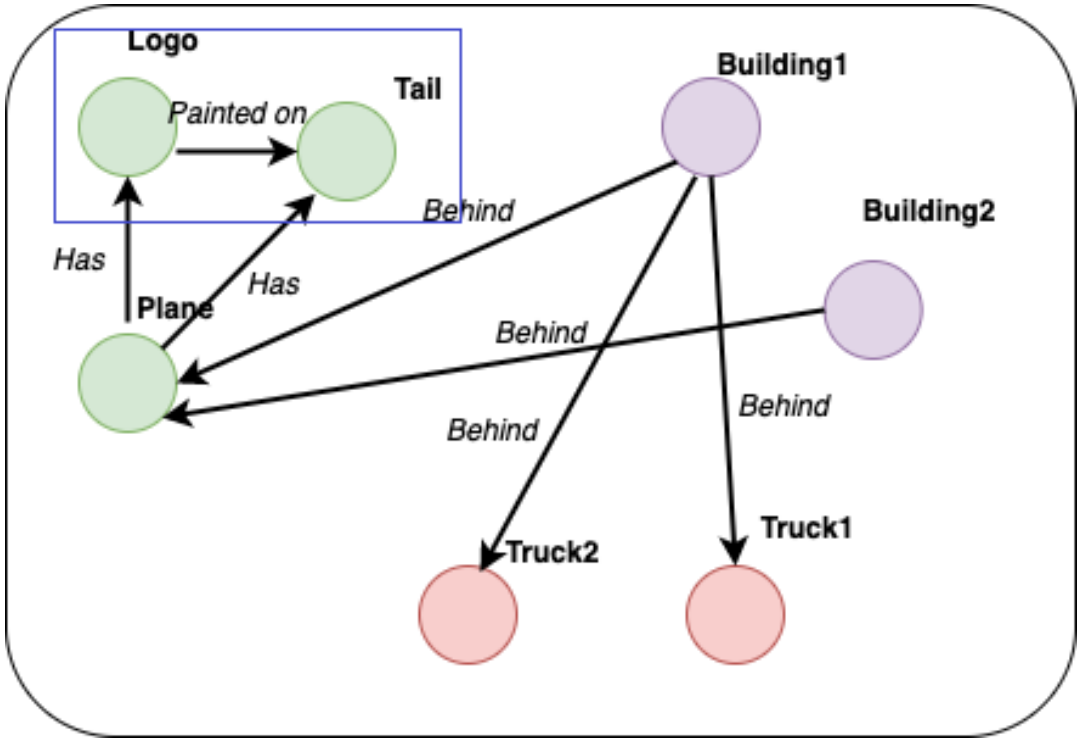
Iterative Scene Graph Generation with Generative Transformers

Sanjoy Kundu, Sathyanarayanan N. Aakur

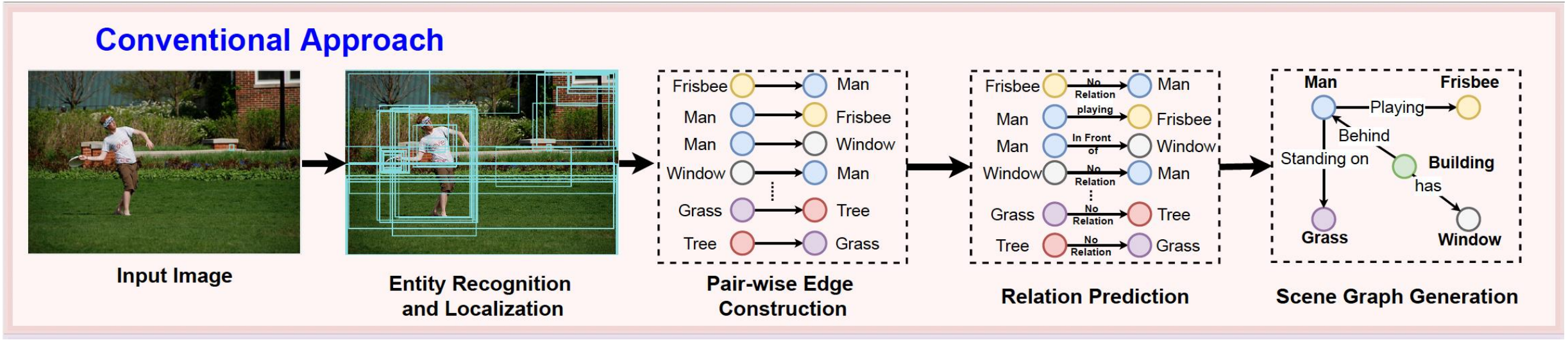
TUE-PM-207

Scene Graphs



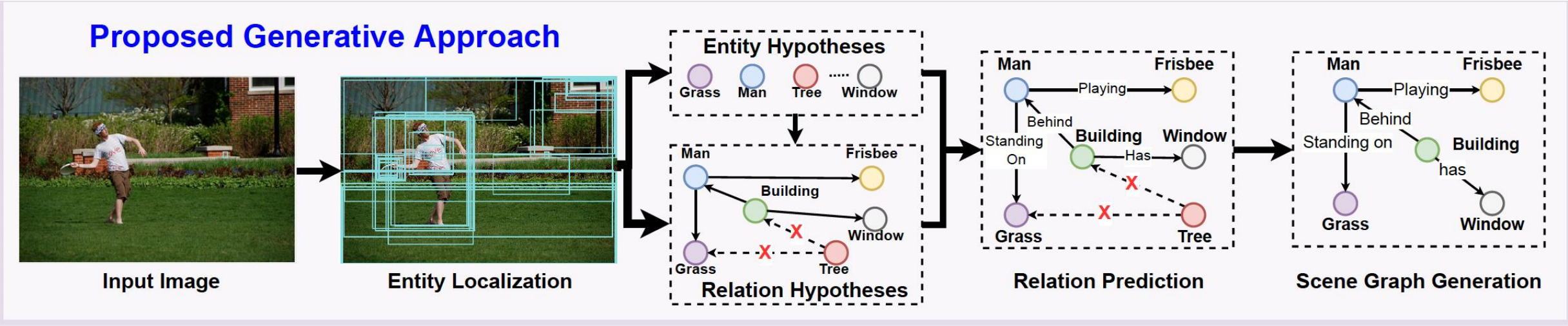
Goal

- conventional approach models pairwise relationships among all detected entities and helps constrain the reasoning to the underlying semantic structure.



Goal

- 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.



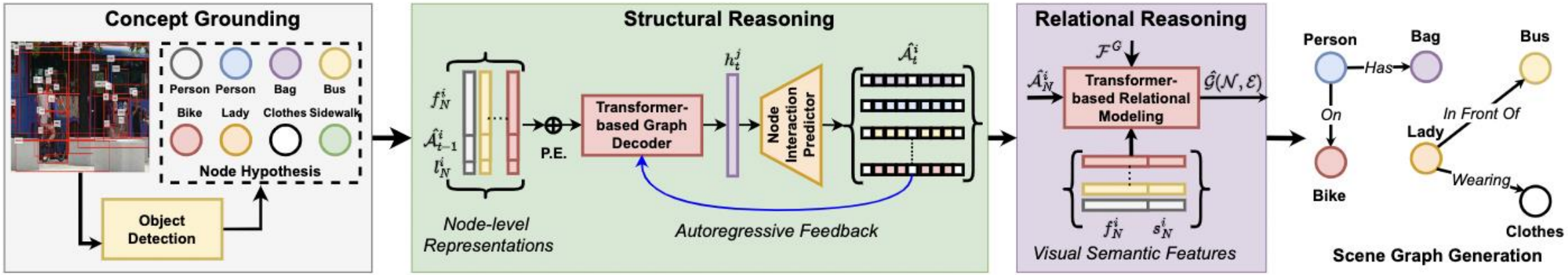
Limitations of current approaches

- 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 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 number of detected concepts.
- Combined, these two issues aggravate the existing long-tail distribution problem in scene graph generation.
- Recent progress in unbiasing 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.

Contributions

- The contributions of this paper are three-fold:
  1. 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
  2. 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
  3. through extensive evaluation on Visual Genome, 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.

How do we do it?



We first ground the concepts in the image data and use a generative transformer decoder network to sample an entity interaction graph before relation or predicate classification using a transformer-based contextualization mechanism for efficient scene graph generation.

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 is unlabeled and merely signifies the plausible existence of a semantic relationship between the connecting nodes v_i and v_j.
- The algorithm is shown here.

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

Input: $\mathcal{V} = v_1, v_2, \ldots v_n \mid v_i = \{l_i, f_N^i, bb_i\}$

Output: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\} = \hat{\mathcal{A}}_N = \{\hat{\mathcal{A}}_N^i\}$

$\mathcal{G} \leftarrow \emptyset$ ▷ *Initialize empty graph*

$\hat{\mathcal{A}}_N \leftarrow \emptyset$ ▷ *Initialize empty adjacency matrix*

$\mathcal{E} \leftarrow \emptyset$ ▷ *Initialize empty edge list*

for each node v_i in $\mathcal{V}$ **do**

  $c_t \leftarrow [f_N^{1:i}; \hat{\mathcal{A}}_N^{1:i}, l_{1:i}]$ ▷ *Context vector for decoding.*

  $c_t \leftarrow c_t + PositionalEncoding(c)$

  $h_t^0 \leftarrow MLP(c_t)$ ▷ *Linear projection*

  $\hat{h}_l^K \leftarrow TransformerDecoder(h_t^0)$

  $h_t^i \leftarrow MLP(\hat{h}_l^K)$ ▷ *Learned feature space*

  $\hat{\mathcal{A}}_N^i \leftarrow Sample(\sigma(MLP(h_t))$ ▷ *v_i's adjacency list*

  $\hat{l}_i \leftarrow Softmax(MLP(h_t^i))$ ▷ *v_i*'s auxiliary label

  $\hat{\mathcal{A}}_N \leftarrow \hat{\mathcal{A}}_N \bigcup \{\hat{\mathcal{A}}_N^i\}$ ▷ *Populate adjacency matrix*

  $\mathcal{E} \leftarrow \mathcal{E} \bigcup EdgeList(\hat{\mathcal{A}}_N^i)$ ▷ *Collect edge list*

end for

$\mathcal{G} \leftarrow \{\mathcal{V}, \mathcal{E}\}$ ▷ *Construct final interaction graph*

Evaluation Setup

- **Data.** We evaluate our approach on Visual Genome. Following prior works, we use the standard scene graph evaluation subset containing 108k images with 150 object (entity) classes sharing 50 types of relationships (predicates).
- **Tasks.** We evaluate our approach on three standard scene graph generation tasks - predicate classification (**PredCls**), scene graph classification (**SGCls**), and scene graph generation (**SGDet**).
  - In PredCls, the goal is to generate the scene graph, given ground truth entities and localization
  - 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. W**e report the mean recall (**mR@K**) metric, since the recall has shown to be biased towards predicate classes with larger amounts of training. We report across different values of K=(50,100). We also report the zero-shot recall to evaluate the generalization capabilities of the SGG models.

Comparison with state-of-the-art

	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 [22]	6.3	7.1	3.7	4.1	3.6	4.2	4.5	5.1
	IMP [33]	9.8	10.5	5.8	6.0	3.8	4.8	7.1	6.5
	MOTIFS [38]	14.0	15.3	7.7	8.2	5.7	6.6	10.0	9.1
	VCTree [30]	17.9	19.4	10.1	10.8	6.9	8.0	12.7	11.6
	KERN [6]	-	19.2	-	10	-	7.3	12.2	-
	R-CAGCN [35]	-	19.9	-	11.1	-	8.8	13.3	-
	Transformer [8]	-	17.5	-	10.2	-	8.8	12.2	-
	Relationformer [26]	-	-	-	-	9.3	10.7	-	-
	RelTR [7]	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 [21]	-	24.2	-	14.6	-	10.8	16.5	-
	IMP+EBML [28]	11.8	12.8	6.8	7.2	4.2	5.4	8.46	7.6
	VCTree+EBML [28]	18.2	19.7	12.5	13.5	7.7	9.1	14.1	12.8
	MOTIFS+EBML [28]	18.0	19.5	10.2	11	7.7	9.1	13.2	12.0
	MOTIFS+TDE [29]	25.5	29.1	13.1	14.9	8.2	9.8	17.9	15.6
	VCTree+TDE [29]	25.4	28.7	12.2	14	9.3	11.1	17.9	15.6
	MOTIFS+CogTree [37]	26.4	29	14.9	16.1	10.4	11.8	19.0	17.2
	VCTree+CogTree [37]	27.6	29.7	18.8	19.9	10.4	12.1	20.6	18.9
	IMP+PPDL [18]	24.8	25.3	14.2	15.9	9.8	10.4	17.2	16.2
	MOTIFS+PPDL [18]	32.2	33.3	17.5	18.2	11.4	13.5	21.7	20.4
	VCTree+PPDL [18]	33.3	33.8	21.8	22.4	11.3	14.4	23.5	22.1
	BGNN [17]	30.4	32.9	14.3	16.5	10.7	12.6	20.7	18.5
	PCPL [34]	35.2	37.8	18.6	19.6	9.5	11.7	23.0	21.1

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.

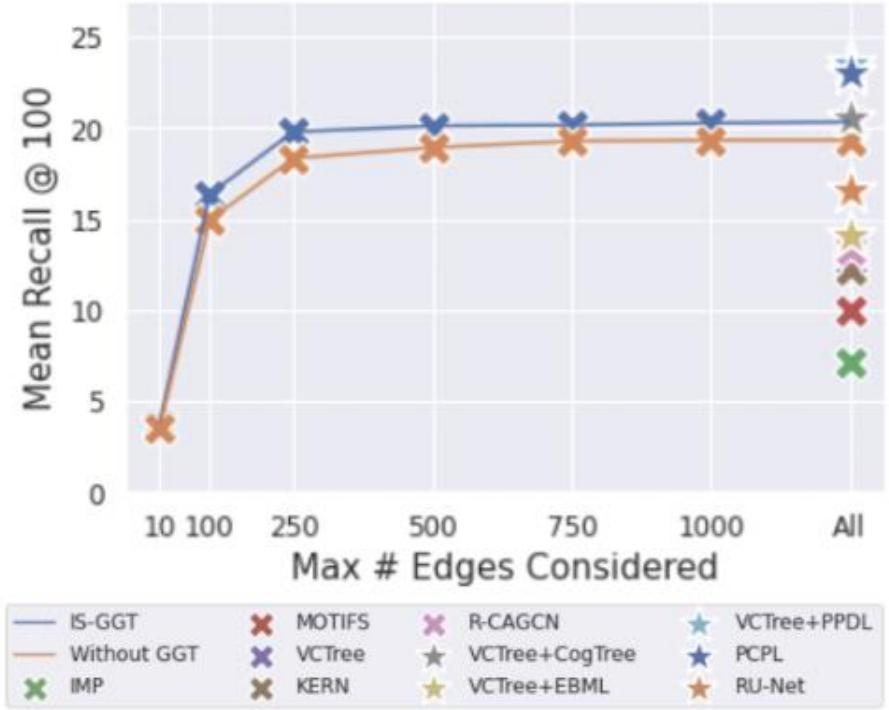
Zero-shot evaluation

Approach	PredCls zR@{20/50}	SGCls zR@{20/50}	SGDet zR@{20/50}	Mean zR@{20/50}
VCTree [30]	1.4 / 4.0	0.4 / 1.2	0.2 / 0.5	0.7 / 1.9
MOTIFS [38]	1.3 / 3.6	0.4 / 0.8	0.0 / 0.4	0.6 / 1.7
FC-SGG [22]	-/7.9	-/1.7	-/0.9	-/3.5
VCTree + EBML [28]	2.3 / 5.4	0.9 / 1.9	0.2 / 0.5	1.1 / 2.6
MOTIFS + EBML [28]	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**

Zero-shot evaluation **on Visual Genome. We report the recall@20 and recall@50 for fair comparison. 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 2x the performance of comparable models without unbiasing such as VCTree and MOTIFS while also outperforming the comparable FC-SGG across all three tasks.**

Impact of Generative Graph Sampling

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*)



We greatly reduce the number of pairwise comparisons made for scene graph generation. Using only 200 edges (i.e., Approximately 20% of all edges), we outperform most state-of-the-art approaches on the mean mR@100 across all tasks.

Qualitative Visualization