat{\mathcal{G}}, \hat{B})$, the next step in our GAN-based pipeline is to generate visual features (Figure 5). To train such a model, we first need to extract real features from the dataset $\mathcal{D} = \{(I, \mathcal{G}, B)\}^N$. Following [74, 82], we use a pretrained and frozen object detector [59] to extract global visual features H from input images. Then, given B and H, we use RoIAlign [24] to extract visual features (V, E) of nodes and edges, respectively. To extract edge features between a pair of nodes, the union of their bounding boxes is used [82]. Since we do not update the detector, we do not need to generate images as in scene-graph-to-image models [31], just intermediate features $\hat{H}, \hat{V}, \hat{E}$.

Main scene graph classification model F. Given extracted (V, E), the main model F predicts a scene graph $\mathcal{G} = (O, R)$, i.e. it needs to correctly assign object labels O to node features V and predicate classes R to edge features E. Our pipeline is not constrained to the choice of F.

Generator G. Our scene-graph-to-features generator G follows the architecture of [31]. First, a scene graph $\hat{\mathcal{G}}$ is processed by a graph convolutional network (GCN) to exchange information between nodes and edges. We found it beneficial to concatenate output GCN features of all nodes with visual features V', where V' are sampled from the set $\{V_{o_i}\}$ precomputed at the previous stage and o_i is the category of node i. By conditioning the generator on visual features, the main task of G becomes simply to align and smooth the features appropriately, which we believe is easier than generating visual features from the categorical distribution. In addition, the randomness of this sampling step injects noise improving the diversity of generated features. The generated node features and the bounding boxes $\hat{B}$ are used to construct the layout followed by feature refinement [31] to generate $\hat{H}$. Afterwards, $(\hat{V}, \hat{E})$ are extracted from $\hat{H}$ the same way as (V, E).

Discriminators D. We have independent discriminators for nodes and edges, D_{node} and D_{edge}, that discriminate real features (V, E) from fake ones $(\hat{V}, \hat{E})$ conditioned on their class as per the CGAN [49, 56]. We add a global discriminator D_{global} acting on feature maps H, which encourages global consistency between nodes and edges. Thus, D_{node} and D_{edge} are trained to match marginal distributions, while D_{global} is trained to match the joint distribution. The right balance between these discriminators should enable the generation of realistic visual features conditioned on OOD scene graphs. Please see § A.1 in *Appendix* for the detailed architectures of D and G.

Losses. To train our generative model, we define several

losses. These include the baseline SG classification loss (1) and ones specific to our generative pipeline (2)-(5). The latter are motivated by a CycleGAN [86] and, similarly, consist of the reconstruction and adversarial losses (2)-(5).

We use an improved **scene graph classification loss** from [36], which is a sum of the node cross-entropy loss $\mathcal{L}^O$ and graph density-normalized edge cross-entropy loss $\mathcal{L}^R$:

$$\begin{aligned}\mathcal{L}_{\text{CLS}} &= \mathcal{L}(F(V,E),\mathcal{G}) = \\ &= \mathcal{L}^O(F(V,E),O) + \mathcal{L}^R(F(V,E),R). \end{aligned} \quad (1)$$

$\mathcal{L}^R$ is computed based on the ratio of foreground (annotated) to background (not annotated) edges in a batch of scene graphs [36]. To improve F by training it on augmented features $(\hat{V},\hat{E})$, we define the **reconstruction (cycle-consistency) loss** analogous to (1):

$$\begin{aligned}\mathcal{L}_{\text{REC}} =& \mathcal{L}(F(G(\hat{\mathcal{G}},\hat{B},V')),\hat{\mathcal{G}}) = \\ =& \mathcal{L}^O(F(\hat{V},\hat{E}),\hat{O}) + \mathcal{L}^R(F(\hat{V},\hat{E}),R). \end{aligned} \quad (2)$$

We do not update G on this loss to prevent its potential undesirable collaboration with F. Instead, to train G as well as D, we optimize **conditional adversarial losses** [49]. We first write these separately for D and G in a general form. So, for some features $\boldsymbol{x}$ and their corresponding class $\boldsymbol{y}$:

$$\begin{aligned}\mathcal{L}^D_{\text{ADV}}(\boldsymbol{x},\boldsymbol{y}) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \\ \mathbb{E}_{\hat{\mathcal{G}}\sim p_{\hat{\mathcal{G}}}(\hat{\mathcal{G}})}[\log(1 - D(G(\hat{\mathcal{G}})|\boldsymbol{y})] \end{aligned} \quad (3)$$

$$\mathcal{L}^G_{\text{ADV}}(\boldsymbol{y}) = \mathbb{E}_{\hat{\mathcal{G}}\sim p_{\hat{\mathcal{G}}}(\hat{\mathcal{G}})}[\log D(G(\hat{\mathcal{G}})|\boldsymbol{y})]. \quad (4)$$

We compute these losses for object and edge visual features by using the discriminators D_{node} and D_{edge}. This loss is also computed for global features H using D_{global}, so that the total discriminator and generator losses are:

$$\begin{aligned}\mathcal{L}^D_{\text{ADV}} &= \mathcal{L}^D_{\text{ADV}}(V,O) + \mathcal{L}^D_{\text{ADV}}(E,R) + \mathcal{L}^D_{\text{ADV}}(H,\emptyset) \\ \mathcal{L}^G_{\text{ADV}} &= \mathcal{L}^G_{\text{ADV}}(O) + \mathcal{L}^G_{\text{ADV}}(R) + \mathcal{L}^G_{\text{ADV}}(\emptyset), \end{aligned} \quad (5)$$

where $\emptyset$ denotes that our global discriminator is unconditional for simplicity. Thus, the total loss to minimize is:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{CLS}} + \mathcal{L}_{\text{REC}}}_{\text{update } F} - \gamma(\underbrace{\mathcal{L}^D_{\text{ADV}}}_{\text{update } D} + \underbrace{\mathcal{L}^G_{\text{ADV}}}_{\text{update } G}), \quad (6)$$

where the loss weight $\gamma = 5$ worked well in our experiments. Compared to a similar work of [73], in our model all of its components (F, D, G) are learned jointly end-to-end.

3.2. Semantic Plausibility of Scene Graphs

Language model. To directly evaluate the quality of perturbations, it is desirable to have some quantitative measure other than downstream SGG performance. We found that a cheap (relative to human evaluation) and effective way to achieve this goal is to use a language model. In particular, we use a pretrained BERT [14] model and estimate the "semantic plausibility" of both ground truth and perturbed scene graphs in the following way. We create a textual query from a scene graph by concatenating all triplets (in a random order). We then mask out one of the perturbed nodes (in case of $\hat{\mathcal{G}}$) or a random node (in case of $\mathcal{G}$) in the triplet, so that BERT can return (unnormalized) likelihood scores for the object category of the masked out token. We have also considered using this strategy to create SG perturbations as an alternative to GRAPHN. However, we did not find it effective for obtaining rare scene graphs, since BERT is not grounded to visual concepts and not aware of what is considered "rare" in a particular SG dataset. For qualitative evaluation and when BERT scores are averaged over many samples, we found them still useful as a rough measure of SG quality. Please see § B.2 in *Appendix* for an example of the BERT-based estimation of scene graph quality.

Hit rate. For perturbed SGs, we compute an additional qualitative metric, which we call the 'Hit rate'. Assuming we perturbed M triplets in total for all training SGs, this metric computes the percentage of the triplets matching an actual annotation in an evaluation test subset (zero-, few- or all-shot).

4. Experiments

4.1. Dataset, Models and Hyperparameters

We use a publicly available SGG codebase[1] for evaluation and baseline model implementations. For the model F, we use Iterative Message Passing (IMP+) [74, 82] and Neural Motifs (NM) [82]. IMP+ shows strong compositional generalization capabilities [36] and, therefore is more explored in this work. We use an improved loss for (1) from [36], so we denote our baselines as IMP++ and NM++. We use the default hyperparameters and identical setups for the baseline models without a GAN and our models with a GAN. We borrow the detector Faster-RCNN with the VGG16 backbone pretrained on Visual Genome (VG) from [82] and use it in all our experiments. We evaluate the models on a standard split of VG [38], with the 150 most frequent object classes and 50 predicate classes, introduced in [74]. The training set has 57723 and the test set has 26446 images. Similarly to [36, 73, 67, 65], in addition to the all-shot (all test scene graphs) case, we define zero-shot, 10-shot and 100-shot test subsets. For each such subset we keep only those triplets in a scene graph that occur 0, 1-10 or 11-100 times during training and remove samples without such triplets, which results in 4519, 9602 and 16528 test scene graphs (and images) respectively. We use a held-out validation set of 5000 images for tuning the hyperparameters.

Baselines. In addition to the IMP++ and NM++ baselines, we evaluate RESAMPLE, REWEIGHT and TDE [67] when combined with IMP++. RESAMPLE samples training images based on the inverse frequency of predicates/triplets [67]. REWEIGHT increases the softmax scores of rare predicate classes (see § B.4 in *Appendix* for details). TDE debiases contextual edge features of a SGG

[1] https://github.com/rowanz/neural-motifs

Table 1. Results on Visual Genome [38] using models based on IMP++ [36]. The top-1 result in each column is **bolded** (ignoring ORACLE-ZS). ORACLE-ZS results are an upper bound estimate of ZS recall obtained by directly using ZS test triplets for perturbations.

MODEL	ZERO-SHOT RECALL		10-SHOT RECALL		100-SHOT RECALL		ALL-SHOT RECALL		
	SGCls	PredCls	SGCls	PredCls	SGCls	PredCls	SGCls	PredCls	SGCls-mR
Baseline (IMP++)	9.27±0.10	28.14±0.05	21.80±0.19	42.78±0.32	40.42±0.02	67.78±0.07	48.70±0.08	77.48±0.09	27.78±0.10
GAN+GRAPHN, $\alpha = 2$	**9.89**±0.15	28.90±0.14	21.96±0.30	**43.79**±0.27	41.22±0.33	69.17±0.24	50.06±0.29	78.98±0.09	27.79±0.48
GAN+GRAPHN, $\alpha = 5$	9.62±0.29	**29.18**±0.33	**22.24**±0.11	43.74±0.10	41.39±0.26	69.11±0.05	50.14±0.21	78.94±0.03	27.98±0.23
GAN+GRAPHN, $\alpha = 10$	9.84±0.17	28.90±0.46	22.04±0.33	43.54±0.36	41.46±0.15	69.13±0.24	50.10±0.23	79.00±0.09	27.68±0.37
GAN+GRAPHN, $\alpha = 20$	9.65±0.15	28.68±0.28	21.97±0.30	43.64±0.20	41.24±0.08	**69.31**±0.17	49.89±0.28	78.95±0.04	27.42±0.36
Ablated models									
GAN (no perturb.)	9.25±0.20	28.66±0.35	22.15±0.21	43.66±0.29	**41.58**±0.20	69.16±0.16	**50.38**±0.28	**79.05**±0.08	28.17±0.08
GAN+RAND	9.71±0.09	28.71±0.40	21.89±0.21	43.33±0.18	41.01±0.32	68.88±0.23	49.83±0.32	78.84±0.10	27.45±0.48
GAN+NEIGH	9.65±0.04	28.68±0.40	21.86±0.23	43.77±0.15	41.25±0.35	69.07±0.09	50.00±0.36	78.94±0.10	27.41±0.51
Other baselines									
REWEIGHT	9.58±0.14	28.27±0.22	22.19±0.09	42.98±0.17	40.00±0.01	65.27±0.13	48.13±0.10	74.68±0.13	**30.95**±0.05
RESAMPLE-predicates	9.13±0.06	27.77±0.10	21.35±0.05	42.14±0.16	39.69±0.06	66.74±0.01	48.23±0.10	76.59±0.05	28.44±0.38
RESAMPLE-triplets	8.94±0.16	27.66±0.14	21.65±0.10	42.60±0.17	39.39±0.08	66.44±0.06	47.77±0.10	76.38±0.14	27.56±0.10
TDE	9.21±0.21	27.91±0.09	21.20±0.16	41.61±0.32	39.72±0.10	65.40±0.21	48.35±0.08	76.22±0.17	28.25±0.21
ORACLE perturbations $\hat{\mathcal{G}}$									
GAN+ORACLE-ZS $\hat{\mathcal{G}}$	10.11±0.34	29.27±0.10	22.05±0.38	43.78±0.09	41.38±0.50	69.06±0.16	50.19±0.36	79.00±0.08	27.91±0.56
GAN+ORACLE-ZS $\hat{\mathcal{G}} + \hat{B}$	10.52±0.31	29.43±0.42	21.98±0.39	43.03±0.13	41.12±0.19	68.73±0.17	50.05±0.35	78.65±0.09	27.52±0.46

Figure 6. Triplet hit rates (§ 3.2) versus the threshold α on four different VG test subsets using our perturbation methods.

model. We use the Total Effect (TE) variant according to Eq. 6 in [67], since applying TDE to IMP++ is not straightforward due to the absence of conditioning on node labels when making predictions for edges in IMP++. REWEIGHT and TDE/TE do not require retraining IMP++.

GAN. To train the generator G and discriminators D of a GAN, we generally follow hyperparameters suggested by SPADE [53]. In particular, we use Spectral Norm [50] for D, Batch Norm [30] for G, and TTUR [25] with learning rates of 1e-4 and 2e-4 for G and D respectively.

Perturbation methods (§ 3.1.1). We found that perturbing $L = 20\%$ nodes works well across the methods, which we use in all our experiments. For NEIGH we use top-k=10 as a compromise between too limited diversity and plausibility. For GRAPHN, we set top-k=5, as the method enables larger diversity even with very small top-k. To train the GAN-based models with GRAPHN, we use frequency threshold $\alpha = [2, 5, 10, 20]$. In addition to the proposed perturbation methods, we also consider so called ORACLE-ZS perturbations. These are created by directly using ZS triplets from the test set (all obtained triplets are the same as ZS triplets, so that zero-shot hit rate is 100%). We also evaluate ORACLE-ZS+$\hat{B}$, which in addition to exploiting test ZS triplets, uses bounding boxes from the test samples corresponding to the resulted ZS triplets. ORACLE-ZS-based results are an upper bound estimate of ZS recall, highlighting the challenging nature of the task.

Evaluation. Following prior work [74, 82, 36, 67], we focus our evaluation on two standard SGG tasks: scene graph classification (**SGCls**) and predicate classification (**PredCls**), using recall (R@K) metrics. The scene graph generation (**SGGen**) results are presented in § B.6 in *Appendix*. Unless otherwise stated, we report results with K=100 for SGCls and K=50 for PredCls, since the latter is an easier task with saturated results for K=100. We compute recall *without* the graph constraint in Table 1, since it is a less noisy metric [36]. We emphasize performance metrics that focus on the ability to recognize rare and novel visual relationship compositions [36, 67, 65]: **zero-shot** and **10-shot** recalls. In Tables 1 and 2, the mean and standard deviations of 3 runs (random seeds) are reported.

4.2. Results

Main SGG results (Table 1). First, we compare the baseline IMP++ to our GAN-based model trained *without* and *with* perturbation methods. Even without any perturbations, the GAN-based model significantly outperforms IMP++, especially on the 100-shot and all-shot recalls. GANs with simple perturbation strategies, RAND (as in [73]) and NEIGH, improve on zero-shots, but at a drop in the 100-shot and all-shot recalls. GANs with GRAPHN further improve ZS and 10-shot recalls, but compared to RAND and NEIGH, also show high recalls on the 100-shots and all-shots.

For GRAPHN, there is a connection between the SGG

Table 2. ZS recall results on VG using the graph constraint evaluation. †The results are obtained with a more advanced feature extractor and, thus, are not directly comparable.

MODEL	SGCls		PredCls	
	zsR@50	zsR@100	zsR@50	zsR@100
FREQ [82]	0.0	0.0	0.1	0.1
KERN [9]	–	1.5	3.9	–
VCTree† [67]	1.9	2.6	10.8	14.3
NM [82]	1.1	1.7	6.5	9.5
NM† [67]	2.2	3.0	10.9	14.5
NM, TDE† [67]	3.4	**4.5**	14.4	18.2
NM, EBM† [65]	1.3	–	4.9	–
NM++ [36]	$1.8_{\pm 0.1}$	$2.3_{\pm 0.1}$	$10.2_{\pm 0.1}$	$13.4_{\pm 0.3}$
NM++, GAN+GRAPHN	$2.5_{\pm 0.1}$	$3.1_{\pm 0.1}$	$14.2_{\pm 0.0}$	$17.4_{\pm 0.3}$
IMP+ [74, 82]	2.5	3.2	14.5	17.2
IMP+, EBM† [65]	3.7	–	18.6	–
IMP++ [36]	$3.5_{\pm 0.1}$	$4.2_{\pm 0.2}$	$18.3_{\pm 0.4}$	$21.2_{\pm 0.5}$
IMP++, TDE	$3.5_{\pm 0.1}$	$4.3_{\pm 0.1}$	$18.5_{\pm 0.3}$	$21.5_{\pm 0.3}$
IMP++, GAN+GRAPHN	$3.7_{\pm 0.1}$	$4.4_{\pm 0.1}$	$19.1_{\pm 0.3}$	$21.8_{\pm 0.4}$
IMP++, GAN+GRAPHN (max)	**3.8**	**4.5**	**19.5**	**22.4**

recall results (Table 1) and triplet hit rates (Fig. 6) for different values of the threshold α. Specifically, GRAPHN with lower α values upsamples more of the rare compositions leading to higher ZS and 10-shot *hit rate* (Fig. 6 a,b) and, as a result, higher ZS and 10-shot *recalls* (Table 1). GRAPHN with higher α values upsamples more of the frequent compositions leading to higher 100-shot and all-shot *hit rates* (Fig. 6 c,d) and, as a result, higher 100-shot and all-shot *recalls*. Compared to RAND and NEIGH, the compositions obtained using GRAPHN have higher triplet hit rates due to better respecting the graph structure and dataset statistics. As a result, GRAPHN shows overall better recalls in SGG, even approaching the ORACLE-ZS model (Table 1). Devising a perturbation strategy universally strong across all metrics is challenging. NEIGH can be viewed as such an attempt, which shows average hit rates for all test subsets, but lower performance in all SGG metrics.

Among the alternatives to our GAN approach, REWEIGHT improves on zero-shots, 10-shots and mean recall (SGCls-mR) (Table 1). However it downweights the class scores of frequent predicates, which directly degrades 100-shot and all-shot recalls. RESAMPLE underperforms on all metrics except for SGCls-mR. The main limitation of RESAMPLE is that when we resample images with rare predicates/triplets, those images are likely to contain annotations of frequent predicates/triplets. Another method, TDE [67], only debiases the predicates similarly to REWEIGHT and RESAMPLE-predicates. So, it may benefit little in recognizing ZS triplets such as $(cup, on, surfboard)$, because the predicate 'on' is the frequent one. ZS compositions with such frequent predicates are abundant in VG (Fig. 1). Thus, debiasing only the predicates fundamentally limits TDE's performance. In contrast, our GAN method does not suffer from this limitation, since we perturb scene graphs

Table 3. Evaluation of generated (fake) node feature using the metrics of "similarity" between two distributions X and Y [39, 51]. The same held-out set of real test features ($Y \sim V$) is used as the reference distribution in all cases. The percentage in the superscripts denotes a relative drop of the average metric when switching from test to test-zs conditioning. For all metrics, higher is better.

DISTRIBUTION X	Fidelity (realism)		Diversity		AVG
	PRECISION	DENSITY	RECALL	COVERAGE	
Real test	0.74	1.02	0.75	0.97	0.87
Real test-zs	0.66	0.99	0.70	0.94	$0.82^{-6\%}$
GAN: Fake test	0.55	0.77	0.42	0.82	0.64
GAN: Fake test-zs	0.47	0.60	0.41	0.75	$0.56^{-13\%}$

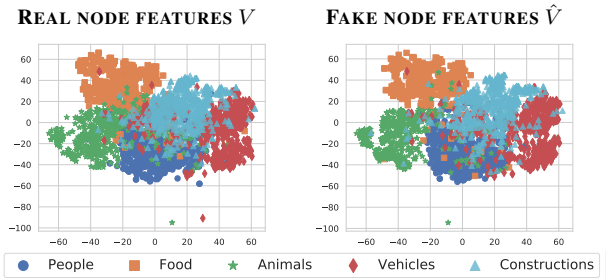

Figure 7. Real *vs* generated node features plotted using t-SNE.

aiming to increase *compositional diversity*, not merely the frequency of rare predicates. As a result, our GAN method improves on all metrics, *especially* on ZS (in relative terms).

Comparison to other SGG works (Table 2). Our GAN approach also improves ZS recall (zsR) of other SGG models, namely NM++. For example in PredCls, GAN+GRAPHN improves zsR of NM++ by 4 percentage points. Compared to the other previous methods presented in Table 2, we obtain competitive ZS results on par or better with TDE [67] and recent EBM [65]. However, it is hard to directly compare to the results reported in [67, 65] due to the different object detectors and potential implementation discrepancies.

Evaluation of generated visual features. We evaluate the quality of generated features of our GAN trained with GRAPHN by comparing the generated (fake) features to the real ones. To obtain fake node features $\hat{V}$, we condition our GAN on test SGs. To obtain real node features V, we apply the pretrained object detector to test images as described in § 3.1.2. First, for the qualitative evaluation of node features, we group features based on the object category's super-type, *e.g.* 'people' includes all features of 'man', 'woman', 'person', etc. When projected on a 2D space using t-SNE [70], the fake features $\hat{V}$ generated using our GAN are clustered similarly to the real features V (Fig. 7). Therefore, qualitatively our GAN generates realistic and diverse features given a scene graph.

Second, we evaluate GAN features quantitatively. For that purpose, we follow [15] and use Precision, Recall [39] and Density, Coverage [51] metrics. These metrics compare the manifolds spanned by real and fake features and do not require any labels. We consider two cases: conditioning our GAN on test SGs and test zero-shot (test-zs) SGs. The moti-

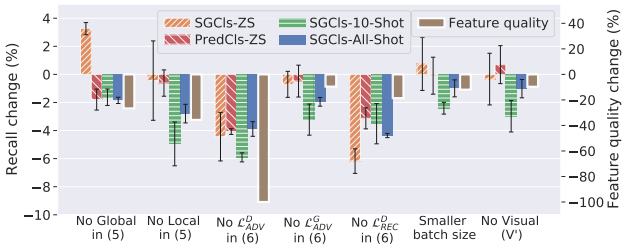

Figure 8. Ablations of our GAN model on SGG and feature quality metrics. Error bars denote standard deviation. For feature quality the average metric on the test-zs SGs from Table 3 is used.

vation is similar to [8]: understand if novel compositions confuse the GAN and lead to poor features, that in our context may result in poor training of the main model F. Indeed, the features generated conditioned on test-zs SGs significantly degrade in quality compared to test SGs, especially in terms of fidelity (Table 3). This result suggests that it is more challenging to produce realistic features for more rare compositions limiting our approach (see § B.8 in *Appendix* for a discussion). The same qualitative and quantitative experiments for edge features $(E, \hat{E})$ and global features $(H, \hat{H})$ confirm our results: (1) when conditioned on test SGs, the generated features are realistic and diverse; (2) conditioning on more rare compositions degrades feature quality (see § B.7).

Ablations (Figure 8). We also performed ablations to determine the effect of the proposed GAN losses (6) and other design choices on the (i) SGG performance and (ii) quality of generated features. As a reference model, we use our GAN model without any perturbations. In general, all ablated GANs degrade both in (i) and (ii) with correlated drops between (i) and (ii). So by improving generative models in future work, we can expect larger SGG gains. One exception is the GAN without the global terms in (5), which performed better on zero-shots despite having lower feature quality. This might be explained as some regularization effect. We also found that this model did not combine well with perturbations.

Evaluating the quality of SG perturbations. We show examples of SG perturbations in Fig. 10 and in *Appendix*. In case of RAND, most of the created triplets are implausible as a result of random perturbations. NEIGH leads to very likely compositions, but less often provides rare plausible compositions. In contrast, GRAPHN can create plausible compositions that are rare or more frequent depending on α.

We also analyzed the quality of real and perturbed SGs using the BERT-based metric (§ 3.2). We found that the overall test set has on average the highest BERT scores, while lower-shot subsets gradually decrease in "semantic plausibility", which aligns with our intuition. We then perturbed all nodes of all test SGs using our perturbation strategies. Surprisingly, real test-zs SGs have very low plausibility close to RAND-based SGs. NEIGH produces SGs of plausibility between real 10-shot and 100-shot SGs. In contrast, with GRAPHN we

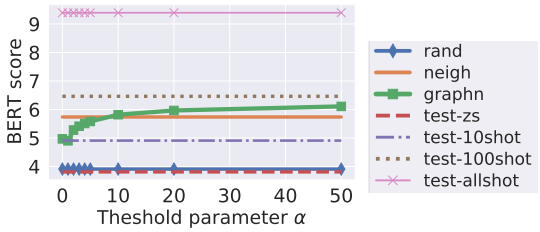

Figure 9. Semantic plausibility (as per BERT) depending on α. These results should be interpreted with caution, because: (1) the variance of scores is very high (not shown); (2) in the zero- and few-shot test subsets the graphs are significantly smaller, which affects the amount of contextual information available to BERT.

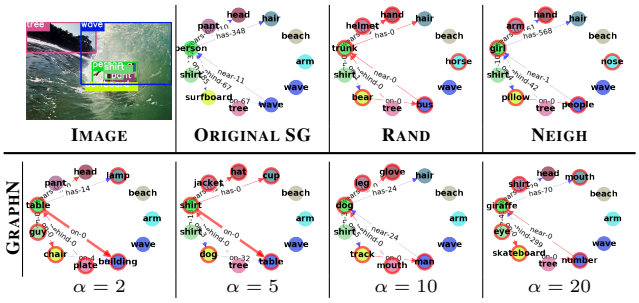

Figure 10. Examples of perturbations (nodes in red) applied to a scene graph. The numbers on edges denote the count of triplets in the training set and a thick red arrow denotes matching a ZS triplet.

can gradually slide between low and high plausibility, which enabled better SGG results. The BERT scores, however, are not tied to the VG dataset. So, semantic plausibility per BERT may be different from the likelihood per VG.

5. Conclusion

We focus on the compositional generalization problem within the scene graph generation task. Our GAN-based augmentation approach can be used with different SGG models and can improve their zero-, few- and all-shot SGG results. To obtain better SGG results using our augmentations, it is important to rely on the structure of scene graphs and tune the augmentation parameters towards a specific SGG metric. Our evaluation confirmed that our augmentations provide plausible compositions and the generator generally produces high-fidelity and diverse features enabling gains in SGG.

Acknowledgments

We thank all the reviewers for their useful feedback. This research was developed with funding from DARPA. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. The authors also acknowledge support from the Canadian Institute for Advanced Research and the Canada Foundation for Innovation. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute: http://www.vectorinstitute.ai/#partners.

References

[1] Aniket Agarwal, Ayush Mangal, et al. Visual relationship detection using scene graphs: A survey. *arXiv preprint arXiv:2005.08045*, 2020. 1, 2

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1

[3] Yuval Atzmon, Jonathan Berant, Vahid Kezami, Amir Globerson, and Gal Chechik. Learning to generalize to new compositions in image understanding. *arXiv preprint arXiv:1608.07639*, 2016. 1

[4] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: what is required and can it be learned? *arXiv preprint arXiv:1811.12889*, 2018. 1

[5] Eugene Belilovsky, Matthew Blaschko, Jamie Kiros, Raquel Urtasun, and Richard Zemel. Joint embeddings of scene graphs and images. 2017. 1

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[7] Cătălina Cangea, Eugene Belilovsky, Pietro Liò, and Aaron Courville. VideoNavQA: Bridging the Gap between Visual and Embodied Question Answering. *arXiv preprint arXiv:1908.04950*, 2019. 1

[8] Arantxa Casanova, Michal Drozdzal, and Adriana Romero-Soriano. Generating unseen complex scenes: are we there yet? *arXiv preprint arXiv:2012.04027*, 2020. 3, 8, 14, 16

[9] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019. 2, 7, 15

[10] Vincent S Chen, Paroma Varma, Ranjay Krishna, Michael Bernstein, Christopher Re, and Li Fei-Fei. Scene graph prediction with limited labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2580–2590, 2019. 2

[11] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 3

[12] Vinay Damodaran, Sharanya Chakravarthy, Akshay Kumar, Anjana Umapathy, Teruko Mitamura, Yuta Nakashima, Noa Garcia, and Chenhui Chu. Understanding the role of scene graphs in visual question answering. *arXiv preprint arXiv:2101.05479*, 2021. 1

[13] Fei Deng, Zhuo Zhi, Donghun Lee, and Sungjin Ahn. Generative scene graph networks. In *International Conference on Learning Representations*, 2021. 3

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[15] Terrance DeVries, Michal Drozdzal, and Graham W Taylor. Instance selection for gans. *arXiv preprint arXiv:2007.15255*, 2020. 7

[16] Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017. 16

[17] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[18] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationships as functions: Enabling few-shot scene graph prediction. In *ArXiv*, 2019. 2

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2, 3

[20] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019. 3

[21] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020. 1

[22] Jiuxiang Gu, Shafiq Joty, Jianfei Cai, Handong Zhao, Xu Yang, and Gang Wang. Unpaired image captioning via scene graph alignments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10323–10332, 2019. 1

[23] 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 Conference on Computer Vision and Pattern Recognition*, pages 1969–1978, 2019. 2

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 6

[26] Marcel Hildebrandt, Hang Li, Rajat Koner, Volker Tresp, and Stephan Günnemann. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020. 1

[27] Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent, 2019. 2

[28] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7986–7994, 2018. 16

[29] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5903–5916. Curran Associates, Inc., 2019. 1, 2

[30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6, 12

[31] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 3, 4, 12

[32] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 1

[33] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015. 1, 3

[34] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020. 3

[35] Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*, 2019. 1

[36] Boris Knyazev, Harm de Vries, Cătălina Cangea, Graham W Taylor, Aaron Courville, and Eugene Belilovsky. Graph density-aware losses for novel compositions in scene graph generation. *British Machine Vision Conference*, 2020. 1, 2, 5, 6, 7, 12, 13, 15

[37] Murat Kocaoglu, Christopher Snyder, Alexandros G Dimakis, and Sriram Vishwanath. CausalGAN: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017. 3

[38] 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. 1, 2, 3, 5, 6, 12, 13

[39] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019. 7

[40] Brenden M Lake. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, pages 9788–9798, 2019. 1

[41] Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. Visual question answering over scene graph. In *2019 First International Conference on Graph Computing (GC)*, pages 45–50. IEEE, 2019. 1

[42] Xiangyang Li and Shuqiang Jiang. Know more say less: Image captioning based on scene graphs. *IEEE Transactions on Multimedia*, 21(8):2117–2130, 2019. 1

[43] 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 International Conference on Computer Vision*, pages 1261–1270, 2017. 2

[44] 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. 2, 14

[45] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European conference on computer vision*, pages 852–869. Springer, 2016. 1, 2

[46] Yichao Lu, Cheng Chang, Himanshu Rai, Guangwei Yu, and Maksims Volkovs. Learning effective visual relationship detector on 1 gpu. *arXiv preprint arXiv:1912.06185*, 2019. 2

[47] Yichao Lu, Cheng Chang, Himanshu Rai, Guangwei Yu, and Maksims Volkovs. Multi-view scene graph generation in videos. *International Challenge on Activity Recognition (ActivityNet) CVPR 2021 Workshop*, 2021. 2

[48] Victor Milewski, Marie-Francine Moens, and Iacer Calixto. Are scene graphs good enough to improve image captioning? *arXiv preprint arXiv:2009.12313*, 2020. 1

[49] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2, 4, 5

[50] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 6, 12

[51] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020. 7

[52] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. *arXiv preprint arXiv:2006.04315*, 2020. 1

[53] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 6

[54] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3

[55] Moshiko Raboh, Roei Herzig, Jonathan Berant, Gal Chechik, and Amir Globerson. Differentiable scene graphs. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1488–1497, 2020. 2

[56] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4, 12

[57] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pages 3236–3246, 2017. 2

[58] Suman Ravuri and Oriol Vinyals. Seeing is not necessarily believing: Limitations of biggans for data augmentation. 2019. 2, 3

[59] 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*, pages 91–99, 2015. 4

[60] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *CVPR 2011*, pages 1745–1752. IEEE, 2011. 2

[61] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. Data augmentation using generative adversarial networks (cyclegan) to improve generalizability in ct segmentation tasks. *Scientific reports*, 9(1):1–9, 2019. 3

[62] Brigit Schroeder and Subarna Tripathi. Structured query-based image retrieval using scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 178–179, 2020. 1

[63] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8376–8384, 2019. 1, 2

[64] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski. Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *International workshop on simulation and synthesis in medical imaging*, pages 1–11. Springer, 2018. 3

[65] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. Energy-based learning for scene graph generation. *arXiv preprint arXiv:2103.02221*, 2021. 2, 5, 6, 7

[66] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *arXiv preprint arXiv:2003.11571*, 2020. 3

[67] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 5, 6, 7, 14, 15

[68] 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 Conference on Computer Vision and Pattern Recognition*, pages 6619–6628, 2019. 2

[69] Subarna Tripathi, Sharath Nittur Sridhar, Sairam Sundaresan, and Hanlin Tang. Compact scene graphs for layout composition and patch retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[70] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7

[71] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019. 16

[72] Dalin Wang, Daniel Beck, and Trevor Cohn. On the role of scene graphs in image captioning. In *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-world kNowledge (LANTERN)*, pages 29–34, 2019. 1

[73] Xiaogang Wang, Qianru Sun, Marcelo ANG, and Tat-Seng CHUA. Generating expensive relationship features from cheap objects. 2019. 2, 5, 6

[74] 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. 2, 4, 5, 6, 7, 12, 15

[75] Pengfei Xu, Xiaojun Chang, Ling Guo, Po-Yao Huang, Xiaojiang Chen, and Alexander G Hauptmann. A survey of scene graph: Generation and application. 2020. 1, 2

[76] 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. 2, 14, 15

[77] 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 (ECCV)*, pages 670–685, 2018. 2

[78] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 1

[79] Xu Yang, Hanwang Zhang, and Jianfei Cai. Shuffle-then-assemble: Learning object-agnostic visual relationship features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 36–52, 2018. 2

[80] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Bridging knowledge graphs to generate scene graphs. In *European Conference on Computer Vision*, pages 606–623. Springer, 2020. 2

[81] Alireza Zareian, Haoxuan You, Zhecan Wang, and Shih-Fu Chang. Learning visual commonsense for robust scene graph generation. *arXiv preprint arXiv:2006.09623*, 2020. 2

[82] 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. 2, 4, 5, 6, 7, 12, 13, 15

[83] Cheng Zhang, Wei-Lun Chao, and Dong Xuan. An empirical study on leveraging scene graphs for visual question answering. *arXiv preprint arXiv:1907.12133*, 2019. 1, 2

[84] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5532–5540, 2017. 2

[85] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019. 2

[86] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 5