IS-GGT: Iterative Scene Graph Generation with Generative Transformers

Sanjoy Kundu
Department of Computer Science
Oklahoma State University
sanjoy.kundu@okstate.edu

Sathyanarayanan N. Aakur
Department of Computer Science
Oklahoma State University
saakurn@okstate.edu

Abstract

Scene graphs provide a rich, structured representation of a scene by encoding the entities (objects) and their spatial relationships in a graphical format. This representation has proven useful in several tasks, such as question answering, captioning, and even object detection, to name a few. Current approaches take a generation-by-classification approach where the scene graph is generated through labeling of all possible edges between objects in a scene, which adds computational overhead to the approach. This work introduces a generative transformer-based approach to generating scene graphs beyond link prediction. Using two transformer-based components, we first sample a possible scene graph structure from detected objects and their visual features. We then perform predicate classification on the sampled edges to generate the final scene graph. This approach allows us to efficiently generate scene graphs from images with minimal inference overhead. Extensive experiments on the Visual Genome dataset demonstrate the efficiency of the proposed approach. Without bells and whistles, we obtain, on average, 20.7% *mean recall (mR@100) across different settings for scene graph generation (SGG), outperforming state-of-the-art SGG approaches while offering competitive performance to unbiased SGG approaches.*

1. Introduction

Graph-based visual representations are becoming increasingly popular due to their ability to encode visual, semantic, and even temporal relationships in a compact representation that has several downstream tasks such as object tracking [4], scene understanding [17] and event complex visual commonsense reasoning [2, 3, 22]. Graphs can help navigate clutter and express complex semantic structures from visual inputs to mitigate the impact of noise, clutter, and (appearance/scene) variability, which is essential in scene understanding. Scene graphs, defined as directed graphs that model the visual-semantic relationships among entities (objects) in a given scene, have proven to be very useful in downstream tasks such as visual question-answering [14, 34], captioning [17], and even embodied tasks such as navigation [27], to name a few.

There has been a growing body of work [7, 10, 29, 33, 36, 38, 41] that has focused on the problem of scene graph generation (SGG), that aims to generate scene graph from a given input observation. However, such approaches have tackled the problem by beginning with a fully connected graph, where all entities interact with each other before pruning it down to a more compact graph by predicting edge relationships, or the lack of one, between each pair of localized entities. This approach, while effective, has several limitations. First, by modeling the interactions between entities with a dense topology, the underlying semantic structure is ignored during relational reasoning, which can lead to poor predicate (relationship) classification. Second, by constructing pairwise relationships between all entities in a scene, there is tremendous overhead on the predicate classification modules since the number of pairwise comparisons can grow non-linearly with the increase in the number of detected concepts. Combined, these two issues aggravate the existing long-tail distribution problem in scene graph generation. Recent progress in *unbiasing* [21, 31–33] has attempted to address this issue by tackling the long-tail distribution problem. However, they depend on the quality of the underlying graph generation approaches, which suffer from the above limitations.

In this work, we aim to overcome these limitations using a two-stage, generative approach called IS-GGT, a transformer-based iterative scene graph generation approach. An overview of the approach is illustrated in Figure 1. Contrary to current approaches to SGG, we leverage advances in generative graph models [5, 23] to first sample the underlying interaction graph between the detected entities before reasoning over this sampled semantic structure for scene graph generation. By decoupling the ideas of graph generation and relationship modeling, we can constrain the relationship classification process to consider only those edges (pairs of entities) that have a higher probability

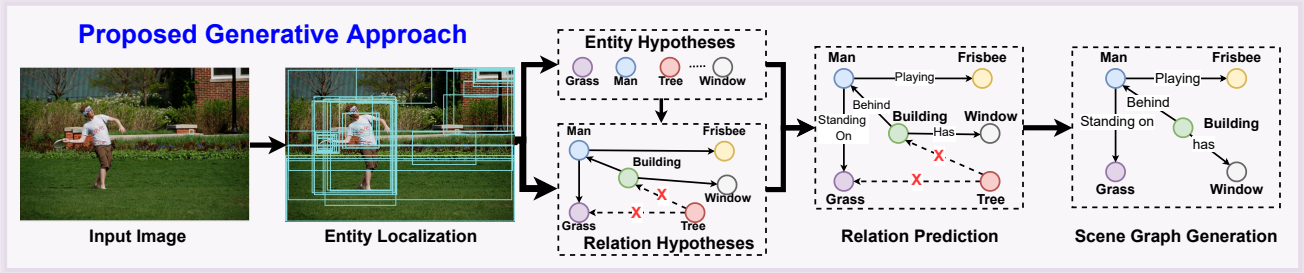

Figure 1. Our goal is to move towards a *generative* model for scene graph generation using a two-stage approach where we first sample the underlying semantic structure between entities before predicate classification. This is different from the conventional approach of modeling pairwise relationships among all detected entities and helps constrain the reasoning to the underlying semantic structure.

of interaction (both semantic and visual) and hence reduce the computational overhead during inference. Additionally, the first step of generative graph sampling (Section 3.2) allows us to navigate clutter by rejecting detected entities that do not add to the semantic structure of the scene by iteratively constructing the underlying entity interaction graph conditioned on the input image. A relation prediction model (Section 3.3) reasons over this constrained edge list to classify the relationships among interacting entities. Hence, the relational reasoning mechanism only considers the (predicted) global semantic structure of the scene and makes more coherent relationship predictions that help tackle the long-tail distribution problem without additional unbiasing steps and computational overhead.

Contributions. The contributions of this paper are three-fold: (i) we are among the first to tackle the problem of scene graph generation using a *graph generative* approach without constructing expensive, pairwise comparisons between all detected entities, (ii) we propose the idea of iterative interaction graph generation and global, contextualized relational reasoning using a two-stage transformer-based architecture for effective reasoning over cluttered, complex semantic structures, and (iii) through extensive evaluation on Visual Genome [19] we show that the proposed approach achieves state-of-the-art performance (without unbiasing) across all three scene graph generation tasks while considering only 20% of all possible pairwise edges using an effective graph sampling approach.

2. Related Work

Scene graph generation, introduced by Johnson *et al.* [17], aims to construct graph-based representations that capture the rich semantic structure of scenes by modeling objects, their interaction and the relationships between them. Most approaches to scene graph generation have followed a typical pipeline: object detection followed by *pairwise* interaction modeling to generate plausible *(Subject, Predicate, Object)* tuples, which represent the labeled edge list of the scene graph. Entity localization (i.e., concept grounding) has primarily been tackled through localization and labeling of images through advances in object detection [6, 28]. The relationship or predicate classification for obtaining the edge list tuples has focused mainly on capturing the global and local contexts using mechanisms such as recurrent neural networks and graph neural networks to result in seminal approaches to scene graph generation such as IMP [36], MOTIFS [41], and R-CAGCN [39]. Single-stage methods such as FC-SSG [25] and Relationformer [29], as well relational modeling approaches such as RelTR [9] have integrated context through transformer-based [35] architectures [8, 18]. However, these approaches fail to explicitly tackle the long-tail distributions prevalent in visual scene graphs as proposed by Tang *et al.* [33] and Chen *et al.* [7].

Unbiased scene graph generation models explicitly tackle this problem by building upon SGG models such as VCTree and MOTIFs to provide better predicate classification. Several approaches have been successfully applied to tackle unbiased generation, such as using external knowledge (VCTree [33] and KERN [7]), counterfactual reasoning (TDE [32]), energy-based loss functions (EBML [31]), modeling predicate probability distributions (PPDL [21] and PCPL [37]), graphical contrastive losses [42], cognitive trees (CogTree [40]), bi-level sampling [20], and regularized unrolling (RU-Net [24]), to name a few. However, these approaches still perform expensive pairwise comparisons to obtain the final scene graph as a collection of tuples rather than directly modeling the underlying semantic structure. Instead of considering graph generation as tuple detection, we build upon an exciting avenue of research in *graph generative models* [5, 13, 15, 23] to directly sample graph structures conditioned on images. By modeling the graph generation process as sequential decoding of adjacency lists, we can effectively model the interaction between detected entities using a simple, directed graph. A transformer-based relation classification model then converts the simple graph into a labeled, weighted, directed graph to generate scene graphs in an iterative, two-stage approach to move beyond edge classification-based detection.

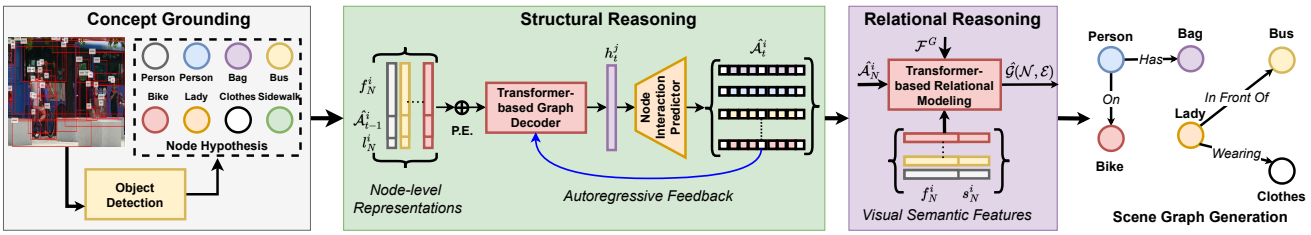

Figure 2. The **overall architecture** of the proposed IS-GGT is illustrated. We first ground the concepts in the image data (Section 3.1) and use a generative transformer decoder network to sample an entity interaction graph (Section 3.2) before relation or predicate classification (Section 3.3) using a transformer-based contextualization mechanism for efficient scene graph generation.

3. Proposed Approach

Overview. We take a two-stage, generative approach to the problem of scene graph generation. The overall approach, called IS-GGT [1], is shown in Figure 2. There are three major components to the approach: (i) concept grounding, (ii) structural reasoning, and (iii) relational reasoning. Based on the idea of generative graph models, we use scene-level localization and entity concept hypothesis (Section 3.1) to first sample the underlying semantic structure of the scene using a generative transformer decoder network (Section 3.2). Once the semantic structure is sampled, the semantic relations (predicates), i.e., the edges, are labeled to characterize the scene graph (Section 3.3).

Problem Statement. Scene graph generation (SGG) aims to generate a graph structure $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ from a given input image I, where $\mathcal{V} = \{v_1, v_2, \dots v_n\}$ is the graph's nodes representing localized entities (objects) in the image and $\mathcal{E} = \{e_1, e_2, \dots e_k\}$ represent the edges that describe the relationship connecting two nodes n_i and n_j. Each node $v_i \in \mathcal{V}$ has two attributes, a label $l_i \in \mathcal{C}_\mathcal{N}$ and a bounding box bb_i, where $\mathcal{C}_\mathcal{N}$ is the space of all possible concepts in an environment. Each edge $e_i \in \mathcal{E}$ is characterized by a label $r_i \in \mathcal{R}_\mathcal{K}$ and an optional assertion score $p(r_i)$, where $\mathcal{R}_K$ is the set of all possible relationships that can be present between the entities $\mathcal{C}_\mathcal{N}$. Current approaches have focused on extracting plausible triplets from an exhaustive search space consisting of all possible edges. Each node is connected to every other node. A relational prediction model is then trained to distinguish between the plausible relationship between the nodes, including *null* relationship. In contrast, we first sample the underlying semantic structure based on the node (entity) hypothesis to model the global context before relationship classification. This helps reduce the computational overload for relationship prediction while restricting the relational reasoning to interactions that are considered to be plausible. We present the proposed framework below.

[1] https://saakur.github.io/Projects/IS_GGT/

3.1. Concept Grounding: Entity Hypotheses

The scene graph generation process begins with entity hypotheses generation, which involves the localization and recognition of concepts in a given image I. Following prior work [33, 36, 41], we use a standard ResNet-based [12], FasterRCNN [28] model as the localization module. The object detector returns a set of n detected entities $v_1, v_2, \dots v_n$, characterized by their location using bounding boxes ($bb_1, bb_2, \dots bb_n \in \mathcal{B}$) and corresponding labels ($l_1, l_2 \dots l_n \mid l_i \in C_N$). These entities ($\mathcal{V}$) serve as our node hypothesis space, over which the scene graph generation is conditioned. Each entity is described by a feature representation (f_N^i) from the underlying ResNet encoder, through ROIAlign [11] using the predicted bounding boxes (ROIs) and the labels are generated through the classification layer from the object detector. Compared to prior work [33, 41], we do not have separate visual encoders for capturing the relationships among concepts at this stage. We allow the entities to be detected and represented independently, which enables us to decouple the ideas of graph prediction and predicate classification.

3.2. Iterative Interaction Graph Generation

At the core of our approach is the idea of graph sampling, where we first model the interactions between the detected entities in a graph structure. This sampled graph is a *simple*, *directed* graph, where the edges are present only between nodes (i.e., the detected entities) that share a semantically meaningful relationship. Each edge e_i is unlabeled and merely signifies the plausible existence of a semantic relationship between the connecting nodes v_i and v_j. Inspired by the success of prior work [5], we model this graph generation process as the autoregressive decoding of the adjacency list $\mathcal{A}_N^i$ for each node v_i, using a transformer network [35]. A simplified pseudocode of the whole process is shown in Algorithm 1. Given an empty graph $\mathcal{G} = \emptyset$, the underlying structural graph is generated through a sequence of edge and node additions. Each step of the decoding process emits an output adjacency list conditioned

Algorithm 1 Scene semantic graph structure sampling using a generative transformer decoder.

Input: $\mathcal{V} = v_1, v_2, \dots v_n \mid v_i = \{l_i, f_N^i, bb_i\}$
Output: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\} = \hat{\mathcal{A}_N} = \{\hat{\mathcal{A}_N^i}\}$

1: $\mathcal{G} \leftarrow \emptyset$ ▷ *Initialize empty graph*
2: $\hat{\mathcal{A}_N} \leftarrow \emptyset$ ▷ *Initialize empty adjacency matrix*
3: $\mathcal{E} \leftarrow \emptyset$ ▷ *Initialize empty edge list*
4: **for** each node v_i in $\mathcal{V}$ **do**
5: $c_t \leftarrow [f_N^{1:i}; \hat{\mathcal{A}_N^{1:i}}, l_{1:i}]$ ▷ *Context vector for decoding.*
6: $c_t \leftarrow c_t + PositionalEncoding(c)$
7: $h_t^0 \leftarrow MLP(c_t)$ ▷ *Linear projection*
8: $\hat{h}_l^K \leftarrow TransformerDecoder(h_t^0)$
9: $h_t^i \leftarrow MLP(\hat{h}_l^K)$ ▷ *Learned feature space*
10: $\hat{\mathcal{A}_N^i} \leftarrow Sample(\sigma(MLP(h_t))$ ▷ *v_i's adjacency list*
11: $\hat{l}_i \leftarrow Softmax(MLP(h_t^i))$ ▷ v_i's auxiliary label
12: $\hat{\mathcal{A}_N} \leftarrow \hat{\mathcal{A}_N} \bigcup \{\hat{\mathcal{A}_N^i}\}$ ▷ *Populate adjacency matrix*
13: $\mathcal{E} \leftarrow \mathcal{E} \bigcup EdgeList(\hat{\mathcal{A}_N^i})$ ▷ *Collect edge list*
14: **end for**
15: $\mathcal{G} \leftarrow \{\mathcal{V}, \mathcal{E}\}$ ▷ *Construct final interaction graph*

upon the visual features f_N^i of each detected node v_i, its hypothesized label l_i and the previously decoded adjacency matrices up to the current step t given by $\hat{\mathcal{A}}_t$. This iterative graph generation process results in an adjacency matrix $\hat{\mathcal{A}} = \{\mathcal{A}_N^1, \mathcal{A}_N^2 \dots \mathcal{A}_N^n, \forall v_i \in \mathcal{V}\}$. The final adjacency matrix is an $N \times N$ matrix that can be sampled by some threshold γ to obtain a binary adjacency matrix. The values where $\hat{\mathcal{A}}_N^i(i,j) = 1$'s indicate that an edge is present between nodes v_i and v_j, which can then be added to the edge list $\mathcal{E}$. The edge list is then sorted by its *energy*, given by $E(e_{ij}) = \sigma(p_i + p_j)$, where p_i and p_j refer to the confidence scores from the detector that provides a measure of confidence about the existence of the concepts v_i and v_j in the image, respectively. The collection of nodes $\mathcal{V}$ and edge list $\mathcal{E}$ provide the underlying semantic structure.

Formally, we define this process as maximizing the probability of observing a scene graph $\mathcal{G}$ conditioned on the input image I, and is given by

$$P(G \mid I) = P(\hat{\mathcal{A}}_N|I) = \prod_{i=1}^{N} p(\hat{\mathcal{A}}_N^i \mid \hat{\mathcal{A}^{1:i}}_N, f_N^{1:i}, l_{1:i}) \tag{1}$$

where we decompose the probability of observing the graph $\mathcal{G}$ as the joint probability over the separate adjacency lists for each node v_i given its visual features f_N^i and label l_i, along with the other nodes that have previously been sampled. Note that the ordering of the nodes can vary greatly; thus, search space to learn the sequence of adjacency lists can grow exponentially with the number of nodes. To this end, we present a fixed ordering of the nodes to be added to the graph based on the confidence score from the object detector to provide a tractable solution. We use a transformer-based decoder model trained in an auto-regressive manner to learn the probability measure.

The decoder is trained using two loss functions - an adjacency loss $\mathcal{L}_\mathcal{A}$ and a semantic loss $\mathcal{L}_\mathcal{S}$. The former is a binary cross-entropy loss between the predicted and actual binary adjacency matrix, while the latter is a cross-entropy loss for node label prediction. Specifically, we define $\mathcal{L}_\mathcal{A} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} -(a_{ij}log(\hat{a_{ij}}) + (1-a_{ij})log(1-\hat{a_{ij}}))$ and $\mathcal{L}_\mathcal{S} = -\sum_j^\mathcal{C} l_j log(p(\hat{l}_j))$, where l_j is the entity's label as predicted by the concept grounding module from Section 3.1 and $\hat{l}_j$ is the softmax probability from the node prediction of the transformer decoder as defined in line 11 of Algorithm 1. Note that we use the semantic loss $\mathcal{L}_\mathcal{S}$ as a mechanism to inject the semantics of the grounded concepts into the decoding process and do not use these predictions (termed *node sampling*) as node labels for the final graph. We observe that node sampling (see Section 4.3) reduces the performance slightly. We attribute it to the fact the object detector has access to the global and image-level context. We train with the combined loss is given by

$$\mathcal{L}_G = \lambda\mathcal{L}_\mathcal{A} + (1-\lambda)\mathcal{L}_\mathcal{S} \tag{2}$$

where λ is a trade-off between semantic and adjacency losses. In our experiments, we set $\lambda = 0.75$ to place more emphasis on the adjacency loss. During training, we use teacher forcing in the transformer decoder and convert the adjacency matrix to binary for tractable optimization.

3.3. Edge Labeling: Relation Prediction

The final step in the proposed approach is predicate (or entity relation) prediction, which involves the labeling of the edges $\mathcal{E}$ in the interaction graph $\mathcal{G}$ generated from Section 3.2. To further refine the interaction graph, we assign an "edge prior" to each sampled edge $e_{ij} \in \mathcal{E}$ between two nodes n_i and n_j. This prior is a function of the confidence scores (c_i and c_j, respectively) obtained from the concept grounding module (Section 3.1) and is given by $E(e_{ij}) = \sigma(c_i \times c_j)$. Finally, we sort the edges based on their edge prior and take the top K edges as the final edge list to represent the scene graph $\mathcal{G}_s$. In our experiments, we set K=250 to provide a tradeoff between inference time and expressiveness, although we find that lower values of K do not reduce the performance (see Section 4.2). Given the final edge list $\mathcal{E}$, we then predict the relationship by maximizing the probability $P(r_k \mid f_N^i, f_N^j, S_N^i, S_N^j, bb_i, bb_j, \mathcal{F}^G)$, where $\mathcal{F}^G$ is the global image context captured by a contextualization mechanism, and r_k is the relationship of the k^{th} edge between nodes n_i and n_j described by their visual features f_N^i and f_N^j, and semantic features S_N^i and S_N^j, respectively. We obtain the contextualized global features

	Approach	PredCls		SGCls		SGDet		Average	Average
		mR@50	mR@100	mR@50	mR@100	mR@50	mR@100	mR@100	mR@50
Without Unbiasing	FC-SSG [25]	6.3	7.1	3.7	4.1	3.6	4.2	4.5	5.1
	IMP [36]	9.8	10.5	5.8	6.0	3.8	4.8	7.1	6.5
	MOTIFS [41]	14.0	15.3	7.7	8.2	5.7	6.6	10.0	9.1
	VCTree [33]	17.9	19.4	10.1	10.8	6.9	8.0	12.7	11.6
	KERN [7]	-	19.2	-	10	-	7.3	12.2	-
	R-CAGCN [38]	-	19.9	-	11.1	-	8.8	13.3	-
	Transformer [10]	-	17.5	-	10.2	-	8.8	12.2	-
	Relationformer [29]	-	-	-	-	9.3	10.7	-	-
	RelTR [9]	21.2	-	11.4	-	8.5	-	-	13.7
	IS-GGT (Ours)	**26.4**	**31.9**	**15.8**	**18.9**	**9.1**	**11.3**	**20.7**	**17.1**
With Unbiasing	RU-Net [24]	-	24.2	-	14.6	-	10.8	16.5	-
	IMP+EBML [31]	11.8	12.8	6.8	7.2	4.2	5.4	8.46	7.6
	VCTree+EBML [31]	18.2	19.7	12.5	13.5	7.7	9.1	14.1	12.8
	MOTIFS+EBML [31]	18.0	19.5	10.2	11	7.7	9.1	13.2	12.0
	MOTIFS+TDE [32]	25.5	29.1	13.1	14.9	8.2	9.8	17.9	15.6
	VCTree+TDE [32]	25.4	28.7	12.2	14	9.3	11.1	17.9	15.6
	MOTIFS+CogTree [40]	26.4	29	14.9	16.1	10.4	11.8	19.0	17.2
	VCTree+CogTree [40]	27.6	29.7	18.8	19.9	10.4	12.1	20.6	18.9
	IMP+PPDL [21]	24.8	25.3	14.2	15.9	9.8	10.4	17.2	16.2
	MOTIFS+PPDL [21]	32.2	33.3	17.5	18.2	11.4	13.5	21.7	20.4
	VCTree+PPDL [21]	33.3	33.8	21.8	22.4	11.3	14.4	23.5	22.1
	BGNN [20]	30.4	32.9	14.3	16.5	10.7	12.6	20.7	18.5
	PCPL [37]	35.2	37.8	18.6	19.6	9.5	11.7	23.0	21.1

Table 1. **Comparison with the state-of-the-art** scene graph generation approaches, with and without unbiasing. We consistently outperform all models that do not use unbiasing and some early unbiasing models across all three tasks while offering competitive performance to current state-of-the-art unbiasing models. Approaches outperforming the proposed IS-GGT are underlined.

$\mathcal{F}^G$ using DETR [6]. The semantic features are obtained through an embedding layer initialized by pre-trained word embeddings of the concept labels $\mathcal{C}$ such as GloVe [26] or ConceptNet Numberbatch [30]. We use an encoder-decoder transformer [35] to model this probability. Specifically, we use a linear projection to map the entity features (visual features F_N^i and localization features bb_i) of each node in the edge $e_k = e_{ij} \in \mathcal{E}$ into a shared visual embedding space by $\hat{h}_v^k = RELU(W_c[f_N^i; bb_i; f_N^j; bb_j])$. A visual-semantic entity embedding is obtained by a linear projection and is given by $\hat{h}_{sv}^k = RELU(W_{sv}[\hat{h}^k; S_N^i, S_N^j])$. An encoder-decoder transformer then takes these visual-semantic features to predict the relationship through a series of attention-based operations given by

$$h_{sv}^k = Att_{enc}^E(Q = K = V = \hat{h}_{sv}^k) \quad (3)$$

$$\hat{h}^k = Att_{dec}^D(Q = \hat{h}_{sv}^k, K = V = \mathcal{F}^G) \quad (4)$$

where $Att_{enc}^E(\dots)$ is a transformer encoder consisting of E multi-headed attention layer $(MHA(Q, K, V){=}W_a[h_1; h_2; \dots h_K])$, as proposed in Vaswani *et al.* [35], where $h_i{=}Attn(Q{=}W_Q X, K{=}W_K X, V{=}W_V X)$. The multi-headed attention mechanism applies a scaled dot product attention operation given by $Attn(Q, K, V){=}Softmax(\frac{QK^T}{\sqrt{D_K}}V)$. The resulting vector h_{sv}^k is then passed through a D-layer transformer decoder that obtains a contextualized representation $\hat{h}^k$ for each edge e_k with respect to the global context $\mathcal{F}^G$. The relationship (or predicate) for each edge is obtained by applying a linear layer on $\hat{h}_k$ followed by softmax to obtain the probability of a relationship $p(\hat{r}_k)$. We train this network using a weighted cross-entropy loss given by

$$\mathcal{L}_R = -w_r \sum_{l=1}^{\mathcal{C}_\mathcal{N}} r_k log(\hat{r}_k) \quad (5)$$

where r_k is the target relationship class, $\hat{r}_k$ is the probability of the predicted relationship class and w_r is the weight given to correct relationship class. In our experiments, we set the weights as the inverse of the normalized frequency of occurrence of each relationship $r_k \in \mathcal{C}_\mathcal{N}$. The weighted cross-entropy allows us to address the long-tail distribution

Approach	PredCls zR@{20/50}	SGCls zR@{20/50}	SGDet zR@{20/50}	Mean zR@{20/50}
VCTree [33]	1.4 / 4.0	0.4 / 1.2	0.2 / 0.5	0.7 / 1.9
MOTIFS [41]	1.3 / 3.6	0.4 / 0.8	0.0 / 0.4	0.6 / 1.7
FC-SGG [25]	-/7.9	-/1.7	-/0.9	-/3.5
VCTree + EBML [31]	2.3 / 5.4	0.9 / 1.9	0.2 / 0.5	1.1 / 2.6
MOTIFS + EBML [31]	2.1 / 4.9	0.5 / 1.3	0.1 / 0.2	0.9 / 2.1
IS-GGT (Ours)	**5.0 / 8.3**	**1.4 / 2.6**	**1.0 / 1.3**	**2.5 / 4.1**

Table 2. **Zero-shot evaluation** on Visual Genome. We report the recall@20 and recall@50 for fair comparison.

of the predicate relationships in the scene graph classification task in a simple yet efficient manner.

Implementation Details. In our experiments, we use a Faster RCNN model with ResNet-101 [12] as its backbone, trained on Visual Genome, and freeze the detector layers [1]. The features extracted from the object detector were 2048 dimensions and were filtered to obtain bounding boxes specific to the target vocabulary. The iterative graph decoder from Section 3.2 has a hidden size of dimension 256 and 6 layers with a sinusoidal positional encoding and is trained for 50 epochs with a learning rate of 0.001. The predicate classifier (Section 3.3) is set to have 256 in its hidden state for both networks, and GloVe embeddings [26] with 300-d vectors are used to derive the semantic features S_N^i. The predicate classifier is trained for 20 epochs with a learning rate of 1×10^{-4}. The training took around 3 hours for both networks on a GPU server with a 64-core AMD Threadripper processer and 2 NVIDIA Titan RTX GPUs.

4. Experimental Evaluation

Data. We evaluate our approach on Visual Genome [19]. Following prior works [7, 33, 36, 41], we use the standard scene graph evaluation subset containing 108k images with 150 object (entity) classes sharing 50 types of relationships (predicates). We use the 70% of the data for training, whose subset of 5,000 images is used for validation, and the remaining 30% is used for evaluation. We evaluate our approach on three standard scene graph generation tasks - predicate classification (**PredCls**), scene graph classification (**SGCls**), and scene graph generation (**SGDet**). The goal of PredCls is to generate the scene graph, given ground truth entities and localization, while in SGCls, the goal is to generate the scene graph, given only entity localization. In SGDet, only the input image is provided, and the goal is to generate the scene graph along with the entity localization.

Metrics and Baselines. Following prior work [7, 9, 33, 38], we report the mean recall (**mR@K**) metric, since the recall has shown to be biased towards predicate classes with larger amounts of training data [7,33]. We report across different values of $K \in \{50, 100\}$ We also present the average mR@K across all tasks to summarize the performance of the scene graph generation models across the three tasks with varying difficulty. We also report the zero-shot recall

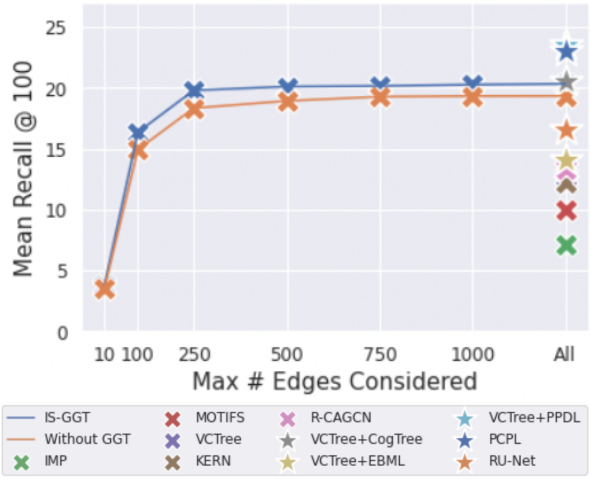

Figure 3. **Impact of graph sampling.** We greatly reduce the number of pairwise comparisons made for scene graph generation. Using only 200 edges ($\sim$ 20% of all edges), we outperform most state-of-the-art approaches on the mean mR@100 across all tasks.

(**zsR@K**, $K \in \{20, 50\}$) to evaluate the generalization capabilities of the SGG models. Finally, we compare against two broad categories of scene graph generation models - those with unbiasing and those without unbiasing. Unbiasing refers to the use of additional training mechanisms, such as leveraging prior knowledge to tackle the long-tail distribution in predicate classification. All numbers are reported under the *with graph constraint* setting.

4.1. Comparison with State-Of-The-Art

We evaluate our approach on the test split of Visual Genome with the mean recall under graph constraints metric (mR@50 and mR@100) and compare with several state-of-the-art scene graph generation approaches, both with and without unbiasing. The results are summarized in Table 1. Without bells and whistles, we significantly outperform approaches that do not use unbiasing across all three tasks. Interestingly, we outperform the closely related, transformer-based ReITR [9] model by 2.7 points in the average mR@50 metric. In comparison with models with unbiasing, we see that we perform competitively to current state-of-the-art models such as PPDL [21], CogTree [40], and BGNN [20], while outperforming some of the earlier approaches to unbiasing such as EBML [31] and TDE [32] across all tasks. Of particular interest is the comparison with RU-Net [24], a scene graph generation model that jointly models unbiasing and generation in a unified framework, as opposed to other approaches, which primarily focus on improving the predicate classification performance of underlying SGG models. We consistently outperform RU-Net across all three tasks, with an average mR@100 improvement of 3.6 absolute points. It is also remark-

Max Edges Considered	PredCls mR@100	SGCls mR@100	SGDet mR@100	Graph Acc. unconst. (*const.*)
10	4.6	3.3	3.5	11.6 (*9.1*)
100	24.3	14.0	10.8	35.1 (*25.3*)
250	30.1	17.5	11.8	44.2 (*30.7*)
500	30.8	17.6	11.9	49.5 (*33.3*)
750	31.0	17.6	11.9	51.4 (*34.4*)
All	31.4	17.6	12.0	52.7 (*34.8*)

Table 3. The **quality of the sampled edges** is quantified using its impact on the three scene graph generation tasks.

G.C.	V.F.	S.F.	G.S.	PredCls	SGCls	SGDet
✓	✓	✗	✓	28.3	16.5	10.3
✓	✓	C.N.B.	✓	28.5	16.8	11.6
✓	✓	GloVe	✓	**30.1**	**17.4**	**11.9**
✓	✗	GloVe	✓	29.2	15.2	10.0
✗	✓	C.N.B	✓	28.5	16.9	11.0
✗	✓	GloVe	✓	29.3	16.9	10.5
✗	✓	GloVe	✗	27.9	16.1	11.0
✓	✓	GloVe	✗	28.5	16.8	11.2
✓	✓	GloVe	W/o E.P.	N/A	17.2	9.3
✓	✓	GloVe	With N.S.	28.5	17.2	8.9

Table 4. **Ablation studies** are presented to quantify each component's impact on mR@100. G.C.: global context, V.F.: visual features, S.F: semantic features, G.S.: graph sampling, C.N.B: ConceptNet Numberbatch, E.P. edge prior, and N.S: node sampling.

able to note the performance difference (in mR@100) between the state-of-the-art unbiasing model (PPDL) and our IS-GGT on PredCls is less than 3%, considering that they are optimized specifically for this task, indicating that the graph sampling approach consistently places the edges in the ground truth scene graph in the top 100 edges.

Zero-Shot Evaluation. We also evaluated the generalization capabilities of our approach by considering the zero-shot evaluation setting. Here, the recall (with graph constraint) was computed only on edges (i.e., subject-predicate-object pairs) that were not part of the training set and summarize the results in Table 2. It can be seen that we outperform approaches with and without unbiasing. Specifically, we obtain and average zero-shot recall of 2.2 (at K=20) and 4.0 (at K=50), which is more than 2× the performance of comparable models without unbiasing such as VCTree and MOTIFS while also outperforming the comparable FC-SGG [25] across all three tasks. It is interesting to note that we also outperform EBML [31], which proposes to mitigate the long-tail distribution using an energy-based loss function. Interestingly, our approach, IS-GGT obtains 21.4 zR@100, *without graph constraint*, which outperforms FC-SGG [25] (19.6), VCTree+TDE [32] (17.6), and MOTIFS+TDE [32] (18.2) which are state-of-the-art unbiasing models in the zero-shot regimen.

4.2. Importance of Graph Sampling.

At the core of our approach is the notion of graph sampling, as outlined in Section 3.2. Hence, we examine its impact on the performance of the proposed IS-GGT in more detail. First, we assess the effect of considering the top K edges based on the edge prior (Section 3.3), which directly impacts the number of edges considered in the final graph for predicate classification. We vary the maximum number of edges considered per predicted scene graph from 10 to 1000 and consider all pairwise comparisons for each detected entity. We assess its impact on the average mean recall (mR@100) across all three tasks (PredCls, SGCls, and SGDet) and summarize the result in Figure 3. As can be seen, we outperform all SGG models that do not use unbiasing while considering only the top 100 edges, which represents $\sim 10\%$ of all possible pairwise combinations while at K=200 edges outperform most models *with* unbiasing. Only PCPL [37] and PPDL [21] outperform IS-GGT, although they consider all (> 1000) combinations.

In addition to the impact on the average mR@100, we also assess the *quality* of the underlying graph sampled with the generative graph transformer decoder. We propose two new metrics, unconstrained and constrained graph accuracy, which measure the quality of the sampled edges. In the former, we measure the accuracy of the underlying structure by when both the nodes and edges are unlabeled and binary. In the latter, we only consider the edges to be unlabeled. Note that, in both metrics, for a node to be "correct", its bounding box must have at least 50% overlap with a corresponding ground truth node. We summarize the results in Table 3. It can be seen that the graph accuracy increases with the number of considered edges while plateauing out at around 500 edges. Interestingly, the constrained accuracy, IS-GGT's theoretical upper bound, is 30.7% with only 250 sampled edges. This is a remarkable metric considering that, on average, the number of total possible edges per image can be more than 1000, and more than 30% of the ground truth edges are part of the top 250 edges. These results indicate that the graph sampling does an effective job.

4.3. Ablation Studies

To assess the impact of each component in the proposed IS-GGT framework, we systematically evaluate the framework's performance by exploring alternatives, including the exclusion of each element. Specifically, we assess the impact of three broad categories - (i) use of semantics, (ii) choice of visual features, and (iii) use of graph sampling. We see that the lack of semantic features has a more significant impact, resulting in a reduction of an average of 1.47% in absolute mR@100 across tasks. In contrast, the choice of semantic features (ConceptNet Numberbatch [30] vs. GloVe [26]) has limited impact. We attribute the success

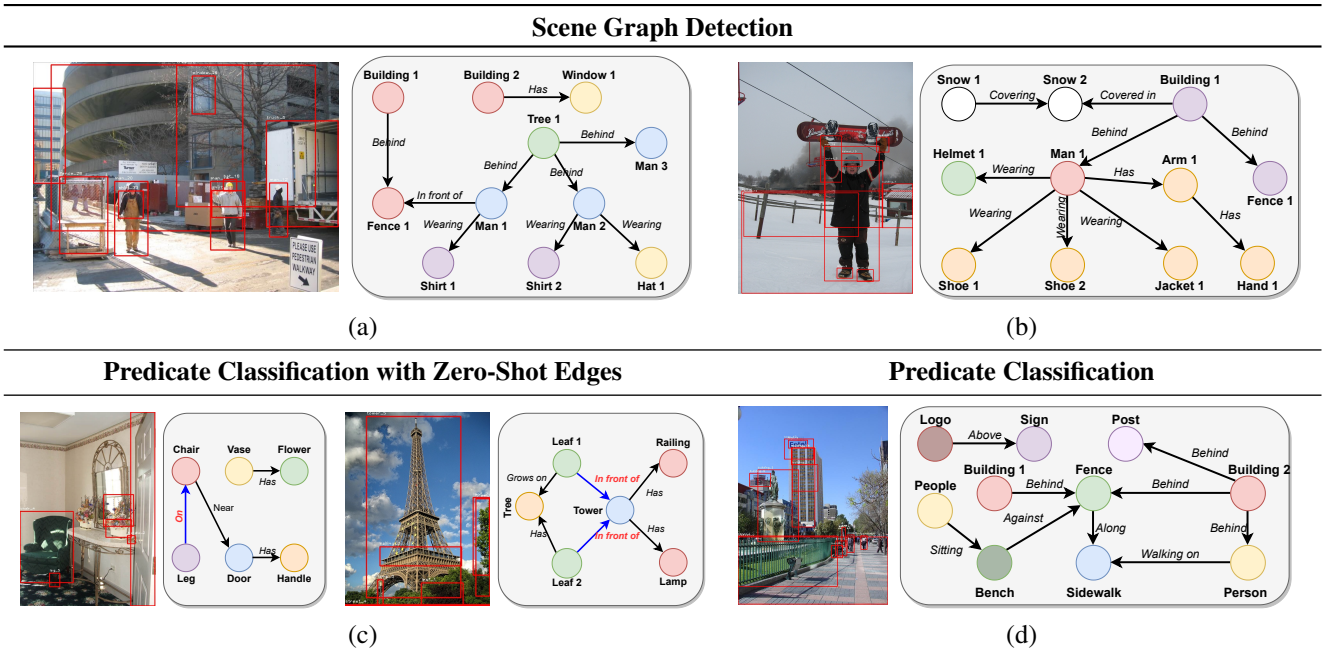

Figure 4. We present **qualitative visualizations** of the scene graphs generated by IS-GGT under (a) scene graph detection setting, (b) predicate classification on images with zero-shot predicates (indicated in blue), and (c,d) predicate classification with complex structures.

of GloVe to its pre-training objective, which ensures that the dot product between GloVe embeddings is proportional to their co-occurrence frequency. This property helps model the potential semantic relationships between nodes using the attention mechanism in relationship prediction model (Section 3.3). Interestingly, we see that adding global context as part of the predicate prediction features (Section 3.3) significantly improves the performance ($\sim$ 1.1% average mR@100), whereas removing visual context altogether also results in a reduction of $\sim$ 1.7% average mR@100. Removing the GGT and removing the edge prior also hurt the performance. However, the recall does not accurately capture the reduction in false alarms due to the lack of edge sampling with a generative model. Finally, we see that using node sampling ($\hat{l}_i$ from Section 3.2) affects SGCls and SGDet. We attribute it to the importance of concept grounding in modeling visual-semantic relationships.

Qualitative Evaluation. We present some qualitative illustrations of some of the scene graphs generated by the proposed approach in Figure 4. In the top row, we present the generated scene graphs under the "detection" setting, where the goal is to both detect entities and characterize the relationships between them. It can be seen that, although there are a large number of detected entities ($\sim$ 28 per image), the graph sampling approach allows us to reject clutter to arrive at a compact representation that captures the underlying semantic structure. Figure 4 (c) shows the generalization capabilities of the proposed approach for predicate classification when previously unseen ("zero-shot") triplets are observed. Finally, we show in Figure 4 (d) that the graph sampling also works under cluttered scenarios, where there is a need to reject nodes that do not add to the scene's semantic structure. We can sample sparse graph structures to express complex semantics without losing expressiveness.

5. Conclusion

In this work, we presented IS-GGT, one of the first works to address the problem of generative graph sampling for scene graph generation. Using a two-stage approach, we first sample the underlying semantic structure of the scene before predicate (relationship) characterization. This decoupled prediction allows us to reason over the constrained (optimal) global semantic structure while reducing the number of pairwise comparisons for predicate classification. Extensive experiments on visual genome indicate that the proposed approach outperforms scene graph generation models without unbiasing while offering competitive performance to those with unbiasing while considering only $\sim$ 20% of the total possible edges. We aim to extend this approach for general graph generation problems such as semantic graphs [2] and temporal graph prediction [4, 16], where capturing the underlying entity interactions can help constrain the search space for complex reasoning.

Acknowledgements. This research was supported in part by the US National Science Foundation grants IIS 2143150, and IIS 1955230.

References

[1] Faster r-cnn with model pretrained on visual genome. https://github.com/shilrley6/Faster-R-CNN-with-model-pretrained-on-Visual-Genome. 6

[2] Sathyanarayanan Aakur, Fillipe DM de Souza, and Sudeep Sarkar. Going deeper with semantics: Video activity interpretation using semantic contextualization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 190–199. IEEE, 2019. 1, 8

[3] Sathyanarayanan N Aakur, Sanjoy Kundu, and Nikhil Gunti. Knowledge guided learning: Open world egocentric action recognition with zero supervision. *Pattern Recognition Letters*, 156:38–45, 2022. 1

[4] Aditi Basu Bal, Ramy Mounir, Sathyanarayanan Aakur, Sudeep Sarkar, and Anuj Srivastava. Bayesian tracking of video graphs using joint kalman smoothing and registration. In *European Conference on Computer Vision*, pages 440–456. Springer, 2022. 1, 8

[5] Davide Belli and Thomas Kipf. Image-conditioned graph generation for road network extraction. *arXiv preprint arXiv:1910.14388*, 2019. 1, 2, 3

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020. 2, 5

[7] 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 (CVPR)*, June 2019. 1, 2, 5, 6

[8] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16372–16382, 2021. 2

[9] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *arXiv preprint arXiv:2201.11460*, 2022. 2, 5, 6

[10] Yuyu Guo, Lianli Gao, Xuanhan Wang, Yuxuan Hu, Xing Xu, Xu Lu, Heng Tao Shen, and Jingkuan Song. From general to specific: Informative scene graph generation via balance adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16383–16392, 2021. 1, 5

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3, 6

[13] Yang He, Ravi Garg, and Amber Roy Chowdhury. Td-road: Top-down road network extraction with holistic graph construction. In *European Conference on Computer Vision*, pages 562–577. Springer, 2022. 2

[14] 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

[15] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 2

[16] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020. 8

[17] 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 (CVPR)*, pages 3668–3678, 2015. 1, 2

[18] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. In *Advances in Neural Information Processing Systems*, 2022. 2

[19] 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 (IJCV)*, 123(1):32–73, 2017. 2, 6

[20] 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 (CVPR)*, pages 11109–11119, June 2021. 2, 5, 6

[21] Wei Li, Haiwei Zhang, Qijie Bai, Guoqing Zhao, Ning Jiang, and Xiaojie Yuan. Ppdl: Predicate probability distribution based loss for unbiased scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, June 2022. 1, 2, 5, 6, 7

[22] Chen Liang, Wenguan Wang, Tianfei Zhou, and Yi Yang. Visual abductive reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15565–15575, 2022. 1

[23] Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 1, 2

[24] Xin Lin, Changxing Ding, Jing Zhang, Yibing Zhan, and Dacheng Tao. Ru-net: Regularized unrolling network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19457–19466, June 2022. 2, 5, 6

[25] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pat-*

tern Recognition (CVPR), pages 11546–11556, June 2021. 2, 5, 6, 7

[26] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. 5, 6, 7

[27] Zachary Ravichandran, Lisa Peng, Nathan Hughes, J Daniel Griffith, and Luca Carlone. Hierarchical representations and explicit memory: Learning effective navigation policies on 3d scene graphs using graph neural networks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9272–9279. IEEE, 2022. 1

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015. 2, 3

[29] Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. *arXiv preprint arXiv:2203.10202*, 2022. 1, 2, 5

[30] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 5, 7

[31] 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 (CVPR)*, pages 13936–13945, 2021. 1, 2, 5, 6, 7

[32] 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 (CVPR)*, pages 3716–3725, 2020. 1, 2, 5, 6, 7

[33] 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 (CVPR)*, June 2019. 1, 2, 3, 5, 6

[34] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2017. 1

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 2, 3, 5

[36] 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 (CVPR)*, pages 5410–5419, 2017. 1, 2, 3, 5, 6

[37] 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*, MM '20, page 265–273, New York, NY, USA, 2020. Association for Computing Machinery. 2, 5, 7

[38] Gengcong Yang, Jingyi Zhang, Yong Zhang, Baoyuan Wu, and Yujiu Yang. Probabilistic modeling of semantic ambiguity for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12527–12536, 2021. 1, 5, 6

[39] 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

[40] J. Yu, Yuan Chai, Yue Hu, and Qi Wu. Cogtree: Cognition tree loss for unbiased scene graph generation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. 2, 5, 6

[41] 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 (CVPR)*, pages 5831–5840, 2018. 1, 2, 3, 5, 6

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