
Relationformer: A Unified Framework for *Image-to-Graph* Generation

Suprosanna Shit^{*1,3}, Rajat Koner^{*2}, Bastian Wittmann¹, Johannes Paetzold¹,
Ivan Ezhov¹, Hongwei Li^{1,3}, Jiazhen Pan¹, Sahand Sharifzadeh²,
Georgios Kaissis¹, Volker Tresp², Bjoern Menze^{1,3}

¹Technical University of Munich, ²Ludwig Maximilian University of Munich,

³University of Zurich

suprosanna.shit@tum.de, koner@dbis.lmu.de

Abstract

A comprehensive representation of an image requires understanding objects and their mutual relationship, especially in *image-to-graph* generation, e.g., road network extraction, blood-vessel network extraction, or scene graph generation. Traditionally, *image-to-graph* generation is addressed with a two-stage approach consisting of object detection followed by a separate relation prediction, which prevents simultaneous object-relation interaction. This work proposes a unified one-stage transformer-based framework, namely Relationformer that jointly predicts objects and their relations. We leverage direct set-based object prediction and incorporate the interaction among the objects to learn an object-relation representation jointly. In addition to existing [obj]-tokens, we propose a novel learnable token, namely [rln]-token. Together with [obj]-tokens, [rln]-token exploits local and global semantic reasoning in an image through a series of mutual associations. In combination with the pair-wise [obj]-token, the [rln]-token contributes to a computationally efficient relation prediction. We achieve state-of-the-art performance on multiple, diverse and multi-domain datasets that demonstrate our approach's effectiveness and generalizability. ²

1 Introduction

An image contains multiple layers of abstractions, from low-level features to intermediate-level objects to high-level complex semantic relations. To gain a complete visual understanding, it is essential to investigate different abstraction layers jointly. An example of such multi-abstraction problem is *image-to-graph* generation, such as road-network extraction [18], blood vessel-graph extraction [42], and scene-graph generation [57]. In all of these tasks, one needs to explore not only the objects or the *nodes*, but also their mutual dependencies or relations as *edges*.

In *spatio-structural* tasks, such as road network extraction (Fig. 1a), nodes represent road-junctions or significant turns, while edges correspond to structural connections, i.e., the road itself. The resulting spatio-structural graph construction is crucial for navigation tasks, especially with regard to autonomous vehicles. Similarly, in 3D blood vessel-graph extraction (Fig. 1b), nodes represent branching points or substantial curves, and edges correspond to structural connections, i.e., arteries, veins, and capillaries. Biological studies relying on a vascular graph representation, such as detecting collaterals [54], assessing structural robustness [22], emphasize the importance of efficient extraction thereof. In case of *spatio-semantic* graph generation, e.g. scene graph generation from natural images (Fig. 1c), the objects denote nodes and the semantic-relation denotes the edges [23]. This graphical

^{*}equal contribution

²code is available at <https://github.com/suprosanna/relationformer>

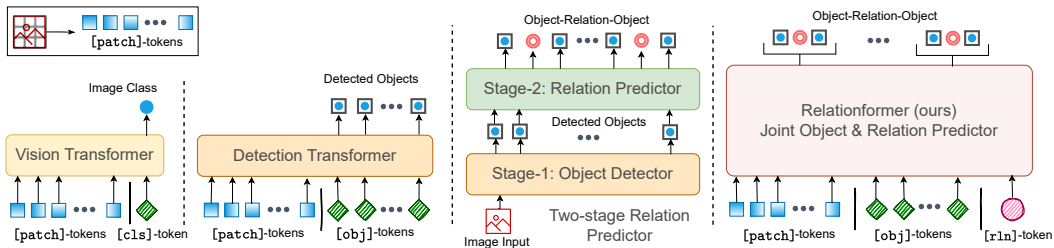


Figure 2: This illustrates a general architectural evolution of transformers in computer vision and how Relationformer advances the concept of a task-specific learnable token one step further. The proposed Relationformer is also shown in comparison to the conventional two-stage relation predictor. The amalgamation of two separate stages not only simplifies the architectural pipeline but also co-reinforces both of the tasks.

2 Related Work

Transformer in Vision: In recent times, transformer-based architectures have emerged as the de-facto gold standard model for various multi-domain and multi-modal tasks such as image classification [13], object detection [7], and out-of-distribution detection [26]. DETR [7] proposed an end-to-end transformer-based object detection approach with learnable object queries ([obj]-tokens) and direct set-based prediction. DETR eliminates burdensome object detection pipelines (e.g., anchor boxes, NMS) of traditional approaches [44] and directly predicts objects. Building on DETR, a series of object detection approaches improved DETR’s slow convergence [64], adapted a pure sequence-to-sequence approach [15], and improved detector efficiency [52]. In parallel, the development of the vision transformer [13] for image classification offered a powerful alternative. Several refined ideas [35, 55] have advanced this breakthrough and transformer in general emerges as a cutting-edge research topic with focus on novel design principle and innovative application. Fig. 2, shows a pictorial overview of transformer-based image classifier, object detector, and relation predictor including our proposed method, which we referred to as Relationformer.

Spatio-structural Graph Generation: In a spatio-structural graph, the most important physical objects are edges, i.e., roads for a road network or arteries and veins in vessel graphs. Conventionally, spatio-structural graph extraction has only been discussed in 2D with little-to-no attention on the 3D counterpart. For 2D road network extraction, the predominant approach is to segment [4, 38] followed by morphological thinning to extract the spatial graph. Only few approaches combine graph level information processing, iterative node generation [3], sequential generative modelling [9], and graph-tensor-encoding [18]. Belli et al. [5] for the first time, adopted attention mechanisms in an auto-regressive model to generate graphs conditioned on binary segmentation. Importantly, to this date, none of these 2D approaches has been shown to scale to 3D.

For 3D vessel-network extraction, segmentation of whole-brain microscopy images [40, 54] has been combined with rule-based graph extraction algorithms [51]. Recently, a large-scale study [42] used the *Voreen* [39] software to extract whole-brain vascular graph from binary segmentation, which required complicated heuristics and huge computational resources. Despite recent works on 3D scene graphs [1] and temporal scene graphs [21], to this day, there exists no learning-based solution for 3D spatio-structural graph extraction.

Considering two spatio-structural image-to-graph examples of vessel-graph and road-network, one can understand the spatial relation detection task as a link prediction task. In link-prediction, graph neural networks, such as GraphSAGE [16], SEAL [62] are trained to predict missing links among nodes using node features. These approaches predict links on a given set of nodes. Therefore, link prediction can only optimize correct graph topology. In comparison, we are interested in joint node-edge prediction, emphasizing correct topology and correct spatial location simultaneously, making the task even more challenging.

Spatio-semantic Graph Generation: Scene graph generation (SGG) [36, 57] from 2D natural images has long been studied to explore objects and their inter-dependencies in an interpretable way.

Context refinement across objects [57, 61], extra modality of features [36, 50] or prior knowledge [48] has been used to model inter-dependencies of objects for relation prediction. RTN [24, 27] was one of the first transformer approaches to explore context modeling and interactions between objects and edges for SGG. Li et al. [30] uses DETR like architecture to separately predict entity and predicate proposal followed by a graph assembly module. Later, several works [12, 37] explored transformers, improving relation predictions. On the downside, such two-stage approaches increase model size, lead to high inference times, and rely on extra features such as glove vector [43] embedding or knowledge graph embedding [49], limiting their applicability. Recently, Liu et al. [34] proposed a fully convolutional one-stage SGG method. It combined a feature pyramid network [32] and a relation affinity field [41, 63] for modeling a joint *object-relation* graph. However, their convolution-based architecture limits the context exploration across objects and relations. Contemporary to us [10] used transformers for the task of SGG. However, their complex pipeline for a separate subject and object further increases computational complexity. Crucially, there has been a significant performance gap between one-stage and two-stage approaches. This paper bridges this gap with simultaneous contextual exploration across objects and relations.

3 Methodology

In this section, we formally define the generalized *image-to-graph* generation problem. Each of the presented relation prediction tasks in Figure 1 is a special instance of this generalized image-to-graph problem. Consider an image space $I \in \mathbb{R}^{D \times \text{\#ch}}$, where $D = \prod_{i=1}^d \dim[i]$ for a d dimensional image and \#ch denotes the number of channels. Now, an image-to-graph generator \mathcal{F} predicts $\mathcal{F}(I) = \mathcal{G}$ for a given image I , where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a graph with vertices (or objects) \mathcal{V} and edges (or relations) \mathcal{E} . Specifically, the i^{th} vertex $v^i \in \mathcal{V}$ has a node or object location specified by a bounding box $v_{\text{box}}^i \in \mathbb{R}^{2 \times d}$ and a node or object label $v_{\text{cls}}^i \in \mathbb{Z}^C$. Similarly, each edge $e^{ij} \in \mathcal{E}$ has an edge or relation label $e_{\text{rln}}^{ij} \in \mathbb{Z}^L$, where we have C number of object classes and L types of relation classes. Note that \mathcal{G} can be both directed and undirected. The algorithmic complexity of predicting graph \mathcal{G} depends on its size, $|\mathcal{G}| = |\mathcal{V}| + |\mathcal{E}|$ which is of order complexity $\mathcal{O}(N^2)$ for $|\mathcal{V}| = N$. It should be noted that object detection as a special case of the generalized image-to-graph generation problem, where $\mathcal{E} = \phi$. In the following, we briefly revisit a set-based object detector before expanding on our rationale and proposed architecture.

3.1 Preliminaries of Set-based Object Detector

Carion et al. [7] proposed DETR, which shows the potential of set-based object detection, building upon an encoder-decoder transformer architecture [56]. Given an input image I , a convolutional backbone [17] is employed to extract high level and down scaled features. Next, the spatial dimensions of extracted features are reshaped into a vector to make them sequential. Afterwards, these sequential features are coupled with a sinusoidal positional encoding [6] to mark a unique position identifier. A stacked encoder layer, consisting of a multi-head self-attention and a feed-forward network, processes the sequential features. The decoder takes N number of learnable object queries ([obj]-tokens) in the input sequence and combines them with the output of the encoder via cross-attention, where N is larger than the maximum number of objects.

DETR utilizes the direct Hungarian set-based assignment for one-to-one matching between the ground truth and the predictions from N [obj]-tokens. The bipartite matching assigns a unique predicted object from the N predictions to each ground truth object. Only matched predictions are considered valid. The rest of the predictions are labeled as \emptyset or ‘background’. Subsequently, it computes the box regression loss solely for valid predictions. For the classification loss, all predictions, including ‘background’ objects, are considered.

In our work, we adopt a modified attention mechanism, namely deformable attention from deformable-DETR (def-DETR) [64] for its faster convergence and computational efficiency. In DETR, complete global attention allows each token to attend to all other tokens and hence capture the entire context in one image. However, information about the presence of an object is highly localized to a spatial position. Following the concept of deformable convolutions [11], deformable attention enables the queries to attend to a small set of spatial features determined from learned offsets of the reference points. This improves convergence and reduces the computational complexity of the attention operation.

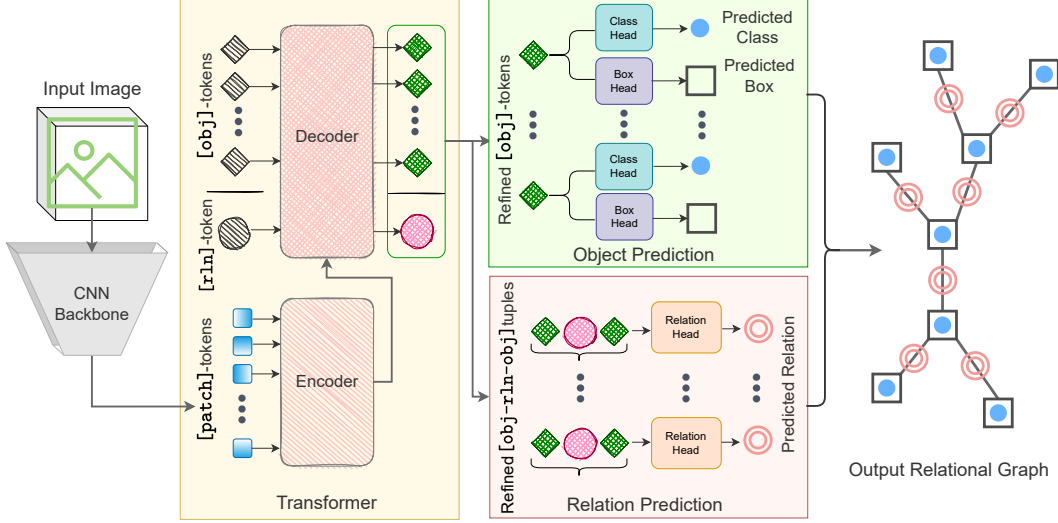


Figure 3: Specifics of the Relationformer architecture. The image is first processed by a feature extractor, which generates [patch]-tokens for the input of the transformer encoder. Next, transformer decoder takes learnable [obj]-tokens and a [rln]-token along with output from encoder. Decoder processes them through a series of self- and cross-attention operations. The object head processes the final [obj]-tokens from the decoder to produce the bounding box and object classes. The relation head takes a tuple of the final [obj-rln-obj]-token combination and classifies their relation. Combining the output of the object and relation head yields the final graph.

Let us consider an image feature map f_I , the q^{th} [obj]-token with associated features f_q , and the reference point x_q . First, for the m^{th} attention head, we need to compute the k^{th} sampling offset Δx_{mqk} based on the query features f_q . Subsequently, the sampled image features $f_I(x_q + \Delta x_{mqk})$ go through a single layer W'_m followed by a multiplication with the attention weight A_{mqk} , which is also obtained from the query features f_q . Finally, another single layer W_m merges all the heads. Formally, the deformable attention operation (DefAttn) for M heads and K sampling points is defined as:

$$\text{DefAttn}(f_q, x_q, f_I) = \sum_{m=1}^M W_m \left[\sum_{k=1}^K A_{mqk} \cdot W'_m f_I(x_q + \Delta x_{mqk}) \right] \quad (1)$$

3.2 Object-Relation Prediction as Set-prediction and Interaction

A joint *object-relation* graph generation requires searching from a pairwise combinatorial space of the maximum number of expected nodes. Hence, a naive joint-learning for *object-relations* requires $\mathcal{O}(N^2)$ number of tokens for N number of objects. This is computationally intractable because self-attention is quadratically-complex to the number of tokens. We overcome this combinatorially challenging formulation with a carefully engineered inductive bias. The inductive bias, in this case, is to exploit learned pair-wise interactions among N [obj]-tokens and combine refined pair-wise [obj]-tokens with an additional $(N + 1)^{th}$ token, which we refer to as [rln]-token. One can think of the [rln]-token as a query to pair-wise object interaction.

The [rln]-token captures the additional context of pair-wise interactions among all valid predicted classes. In this process, related objects are incentivized to have a strong correlation in an embedding space of, and unrelated objects are penalized to be dissimilar. The [rln]-token attends to all N [obj]-tokens along with contextualized image features that enrich its local pairwise and global image reasoning. Finally, we classify a pair-wise relation by combining the pair-wise [obj]-tokens with the [rln]-token. Thus, instead of $\mathcal{O}(N^2)$ number of tokens, we only need $N + 1$ tokens in total. These consist of N [obj]-tokens and one [rln]-token. This novel formulation allows relation detection with a marginally increased cost compared to one-stage object detection.

Here, one could present a two-fold argument: 1) There is no need for an extra token as one could directly classify joint pairwise [obj]-tokens. 2) Instead of one single [r1n]-token, one could use as many as the number of possible object-pairs. To answer the first question, we argue that relations can be viewed as a higher order topological entity compared to objects. Thus, to capture inter-dependencies among the relations the model requires additional expressive capacity, which can be shared among the objects. The [r1n]-token reduces the burden on the [obj]-tokens by specializing exclusively on the task of relation prediction. Moreover, [obj]-tokens can also attend to the [r1n]-token and exploit a global semantic reasoning. This hypothesis has been confirmed in our ablation. For the second question, we argue that individual tokens for all possible object-pairs will lead to a drastic increase in the decoder complexity, which may result in computationally intractability.

3.3 Relationformer

The *Relationformer* architecture is intuitive and without any bells and whistles, see Fig. 3. We have four main components: a backbone, a transformer, an object detection head and a relation prediction head. In the following, we describe each of the components and the set-based loss formulations specific to joint *object-relation* graph generation in detail.

Backbone: Given the input image I , a convolutional backbone [17] extracts features $f_I \in \mathbb{R}^{D_f \times \# \text{emb}}$, where D_f is the spatial dimensions of the features and $\# \text{emb}$ denotes embedding dimension. Further, this feature dimension is reduced to d_{emb} , the embedding dimension of the transformer, and flattened by its spatial size. The new sequential features coupled with the sinusoidal positional encoding [6] produce the desired sequence which is processed by the encoder.

Transformer: We use a transformer encoder-decoder architecture with deformable attention [64], which considerably speeds up the training convergence of DETR by exploiting spatial sparsity of the image features.

Encoder: Our encoder remains unchanged from [64], and uses multi-scale deformable self-attention. We use a different number of layers based on each task’s requirement, which is specified in detail in the supplementary material.

Decoder: We use $N + 1$ tokens for the joint *object-relation* task as inputs to the decoder, where N represents the number of [obj]-tokens preceded by a single [r1n]-token. Contextualized image features from the encoder serve as the second input of our decoder. In order to have a tractable computation and to leverage spatial sparsity, we use deformable cross-attention between the joint tokens and the image features from the encoder. The self-attention in the decoder remains unchanged. The [obj]-tokens and [r1n]-token go through a series of multi-hop information exchanges with other tokens and image features, which gradually builds a hierarchical object and relational semantics. Here, [obj]-tokens learn to attend to specific spatial positions, whereas the [r1n]-token learns how objects interact in the context of their semantic or global reasoning.

Object Detection Head: The object detection head has two components. The first one is a stack of fully connected network or multi layer-perceptron (MLP), which regresses the location of objects, and the second one is a single layer classification module. For each refined [obj]-token o^i , the object detection head predicts an object class $\tilde{v}_{\text{cls}}^i = \mathbf{W}_{\text{cls}}(o^i)$ and an object location $\tilde{v}_{\text{box}}^i = \text{MLP}_{\text{box}}(o^i)$, $\tilde{v}_{\text{box}}^i \in [0, 1]^{2 \times d}$ in parallel, where d represents the image dimension, \mathbf{W}_{cls} is the classification layer, and MLP_{box} is an MLP. We use the normalized bounding box co-ordinate for scale invariant prediction. Note that for the spatio-structural graph, we create virtual objects around each node’s center by assuming an uniform bounding box with a normalized width of Δx .

Relation Prediction Head: In parallel to the object detection head, the input of the relation head, given by a pair-wise [obj]-token and a shared [r1n]-token, is processed as $\tilde{e}_{\text{rln}}^{ij} = \text{MLP}_{\text{rln}}(\{o^i, r, o^j\}_{i \neq j})$. Here, r represents the refined [r1n]-token and MLP_{rln} a three-layer fully-connected network headed by layer normalization [2]. In the case of directional relation prediction (e.g., scene graph), the *ordering* of the object token pairs $\{o^i, r, o^j\}_{i \neq j}$ determines the direction $i \rightarrow j$. Otherwise (e.g., road network, vessel graph), the network is trained to learn object token *order* invariance as well.

Table 1: Brief summary of the datasets used in our experiment. For more details regarding dataset preparation, please refer to supplementary material.

Dataset	Description				Data Split		
	Edge Type	2D/3D	Image Type	Image Size	Train	Val	Test
Toulouse [5]	Undirected	2D	Binary	64x64	80k	12k	19k
20 US cities[18]	Undirected	2D	RGB	128x128	124k	13k	25k
Synthetic vessel [53]	Undirected	3D	Grayscale	64x64x64	54k	6k	20k
Visual Genome [28]	Directed	2D	RGB	800x800	57k	5k	26k

3.4 Loss Function

For object detection, we utilize a combination of loss functions. We use two standard box prediction losses, namely the ℓ_1 regression loss (\mathcal{L}_{reg}) and the generalized intersection over union loss ($\mathcal{L}_{\text{gIoU}}$) between the predicted \tilde{v}_{box} and ground truth v_{box} box coordinates. Besides, we use the cross-entropy classification loss (\mathcal{L}_{cls}) between the predicted class \tilde{v}_{cls} and the ground truth class v_{cls} .

Stochastic Relation Loss: In parallel to object detection, their pair-wise relations are classified with a cross-entropy loss. Particularly, we only use predicted objects that are assigned to ground truth objects by the Hungarian matcher. When two objects have a relation, we refer to their relation as a ‘valid’-relation. Otherwise, the relation is categorized as ‘background’. Since ‘valid’-relations are highly sparse in the set of all possible permutations of objects, computing the loss for every possible pair is burdensome and will be dominated by the ‘background’ class, which may hurt performance. To alleviate this, we randomly sample three ‘background’-relations for every ‘valid’-relation. From sampled ‘valid’- and ‘background’-relations, we obtain a subset \mathcal{R} of size M , where $\mathcal{R} \subseteq {}^N P_2$. To this end, \mathcal{L}_{rln} represents a classification loss for the predicted relations in \mathcal{R} . The total loss for simultaneous *object-relation* graph generation is defined as:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \sum_{i=1}^N \left[\mathbb{1}_{v_{\text{cls}}^i \notin \emptyset} (\lambda_{\text{reg}} \mathcal{L}_{\text{reg}}(v_{\text{box}}^i, \tilde{v}_{\text{box}}^i) + \lambda_{\text{gIoU}} \mathcal{L}_{\text{gIoU}}(v_{\text{box}}^i, \tilde{v}_{\text{box}}^i)) \right] \\ & + \lambda_{\text{cls}} \sum_{i=1}^N \mathcal{L}_{\text{cls}}(v_{\text{cls}}^i, \tilde{v}_{\text{cls}}^i) + \lambda_{\text{rln}} \sum_{\{i,j\} \in \mathcal{R}} \mathcal{L}_{\text{rln}}(e_{\text{rln}}^{ij}, \tilde{e}_{\text{rln}}^{ij}) \end{aligned} \quad (2)$$

where λ_{reg} , λ_{gIoU} , λ_{cls} and λ_{rln} are the loss functions specific weights.

4 Experiments

4.1 Datasets

We conducted experiments on four public datasets for the tasks of road network generation (20 US cities [18], Toulouse [5]), 3D synthetic vessel graph generation [53], and scene-graph generation (Visual Genome [28]). The road and vessel graph generation datasets are spatio-structural with a binary node and edge classification task, while the scene-graph generation dataset is spatio-semantic and has 151 node classes and 51 edge classes, including ‘background’ class.

4.2 Evaluation Metrics

Given the diversity of tasks at hand, we resort to widely-used task-specific metrics. Following is a brief description, while details can be found in the supplementary material. For *Spatio-Structural Graphs*, we use four different metrics to capture spatial similarity alongside the topological similarity of the predicted graphs. 1) *Street Mover Distance (SMD)*[5] computes a Wasserstein distance between predicted and ground truth edges; 2) *TOPO Score*[18] includes precision, recall, and F-1 score for topological mismatch; 3) *Node Detection* yields mean average precision (mAP) and mean average recall (mAR) for the node; and 4) *Edge Detection* yields mAP and mAR for the edges. For *Spatio-semantic Graphs*, the scene graph detection (SGDet) metric is the most challenging and appropriate for joint object-relation detection tasks, because it does not need apriori knowledge about object location or class label. Hence, we compute recall@K, mean-recall@K, and no-graph constraint (ng)-recall@K for $K = \{20, 50, 100\}$ on the SGDet task following Zellers et al. [61]. Further, we evaluate the quality of object detection using average precision, AP@50 (IoU=0.5) [31].

Table 2: Quantitative comparison of Relationformer with the different baselines for undirected graph generation datasets. Relationformer achieves a near-perfect solution for the Toulouse dataset and improves the results on the 20 US Cities dataset over baseline models. Relationformer translates a similar trend in 3D and significantly outperforms the heuristic-based approach on the synthetic vessel dataset.

Dataset	Model	Graph-level Metrics				Node Det.		Edge Det.	
		SMD ↓	Prec. ↑	Rec. ↑	F1 ↑	mAP ↑	mAR ↑	mAP ↑	mAR ↑
Toulouse (2D)	RNN [5]	0.04857	65.41	57.52	61.21	0.50	5.01	0.21	2.56
	GraphRNN [5]	0.02450	71.69	73.21	72.44	1.34	4.15	0.34	1.01
	GGT [5]	0.01649	86.95	79.88	83.26	2.94	13.31	1.62	9.75
	Relationformer	0.00012	99.76	98.99	99.37	94.59	96.76	83.30	89.87
20 US Cities (2D)	RoadTracer[3]	N.A.	78.00	57.44	66.16	N.A.	N.A.	N.A.	N.A.
	Seg-DRM[38]	N.A.	76.54	71.25	73.80	N.A.	N.A.	N.A.	N.A.
	Seg-Orientation[4]	N.A.	75.83	68.90	72.20	N.A.	N.A.	N.A.	N.A.
	Sat2Graph [18]	N.A.	80.70	72.28	76.26	N.A.	N.A.	N.A.	N.A.
	Relationformer	0.04939	85.28	77.75	81.34	29.25	42.84	33.19	13.45
Synthetic Vessel (3D)	U-net[46]+heuristics	0.01982	N/A	N/A	N/A	18.94	29.81	17.88	27.63
	Relationformer	0.01107	N/A	N/A	N/A	78.51	84.34	78.10	82.15

*N.A. indicates scores are not readily available. † N/A indicates that the metric is not applicable.

4.3 Results

Spatio-structural Graph Generation: In spatio-structural graph generation, both correct graph topology and spatial location are equally important. Note that the objects here are represented as points in 2D/3D space. For practical reasons, we assume a hypothetical box of $\Delta x = 0.2$ around these points and treat these boxes as objects.

The Toulouse dataset poses the least difficulty as we can predict a graph from a binary segmentation image. We notice that existing methods perform poorly. Our method improves the SMD score by three orders of magnitude. All other metrics, such as TOPO-Score (prec., rec., and F-1), indicate near-optimal topological accuracy of our method. At the same time, our performance in node and edge mAP and mAR is vastly superior to all competing methods. For the more complex 20 U.S. cities dataset, we observe a similar trend. Note that due to the lack of existing scores from competing methods (SMD, mAP, and mAR), we only compare the TOPO scores, which we outperform by a significant margin. However, when compared to the results on the Toulouse dataset, Relationformer yields lower node detection scores on the 20 U.S. cities dataset, which can be attributed to the increased dataset complexity. Furthermore, the edge detection score also deteriorates. This is due to the increased proximity of edges, i.e., parallel roads.

For 3D data, such as vessels, no learning-based comparisons exist. Hence, we compare to the current best practice [51], which relies on segmentation, skeletonization, and heuristic pruning of the dense skeleta extracted from the binary segmentation [14]. The purpose of pruning is to eliminate redundant neighboring nodes, which is error-prone due to the voxelization of the connectivity, leading to poor performances. Table 2 clearly depicts how our method outperforms the current method. Importantly, we find that our method effortlessly translates from 2D to 3D without major modifications. Moreover, our 3D model is trained end-to-end from scratch without a pre-trained backbone. To summarize, we propose the first reliable learning-based 3D spatio-structural graph generation method and show how it outperforms existing 2D approaches by a considerable margin.

Scene Graph Generation: We extensively compare our method to numerous existing methods, which can be grouped based on three concepts. One-stage methods, two-stage methods utilizing only image features, and two-stage methods utilizing extra features. Importantly, Relationformer represents a one-stage method without the need for extra features. We find that Relationformer outperforms all one stage methods in Recall and ng-Recall despite using a simpler backbone. In terms of mean-Recall, a metric addressing dataset bias, we outperform [34] and our contemporary [10] @50 and perform close to [10] @20.

In terms of object detection performance, we achieve an AP@50 of 26.3, which is close to the best performing one- and two-stage methods, even though we use a simpler backbone. Note that

Table 3: Quantitative results of Relationformer in comparison with state-of-the-art methods on the Visual Genome dataset. Relationformer achieves new one-stage state-of-the-art results and bridges the performance gap with two-stage models, while reducing model complexity and inference time significantly without the need for any extra features (e.g., glove vector, knowledge graph, etc.). Importantly, Relationformer outperforms two-stage models that previously reported mean-Recall@100 and ng-Recalls. Note that ‘-’ indicates that the corresponding results are not available to us.

	Method	Extra Feat.	Recall			mean-Recall			ng-Recall			AP @50	#param (M)↓	FPS ↑	
			@20	@50	@100	@20	@50	@100	@20	@50	@100				
Two-Stage	MOTIFS [61]	✓	21.4	27.2	30.5	4.2	5.7	6.6	-	3	0.5	35.8	20.0	240.7	6.6
	KERN [8]	✓	22.3	27.1	-	-	6.4	-	-	30.9	35.8	20.0	405.2	4.6	
	GPS-Net [33]	✓	22.3	28.9	33.2	6.9	8.7	9.8	-	-	-	-	-	-	
	BGT-Net [12]	✓	23.1	28.6	32.2	-	-	9.6	-	-	-	-	-	-	
	RTN [24]	✓	22.5	29.0	33.1	-	-	-	-	-	-	-	-	-	
	BGNN [29]	✓	23.3	31.0	-	7.5	10.7	-	-	-	-	29.0	341.9	2.3	
	GB-Net [60]	✓	-	26.3	29.9	-	7.1	8.5	-	29.3	35.0	-	-	-	
	IMP+ [58]	✗	14.6	20.7	24.5	2.9	3.8	4.8	-	22.0	27.4	20.0	203.8	10.0	
	G-RCNN[59]	✗	-	11.4	13.7	-	-	-	-	28.5	35.9	23.0	-	-	
One-Stage	FCSGG [34]	✗	16.1	21.3	-	2.7	3.6	-	16.7	23.5	29.2	28.5	87.1	8.3*	
	RelTR [10]	✗	20.2	25.2	-	5.8	8.5	-	-	-	-	26.4	63.7	16.6	
	Relationformer	✗	22.2	28.4	31.3	4.6	9.3	10.7	22.9	31.2	36.8	26.3	92.9	18.2*	

#param are taken from [10]. * Frame-per-second (FPS) is computed in Nvidia GTX 1080 GPU.

the object detection performance varies substantially across multiple backbones and object detectors. For example, BGNN [29] uses X-101FPN, FCSGG [34] uses HRNetW48-5S-FPN, whereas Relationformer and its contemporary RelTR [10] use a simple ResNet50 [17] backbone.

Comparing our Relationformer to two-stage models, we outperform all models that use no extra features in all metrics. Moreover, we perform almost equal to the remaining two-stage models, which use powerful backbones [29], bi-label graph resampling [29], custom loss functions [33], and extra features such as word [24] or knowledge graph embeddings [8]. Therefore, we can claim that we achieve competitive performances without custom loss functions or extra features while using significantly fewer parameters. We also achieve much faster processing times, measured in frames per second (FPS)(see Table 4.3). For example, BGNN [29], which was the top performer in a number of metrics, requires three times more parameters and is an order of magnitude slower than our method.

Fig. 4 shows qualitative examples for all datasets used in our experiments. Qualitative and quantitative results from both spatio-structural and spatio-semantic graph generation demonstrate the efficiency of our approach and the importance of simultaneously leveraging [obj]-tokens and the [r1n]-token. Relationformer achieves benchmark performances across a diverse set of image-to-graph generation tasks suggesting its wide applicability and scalability.

4.4 Ablation Studies

In our ablation study, we focus on two aspects. First, how the [r1n]-token and relation-head guide the graph generation; second, the effect of the sample size in training transformers from scratch. We select the complex 3D synthetic vessel and Visual Genome datasets for the ablation. Further ablation experiments can be found in the supplementary material.

In Table 4, we evaluate the importance of the [r1n]-token and different relation-head types. First, we train def-DETR only for object detection as proposed in [7, 64], second, we evaluate Relationformer w/ and w/o [r1n]-token and use a linear relation classification layer (models w/o the [r1n]-token use only concatenated pair-wise [obj]-tokens for relation classification). Third, we replace the linear relation head with an MLP and repeat the same w/ and w/o [r1n]-tokens.

We observe that a linear relation classifier w/o [r1n]-token is insufficient to model the mutual relationships among objects and diminishes the object detection performance as well. In contrast, we see that the [r1n]-token significantly improves performance despite using a linear relation classifier. Using an MLP instead of a linear classifier is a better strategy whereas the Relationformer w/ [r1n]-token shows a clear benefit. Unlike the linear layer, we hypothesize that the MLP provides a separate and adequate embedding space to model the complex semantic relationships for [obj]-tokens and our [r1n]-token.

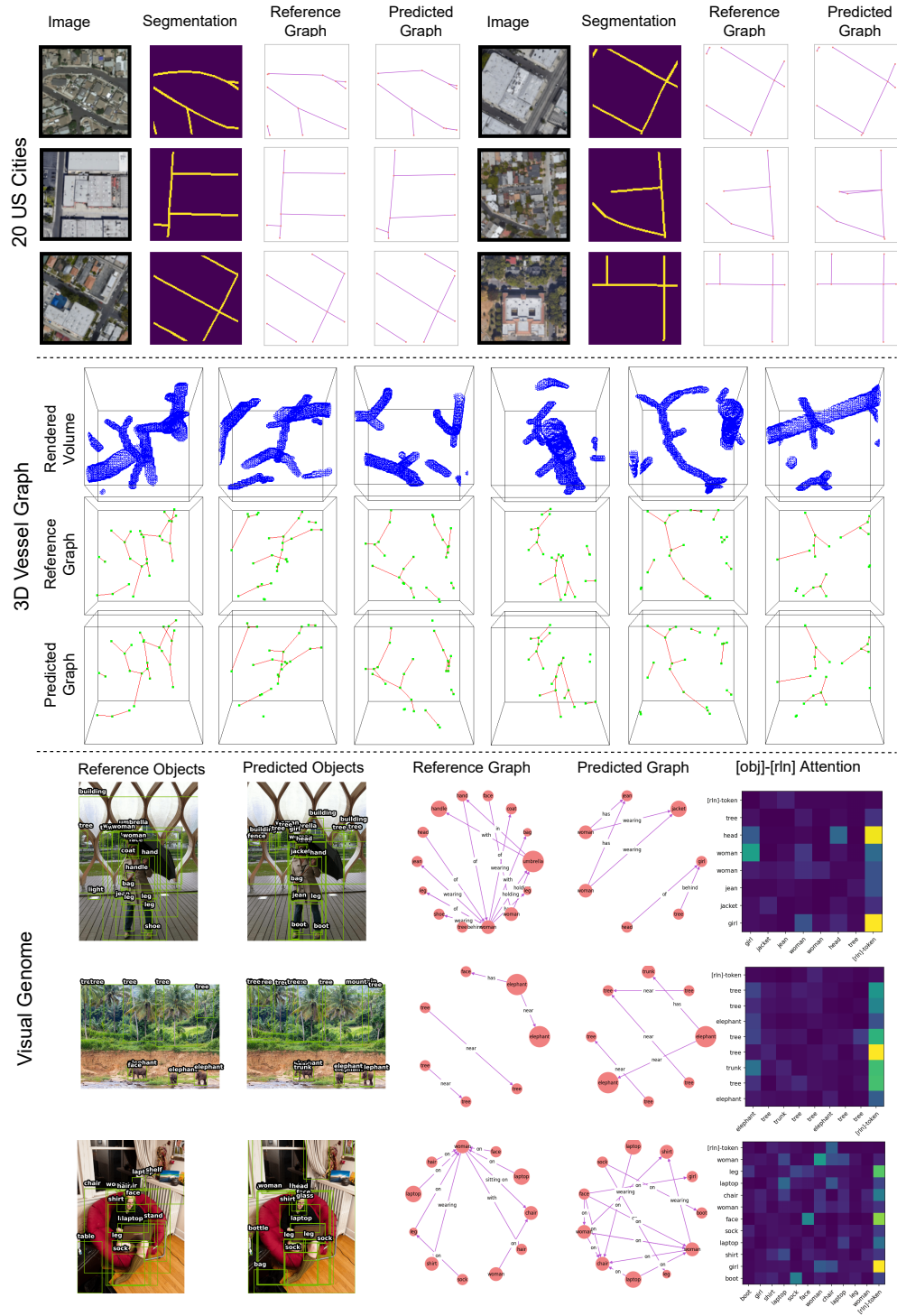


Figure 4: Qualitative results (better viewed zoomed in) from road-network, vessel-graph, and scene-graph generation experiments. Across all datasets, we observe that Relationformer is able to produce correct results. The segmentation map is given for better interpretability of road network satellite images. For vessel-graphs, we surface-render the segmentation of corresponding greyscale voxel data. For scene graphs, we visualize the attention map between detected [obj]-tokens and [r1n]-token, which shows that the [r1n]-token actively attends to objects that contribute to relation formation.

Table 4: Ablation on the [r1n]-token and relation head type on Visual Genome. We observe [r1n]-token significantly improves relation prediction for both types of relation heads. Importantly, the improvement is larger for the linear classifier than for the MLP.

Model	[r1n]-token	Rel. Head	SGDet Recall			
			AP @50	@20	@50	@100
def-DETR	N/A	N/A	26.4	N/A	N/A	N/A
Relationformer	✗	Linear	24.1	16.6	22.0	25.2
Relationformer	✓	Linear	25.3	20.1	25.4	28.3
Relationformer	✗	MLP	26.0	19.2	26.4	29.4
Relationformer	✓	MLP	26.3	22.2	28.4	31.3

Table 5: Ablation on the [r1n]-token and train-data size on synthetic vessel. We observe that [r1n]-token significantly improves both node and edge detection. Additionally, the scores improves with increased train-data size, suggesting further room for improvement by training on more data.

Model	[r1n]-token	Train Data	SMD	Node Det.		Edge Det.	
				mAP	mAR	mAP	mAR
def-DETR	N/A	100%	N/A	77.5	83.5	N/A	N/A
Relationformer	✗	100%	0.0129	75.5	81.6	76.3	80.4
Relationformer	✓	25%	0.0138	17.0	32.1	11.5	28.3
Relationformer	✓	50%	0.0124	39.2	53.5	33.3	48.9
Relationformer	✓	100%	0.0110	78.5	84.3	78.1	82.1

From ablation on 3D vessel (Table 5), we draw the same conclusion that [r1n]-token significantly improve over Relationformer w/o [r1n]-token. Further, a high correlation between performance and train-data size indicates additional room for improvement by increasing the sample size while training from scratch.

4.5 Limitations and Outlook:

In this work, we only use bipartite object matching, and future work will investigate graph-based matching [45]. Additionally, incorporating recent transformer-based backbones, i.e., Swin-transformer [35] could further boost the performance without compromising the simplicity.

5 Conclusion

Extraction of structural- and semantic-relational graphs from images is the key for image understanding. We propose Relationformer, a unified single-stage model for direct *image-to-graph* generation. Our method is intuitive and easy to interpret because it is devoid of any hand-designed components. We show consistent performance improvement across multiple *image-to-graph* tasks using Relationformer compared to previous methods; all while being substantially faster and using fewer parameters which reduce energy consumption. Relationformer opens up new possibilities for efficient integration of a *image-to-graph* models to downstream applications in an end-to-end fashion. We believe that our method has the potential to shed light on many previously unexplored domains and can lead to new discoveries, especially in 3D.

Acknowledgement

Suprosanna Shit is supported by TRABIT (EU Grant: 765148). Bjoern Menze gratefully acknowledges the support of the Helmut Horten Foundation.

References

- [1] Iro Armeni et al. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5664–5673, 2019.
- [2] Jimmy Lei Ba et al. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [3] Favyen Bastani et al. Roadtracer: Automatic extraction of road networks from aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2018.
- [4] Anil Batra et al. Improved road connectivity by joint learning of orientation and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10385–10393, 2019.
- [5] Davide Belli and Thomas Kipf. Image-conditioned graph generation for road network extraction. *arXiv preprint arXiv:1910.14388*, 2019.

- [6] Irwan Bello et al. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.
- [7] Nicolas Carion et al. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.
- [8] Tianshui Chen et al. Knowledge-embedded routing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6171, 2019.
- [9] Hang Chu et al. Neural turtle graphics for modeling city road layouts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4522–4530, 2019.
- [10] Yuren Cong et al. Reltr: Relation transformer for scene graph generation. *arXiv preprint arXiv:2201.11460*, 2022.
- [11] Jifeng Dai et al. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.
- [12] Naina Dhingra et al. Bgt-net: Bidirectional gru transformer network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2150–2159, 2021.
- [13] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [14] Dominik Drees et al. Scalable robust graph and feature extraction for arbitrary vessel networks in large volumetric datasets. *arXiv preprint arXiv:2102.03444*, 2021.
- [15] Yuxin Fang et al. You Only Look at One Sequence: Rethinking Transformer in Vision through Object Detection. *arXiv preprint arXiv:2106.00666*, 2021.
- [16] William L Hamilton et al. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.
- [17] Kaiming He et al. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [18] Songtao He et al. Sat2Graph: road graph extraction through graph-tensor encoding. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020.
- [19] Marcel Hildebrandt et al. Scene graph reasoning for visual question answering. *arXiv preprint arXiv:2007.01072*, 2020.
- [20] Jie Hu et al. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [21] Jingwei Ji et al. 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.
- [22] Xiang Ji et al. Brain microvasculature has a common topology with local differences in geometry that match metabolic load. *Neuron*, 109(7):1168–1187, 2021.
- [23] Justin Johnson et al. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.
- [24] Rajat Koner et al. Relation transformer network. *arXiv preprint arXiv:2004.06193*, 2020.
- [25] Rajat Koner et al. Graphhopper: Multi-hop scene graph reasoning for visual question answering. In *International Semantic Web Conference*, pages 111–127. Springer, 2021.
- [26] Rajat Koner et al. Oodformer: Out-of-distribution detection transformer. *arXiv preprint arXiv:2107.08976*, 2021.

- [27] Rajat Koner et al. Scenes and surroundings: Scene graph generation using relation transformer. *arXiv preprint arXiv:2107.05448*, 2021.
- [28] Ranjay Krishna et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.
- [29] Rongjie Li et al. 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*, pages 11109–11119, 2021.
- [30] Rongjie Li et al. Sgtr: End-to-end scene graph generation with transformer. *arXiv preprint arXiv:2112.12970*, 2021.
- [31] Tsung-Yi Lin et al. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [32] Tsung-Yi Lin et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [33] Xin Lin et al. Gps-net: Graph property sensing network for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3746–3753, 2020.
- [34] Hengyue Liu et al. Fully convolutional scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11546–11556, 2021.
- [35] Ze Liu et al. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.
- [36] Cewu Lu et al. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016.
- [37] Yichao Lu et al. Context-aware scene graph generation with seq2seq transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15931–15941, 2021.
- [38] Gellért Mátyus et al. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE international conference on computer vision*, pages 3438–3446, 2017.
- [39] Meyer-Spradow et al. Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications*, 29(6):6–13, 2009.
- [40] Arttu Miettinen et al. Micrometer-resolution reconstruction and analysis of whole mouse brain vasculature by synchrotron-based phase-contrast tomographic microscopy. *bioRxiv*, 2021.
- [41] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. *Advances in neural information processing systems*, 30, 2017.
- [42] Johannes C Paetzold et al. Whole brain vessel graphs: A dataset and benchmark for graph learning and neuroscience. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [43] Jeffrey Pennington et al. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [44] Shaoqing Ren et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [45] Michal Rolínek et al. Deep graph matching via blackbox differentiation of combinatorial solvers. In *European Conference on Computer Vision*, pages 407–424. Springer, 2020.
- [46] Olaf Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

- [47] Matthias Schneider et al. Tissue metabolism driven arterial tree generation. *Med Image Anal.*, 16(7):1397–1414, 2012.
- [48] Sahand Sharifzadeh et al. Classification by attention: Scene graph classification with prior knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5025–5033, 2021.
- [49] Sahand Sharifzadeh et al. Improving scene graph classification by exploiting knowledge from texts. *arXiv preprint arXiv:2102.04760*, 2021.
- [50] Sahand Sharifzadeh et al. Improving visual relation detection using depth maps. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3597–3604. IEEE, 2021.
- [51] Suprosanna Shit et al. cldice-a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16560–16569, 2021.
- [52] Hwanjun Song et al. ViDT: An Efficient and Effective Fully Transformer-based Object Detector. *arXiv preprint arXiv:2110.03921*, 2021.
- [53] Giles Tetteh et al. Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *Frontiers in Neuroscience*, 14:1285, 2020.
- [54] Mihail Ivilinov Todorov et al. Machine learning analysis of whole mouse brain vasculature. *Nature methods*, 17(4):442–449, 2020.
- [55] Hugo Touvron et al. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.
- [56] Ashish Vaswani et al. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [57] Danfei Xu et al. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [58] Danfei Xu et al. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.
- [59] Jianwei Yang et al. Graph r-cnn for scene graph generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 670–685, 2018.
- [60] Alireza Zareian et al. Bridging knowledge graphs to generate scene graphs. In *European Conference on Computer Vision*, pages 606–623. Springer, 2020.
- [61] Rowan Zellers et al. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840, 2018.
- [62] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks, 2018.
- [63] Xingyi Zhou et al. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.
- [64] Xizhou Zhu et al. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Transformer and Deformable-DETR

The core of a transformer [56] is the attention mechanism. Let us consider an image feature map \mathbf{f}_I , the q^{th} query with associated features \mathbf{f}_q and k^{th} key with associated image features \mathbf{f}_I^k . One can define the multi-head attention for M number of heads and K number of key elements as

$$\text{MultiHeadAttn}(\mathbf{f}_q, \mathbf{f}_I) = \sum_{m=1}^M \mathbf{W}_m \left[\sum_{k=1}^K \mathbf{A}_{mqk} \cdot \mathbf{W}'_m \mathbf{f}_I^k \right]$$

where \mathbf{W}'_m and \mathbf{W}_m are learnable weights. The attention weights $\mathbf{A}_{mqk} \propto \exp \left\{ \frac{\mathbf{f}_q^\top \mathbf{W}''_m{}^\top \mathbf{W}'''_m \mathbf{f}_I^k}{\sqrt{d_k}} \right\}$

are normalized as $\sum_{k=1}^K \mathbf{A}_{mqk} = 1$, where $\mathbf{W}''_m, \mathbf{W}'''_m$ are also learnable weights and d_k is the temperature parameter. To differentiate position of each element uniquely, \mathbf{f}_q and \mathbf{f}_I are given a distinct positional embedding.

In our work, we use the multi scale deformable attention [64] for L number of level as

$$\text{MSDefAttn}(\mathbf{f}_q, \mathbf{x}_q, \{\mathbf{f}_I^l\}_{l=1}^L) = \sum_{m=1}^M \mathbf{W}_m \left[\sum_{l=1}^L \sum_{k=1}^K \mathbf{A}_{mlqk} \cdot \mathbf{W}'_m \mathbf{f}_I^l(\phi_l(\mathbf{x}_q) + \Delta \mathbf{x}_{mlqk}) \right]$$

where ϕ_l rescales the normalized reference point coordinates appropriately in the corresponding image space.

B Dataset

Here we describe the individual datasets used in our experimentation in detail. We also elaborate on generating train-test sets for our experiments. For 20 U.S. Cities and 3D synthetic vessel we extract overlapping patches from large images. This provides us a large enough sample size to train our Relationformer from scratch. Since, a DETR like architecture is not translation invariant because of learned [obj]-tokens in the decoder, extracting overlapping patches drastically increases the effective sample size within a limited number of available images.

B.1 Toulouse Road Network

The Toulouse Road Network dataset [5] is based on publicly available satellite images from Open Streetmap and consists of semantic segmentation images with their corresponding graph representations. For our experiments we use the same split as in the original dataset paper with 80,357 samples in the training set, 11,679 samples in the validation set, and 18,998 samples in the test set [5].

B.2 20 U.S. Cities Dataset

For the 20 U.S. Cities dataset [18], there are 180 images with a resolution of 2048x2048. We select 144 for training, 9 for validation, and 27 for testing. From those images, we extract overlapping patches of size 128x128 to construct the final train-validation-test split. We crop the RGB image and the corresponding graph followed by a node simplification. Following Belli et al. [5], we prune the dense nodes by computing the angle between two road-segments at each node of degree 2 and only keep a node if the road curvature is less than 160 degrees. This allows eliminating redundant nodes and simplifying the graph prediction task. Fig. 5 illustrates the pruning process.

B.3 3D Synthetic Vessels

Our synthetic vessel dataset is based on publicly available synthetic images generated in Tetteh et al. [53]. In this dataset, the ground truth graph was generated by [47] and from that, corresponding voxel-level semantic segmentation data was generated. Grey valued data was obtained by adding different noise levels to the segmentation map. Specifically, we train on greyscale "images" and their corresponding vessel graph representations, where each node represents a bifurcation point, and the edges represent their connecting vessels. The whole dataset contains 136 3D volumes of size 325x304x600. First, we choose 40 volumes to create a train and validation set and next pick 10

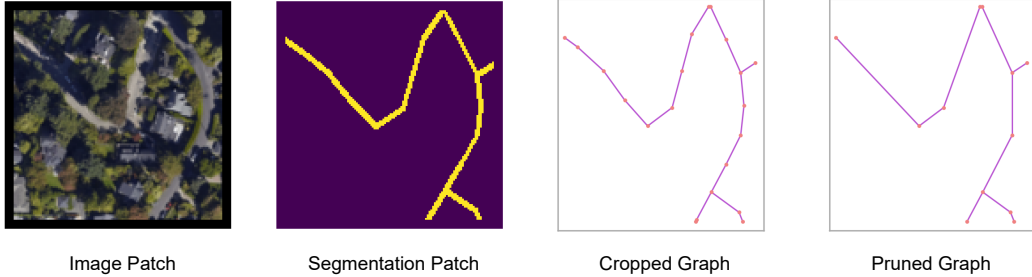


Figure 5: Preprocessing steps for the 20 U.S. Cities dataset. The same steps are followed in the 3D Synthetic Vessel dataset curation.

volumes for the test set. From this, we extract overlapping patches of size $64 \times 64 \times 64$ to construct the final train-validation-test set. Similar to the 20 U.S. cities dataset, we prune nodes having degree 2 based on the angle between two edges.

B.4 Visual Genome

Visual Genome is one of the largest scene graph datasets consisting of 108,077 natural images [28]. However, the original dataset suffers from multiple annotation errors and improper bounding boxes. Lu et al. [36] proposed a refined version of Visual Genome with the most frequent occurring 150 objects classes and 50 relation categories. It also proposed its own train/val/test splits and is the most widely used data-split [24, 33, 34, 61] for SGG. For fair comparison, we only train on the Visual Genome dataset and do **not** use any pre-training.

C Metrics Details

Metrics for Spatio-Structural Graph: We use three different kinds of metrics to capture spatial similarity alongside the topological similarity of the predicted graphs. The graph-level metrics include; 1) *Street Mover Distance (SMD)*: SMD[5] compute Wasserstein distance between the uniformly sampled fixed number of points (See Fig. 6) from the predicted and ground truth edges; and 2) *TOPO Score*: TOPO Score[18] computes precision, recall, and F-1 score for topological mismatch in terms of the false-positive and false-negative topological loop. Alongside, we use 3) *Node Detection*: For this, we report mean average precision (mAP) and mean average recall (mAR) over a threshold range [0.5,0.95,0.05] for node box prediction. Similarly, we use 4) *Edge Detection*: We compute the mAP and mAR for the edge in the same way as above. The edge boxes are constructed from the center points of two connecting nodes (See Fig. 6). For vertical and horizontal edges we assume an hypothetical width of 0.15 to avoid objects with near zero width.

Metrics for Spatio-Semantic Graph: We evaluate Relationformer on the most challenging Scene Graph Detection(SGDet) metrics and its variants. Unlike other scene graph metrics like Predicate Classification (PredCls) or Scene Graph classification (SGCls), SGDdet does not use apriori information on class label or object spatial position and does not rely on complex ROI-align based spatial features. SGDdet jointly measures the predicted boxes (with 50% overlaps) class labels of an object, and relation labels. The variants of SGDdet include 1) *Recall*: Recall at the different K (20, 50 and 100) of predicted relation that reflects overall relation prediction performance, 2) *Mean-Recall*: mean-Recall computes mean of each relation class-wise recall that reflects the performance under the relational imbalance or long-tailed distribution of relation class, 3) *ng-Recall*: ng-Recall is recall w/o graph constraints on the prediction, which takes the top-k predictions instead of just the top-1. Additionally, we use 4) *AP@50*: Average precision at 50% threshold of IOU reflects an average object detection performance.

D Model Details

Table 6, describes the backbone and important parameters of the Relationformer. We experiment with different ResNet backbones to show the flexibility of our Relationformer. In order to reduce energy consumption, we use the lighter ResNet50 for most 2D datasets. For the 3D experiment, we used

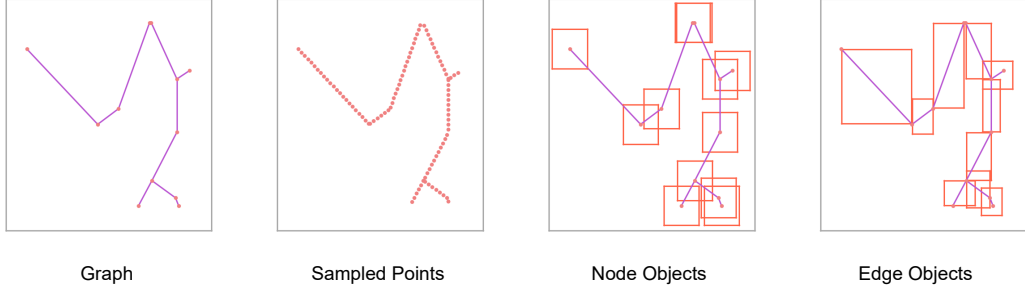


Figure 6: Sampled points, node objects and edge objects for computing different spatio-structural graph metrics. The same notion is used for 3D graphs.

Table 6: The model parameters used in Relationformer experiments across the various datasets. Specifically, we list details on the backbone and the transformer’s number of layers, feature dimension and other details.

DataSet	Backbone	Transformer				MLP Dim
		Enc. Layer	Dec. Layer	# [obj]-tokens	d_{emb}	
Toulouse	ResNet-50	4	4	20	256	512
20 US cities	ResNet-101	4	4	80	512	1024
Synth Vessel	SE-Net	4	4	80	256	1024
Visual Genome	ResNet-50	6	6	200	512	2048

Squeeze-and-Excite Net [20]. We used the number of encoder and decoder layers and the number of [obj]-tokens in the increasing order of dataset complexity. We find that four transformer layers and 20 [obj]-tokens suffice for Toulouse, while we need four transformer layers and 80 [obj]-tokens are required for 20 U.S. cities and synthetic vessel datasets. We need 6 layers of transformer and 200 [obj]-tokens for the visual genome. The ablation on the number of transformer layers and number of [obj]-tokens are shown in the next section.

E Training Details

Table 7: A list of the important set of parameters used in Relationformer for respective training. Furthermore, we list the weights for bipartite matching costs and training losses.

DataSet	Batch Size	Learning rate	Epoch	Cost Coeff.			Loss Coeff.			
				cls	reg	gIoU	λ_{reg}	λ_{gIoU}	λ_{cls}	λ_{rln}
Toulouse	64	10^{-4}	50	2	5	0	5	2	2	1
20 US cities	32	10^{-4}	100	3	5	0	5	2	3	4
3D Vessel Net	48	10^{-4}	100	2	5	0	2	3	3	4
Visual Genome	16	10^{-4}	25	3	2	3	2	2	4	6

Table 7, summarizes some principal parameters we use in the training. We use AdamW optimizer with a step learning rate. For scene graph generation, we use the prior statistical distribution or frequency-bias [61] of relation for each subject-object pair. To minimize the data imbalance for a relation label present in the Visual Genome, we use log-softmax distribution [33] to soften the frequency bias. Finally, we add this distribution with the predicted relation distribution from the relation head. For the spatio-structural dataset, we set the cost coefficient for the GIoU in the bipartite matcher to be zero because we assume 0.2 widths of the normalized box for each node. Hence, ℓ_1 cost is sufficient to consider for the spatial distances.

F More Ablation Studies on [obj]-tokens and Transformer

We conduct two more ablation studies on Visual Genome for analyzing the influence of [obj]-tokens and optimal number of layers in transformer for the joint graph generation. Furthermore Figure. 7 gives additional insight how [r1n]-token is beneficial for joint object-relation graph.

Table 8: Impact of the [obj]-tokens on joint object and relation detection.

#[obj]-tokens	AP@50	R@20	R@50	R@100
75	25.1	20.6	26.1	29.5
100	25.8	21.1	27.4	30.6
200(ours)	26.3	22.2	28.4	31.3
300	26.3	21.9	27.9	31.0

Table 9: Impact of the transformer’s layers on joint object and relation detection

# layer	AP@50	R@20	R@50	R@100
4	24.6	20.5	26.5	28.8
5	25.2	21.0	27.2	29.9
6(ours)	26.3	22.2	28.4	31.3

As shown in Table 8, it can be observed that increasing [obj]-tokens does increase object and relation detection performance. However, it becomes relatively stable with increasing object queries. DETR-like architectures rely on an optimal number of [obj]-tokens to balance positive and negative samples which also helps in object detection as observed in [7]. Thus, in a joint object and relation prediction, a gain might come from optimal number [obj]-tokens, as relation prediction is linearly co-related to object detection performance. It demonstrates that joint object and relation detection can perfectly coexist without hurting the object detection performance. Instead, it can exploit [obj]-tokens enriched with global relational reasoning for efficient relation extraction.

During the ablation with transformer layers, we observe decreasing number of transformer layers shows an initial gain in object and relation detection. However, they lead to early plateau and inferior performance as depicted in table 9. One intuitive reason is that with less parameter and insufficient contextualization Relationformer quickly learn the initial biases present in both object and relation detection and failed to learn the complex global scenario. We use the same number of layers for both encoder and decoder.

G Qualitative Results

Fig. 8 and 9 shows additional qualitative example from our experiments.

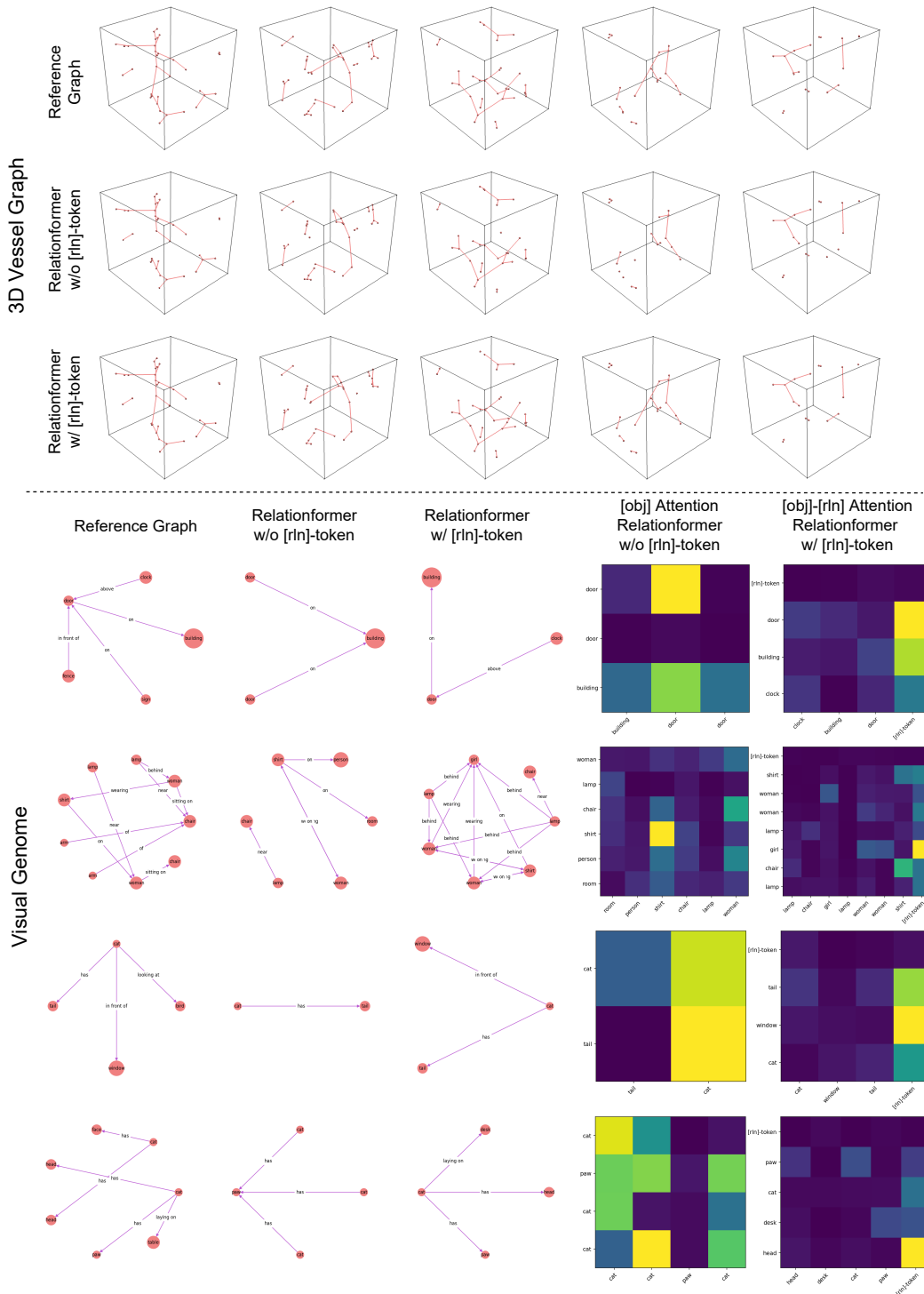


Figure 7: Typical qualitative results (please zoom in) from our ablation on the synthetic vessel-graph and visual genome datasets. We observe that Relationformer w/o [rln]-token is missing vessel edges while Relationformer w/ [rln]-token produces correct edges. For visual genome, we can see w/o [rln]-token the [obj]-tokens have to carry extra burden for relation prediction and sometimes fail to incorporate the global relation. However, the inclusion of [rln]-token provides an additional path to flow relation information that benefits the joint object and relation detection.

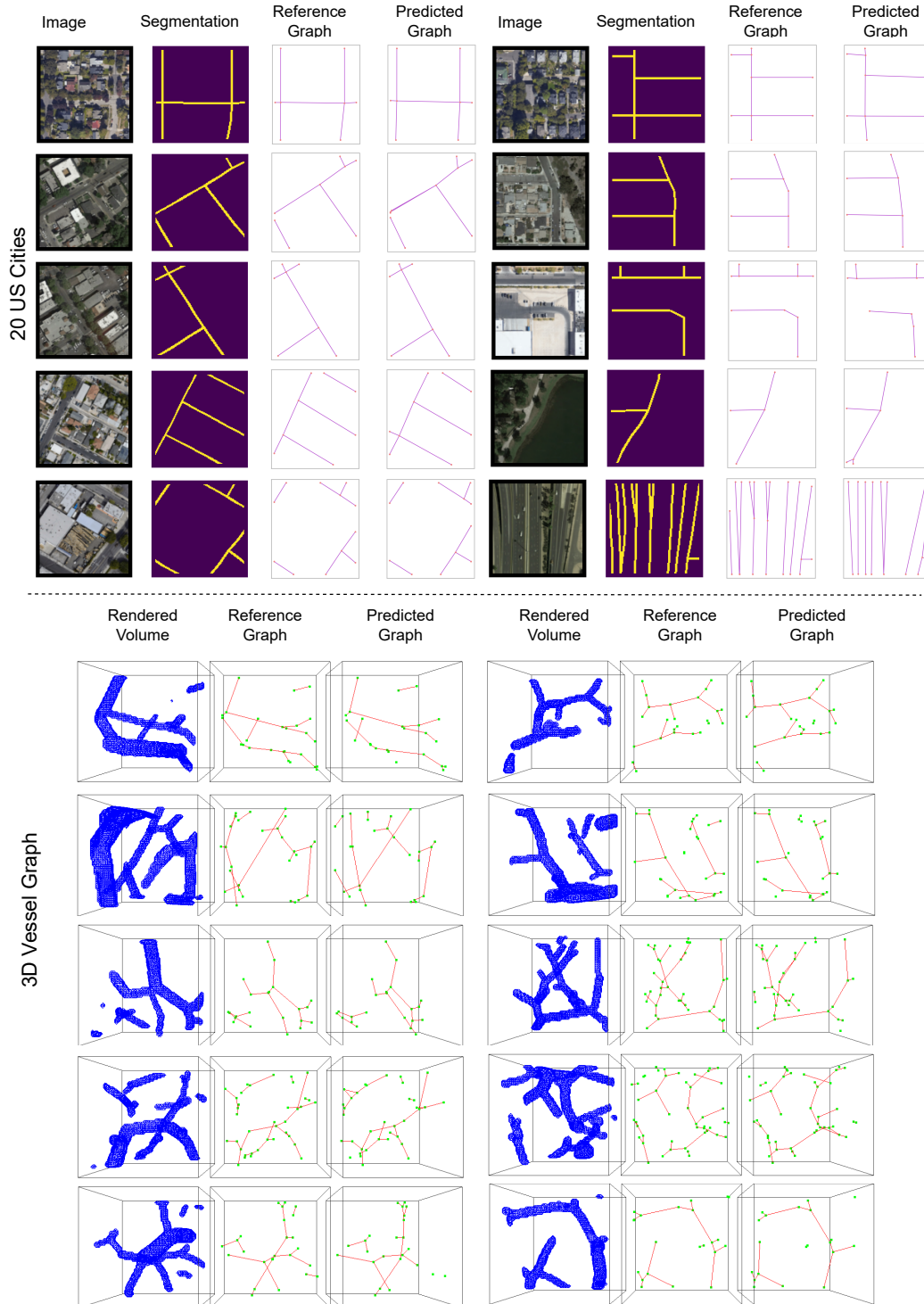


Figure 8: Qualitative results (please zoom in) for the 20 US cities road-network and synthetic vessel-graph experiments. We observe that Relationformer is able to produce correct results. The segmentation map is given for better interpretability of road network satellite images. For vessel-graphs, we surface-render the segmentation of the corresponding greyscale voxel data.

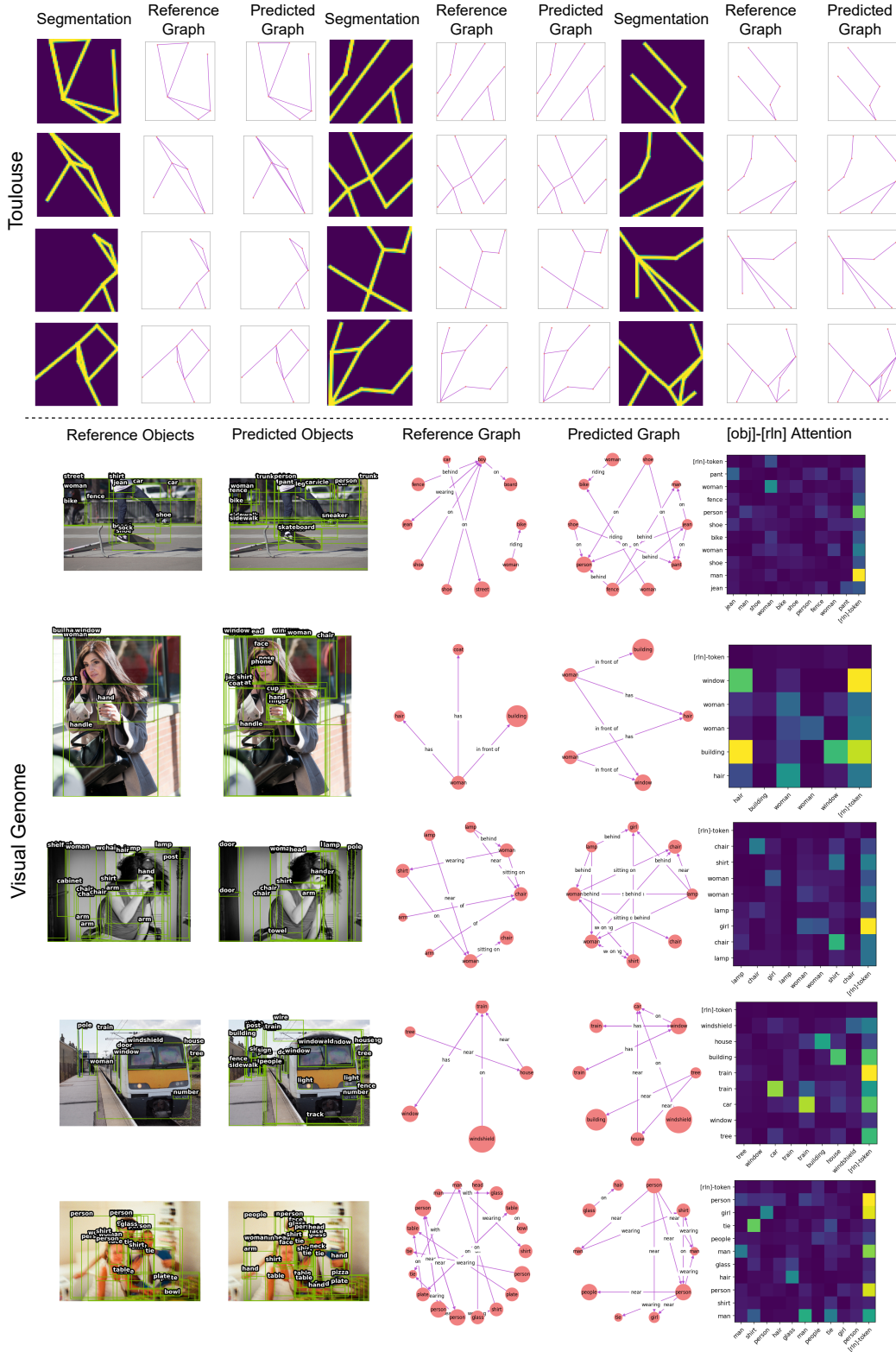


Figure 9: Qualitative results (please zoom in) from the Toulouse road-network and scene-graph generation experiments. For both datasets, we observe that Relationformer is able to generate an accurate graph. For scene graphs, we visualize the attention map between detected [obj]-tokens and [rln]-token, which shows that the [rln]-token actively attends to objects that contribute to relation formation.