EGTR: Extracting Graph from Transformer for Scene Graph Generation

Supplementary Material

Overview of Supplementary Material

This supplementary material provides detailed information not covered in the main manuscript due to space constraints. First, Sec. 1 describes the bipartite matching and the detailed loss function for object detection. Then, in Sec. 2, we introduce the evaluation methods for each dataset used in our study: Visual Genome and Open Image V6. Sec. 3 provides the implementation details. Finally, Sec. 4 shows the results of various additional experiments.

1. Method Details

1.1. Bipartite Matching

We apply the bipartite matching used in DETR [1] to match N predicted objects set $\{\hat{v}_i\}_{i=1}^{N}$ and M ground truth objects set $\{v_i\}_{i=1}^{M}$. Since N is set large enough to handle all objects appearing in the image, we pad the ground truth objects with ϕ (no object). Subsequently, we find the best permutation σ of N predicted objects that minimizes the bipartite matching costs as follows:

$$\sigma = \underset{\hat{\sigma} \in \mathcal{S}_N}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(v_i, \hat{v}_{\hat{\sigma}(i)}), \tag{1}$$

where $\mathcal{S}_N$ denotes all possible permutations of N predicted objects. From the best permutation σ, we denote the permutated predictions as $\{\hat{v}'_i\}_{i=1}^{N}$, where $\hat{v}'_i = \hat{v}_{\sigma(i)}$. The matching cost $\mathcal{L}_{\text{match}}$ is defined as $\mathcal{L}_{\text{match}}(v_i, \hat{v}'_i) = \lambda_{\text{c}} \mathcal{L}^c_{\text{match}}(v^c_i, \hat{v}'^c_i) + \lambda_{\text{b}} \mathcal{L}^b_{\text{match}}(v^b_i, \hat{v}'^b_i)$ with class matching cost $\mathcal{L}^c_{\text{match}}$ and box matching cost $\mathcal{L}^b_{\text{match}}$. Note that the matching cost is not considered when v^c_i is ϕ. $\mathcal{L}^c_{\text{match}}$ is defined as negative likelihood and $\mathcal{L}^b_{\text{match}}$ is defined as $\mathcal{L}^b_{\text{match}}(v^b_i, \hat{v}'^b_i) = \lambda_{\text{IoU}} \mathcal{L}_{\text{IoU}}(v^b_i, \hat{v}'^b_i) + \lambda_{\text{L1}} ||v^b_i, \hat{v}'^b_i||_1$, where $\mathcal{L}_{\text{IoU}}$ indicates generalized intersection over union (IoU) loss [11]. Note that $\mathcal{L}^c_{\text{match}}$ is changed to fit the focal loss [8] for Deformable DETR [20].

1.2. Object Detection Loss

We use the bipartite matching loss proposed in DETR for object detection. From the permutated predictions $\{\hat{v}'_i\}_{i=1}^{N}$, we compute the loss for N matched pairs of the predicted objects and the ground truth objects as follows:

$$\mathcal{L}_{\text{od}} = \sum_{i=1}^{N} [\lambda_c \mathcal{L}_c(v^c_i, \hat{v}'^c_i) + \mathbb{1}_{v^c_i \neq \phi}(\lambda_b \mathcal{L}_b(v^b_i, \hat{v}'^b_i))], \tag{2}$$

where the loss consists of class loss $\mathcal{L}_c$ and box loss $\mathcal{L}_b$ that is the same as box matching cost $\mathcal{L}^b_{\text{match}}$. For $\mathcal{L}_c$, cross-entropy loss is used for DETR, and focal loss is used for Deformable DETR.

2. Evaluation Details

2.1. Visual Genome

Among various evaluation methods for the scene graph generation task, including scene graph detection (SGDET), scene graph classification (SGCLS), and predicate classification (PRDCLS) [17], we adopt the SGDET evaluation protocol due to its rigorous and comprehensive nature compared to other methods. Unlike SGCLS or PRDCLS, where ground truth categories or box coordinates of objects are given, SGDET evaluates the performance of entity categories and box coordinates for each subject and object, as well as predicate categories collectively. We use the widely adopted value of 50% as the thresholding parameter for IoU incorporated in SGDET. We also adopt the graph constraint evaluation method proposed in Zellers *et al.* [18], which enforces a limit of one predicted predicate between a given subject and object entity. We select the top 1 predicate for each object pair, as determined by multiplying the predicate score $\hat{G}_{ijk}$ by the connectivity score $\hat{E}_{ij}$ and multiplying the corresponding class scores of the subject $\hat{v}^c_i$ and object entity $\hat{v}^c_j$. We use recall at k (R@k) and mean recall at k (mR@k) [14] as evaluation metrics. mR@k is a balanced version of R@k in that mR@k can compensate for the bias of predicate categories by aggregating for all predicate categories.

2.2. Open Image V6

Open Image V6 is also assessed in the SGDET setting. We adopt recall and weighted mean AP (wmAP) following the standard settings proposed in Zhang *et al.* [19]. For recall, micro-R@50 is used following previous studies [2, 6, 7, 15]. For wmAP considering the ratio of each predicate category as the weight, wmAP of relationships (wmAP$_{\text{rel}}$) and wmAP of phrases (wmAP$_{\text{phr}}$) are adopted. wmAP$_{\text{rel}}$ evaluates whether both the subject entity and object entity boxes have IoU greater than 50% with the corresponding ground truth boxes. wmAP$_{\text{rel}}$ evaluates the single box that encloses the boxes of the subject entity and the object entity. The final score is calculated by $0.2 \times$ micro-R@50 $+ 0.4 \times$ wmAP$_{\text{rel}}$ $+ 0.4 \times$ wmAP$_{\text{phr}}$. During model inference, we select the top 2 predicates for each object pair following previous works [7, 15].

3. Implementation Details

Our object detector backbone is based on Deformable DETR [20] with ResNet-50 [3]. To shorten the convergence time, we pre-train the object detector backbone us-

f type	# Params(M)	R@50	mR@50
dot product attention	**41.3**	25.9	6.2
dot product	**41.3**	27.4	6.8
Hadamard product	41.5	29.1	7.2
sum	41.5	29.5	7.3
concat	41.6	**29.9**	**7.9**

Table 1. **Ablation for relation function.** Since concat represents the relationship between the attention query and attention key without loss of information, it shows the best performance among all f variants.

$\mathcal{L}_{\text{con}}$	connectivity score	R@50	mR@50
		29.4	7.3
✓		29.4	7.6
✓	✓	**30.2**	**7.9**

Table 2. **Ablation for connectivity loss and score.** $\mathcal{L}_{\text{con}}$ denotes whether the model is trained with loss for connectivity prediction. connectivity score denotes whether we use connectivity score for sorting predicted relation triplets in evaluation. Results show that using connectivity loss and score both improve performance.

λ_{rel}	5	10	**15**	20
R@50	29.1	29.1	**29.4**	28.8
mR@50	7.3	7.3	**7.3**	7.0

Table 3. **Experiments for relation extraction loss coefficient λ_{rel}.** We fix λ_{con} to 0 and conduct experiments on λ_{rel} first.

λ_{con}	0	15	**30**	45
R@50	29.4	29.7	**30.2**	29.6
mR@50	7.3	7.4	**7.9**	7.4

Table 4. **Experiments for connectivity prediction loss coefficient λ_{con}.** λ_{rel} is set to the optimal value 15 from Tab. 3.

ing 8 V100 GPUs with a batch size of 32, employing AdamW [10] optimizer with a default learning rate of 10^{-4} and a decreased learning rate of 10^{-5} for ResNet-50. EGTR is trained on 8 V100 GPUs with a batch size of 64, using a learning rate of 2×10^{-4} for the relation extractor and scaling down the learning rates for the object detector and ResNet-50 by 100 and 1000 times, respectively. Following the original DETR training scheme, we adopt a learning rate schedule that reduces the learning rate by a factor of 10 after the model has trained to some extent. Instead of a fixed learning schedule, we apply an adaptive schedule through early stopping. For the object detector pretraining, we set the maximum number of epochs to 150 for the first schedule and 50 for the second schedule. In the main EGTR training, we configure the first schedule for 50 epochs and the second schedule for 25 epochs. For the hyperparameters of the bipartite matching and object detection loss, we follow the configurations of the original Deformable DETR. λ_{c}, λ_{b}, λ_{IoU}, and λ_{L1} are set to 2, 1, 2, and 5, respectively. For Open Images V6, we only apply hard negative sampling, which is assumed to contribute to mAP performance.

4. Additional Results

4.1. Ablation Studies

Relation Function. We conduct an ablation study on the relation function f, as presented in Tab. 1. To evaluate the impact of different relation functions, we conduct experiments using only the self-attention relation sources $[R_a^1; ...; R_a^L]$ without the final relation source R_z. Furthermore, we do not use linear weights W_S^l and W_O^l for relation source representations; therefore, using the dot product attention function entails utilizing the self-attention weights in their original form. Surprisingly, using only the attention weights of the object detector shows consistently high results, supporting our hypothesis that self-attention contains information relevant to relations. Additionally, excluding only the softmax function from dot product attention significantly improves performance. We also explore different element-wise functions for the relation function, including Hadamard product, sum, and concat. Among these, concat, which preserves the representations of attention query and key, exhibits the best performance.

Connectivity Prediction. We perform ablation studies on the connectivity loss $\mathcal{L}_{\text{con}}$ used during training and the connectivity score employed during inference. As outlined in Tab. 2, both connectivity loss and connectivity score contribute to the performance improvements. These findings indicate that connectivity loss serves as a hint loss for the relation extraction loss, and the connectivity score effectively filters out candidates of object pairs that are less likely to have relations.

4.2. Model Selection

Loss Function. Due to the vast search space for the hyperparameters of the loss function, we first set the connectivity prediction loss coefficient λ_{con} to 0 and explore the relation extraction loss coefficient λ_{rel} in increments of 5, which is the bounding box L1 loss coefficient λ_{L1}, as shown in Tab. 3. Then we explore λ_{con} in multiplies of tuned λ_{rel} as shown in Tab. 4. Since the relation tensor is sparse, relatively high loss coefficients improve the performance.

Adaptive Smoothing. To choose a hyperparameter α representing the minimum uncertainty for adaptive smoothing,

α	R@50	mR@50
10^{-13}	30.1	7.8
$\mathbf{10^{-14}}$	**30.2**	**7.9**
10^{-15}	30.0	7.8

Table 5. **Experiments for adaptive smoothing hyperparameter α.** We set the hyperparameter range of α with the validation uncertainty of the initialized model. Hyperparameters within the range have similar performances; however, 10^{-14} shows the best performance.

k_{neg}	k_{non}	R@50	mR@50
10	10	29.6	**8.2**
20	20	29.9	**8.2**
40	40	30.0	8.1
80	**80**	**30.2**	7.9
160	160	29.9	7.7

Table 6. **Experiments for sampling hyperparameter k_{neg} and k_{non}.** We choose k_{neg} and k_{non} as 80 which shows the best R@50.

we first set the hyperparameter range for α. Since the uncertainty measured through the bipartite matching is sensitive to related configurations such as the number of object queries N and weights used to calculate matching cost $\mathcal{L}_{\text{match}}$, we devise the method to explore the hyperparameter range in advance. We find the hyperparameter range by measuring the validation uncertainty when the model is initialized. To reflect the situation in which the model is initialized with random weights, the hyperparameter range is chosen so that the valid uncertainty can cover a wide range between 0 and 1. We experiment with α of 10^{-13}, 10^{-14}, and 10^{-15} corresponding to valid uncertainties of 0.844, 0.487, and 0.135, respectively. Results shown in Tab. 5 demonstrate that the performance is relatively robust regardless of the hyperparameters. Judging from the fact that 10^{-14} where initial valid uncertainty is 0.487 performs the best, setting initial valid uncertainty close to 0.5 might be suitable for the situation where the model weights are randomly initialized.

Sampling Methodology. We explore hyperparameters k_{neg} and k_{non} for negative sampling and non-matching sampling, respectively. To narrow down the hyperparameter range, we set k_{neg} equal to k_{non}. Results in Tab. 6 illustrate a trade-off, where increasing the sampling coefficients enhances the amount of information on triplets not representing the ground truth relations, and decreasing the coefficient reduces the sparsity of the ground truth relation graph. We select k_{neg} and k_{non} as 80, yielding the best R@50.

In addition to sampling hyperparameters, we perform comprehensive experiments on sampling options, as presented in Tab. 7. For the negative region and non-matching region, we explore the following three options: one that considers the entire region without sampling (-), another that considers only a portion of it through sampling (80), and a third that does not consider the region (0). The results indicate that considering the entire region or ignoring it is suboptimal, and sampling in both regions is crucial for performance.

k_{neg}	k_{non}	R@50	mR@50
0	80	29.4	7.4
80	**80**	**30.2**	**7.9**
-	80	30.0	**7.9**
80	0	29.7	6.8
80	**80**	**30.2**	**7.9**
80	-	29.8	7.2
-	0	29.7	7.0
-	-	29.7	7.2
0	-	29.2	6.8

Table 7. **Experiments for sampling options.** "-" denotes that we use the whole region without sampling. 0 indicates that the region is not considered. Sampling from both negative and non-matching regions shows the best performance.

4.3. Analysis

Backbone. To observe whether the performance improves with a heavier backbone, we conduct experiments using ResNet-101 as the backbone instead of ResNet-50. It shows improved object detection performance and relation extraction performance: AP50 32.3 (+1.5), R@50 30.8 (+0.6), and mR@50 8.1 (+0.2). However, the improvement in the relation extraction performance is relatively lower compared to the enhancement in the object detection performance. We speculate that the capacity of the Transformer decoder might be more crucial than the CNN backbone for the performance of the relation extraction.

Adaptive Smoothing. Since proposed adaptive smoothing can be applied to any one-stage SGG model that utilizes the explicit object detector, we apply the technique to Relationformer [12] and SGTR [7], where object detection loss is used and detected objects are related to relation extraction. We conduct experiments using publicly available code and adapt the technique based on the characteristics of each model. Since Relationformer uses softmax cross-entropy with an additional "no relation" class for the relation extraction, we apply smoothing for the ground truth relation class and compensate the target value of the "no relation" class by the same amount. In our reproduced experiments, it shows improved performance: R@50 $26.61(+0.12)$, mR@50 $8.54(+0.71)$, and ng-R@50 $28.84(+0.76)$ where

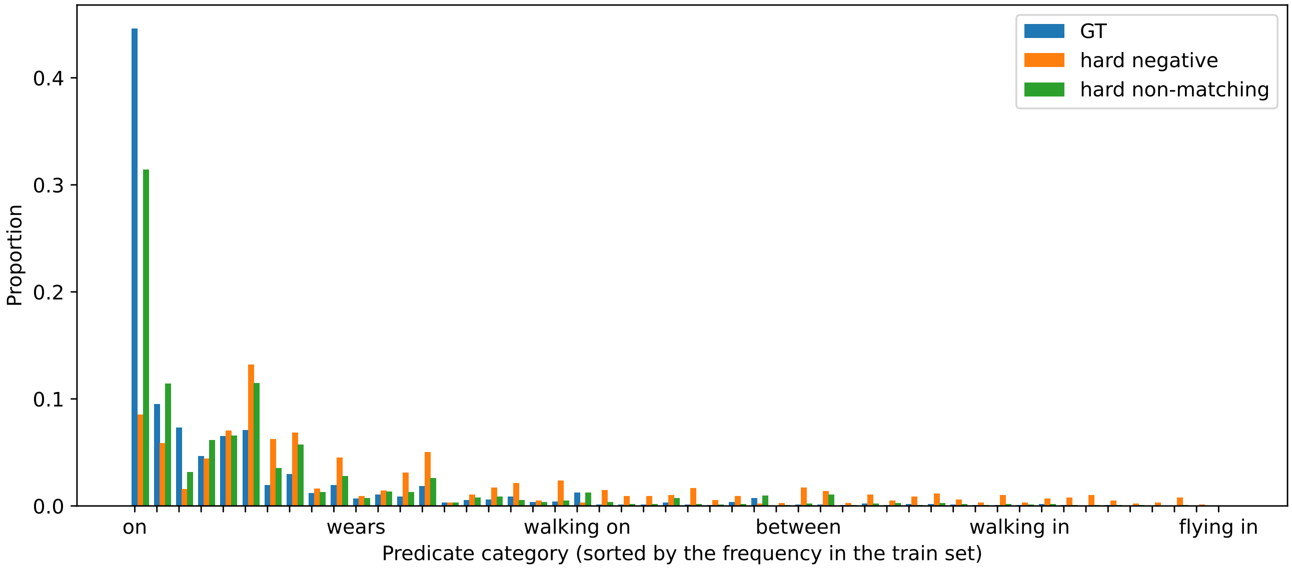

Figure 1. **The comparison of predicate category distribution based on graph regions.** We compare the predicate categories of GT, hard negatives, and hard non-matchings for the validation dataset using histograms. We sort the predicate categories based on their frequency in the training dataset.

ng denotes no graph constraints. Since SGTR matches detected objects with triplets through graph assembling, we apply relation smoothing on predicate labels based on the uncertainties of the detected objects matching the subjects and objects in triplets. Our smoothing method enhances overall performance: R@50 $24.36(+0.04)$ and mR@50 $12.88(+0.75)$. The results demonstrate the generality of the adaptive smoothing. Exploring the possibility of applying the adaptive smoothing based on matching costs of subjects and objects for triplet detection models that do not use an explicit object detector could be an interesting avenue for future research.

Sampling Methodology. We compare the distribution of predicate categories for hard negatives and hard non-matchings with that of the GT as shown in Fig. 1. Since the non-matching region is composed of object candidates that do not match with the ground truth objects and object candidates that closely resemble ground truth objects are selected as hard non-matchings, hard non-matchings exhibit a prevalence of the head predicate categories similar to the GT. On the contrary, hard negatives exhibit a relatively lower proportion of the head categories, and tail categories are more frequently selected. As the negative region is constructed from object candidates that match the ground truth objects, it seems that hard negatives are selected from tail classes that are likely to exist in reality but are not annotated.

Gated Sum. We examine the utilization of relation source representations from each layer in the gating mechanism on Fig. 2. Remarkably, the gate values for the first self-attention source representations, which precede the cross-attention layer and do not utilize any image information, are close to 1.0. Since object queries are trained to reflect the diverse distribution of objects in the training data [13], the relationship information between object queries appears to be utilized as a primary bias, reflecting the various object relationships in the training data. After passing through the cross-attention layer once, the gate values of the second self-attention source representations are very low. However, they gradually increase as the model incorporates image in-

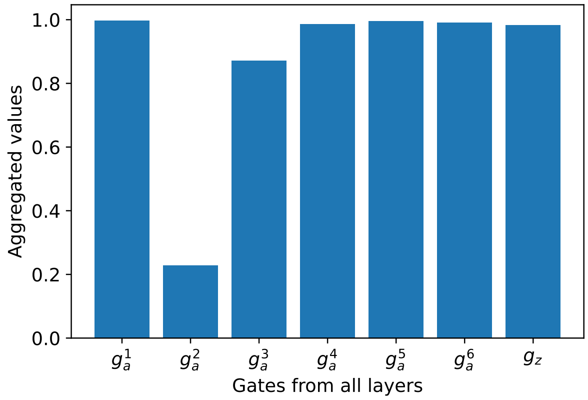

Figure 2. **Aggregated gate values from all layers**. We aggregate the $N \times N$ shaped gate matrices for each layer and report the average over the entire validation dataset.

Model	Backbone	# of params (M)	FPS	MACs (G)
FCSGG [9]	-	87.1	6.0	655.7
RelTR [2]	DETR-50	63.7	13.4	**67.6**
SGTR [7]	DETR-101	117.1	6.2	127.0
Iterative SGG [4]	DETR-101	93.5	6.0	130.3
Relationformer [5]	DDETR-50	92.9	8.5	336.7
EGTR (Ours)	DDETR-50	**42.5**	**14.7**	132.4
SSR-CNN [15]	SRCNN-X101-FPN	274.3	4.0	297.7

Table 8. **Efficiency of one-stage SGG models**. In addition to FPS, we measure MACs, which account for theoretical complexity. "-50" represents ResNet50, "-101" denotes ResNet-101, and "-X101-FPN" signifies ResNeXt-101-FPN [16]. "DDETR" corresponds to Deformable DETR [20], and "SRCNN" corresponds to Sparse-RCNN [13]. For a fair comparison, the image size is set to a minimum of 600 for the shortest side and a maximum of 1000 for the longest side. FPS is measured in a single V100.

formation well. In particular, from the fourth self-attention source representations, the gate values are higher than those of the relation source representations in the final layer.

4.4. Efficiency

As depicted in Tab. 8, we report Multiply-ACcumulation (MACs) to assess efficiency in addition to the number of parameters and frames per second (FPS). MACs quantify the number of multiply and accumulate operations performed by a neural network during the inference phase. It is worth noting that MACs are estimated to be roughly half the number of Floating Point Operations (FLOPs). For a fair comparison, the image size is set to a minimum of 600 for the shortest side and a maximum of 1000 for the longest side.

It seems that EGTR has relatively high MACs, considering the superior efficiency in terms of the number of parameters and FPS. However, it is noteworthy that our MACs are primarily attributed to the Deformable DETR backbone, and additional MACs from our relation extractor are only 16.8G. With 100 object queries, Deformable DETR-50 shows 115.0G MACs, compared to DETR-50 with 56.1G. Although Deformable DETR has more than twice the theoretical complexity compared to DETR, we opt for Deformable DETR due to its notably enhanced convergence speed [20]. Leveraging Deformable DETR as a backbone, we use a lightweight relation extractor composed of only 2.5M parameters, resulting in the fastest inference speed.

4.5. PredCls & SGCls

To assess how well the model can capture the structure of the scene given the ground truth objects information, we provide results for PredCls and SGCls. As they were introduced in the two-stage SGG models to measure relation prediction given ground truth objects, measuring them in one-stage SGG models may involve some arbitrariness. We reviewed one-stage studies [2, 9] that had reported PredCls and SGCls, and carefully designed measurements for one-stage SGG models. We perform bipartite matching for object queries with ground truth objects and replace the prediction of the matched object queries with the corresponding ground truth objects' labels. Note that the representations of object queries used for the relation prediction remain unchanged. As shown in Tab. 9, EGTR performs well in both PredCls and SGCls settings. It demonstrates that SGDet performance of EGTR does not solely depend on high object detection performance but also relation extraction performance.

Models	AP50	R@50	mR@50
FCSGG [9]	28.5	41.0 / 23.5 / 21.3	6.3 / 3.7 / 3.6
RelTR [2]	26.4	*36.0 / 30.5 / 25.2*	*10.8 / 9.3 /* ***8.5***
EGTR (Ours)	**30.8**	**54.3 / 39.8 / 30.2**	**16.6 / 11.9** / 7.9

Table 9. **Comparison with one-stage SGG models on Visual Genome test set.** We report results for PredCls, SGCls, and SGDet settings, separated by "/". *Italic* denotes that we remeasured the score with a publicly available model checkpoint for a fair comparison: the ground truth objects are utilized rather than ground truth triplets in the original RelTR report.

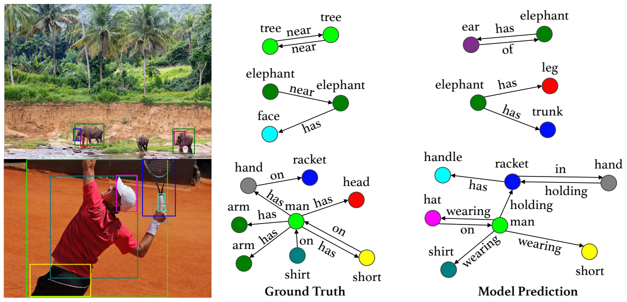

Figure 3. **Qualitative Analysis**. For visualization, we select the same number of predicted triplets as the ground truth triplets within the Visual Genome validation dataset.

4.6. Zero-shot Performance

We have noticed that popular technique frequency baseline [18] directly influences the zero-shot performance in our model. Without the frequency baseline, EGTR demonstrates a commendable zR@50 performance with a score of 2.1 despite a decrease of 0.2 points in R@50 and 0.6 points in mR@50.

4.7. Qualitative Results

In Fig. 3, we present qualitative examples of the Visual Genome validation dataset. The depicted results illustrate the capability of our methodology to generate relationships that are both plausible and semantically rich.

Contribution of Authors

Jinbae Im initiated and led the project, proposed the main ideas, and made significant contributions throughout the process, including implementation, experiments, and manuscript writing. **JeongYeon Nam** implemented experimental ideas, conducted a major part of the experiments, and made significant contributions to the manuscript writing and the development of the paper's direction. **Nokyung Park** managed the reproduction and validation of other models, conducted performance evaluations and comparisons, and contributed to enhancing the model's performance and manuscript writing. **Hyungmin Lee** designed and conducted a proof of concept experiment associated with the model architecture and implemented and conducted experiments mainly related to relation prediction. **Seunghyun Park** co-initiated the project, advised it from its inception, participated in manuscript writing, and significantly contributed to shaping the project's direction as a senior researcher.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1

[2] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE TPAMI*, 2023. 1, 5

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[4] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. In *NeurIPS*, 2022. 5

[5] Rajat Koner, Suprosanna Shit, and Volker Tresp. Relation transformer network. *ECCV*, pages 422–439, 2022. 5

[6] Rongjie Li, Songyang Zhang, Bo Wan, and Xuming He. Bipartite graph network with adaptive message passing for unbiased scene graph generation. In *CVPR*, pages 11109–11119, 2021. 1

[7] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, pages 19486–19496, 2022. 1, 3, 5

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017. 1

[9] Hengyue Liu, Ning Yan, Masood Mortazavi, and Bir Bhanu. Fully convolutional scene graph generation. In *CVPR*, pages 11546–11556, 2021. 5

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[11] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019. 1

[12] 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. In *ECCV*, pages 422–439, 2022. 3

[13] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, pages 14454–14463, 2021. 4, 5

[14] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *CVPR*, pages 6619–6628, 2019. 1

[15] Yao Teng and Limin Wang. Structured sparse r-cnn for direct scene graph generation. In *CVPR*, pages 19437–19446, 2022. 1, 5

[16] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 5

[17] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 5410–5419, 2017. 1

[18] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018. 1, 5

[19] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *CVPR*, pages 11535–11543, 2019. 1

[20] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 5