
Proceedings Paper:
VANICA, GEORGE and BORS, ADRIAN GHEORGHE (2026) OV-SGT: Open Vocabulary Semantic Graph Transformer for Scene Graph Generation. In: IEEE/CVF WACV workshop on Scene Graph for Structured Intelligence. IEEE, Tucson, AZ, USA, pp. 1685-1694.

OV-SGT: Open Vocabulary Semantic Graph Transformer for Scene Graph Generation

George Vanica Adrian G. Bors

Department of Computer Science, University of York, York YO10 5GH, UK
{gfv501, adrian.bors}@york.ac.uk

Abstract

Scene graph generation bridges visual perception and semantic understanding, but existing approaches face two challenges: closed vocabularies that limit real-world applicability and long-tail predicate distributions where common relationships dominate training data. We introduce the Open Vocabulary Semantic Graph Transformer (OV-SGT), which addresses both challenges through CLIP-aligned representation learning. Our method learns relationship embeddings within CLIP's semantic space, enabling zero-shot generalization to unseen predicates. Key contributions include: (1) a node-edge fusion strategy preserving relationship directionality; (2) graph Laplacian eigenvector-based positional encoding capturing structural context; and (3) a multi-component loss combining contrastive, semantic, triplet, and focal objectives for zero-shot transfer while handling class imbalance. Experiments on Visual Genome demonstrate state-of-the-art performance, with significant gains on mean Recall@K metrics reflecting improved rare predicate recognition.

1. Introduction

Scene graph generation (SGG) bridges the gap between scene images and their semantic representations, enabling an understanding of what is depicted in the picture. It takes into account the 3D composition of the scene, the layout, and the semantic as well as the spatial relationship between the objects in the scene. By identifying objects and their relationships, scene graphs provide a structured representation that captures the rich relational context in visual scenes. The scene graph generation task represents the basis for various high level tasks such as: image retrieval [24], video and image captioning [9], image generation [11], visual question answering [6], 3D scene representation [28], scene-based reasoning [1], visual relationship detection [22], human-object interaction [13] and 3D scene synthesis [8], and is essential for robotics and autonomous driving among others.

Despite significant progress in this field, existing SGG approaches typically operate within a closed vocabulary setting, where models are constrained to recognize a predefined set of object categories and relationship types. Such limitations restrict their real-world applicability, as natural scenes often contain a large variety of objects and relationships that may not be represented in the training vocabulary. Most current methods struggle with long-tail distributions of relationship predicates, where a small number of common relationships (such as "on," "has," "in") dominate training data, resulting in poor performance in rare but meaningful relationships.

In this paper, we present the Open Vocabulary Semantic Graph Transformer (OV-SGT), a novel approach to scene graph generation. Our approach begins by detecting objects using state-of-the-art object detection models and enriching their representations with caption-derived features. We then construct a graph structure where nodes represent detected objects and edges represent potential relationships. The core of our method is a specialized graph transformer that processes object-relationship structures. To address the challenge of predicate imbalance, we incorporate frequency-based predicate weighting in our multi-component loss function.

We evaluate OV-SGT on the standard VG-50 configuration [30] with 150 object categories and 50 relationship predicates benchmark. We also evaluate the model on the VRD [22] and GQA [10] datasets. Our model demonstrates significant improvements over others, particularly through the ability to capture rare relationship predicates. The experimental results show that OV-SGT achieves competitive performance in both overall Recall@K metrics and Mean Recall@K metrics, which reflect the performance across the full spectrum of relationship types. Through extensive ablation studies, we analyze the impact of our design choices to these performance gains.

The primary contributions of this paper are:

- A CLIP-aligned training framework that learns relationship embeddings in CLIP's semantic space, enabling open-vocabulary scene graph generation and zero-shot generalization to unseen predicates.
- A specialized graph transformer with a node-edge fusion strategy that concatenates contextually-enhanced source node, target node, and edge attribute features to preserve relationship directionality while incorporating global scene context.
- A graph Laplacian eigenvector-based positional encoding that captures both local object proximity and global structural context within the scene graph.
- A multi-component loss function combining contrastive, semantic, triplet, and focal objectives that addresses class imbalance while maintaining CLIP alignment for zero-shot transfer.

2. Related Work

Scene graph generation research follows two main directions. The first employs a two-stage pipeline where the object detector is pre-trained and frozen during relationship prediction. The second jointly predicts objects and relationships end-to-end, using region proposals as input.

Transformers have proven effective for SGG due to their ability to capture long-range dependencies. Scene Graph Generation Transformer (SGTR) [16] generates subject, object, and predicate proposals followed by graph assembly. Iterative Scene Graph Generation (IterSGG) [12] uses a three-stream transformer with cross-stream attention and Hungarian matching [15] for triplet alignment. Our work builds on these foundations while introducing CLIP-aligned training for open-vocabulary capability.

3. Methodology

3.1. Problem formulation

We formulate the scene graph generation task as a sequence of probabilistic estimation chaining steps comprising of : object region extraction, object identification, relationship proposal and graph labeling. The statistical formulation of the scene graph generation task can be expressed [29], as follows:

$$\mathcal{P}(\mathcal{S} \,|\, \mathcal{I}) = \mathcal{P}(\mathcal{V} \,|\, \mathcal{I})\, \mathcal{P}(\mathcal{O} \,|\, \mathcal{V}, \mathcal{I})\, \mathcal{P}(\mathcal{R} \mid \mathcal{V}, \mathcal{O}, \mathcal{I}), \quad (1)$$

where $\mathcal{P}(\mathcal{S} \,|\, \mathcal{I})$ is the scene $\mathcal{S}$ graph extraction task probability, given an image $\mathcal{I}$, $\mathcal{P}(\mathcal{V} \,|\, \mathcal{I})$ is the object region proposal $\mathcal{V}$, $\mathcal{P}(\mathcal{O} \,|\, \mathcal{V}, \mathcal{I})$ represents the object class identification for the previously identified bounding boxes (object proposal) $\mathcal{O}$, while $\mathcal{P}(\mathcal{R} \,|\, \mathcal{V}, \mathcal{O}, \mathcal{I})$ represents the relationships $\mathcal{R}$ extraction, given the image of the scene, object proposals, and object classes, defined by their corresponding probabilities.

Like SGTR [16] and IterSGG [12], our model uses a convolutional backbone for image feature extraction. These image features are then used for downstream modules.

3.2. Model Pipeline

The scene graph generation model is illustrated in Fig. 1. Our approach combines object detection with CLIP-aligned relationship classification to enable open-vocabulary scene graph generation.

Inference Pipeline. The image is first processed by a DEtection TRansformer (DETR) [2] object detection module with frozen weights, producing bounding boxes, object class predictions, and visual features for detected objects. These objects form the nodes of a complete graph, which is then pruned using a k-nearest neighbors algorithm based on spatial proximity. The pruned graph is processed by our Semantic Graph Transformer, which outputs relationship embeddings for each edge. For predicate classification, these embeddings are compared against CLIP text embeddings of predicates using temperature-scaled cosine similarity, enabling open-vocabulary relationship detection without being constrained to a fixed predicate set.

Training Pipeline. We employ a graph matching module that aligns predicted objects with ground truth annotations using Hungarian matching enhanced by both spatial (IoU) and semantic similarity. The matched graph provides supervision for our multi-component CLIP-aligned loss function. Our training framework supports zero-shot learning: a configurable subset of predicates (default 20%) is held out during training, with edges involving these predicates excluded from the loss computation while similarity is computed against all predicates. This allows the model to learn the full semantic embedding space and generalize to novel relationship types.

CLIP Alignment. Our approach, unlike traditional scene graph methods that learn class-specific projection weights, learns to position relationship embeddings within CLIP's pre-trained semantic space. The Semantic Graph Transformer's output projection is trained via contrastive learning to align with CLIP text embeddings of predicates. This design choice provides two key benefits: (1) the model inherits CLIP's semantic understanding of predicates, enabling meaningful predictions even for predicates not seen during training, and (2) the learned embeddings preserve semantic relationships between predicates (*e.g.*, "riding" and "sitting on" remain semantically related in the embedding space).

3.3. Inference pipeline

3.3.1. Graph Building

The graph building process in our OV-SGT model proceeds through the following steps:

1. **Object Detection:**

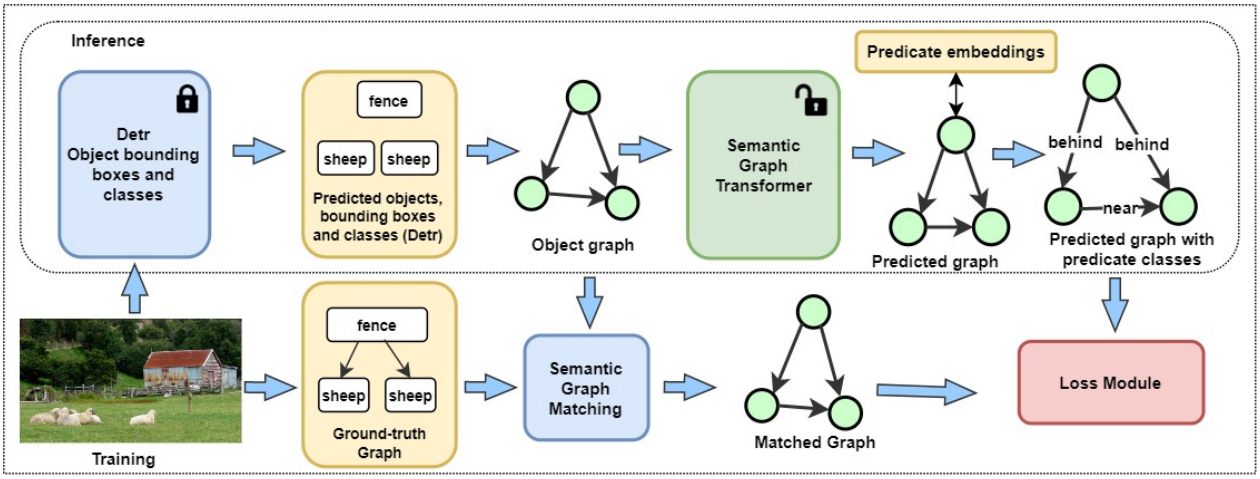


Figure 1. OV-SGT model architecture. In the inference path of our OV-SGT model, objects detected with a DETR object detection module are used for building the predicted object graph, which is subsequently pruned with a k-NN algorithm. The pruned graph is then passed to the Semantic Graph Transformer, after which we use a temperature-scaled cosine similarity matching module to generate relationship classes. In the training path we compute a matched ground truth graph and pass it to the Loss Module.

We utilize DETR to detect objects in the input image, generating bounding boxes, object class predictions, and visual features for each detected object. These objects serve as nodes in the scene graph.

2. **Graph Construction:**
   We prune the complete graph by using a k-nearest neighbors (k-NN) pruning strategy to selectively keep edges between spatially close objects. The k-NN algorithm uses the Euclidean distance between object bounding box centers. The k-NN approach significantly reduces computational complexity from $O(n^2)$ to $O(kn)$ edges, providing a strong spatial prior (as objects that are close to each other are more likely to have meaningful relationships), and maintains the most important connections while filtering out unlikely relationships between distant objects. The value of k controls the sparsity of the resulting graph, balancing computational efficiency with the model's ability to capture all relevant relationships. In the ablation study we show results for fully connected graphs and for $k = 10$.
3. **Edge Feature Computation:**
   For each object-neighbor pair, we compute a rich edge feature representation by concatenating normalized features of the source and target objects, normalized relative spatial position between object centers, and Intersection over Union (IoU) between the bounding boxes.

3.3.2. Semantic Graph Transformer

The resulting graph is passed to the Semantic Graph Transformer, which is detailed in Section 3.5.

3.3.3. Cosine Similarity for Predicate Classification

The Semantic Graph Transformer outputs relationship embeddings $\mathbf{r}_{ij} \in \mathbb{R}^{512}$ for each edge, positioned within CLIP's semantic space through our contrastive training objective. For predicate classification, we compute temperature-scaled cosine similarity between relationship embeddings and CLIP text embeddings of predicates:

$$s_{ij,k} = \frac{\mathbf{r}_{ij} \cdot \mathbf{p}_k}{\|\mathbf{r}_{ij}\|\|\mathbf{p}_k\|} \cdot \frac{1}{\tau}, \tag{2}$$

where $\mathbf{p}_k$ is the CLIP text embedding of the k-th predicate and τ is a temperature parameter. This formulation enables open-vocabulary relationship detection: at inference time, we can compute similarity against any set of predicates—including those not seen during training—by simply encoding their text descriptions with CLIP's text encoder. We evaluate using the top-3 scoring predicates per edge ($M = 3$).

3.4. Training pipeline

3.4.1. Semantic Graph Matching

The graph matching process aligns predicted relationships with ground truth annotations. The resulting matched graph is only used for the loss calculation. Matching enables effective supervision during training by establishing correspondences between predicted and ground-truth objects. Since raw outputs from object detectors don't directly map to ground-truth annotations, matching allows us to compare predicted relationships with their ground-truth counterparts.

Graph matching addresses the gap between predicted object locations and discrete ground truth annotations. We

used the Hungarian Matching Algorithm for our bipartite matching, which leverages both IoU and semantic similarity measures. It handles slight misalignments in object detection and ensures that relationship prediction isn't unfairly penalized for minor localization errors while maintaining the open vocabulary quality of our pipeline. The matching cost $E_{i,j}$ combines IoU-based spatial overlap with the semantic similarity between object class predictions:

$$E_{i,j} = \begin{array}{l} -(\alpha \cdot \text{IoU}(b_i, b_j) \\ +(1-\alpha) \cdot \text{semantic\_similarity}(c_i, c_j)) \end{array} \quad (3)$$

where $\alpha = 0.7$ controls the balance between spatial and semantic components, while the semantic similarity is computed using string matching techniques. b_i, b_j are the bounding boxes for objects i and j, while c_i, c_j are their corresponding classes.

3.5. Semantic Graph Transformer Architecture

The Semantic Graph Transformer consists of the following components, shown in Figure 2:

3.5.1. Positional Encoding

We incorporate positional information to enhance the transformer's understanding of spatial and structural relationships between objects. We employ graph Laplacian eigenvector-based positional encoding, which captures both local proximity and global connectivity patterns within the scene graph.

We construct an adjacency matrix where edge weights are determined using inverse distance between objects:

$$A_{ij} = \frac{1}{1 + d(c_i, c_j)}, \quad (4)$$

where $d(c_i, c_j)$ is the Euclidean distance between the centers of objects i and j. We compute the normalized Laplacian matrix:

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2} \quad (5)$$

where L is the Laplacian matrix and D is the degree matrix. We extract the k smallest eigenvectors from the normalized Laplacian, setting $k = 5$ based on the average of approximately 35 objects per image in Visual Genome. The positional encoding for node j is:

$$\mathbf{p}(j) = [\mathbf{v}_1[j], \mathbf{v}_2[j], \ldots, \mathbf{v}_k[j], x_c/W, y_c/H] \quad (6)$$

where $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ are the k smallest eigenvectors, and $(x_c/W, y_c/H)$ are the normalized center coordinates appended for spatial awareness. This formulation captures structural information about each node's position within the graph topology while retaining basic spatial grounding through the center coordinates.

3.5.2. Input processing

Node features are normalized, then processed through an embedding layer comprising linear transformations, normalization, ReLU activation, and dropout. Edge attributes are similarly processed through projection and embedding layers.

3.5.3. Transformer layers.

Semantic Graph Transformer uses 6 Transformer layers inspired by the Unified Message Passing Model (UniMP) [25]. Each layer implements multi-head attention mechanisms (with 8 heads) that operate directly over the graph structure while preserving edge attribute information during message passing. While the original UniMP focused on semi-supervised node classification tasks, we adapt it for relationship reasoning in scene graphs by enhancing the model's ability to capture both local interactions between adjacent objects and more complex, long-range dependencies.

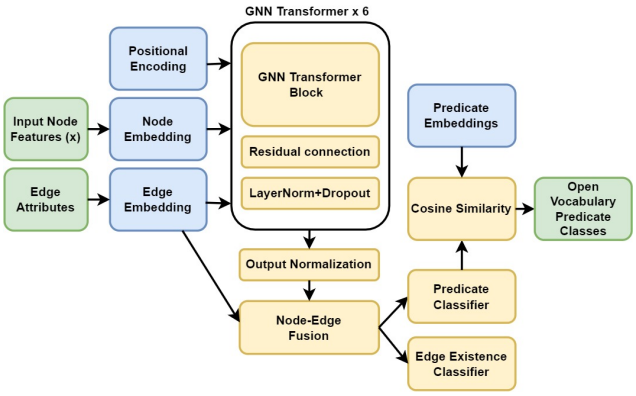


Figure 2. Semantic Graph Transformer.

3.5.4. Edge Feature Processing

After the transformer layers, our model implements a distinctive node-edge fusion mechanism to construct comprehensive relationship representations. While conventional approaches typically process node and edge features separately or rely solely on node-level information, our method explicitly combines three key components: the transformed source node features $\mathbf{h}_i$, the transformed target node features $\mathbf{h}_j$, and the processed edge attributes $\mathbf{e}_{ij}$. For each potential relationship between nodes i and j, we construct the fused representation as:

$$\mathbf{r}_{ij} = [\mathbf{h}_i \oplus \mathbf{h}_j \oplus \mathbf{e}_{ij}] \quad (7)$$

where $\oplus$ denotes concatenation and $\mathbf{r}_{ij} \in \mathbb{R}^{3d}$ for hidden dimension d. This concatenation strategy serves two critical purposes. First, it preserves the inherent directionality of relationships by maintaining distinct representations for source and target nodes, enabling the model to differentiate between relations such as "person riding horse" versus

"horse carrying person". Second, it integrates the contextual information that has been propagated through the graph during transformer processing, allowing each relationship representation to benefit from the broader scene context. The resulting unified representation $\mathbf{r}_{ij}$ captures both the individual semantic properties of the connected objects and their spatial-contextual interaction patterns, providing a rich foundation for subsequent relationship classification.

3.5.5. Predicate classification head.

The predicate classification is a linear projection that maps edge features to a 512-dimensional embedding space. The model learns this projection during training by optimizing the multi-label focal loss, which encourages edge embeddings to have high cosine similarity with their corresponding predicate embeddings. The loss function effectively trains the edge projection to align with the semantic space from CLIP [23]. This approach is conceptually similar to CLIP's own training process, where visual and text embeddings are trained to align in a shared space, but in our case, we are aligning graph structure embeddings with CLIP's text embeddings.

3.6. Predicate Weighting and Debiasing

Scene graph datasets exhibit severe class imbalance, where common predicates like "on" and "has" dominate while semantically rich predicates appear infrequently. We address this through frequency-based weighting that assigns higher importance to rare predicates, incorporated into our focal loss component.

During inference, we apply the Total Direct Effect (TDE) debiasing [26] to separate visual evidence from statistical bias. The debiased prediction $Y_{TDE} = Y(X, Z) - Y(X^*, Z) + Y(X^*, Z^*)$ removes dataset bias while preserving genuine visual evidence. This two-pronged approach—frequency weighting during training and TDE during inference—significantly improves Mean Recall@K metrics.

3.7. Loss Function

Our model employs a multi-component loss function designed for CLIP-aligned open-vocabulary scene graph generation. The total loss combines four complementary objectives that encourage semantic alignment, fine-grained discrimination, and robustness to class imbalance:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{cont}}\mathcal{L}_{\text{cont}} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} + \lambda_{\text{trip}}\mathcal{L}_{\text{trip}} + \lambda_{\text{focal}}\mathcal{L}_{\text{focal}}, \quad (8)$$

where $\lambda_{\text{cont}} = 1.0$, $\lambda_{\text{sem}} = 0.1$, $\lambda_{\text{trip}} = 0.5$, and $\lambda_{\text{focal}} = 0.3$ control the contribution of each component.

3.7.1. CLIP-Aligned Contrastive Loss

The core of our approach is a contrastive loss that aligns learned relationship embeddings with CLIP's text embeddings of predicates. Similar to CLIP's training objective [23], we treat each relationship as a (visual, text) pair:

$$\mathcal{L}_{\text{cont}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(s_{i,y_i}/\tau)}{\sum_{j=1}^{K}\exp(s_{i,j}/\tau)}, \quad (9)$$

where $s_{i,j} = \frac{\mathbf{r}_i \cdot \mathbf{p}_j}{\|\mathbf{r}_i\|\|\mathbf{p}_j\|}$ is the cosine similarity between the i-th relationship embedding $\mathbf{r}_i$ and the j-th predicate embedding $\mathbf{p}_j$, y_i is the ground truth predicate index for edge i, $\tau = 0.07$ is a temperature parameter, N is the number of edges, and K is the number of predicates. This formulation enables open-vocabulary generalization by learning to position relationship embeddings within CLIP's semantic space rather than learning class-specific weights.

3.7.2. Semantic Regularization Loss

To encourage compositional understanding, we introduce a semantic regularization loss that preserves the semantic structure of CLIP's predicate space:

$$\mathcal{L}_{\text{sem}} = \|\mathbf{DS} - \mathbf{D}\|_F^2, \quad (10)$$

where $\mathbf{D} \in \mathbb{R}^{N\times K}$ is the softmax-normalized prediction distribution over predicates, $\mathbf{S} \in \mathbb{R}^{K\times K}$ is the semantic similarity matrix between predicate embeddings in CLIP space, and $\|\cdot\|_F$ denotes the Frobenius norm. This loss encourages the model to produce predictions that respect semantic relationships between predicates—if "riding" and "sitting on" are similar in CLIP space, their prediction scores should also be correlated.

3.7.3. Hard Negative Triplet Loss

To improve fine-grained discrimination between similar predicates, we employ a triplet loss with hard negative mining:

$$\mathcal{L}_{\text{trip}} = \frac{1}{N}\sum_{i=1}^{N}\max(0, m - s_{i,y_i} + s_{i,h_i}), \quad (11)$$

where s_{i,y_i} is the similarity to the ground truth predicate, s_{i,h_i} is the similarity to the hard negative (the highest-scoring incorrect predicate), and $m = 0.3$ is the margin. This loss explicitly pushes the model to separate the correct predicate from its most confusing alternative.

3.7.4. Weighted Focal Loss

To address the severe class imbalance in scene graph datasets, we incorporate the focal loss [20] with frequency-based predicate weighting:

$$\mathcal{L}_{\text{focal}} = -\frac{1}{N^+}\sum_{i=1}^{N}\sum_{j=1}^{K} w_j \alpha_{i,j}(1 - p_{i,j})^{\gamma}\text{BCE}(p_{i,j}, y_{i,j}), \quad (12)$$

where $p_{i,j} = \sigma(s_{i,j}/\tau')$ is the predicted probability with $\tau' = 0.1$, $y_{i,j}$ is the binary ground truth label, $\gamma = 2.0$ is the focusing parameter, $\alpha_{i,j}$ balances positive and negative examples, w_j is the frequency-based weight for predicate j, and N^+ is the number of positive examples. Critically, we normalize by the number of positive examples rather than all elements, preventing the loss from being dominated by the overwhelming number of negative pairs.

The frequency-based weights follow an inverse frequency scheme:

$$w_j = \min\left(\frac{f_{\max}}{f_j + 1}, w_{\max}\right), \tag{13}$$

where f_j is the frequency of predicate j, $f_{\max}$ is the maximum frequency, and $w_{\max} = 10$ prevents excessive weighting of extremely rare predicates.

4. Experiments

In this section we provide the experimental results for the proposed OV-SGT: Open Vocabulary Semantic Graph Transformer for Scene Graph Generation.

4.1. Implementation

We implemented our model using PyTorch and PyTorch Geometric for efficient graph neural network operations. For training, we used the Adam optimizer with a learning rate of 3*1e-3 and weight decay of 1e-5 to minimize overfitting. Our batch size was set to 32 images. We initialized our transformer layers using Xavier uniform initialization to ensure proper gradient flow through the deep network architecture. To prevent exploding gradients, we applied gradient clipping with a maximum norm of 1.0. During training, we employed mixed-precision (FP16) to improve computational efficiency without sacrificing model performance. For data augmentation, we used horizontal flipping with probability 0.5, while maintaining relationship semantics that depend on spatial orientation. We implemented our model in distributed data-parallel mode using PyTorch's DDP framework to maximize GPU utilization during training.

4.2. Dataset

We ran experiments on three datasets: Visual Genome [14], Visual Relationship Detection (VRD) [22] and GQA dataset [10].

For Visual Genome, we consider a curated subset of images, consisting of 84,000 images selected for training, plus an additional 4,800 images reserved for validation, totaling 88,800 images selected from the original collection. This curated dataset maintains the rich semantic annotations that makes Visual Genome valuable while addressing some of the inconsistencies and the over-crowding with repeated identical boxes identifying same objects, present in the original dataset. The VRD dataset contains 5000 images with 37,993 relationships. This dataset contains 100 object categories and 70 predicate categories connecting those objects together. GQA dataset contains over 110K images with scene graph annotations and features a larger and more diverse predicate vocabulary compared to Visual Genome, providing a challenging testbed for generalization. We consider the standard train/validation split for evaluation.

4.3. Analysis of results

We evaluate our model on the Visual Genome dataset [14] using the common VG-50 configuration [30] with 150 object categories and 50 relationship predicates. The baseline models that we compare our results against, are using the same benchmark.

Three standard evaluation metrics are commonly used: (1) Predicate Classification (PredCLS): identifies relationships between objects when both object locations and labels are provided. (2) Scene Graph Classification (SGCLS): determines both object types and their relationships using only the given object locations. (3) Scene Graph Detection (SGDET): performs complete scene understanding by simultaneously detecting object locations, identifying their categories, and predicting their relationships.

In Fig. 3 we provide three graph visualization results in the context of three different scenes. These results show that the proposed model provides high confidence results for a variety of scene relationships : human-animal in Fig. 3(a), human-object in Fig. 3(b) and object-object in Fig. 3(c).

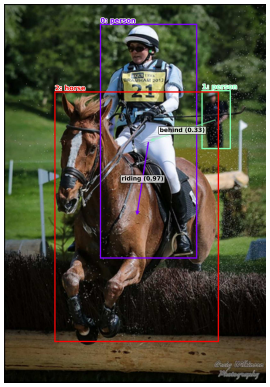
4.4. Ablation studies

We have conducted several ablation studies in order to evaluate the contribution of different components in our model, examining the k-NN graph construction, positional embeddings, predicate weighting, and transformer architecture. The results are provided in Table 3. The results for ablation studies to evaluate the contribution of each loss component and design choice are shown in Table 4.

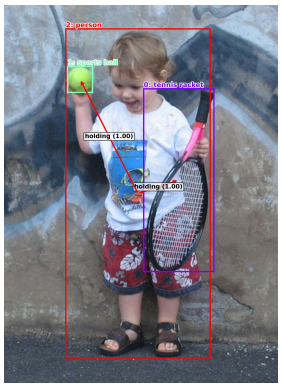
Loss Component Analysis. We systematically remove each loss component to assess its contribution. Removing the **triplet loss** causes the largest performance drop (R@50: 44.7→36.0, mR@50: 36.0→21.4), demonstrating its critical role in fine-grained discrimination between similar predicates. The hard negative mining mechanism helps the model distinguish confusing predicate pairs.

Removing the **contrastive loss** maintains reasonable R@K but significantly degrades mR@K (36.0→27.9), indicating that CLIP alignment is essential for balanced performance across predicate classes, particularly rare ones.

Interestingly, removing the **semantic regularization loss** slightly improves R@K metrics while modestly reducing mR@K. This suggests the semantic consistency con-



(a) Person riding horse
$person \rightarrow riding \rightarrow horse$ (0.97)
$person \rightarrow behind \rightarrow person$ (0.33)



(b) Child with ball and racket
$person \rightarrow holding \rightarrow sports\ ball$ (1.00)
$person \rightarrow holding \rightarrow tennis\ racket$ (1.00)

(c) Cake and knife on table
$cake \rightarrow on \rightarrow dining\ table$ (0.90)
$knife \rightarrow on \rightarrow dining\ table$ (0.65)

Figure 3. Scene graph visualizations showing everyday interactions. Our model detects relationships with high confidence scores shown in parentheses.

Model	SGDET						SGCLS					
	R@20	R@50	R@100	mR@20	mR@50	mR@100	R@20	R@50	R@100	mR@20	mR@50	mR@100
DT2-ACBS [7]	-	15.0	16.3	-	22.0	24.4	-	-	-	-	-	-
G-RCNN [27]	19.4	25.2	31.6	-	-	-	29.0	31.6	31.9	-	-	-
GPS-Net [21]	22.3	28.9	33.2	6.9	8.7	-	41.8	42.3	42.3	10.0	11.8	-
KERN [4]	22.3	27.1	35.8	-	6.4	-	32.2	36.7	49.0	-	9.4	-
Neural Motifs [30]	21.4	27.2	30.3	4.2	5.7	-	32.9	35.8	36.5	6.3	7.7	-
IterSGG [12]	-	24.2	29.7	-	19.5	23.4	-	-	-	-	-	-
RelTR [5]	21.2	27.5	-	6.8	10.8	-	29.0	36.6	-	7.7	11.4	-
DAC [17]	29.8	36.4	41.2	7.2	18.3	27.6	38.9	40.2	40.3	10.1	12.8	13.2
RMP-Net [3]	28.1	34.8	39.7	6.8	17.6	26.4	37.2	39.1	39.8	9.7	12.1	12.9
IETrans [34]	30.5	37.2	42.1	8.1	19.8	28.6	**39.5**	**41.5**	**42.0**	**11.2**	**14.1**	**14.8**
OV-SGT (ours)	**38.4**	**44.7**	**48.9**	**33.7**	**36.0**	**36.8**	-	-	-	-	-	-
OV-SGT (zero-shot)	-	6.8	14.7	-	-	-	-	-	-	-	-	-

Table 1. Scene Graph Generation results on Visual Genome dataset for Scene Graph Detection and Scene Graph Classification metrics. Bold indicates best performance.

Dataset	**Method**	**R@50**	**R@100**
VRD	VTransE [32]	19.4	22.4
	ViP-CNN [18]	17.3	20.0
	VRL [19]	18.2	20.8
	MF-URLN [31]	23.9	26.8
	RelDN [33]	25.3	28.6
	GPS-Net [21]	27.8	31.7
	RelTR [5]	29.2	32.2
	OV-SGT (ours)	28.6	31.4
GQA	OV-SGT (ours)	65.6	72.3
GQA	OV-SGT (zero-shot)	7.2	10.8

Table 2. SGDet results on VRD and GQA datasets.

straint may occasionally conflict with discriminative learning, though it contributes to more balanced predicate coverage.

Configuration	R@			mR@		
	20	50	100	20	50	100
Graph Construction						
k-NN with k=10	25.1	30.6	36.1	4.1	13.1	21.2
No pruning	14.0	16.7	20.2	2.2	7.0	12.2
Positional Embedding						
Scaled coordinates	24.1	30.0	35.3	3.9	12.9	21.2
Predicate Weighting						
No weighting	21.9	25.3	29.2	1.7	4.9	8.5
Full model						
OV-SGT	**38.4**	**44.7**	**48.9**	**33.7**	**36.0**	**36.8**

Table 3. Ablation study results comparing different model configurations.

CLIP Configuration. Fine-tuning CLIP embeddings instead of keeping them frozen dramatically degrades performance (R@50: 44.7→27.1). This validates our design choice: the pre-trained CLIP embeddings provide a well-structured semantic space that should be preserved. Fine-tuning destroys this structure, collapsing the embedding

Configuration	R@			mR@		
	20	50	100	20	50	100
OV-SGT (full model)	**38.4**	**44.7**	**48.9**	**33.7**	**36.0**	**36.8**
Loss Component Ablations						
w/o Contrastive loss	39.1	44.8	48.3	23.3	27.9	28.0
w/o Semantic loss	41.0	46.7	49.6	31.4	33.2	33.3
w/o Triplet loss	30.5	36.0	38.9	19.7	21.4	21.7
CLIP Configuration						
Fine-tuned CLIP	20.3	27.1	32.3	20.8	26.4	29.2

Table 4. Ablation study on loss components and CLIP configuration. Removing triplet loss causes the largest performance drop. Fine-tuning CLIP significantly degrades performance, validating our frozen CLIP design.

space and eliminating the model's ability to generalize to novel predicates.

Zero-Shot Generalization. We hold out 20% of predicates (10 classes) during training. OV-SGT achieves R@50=6.8% and R@100=14.7% on these unseen predicates (Table 1), demonstrating genuine zero-shot transfer through CLIP alignment.

k-NN Graph Construction. We tested the k-NN graph construction, which is explained in Section 3.3 and the results are provided in the second section of results from Table 3. The k-NN graph construction experiments demonstrate the critical importance of selective edge pruning. Using $k = 5$ nearest neighbors significantly outperforms the fully connected graph approach (no pruning) across all metrics. This confirms our hypothesis that selective edge construction provides a strong inductive bias that aligns with the reality of visual relationships, where objects that are spatially proximate are more likely to interact.

Positional Embeddings. We compare our Laplacian eigenvector-based positional encoding against simple scaled box coordinates. The results in Table 3 show that Laplacian eigenvector encodings outperform scaled coordinates across all metrics. This demonstrates the value of incorporating structural graph information: while scaled coordinates only capture absolute spatial positions, Laplacian eigenvectors encode each node's relationship to the overall graph topology, capturing both local clustering and global connectivity patterns. The center coordinates appended to the eigenvectors provide sufficient spatial grounding without requiring the full bounding box representation.

Predicate Weighting. We evaluate the impact of frequency-based predicate weighting, defined by the per-class weights w_j in Eq. (13), which are applied within the focal loss component. The results in Table 3 show substantial benefits for the mean recall metrics when weighting is enabled. This confirms the effectiveness of our inverse-frequency weighting strategy in addressing the long-tail distribution of relationship predicates, significantly improving performance on rare relationship classes without compromising performance on common ones. The weights are capped at $w_{\max} = 10$ to prevent numerical instability from extremely rare predicates.

4.5. Training computation efficiency

We conducted all experiments on a system with 8 NVIDIA H100 GPU with 80GB of VRAM, supported by 26 vCPUs and 200GB of RAM. The training process was executed over 45 epochs on the 84,000-image training set. Each epoch required approximately 1.5 hours of computation time. During training, we used mixed-precision training to optimize GPU utilization.

5. Limitations and Future Work

Despite the strong performance of our OV-SGT model, several limitations remain. First, our approach still relies on a pre-trained object detector, inheriting any biases or limitations. Second, while our k-NN graph construction significantly reduces computational complexity, it may occasionally miss meaningful long-range relationships between distant objects in a scene. Third, our model requires substantial computational resources for training (approximately 3 days on 8 H100 GPU), limiting accessibility for those with constrained computing budgets. Also, the open vocabulary capabilities, while more flexible than closed vocabulary models, are ultimately bounded by the semantic space of the underlying CLIP model.

6. Conclusions

In this paper, we have presented the Open Vocabulary Semantic Graph Transformer for Scene Graph Generation (OV-SGT). The key innovation lies in our graph construction methodology, which fuses contextually-enhanced information from source nodes, target nodes, and edge attributes within the global scene context. Our loss function effectively addresses the long-tail distribution of relationship predicates by accounting for rare relationships that are often overlooked by existing methods. We have provided the results from an extensive series of experiments showing the advantages of the proposed methodology and its implementation.

References

[1] Frank W. Bergmann and Brian Fenton. 2015. Scene Based Reasoning. In *Proc. Artificial General Intelligence (AGI), vol. LNCS 9205*. 25–34. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey

Zagoruyko. 2020. End-to-End Object Detection with Transformers. In *Proc. European Conference on Computer Vision (ECCV), vol. LNCS 12346*. 213–229. 2

[3] Shuai Chen, Xiaoyan Zhang, Liang Zhang, and Zan Gao. 2023. Unbiased Heterogeneous Scene Graph Generation with Relation-aware Message Passing Neural Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7

[4] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. 2019. Knowledge-embedded Routing Network for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6156–6164. 7

[5] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. 2023. RelTR: Relation Transformer for Scene Graph Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 9 (2023), 11169–11183. 7

[6] Vinay Damodaran, Sharanya Chakravarthy, Akshay Kumar, Anjana Umapathy, Teruko Mitamura, Yuta Nakashima, Noa García, and Chenhui Chu. 2021. Understanding the Role of Scene Graphs in Visual Question Answering. *arXiv preprint arXiv:2101.05479* (2021). 1

[7] Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. 2021. Learning of Visual Relations: The Devil is in the Tails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 15384–15393. 7

[8] Matthew Fisher, Manolis Savva, Yangyan Li, Pas Hanrahan, and Matthias Nießner. 2015. Activity-centric scene synthesis for functional 3D scene modeli. *ACM Transactions on Graphics* 34, 6 (2015), 1–13. 1

[9] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. Image Captioning: Transforming Objects into Words. In *Advances in Neural Information Processing Systems (NeurIPS)*. 11137–11147. 1

[10] Drew A. Hudson and Christopher D. Manning. 2019. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 6693–6702. 1, 6

[11] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image Generation from Scene Graphs. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 1219–1228. 1

[12] Siddhesh Khandelwal and Leonid Sigal. 2022. Iterative Scene Graph Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*. 24295–24308. 2, 7

[13] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J. Kim. 2021. HOTR: End-to-End Human-Object Interaction Detection with Transformers. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 74–83. 1

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision* 123 (2016), 32–73. 6

[15] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97. 2

[16] Rongjie Li, Songyang Zhang, and Xuming He. 2021. SGTR: End-to-end Scene Graph Generation with Transformer. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 19464–19474. 2

[17] Yiming Li, Xueming Li, Xiaolong Yang, Jingming Guo, and Xin Wang. 2023. Decompose, Adjust, Compose: Effective Normalization for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7

[18] Yikang Li, Wanli Ouyang, and Xiaogang Wang. 2017. ViP-CNN: A Visual Phrase Reasoning Convolutional Neural Network for Visual Relationship Detection. *arXiv preprint arXiv:1702.07191* (2017). 7

[19] Xiaodan Liang, Lisa Lee, and Eric P. Xing. 2017. Deep Variation-structured Reinforcement Learning for Visual Relationship and Attribute Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 848–857. 7

[20] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. *Proc. IEEE International Conference on Computer Vision (ICCV)* (2017), 2999–3007. 5

[21] Xin Lin, Changxing Ding, Jinquan Zeng, and Dacheng Tao. 2020. GPS-Net: Graph Property Sensing Network for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3746–3753. 7

[22] Cewu Lu, Ranjay Krishna, Michael S. Bernstein, and Li Fei-Fei. 2016. Visual Relationship Detection with Language Priors. In *Proc. European Conference on Computer Vision (ECCV), vol. LNCS 9905*. 852–869. 1, 6

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language

Supervision. In *Proc. Int. Conf. on Machine Learning (ICML), vol, PMLR 139*. 8748–8763. 5

[24] Sahana Ramnath, Amrita Saha, Soumen Chakrabarti, and Mitesh M. Khapra. 2019. Scene Graph based Image Retrieval - A case study on the CLEVR Dataset. In *Proc. ICCV Workshop - Linguistics Meets Image and Video Retrieval*. 1

[25] Yunsheng Shi, Zhengjie Huang, Wenjin Wang, Hui Zhong, Shikun Feng, and Yu Sun. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*. 1548 –1554. 4

[26] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. 2020. Unbiased Scene Graph Generation from Biased Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3716–3725. 5

[27] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Graph r-cnn for scene graph generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 670–685. 7

[28] Bangguo Yu, Chongyu Chen, Fengyu Zhou, Fang Wan, Wenmi Zhuang, and Yang Zhao. 2020. A Bottom-up Framework for Construction of Structured Semantic 3D Scene Graph. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), 8224–8230. 1

[29] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2017. Neural Motifs: Scene Graph Parsing with Global Context. *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5831–5840. 2

[30] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5831–5840. 1, 6, 7

[31] Yaohui Zhan, Jun Yu, Ting Yu, and Dacheng Tao. 2019. On Exploring Undetermined Relationships for Visual Relationship Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5128–5137. 7

[32] Hanwang Zhang, Zaw Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual Translation Embedding Network for Visual Relation Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5532–5540. 7

[33] Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. 2019. Graphical Contrastive Losses for Scene Graph Parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11527–11535. 7

[34] Xin Zhang, Yuan Yuan, Yawei Luo, and Yang Xiao. 2024. IETrans: Instance-level Edge Transformer for Scene Graph Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7