Segment Anything Model for Road Network Graph Extraction

Congrui Hetang
Carnegie Mellon University
congruihetang@gmail.com

Haoru Xue
Carnegie Mellon University
haorux@andrew.cmu.edu

Cindy Le
Columbia University
xl2738@columbia.edu

Tianwei Yue
Carnegie Mellon University
tyue@alumni.cmu.edu

Wenping Wang
Carnegie Mellon University
wenpingw@alumni.cmu.edu

Yihui He
Carnegie Mellon University
he2@alumni.cmu.edu

Abstract

We propose SAM-Road, an adaptation of the Segment Anything Model (SAM) [27] for extracting large-scale, vectorized road network graphs from satellite imagery. To predict graph geometry, we formulate it as a dense semantic segmentation task, leveraging the inherent strengths of SAM. The image encoder of SAM is fine-tuned to produce probability masks for roads and intersections, from which the graph vertices are extracted via simple non-maximum suppression. To predict graph topology, we designed a lightweight transformer-based graph neural network, which leverages the SAM image embeddings to estimate the edge existence probabilities between vertices. Our approach directly predicts the graph vertices and edges for large regions without expensive and complex post-processing heuristics and is capable of building complete road network graphs spanning multiple square kilometers in a matter of seconds. With its simple, straightforward, and minimalist design, SAM-Road achieves comparable accuracy with the state-of-the-art method RNGDet++[57], while being 40 times faster on the City-scale dataset. We thus demonstrate the power of a foundational vision model when applied to a graph learning task. The code is available at https://github.com/htcr/sam_road.

1. Introduction

Road network graphs are spatial representations of the structure and layout of road networks. They are typically stored in a vectorized format [20], consisting of vertices and edges. The vertices may represent intersections, and edges could stand for road segments. Large-scale road network graphs are vital for various applications: they enable navigation systems like Google Maps to determine optimal routes, assist in path planning for autonomous vehicles [18, 63], and help city planners in traffic analysis and optimization [4], to name a few. These applications call for accurate and efficient methods to automatically create such graphs, as they require scaling to huge regions and even near-continuous updating [10], which are astoundingly expensive when manually done. Therefore, systems for automatically generating such maps have tremendous application value and are under active research.

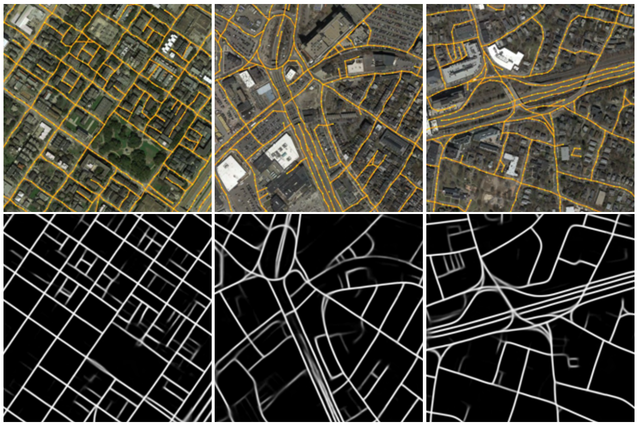


Figure 1. SAM-Road effectively predicts accurate road network graphs for dense urban regions, including roads with complex and irregular shapes, bridges, and multi-lane freeways. The corresponding segmentation masks are sharp and clear.

Recently, the rapid growth of foundational models [2, 44, 45, 50] showcased their impressive capabilities. These models, which leverage flexible, high-capacity, and scalable architectures such as Transformers [51], are pre-trained through effective self-supervision [22] methods and unprecedentedly large datasets. This endows them with robust semantic reasoning and generalization. Segment Anything Model (SAM) [27] is such a foundational vision model. Trained with millions of images and billions of masks, it demonstrates unparalleled semantic segmentation capabilities. This raises intriguing questions: How can SAM be applied to the prediction of road network graphs from satellite images, and how good can it be?

In this work, we answer these questions by introducing the SAM-Road model, which adapts the SAM for generating large-scale, vectorized road network graphs. Incorporating domain knowledge from previous research in satellite mapping, we divide the problem into two main components: geometry prediction and topology reasoning.

We model graph geometry with a set of 2D vertices that, when densely sampled, accurately reflect the graph's overall shape. The SAM-Road model first predicts dense segmentation masks to indicate the likelihood of road elements such as lane segments and intersections, then it employs simple non-maximum suppression to convert the pixels into vertices of the desired density. Leveraging the inherent semantic segmentation capabilities of SAM, this method can effectively capture highly complex shapes (see Figure 1), which are common in dense urban areas.

A notable challenge for segmentation-based mapping approaches is the difficulty of inferring topology from dense imagery. This branch of methods often relied on slow, complex and error-prone post-processing heuristics. Inspired by recent advances in graph learning [14, 48], we developed a transformer-based graph neural network as the second stage of our model. This network focuses on predicting the local subgraph around each vertex and determining connectivity with nearby vertices to establish the overall graph topology. It utilizes relative vertex positions and image embeddings from the SAM backbone to guide its predictions.

Despite its straightforward design, SAM-Road achieves accuracy comparable to more complex state-of-the-art systems on two widely recognized satellite mapping datasets: City-scale [23] and SpaceNet [17]. Moreover, for large spatial areas spanning multi-square kilometers, its architecture supports high degrees of parallelism and rapid GPU inference, achieving speeds up to 80 times faster than existing methods. We hope that this work will inspire further exploration of foundational vision models in remote sensing and graph learning tasks.

2. Related Works

2.1. SAM and Its Applications

In 2023, Segment Anything Model [27] was proposed as a foundational model for image segmentation, showcasing impressive zero-shot and generalization capabilities. Through fine-tuning or direct adoption, SAM has been used in object detection [34], image inpainting [59], segmentation of medical images [26, 37, 53, 61], and remote sensing tasks [11]. Existing adaptations of SAM in remote sensing have focused more on simple segmentation and have not yet been applied to the production of road network graphs.

2.2. Road Network Graph Prediction

Research on road network graph detection dates back to 2010 [40]. Representative methods fall into two categories: segmentation-based and graph-based approaches.

Segmentation-based methods [6, 23, 38] treat the task as a dense mask prediction. They represent the road network graph structure through one or more images, each detailing aspects such as road existence, intersections, orientation [6], and connectivity [23]. Post-processing heuristics, such as thinning [13, 62] and path-finding [28], are then employed to extract the vectorized graph structure. Benefits of this approach include 1) the ability of segmentation masks to represent complex geometries as a bottom-up volumetric representation [47], and 2) ease of parallel patch-wise inference for large areas, and subsequent result aggregation for refinement. However, the challenge of topology prediction persists: handcrafted heuristics often fail with poor mask quality; even with high-quality masks, deriving topology from them remains ill-formed. There exist no universal heuristics for all complex road structures, like multi-way intersections, multi-lane highways, and overpasses. Moreover, the heuristic tends to rely on CPU-intensive logic, which often becomes the inference speed bottleneck.

Graph-based methods have gained popularity recently, offering a more end-to-end approach. Unlike methods that use intermediate representations like mask images, they directly predict graph nodes and edges in vectorized form. Leading examples include RoadTracer [5], RNGDet [56], and RNGDet++ [57], with similar advancements in high-definition map generation for autonomous vehicles[33, 35, 39]. These methods reduce dependence on handcrafted graph generation rules, largely leveraging DETR-like[9, 33, 35, 60] techniques for geometric element prediction or adopting an autoregressive [5, 39, 56, 57] approach for incremental graph construction. Despite their strengths and contributions to the state-of-the-art [57], limitations exist: 1) DETR-like methods struggle with more than a few dozen entities due to the $O(N^2)$ computational complexity of transformer layers, limiting their applicability to city-scale road network graphs with potentially thousands of nodes and edges, and 2) autoregressive methods are difficult to parallelize as they rely on the outcomes of previous steps, significantly slowing down the process.

Our method combines the advantages of segmentation-based and graph-based approaches. It harnesses the exceptional capabilities of SAM to generate a high-quality mask for geometry prediction, and uses a transformer-based graph neural network to directly produce graph structures without handcrafted post-processing heuristics.

2.3. Graph Representation and Learning

Graph representation and learning [21] involves mapping data to graph structures and applying learning algorithms to understand complex relationships within. Significant advancements have been made in this area with the development of Graph Neural Networks (GNNs) [54], Graph Convolutional Networks (GCNs) [29], and Transformers adapted for graph data [14]. Entities with rich structures can be represented as graphs and predicted by deep nets, such as scene graphs [19], human keypoints [32, 55], meshes [41], and in our case, road networks. The goal is to predict whether a graph edge (road segment) exists between a pair of nodes (vertices). For this type of task, GCN is a suitable architecture choice, as they offer powerful mechanisms for aggregating local subgraph information and understanding node relationships. With multiple layers, long-range dependencies can be captured too. In SAM-Road, we adopt Transformers as a special form of GCN: their self-attention mechanism has a simple form and can automatically select the most relevant context [3, 18, 49, 51] without any preset structure.

3. Method

3.1. Overall Architecture

The overall structure of SAM-Road is shown in Figure 2. It contains an image encoder taken from the pre-trained SAM [27], a geometry decoder, and a topology decoder. The model takes as input an RGB satellite imagery. First, the image encoder produces the image feature embeddings. Then, the geometry decoder predicts the per-pixel existence probability, for both roads and intersections. The set of graph vertices $\mathbf{V}\{v_i \in \mathbb{R}^2\}$ representing 2D locations is extracted from these masks with a simple non-maximum suppression process, detailed in Algorithm 1. Given the predicted vertices, the topology decoder goes over each of them and determines whether it should connect to its nearby vertices within a given radius R_{nbr}, given its local context. For an edge (v_i, v_j), it predicts the probability that it exists. One edge may be predicted more than once, its final score will be the average. Eventually, the road network graph $\mathbf{G}$ is predicted as the sets of vertex $\mathbf{V}$ and edges $\mathbf{E}$.

3.2. Image Encoder

The image encoder is taken from a pre-trained Segment Anything Model. We use the smallest ViT-B variant, which has around 80M trainable parameters. It uses a ViT [16] architecture adapted for high-resolution images, as described in ViTDet [30]. The image encoder converts an $(H_{\text{img}}, W_{\text{img}}, 3)$ RGB image into a $(H_{\text{img}}/16, W_{\text{img}}/16, D_{\text{feat}})$ feature map, for the decoders to consume. The image is first divided into 16×16 non-overlapping patches, then each patch is encoded into an embedding vector, producing an $(H_{\text{img}}/16, W_{\text{img}}/16, D_{\text{feat}})$ tensor. A stack of 12 multi-head self-attention layers processes this tensor to the final feature map, alternating between windowed [36] and global self-attention. The feature size stays constant along the way. During training, we fine-tune the entire image encoder with $0.1\times$ base learning rate to adapt it to satellite imagery.

3.3. Geometry Decoder

The graph geometry prediction is formulated as a dense semantic segmentation task. There are two main benefits: First, this formulation leverages the extraordinary power of SAM; Second, per-pixel bottom-up representation can handle arbitrarily complex road structures.

The mask decoder has a minimalist design: it's simply 4 transposed convolution layers with 3×3 kernels and stride 2, each doubling the spatial feature resolution and decreasing the channel number. Eventually, it produces two probability maps as an $(H_{\text{img}}, W_{\text{img}}, 2)$ tensor, with the same size as the input image, representing the existence probability of intersection points and roads. This mask decoder contains about 170K trainable parameters.

After acquiring the masks, the graph vertices are extracted from them. This process converts the dense mask images into a set of sparse vertices, with roughly the same interval d_v in between. d_v is selected to be sparse while not hurting geometry accuracy. It's implemented with simple non-maximum suppression: we first drop the pixels under a probability threshold t, then traverse them by a descending order of their probability. Pixels within a d_v radius of the current one are removed. The (x, y) locations of the remaining pixels form the graph vertices $\mathbf{V}\{v_i \in \mathbb{R}^2\}$. See Algorithm 1.

Algorithm 1 Non-Maximum Suppression of Vertices

```
$\mathbf{V} \leftarrow \emptyset$
$t \leftarrow$ threshold value
$d_v \leftarrow$ radius for non-maximum suppression
**for** each pixel in the image **do**
    **if** pixel value $> t$ **then**
        Add pixel coordinates $(x, y)$ to $\mathbf{V}$
    **end if**
**end for**
Sort $\mathbf{V}$ by pixel values in descending order
**for** each $(x, y)$ in $\mathbf{V}$ **do**
    **for** each $(x', y')$ after $(x, y)$ **do**
        $d \leftarrow$ distance between $(x', y')$ and $(x, y)$
        **if** $d < d_v$ **then**
            Remove $(x', y')$ from $\mathbf{V}$
        **end if**
    **end for**
**end for**
```

We predict masks for both intersections and roads for more accurate graph structures at intersections. If only the road mask existed, there would be no guarantee that the center point of an intersection would be kept, producing error patterns like Figure 6. To mitigate this: 1) Vertices are extracted from both masks with the same NMS algorithm. 2) The two sets of vertices are joined, with all intersection ver-

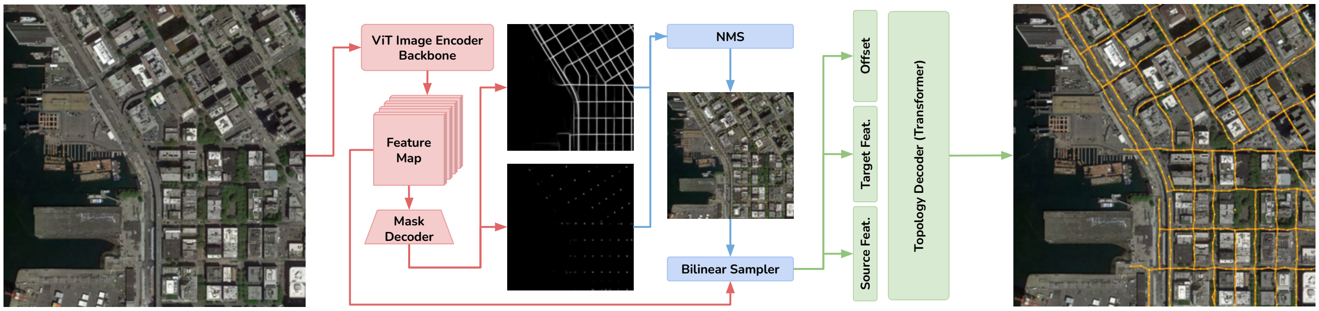
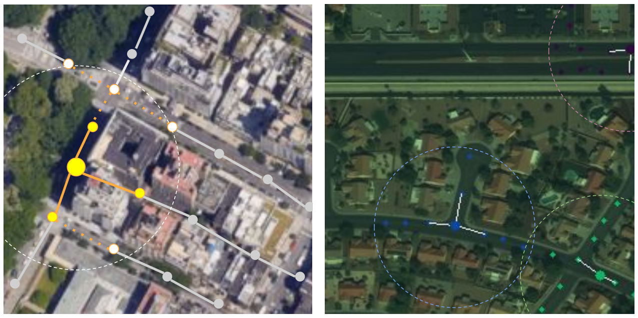


Figure 2. The architecture of our approach, SAM-Road. It contains an image encoder taken from the pre-trained SAM [27], a geometry decoder, and a topology decoder. It directly predicts vectorized graph vertices (yellow) and edges (orange) from an input RGB satellite imagery. Better zoom-in and view in color.



(a) Topology label example (b) Actual topology samples

Figure 3. Illustrating the definition of topology labels. In (a), the white dashed circle represents R_{nbr}; the large dot is the source node, and the smaller yellow dots are the target nodes. Orange lines are connected pairs. In (b), a few real topology samples used for training are shown. The query for one source node is shown in the same color. White lines are positive labels and pairs without lines are negative.

tices assigned a higher score than any road vertices. 3) The joined set is then NMS-processed again to produce the final result. This ensures intersection points are kept as much as possible.

3.4. Topology Decoder

The topology decoder ”wires up” the predicted graph vertices into the correct structure. It is a transformer-based graph neural network that predicts the existence of edges. It predicts the edge existence probability in small local subgraphs around each vertex. Specifically, for a given source vertex, up to $N_{\textbf{nbr}}$ nearest vertices are found within a radius of R_{nbr}. These form the target vertices. The topology decoder then predicts whether the source vertex shall connect with each of the targets, based on their spatial layout and image context.

The connection here is defined as ”whether two vertices are immediate neighbors on the graph”. That is, imagine a breadth-first-search on the road network graph from the source vertex, which stops expanding whenever a) it hits a target vertex or b) the depth (search radius) exceeds R_{nbr} - a target vertex is only connected to the source if it is visited by the search. This is further illustrated in Figure 3.

We formulate the topology prediction task as a binary classification problem on the $(v_{\text{src}}, v_{\text{tgt}})$ vertex pairs, conditioned on the image context. The input of the decoder is a sequence of high-dimensional feature vectors $\{(f^{\text{src}}, f^{\text{tgt}}_k, \vec{d_k}) \mid 0 \leq k < N_{\text{nbr}}\}$ where f^{src} and f^{tgt}_k are the vertex features. They are image embedding vectors acquired by bilinear sampling from the SAM image feature map at the source and target vertex locations. $\vec{d_k}$ is the offset from the source to the k-th target, encoding the relative spatial layout of the vertices of interest. These vectors are concatenated to a tensor shaped $(N_{\textbf{nbr}}, 2D_{\text{feat}} + 2)$, then projected to a $(N_{\textbf{nbr}}, D_{\text{feat}})$ feature tensor. We treat the $N_{\textbf{nbr}}$ dimension as sequence length and pass it through 3 multi-head self-attention layers with ReLU activations for message-passing to understand the multi-hop structures. The interacted feature sequence shaped $(N_{\textbf{nbr}}, D_{\text{feat}})$ is fed into a linear layer to get the $N_{\textbf{nbr}}$ binary classification logits. A sigmoid layer turns these into (0, 1) probabilities, indicating how likely the edge exists.

3.5. Label Generation

Mask Labels. For road mask labels, we rasterize the ground-truth road lines, by drawing each edge as a line segment, with a width of 3 pixels. The pixels covered by the line segments are set to 1, and others are 0. For intersection labels, we find all the graph vertices with a degree not equal to 2 and render them as circles with a radius of 3 pixels. This is partially inspired by the OpenPose [8] work which represents human keypoint graphs as heatmaps.

Topology Labels. During training, we don’t run the vertex extraction process. The topology decoder is trained in a teacher-forcing [52] manner, where the vertices being asked are not from model prediction, but sampled from ground-truth road network graphs to emulate the predictions. This

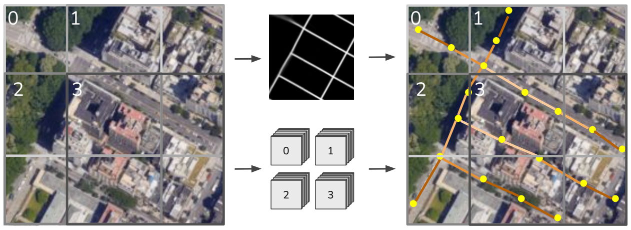


Figure 4. SAM-Road can predict the entire road network graph for arbitrarily large regions by operating in a sliding-window manner. 0-3 represent 4 overlapping windows. It first extracts the global nodes, caches the per-window embeddings, and then aggregates the per-window edge predictions.

is done by first subdividing the ground-truth graph and then running the same NMS procedure as the inference stage. To emulate various NMS results, a uniform random score is assigned to each subdivision vertex.

Having the emulated vertex predictions, we randomly sample $N_{\textbf{sample}}$ source vertices and apply the rules described in section 3.4 to find its targets and connectivity labels. Further, a small random Gaussian perturbation is applied to the vertex coordinates to emulate the prediction noise for better generalization.

The satellite imagery in the datasets used covers large square areas up to 4 square kilometers [23], therefore we randomly crop the RGB image, ground-truth masks, and graphs into smaller patches to get more training samples and keep memory consumption manageable.

3.6. Sliding-window Inference for Large Regions

SAM-Road can predict the entire road network graph for arbitrarily large regions by operating in a sliding-window manner, as shown in Figure 4. The predictions within each window can be aggregated to improve accuracy. Fusing multiple observations is a common practice in vision applications [15, 28, 31, 43] to effectively suppress noise. For SAM-Road, this applies to both geometry and topology.

For geometry, the per-window masks are fused to a large mask before vertex extraction, where each pixel value is the sum of all observed probabilities divided by the time it is observed. The NMS process is run on the fused global mask to get the global graph vertices.

For topology, when it comes to large regions, the topology decoder is run in a second pass after extracting the global vertices. The per-window image feature maps are cached, and for each window, the topology decoder infers the graph edges for the global vertices within that window, based on its image feature map. Since the vertices here are global, each edge prediction within each window can vote towards an edge in the global graph. The final edge probability in the global graph is the average of all observations similar to the mask.

It's also worth noting that the per-window inferences are completely independent of each other and can be done fully in parallel. This enables SAM-Road to be significantly faster (See Table 2) than the state-of-the-art RNGDet++ [57] that reconstructs the graph in an auto-regressive manner. The ease of multi-window aggregation for quality refinement, akin to dense semantic segmentation, is also uncommon for typical graph-based methods. SAM-Road can flexibly trade-off between speed and accuracy, by varying the stride size in sliding-window inference, as shown in Table 3.

4. Experiments

4.1. Datasets

We conduct our experiments on two datasets: City-scale [23] and SpaceNet [17]. The City-scale dataset includes 180 satellite images of 20 U.S. cities, each image has 2048 × 2048 pixels, and 29 are for testing. The SpaceNet dataset contains 2549 images of 400 × 400 pixels of cities around the world including Shanghai, Las Vegas, and more. 382 of them are for testing.

Both datasets have a 1 meter/pixel resolution. The ground-truth vector graphs of the road network are supplied. The two datasets feature diverse environments and road network patterns, facilitating conclusive experiments.

4.2. Metrics

We employ TOPO [7], an evaluation metric tailored for road network graphs. TOPO randomly samples candidate vertices in the ground truth and finds its correspondence in the prediction. It then compares the similarity of reachable subgraphs from the same vertex of the two graphs in terms of precision, recall, and F1. It focuses on geometric accuracy with a heavy penalty for incorrect disconnections.

We also utilize APLS (Average Path Length Similarity) [17] to evaluate the topological correctness. For a random vertex pair (v_1, v_2) on the ground truth and their correspondences in the prediction $(\hat{v_1}, \hat{v_2})$, we evaluate the model by comparing the shortest distance between (v_1, v_2) and between $(\hat{v_1}, \hat{v_2})$. Smaller distance difference indicates high topological similarity.

4.3. Implementation Details

For both datasets, d_v is 16 pixels (meters), R_{nbr} is 64 pixels (meters), N_{nbr} is 16, D_{feat} is 128. At training time, For City-scale, we sample image patches of 512 × 512 pixel, the batch size is 16 and we sample 512 source points for topology query per image patch. For SpaceNet, the batch size is 64 due to using image patches 256 × 256 pixel. We sample 128 source points per patch. When there are fewer than N_{nbr} available target nodes to query, we use attention masking to ensure the interaction only happens between the valid vertices.

We applied simple augmentations to boost data diversity. 1) Rotational: we randomly rotate the patch by the multiple of 90 degrees. 2) Translational: different from previous works that usually pre-crops the patches by a fixed grid and stores them to disk, we load the entire dataset in memory, and randomly sample patches in continuous spatial coordinates. This can be seen as a random-translation augmentation.

Masks and topology prediction are essentially binary classifications. We use the vanilla binary cross entropy loss for all of them and don't apply any loss re-weighting in this work. We take the mean loss of all valid entries. The three sub-tasks have equal loss weight, and the total loss is just adding them together.

We use the Adam optimizer with base LR of 0.001, which applies to the randomly initialized mask decoder and topology decoder. We use the default weight initialization of PyTorch. The image encoder is fine-tuned with $0.1\times$ base LR. LR is constant during training, with no scheduling tricks applied. We train SAM-Road on the two datasets respectively till validation metrics plateaus.

At inference time, we use 16x16 sliding window inference for the main results. To determine the threshold for the binary classifiers (intersection, road, edge connection), we find the threshold that gives the highest F1 score on the validation set. Note that this is just for isolating away the effect of threshold choice in the experiments, and is not critical for SAM-Road performance, as evidenced by the result that just uses 0.5 for everything in Table 4 (A vs H).

All experiments are conducted on one RTX 4090 GPU.

4.4. Evaluating Road Network Prediction

Qualitative results of SAM-Road predicting large-scale road network graphs can be found in Figure 5. The results are shown side-by-side with two baselines and the ground-truths. Some error examples can be found in Figure 7. Overall, SAM-Road predicts highly accurate road networks even under very challenging circumstances, e.g. many blocks and intersections in dense urban areas, curvy roads with irregular shapes, overpasses, and multi-lane highways.

We benchmark SAM-Road on City-scale and SpaceNet benchmarks against other methods, quantitative results are shown in Table 1. We compare several baselines, including segmentation-based (Seg-UNet, Seg-DRM, Seg-Improved, Seg-DLA, Sat2Graph) and graph-based (RoadTracer, RNGDet, RNGDet++). The TOPO metric, which evaluates local graph structure similarity, is on par with state-of-the-art, RNGDet++, despite that SAM-Road has a much simpler structure. The APLS metric of SAM-Road achieves a new state-of-the-art. APLS captures long-range topological and geometrical structure - this indicates the effectiveness of our transformer-based topology decoder and graph representation.

Such performance should largely be attributed to SAM, the powerful foundational vision model. As shown in Figure 1, the predicted masks are sharp and clear, enabling precise geometry prediction. The SAM image features are also informative vertex embeddings containing rich semantic meanings, as evident in the accurate topology predictions.

4.5. Speed and Accuracy Trade-off

SAM-Road is also highly efficient, thanks to its parallelized inference and that it doesn't require complex CPU-heavy post-processing heuristics. We measure the inference time to produce the complete graphs for the test sets of both datasets. The main results use 16×16 windows and are already $40\times$ faster than RNGDet++ on the City-scale dataset, and $10\times$ faster on the SpaceNet dataset, as shown in Table 2. As mentioned in Section 3.6, SAM-Road can trade accuracy for more speed by sparsifying the sliding windows. Table 3 shows the result such trade-off. Using fewer windows can further provide $2\times$ to $4\times$ speed-up, with a minor accuracy drop.

4.6. Ablation Studies

We conduct ablation experiments to study the effects of the key design choices on the City-scale dataset. The results are shown in Table 4.

How important is using the pre-trained SAM model? A vs B proves it is critical. We repeated the experiment with the same ViT-B architecture with only ImageNet1K and MAE pre-training [30], and the results were far worse. This is not surprising, as City-scale and SpaceNet datasets are quite small in this era, especially when using large patch sizes (E.g. 512), resembling few-shot learning. The large-scale pre-training on datasets like SA-1B used by SAM seems critical for the generalization capability. Maybe it's due to this reason that the baseline methods have to rely on smaller patches for more training examples and adopt weaker backbones with more inductive bias like CNNs.

We also studied the importance of the topology decoder's design choices.

Whether using a transformer. A vs C: we tried removing it and simply connecting a dense layer directly to the pair features. This makes the query unaware of other targets. Both geometry and topology performance drops. This is understandable: all nodes in the subgraph being asked shall be visible to the net, otherwise, there are ambiguities about whether two nodes shall connect given the definition in Section 3.4.

Whether taking the vertex offsets as input. A vs D shows a slight performance drop. Without the offset, the topology decoder no longer has a clear view of the local ge-

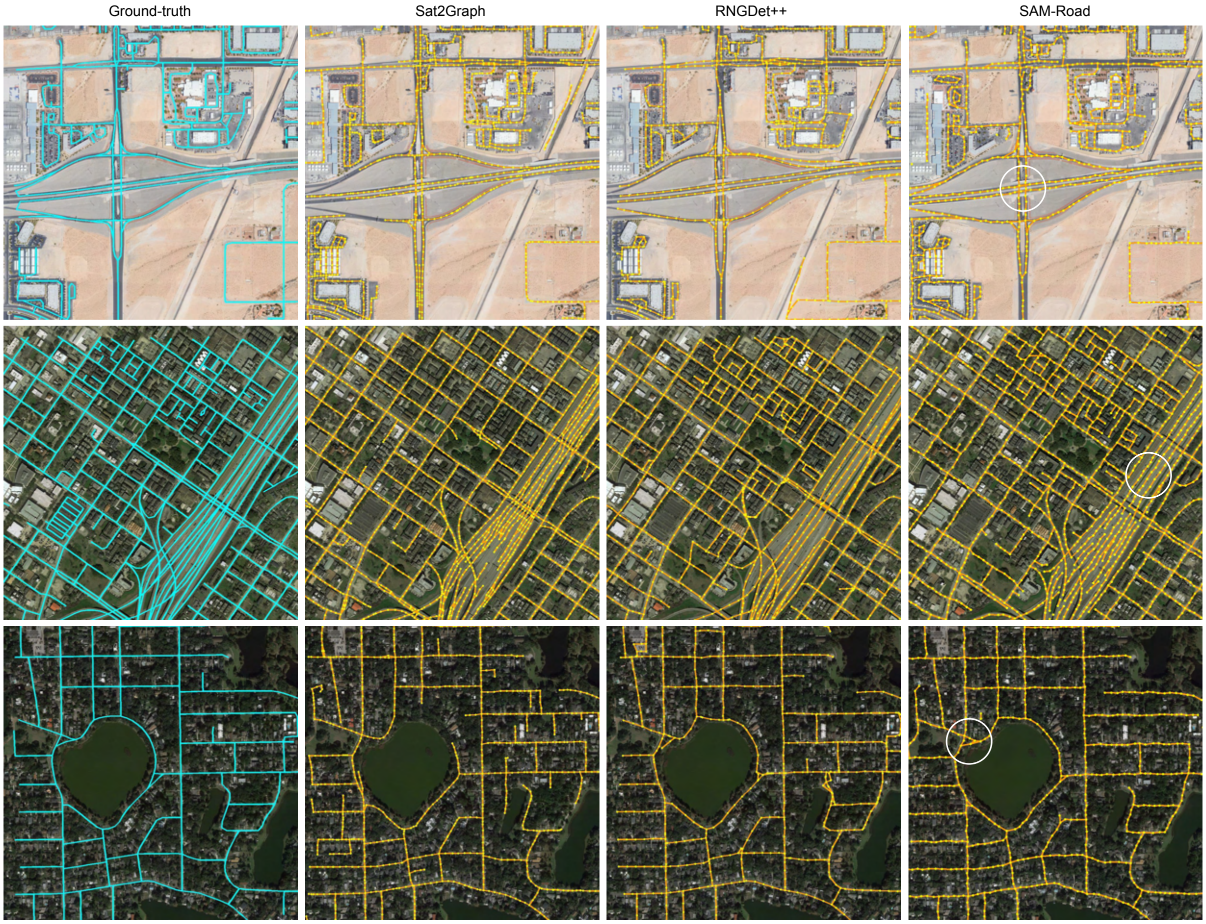


Figure 5. The visualized road network graph predictions of SAM-Road and two baseline methods. Better zoom-in and view in color. Overall, SAM-Road generates highly accurate predictions. The circles highlight especially challenging spots: in the first area, SAM-Road correctly predicts the overpass structure. In the second one, SAM-Road gives superior results for the parallel freeways. The third spot shows an irregular intersection where the two baselines fail.

	City-scale Dataset				SpaceNet Dataset			
Methods	Prec.↑	Rec.↑	F1↑	APLS↑	Prec.↑	Rec.↑	F1↑	APLS↑
Seg-UNet [46]	75.34	65.99	70.36	52.50	68.96	66.32	67.61	53.77
Seg-DRM [38]	76.54	71.25	73.80	54.32	82.79	72.56	77.34	62.26
Seg-Improved [6]	75.83	68.90	72.20	55.34	81.56	71.38	76.13	58.82
Seg-DLA [58]	75.59	72.26	73.89	57.22	78.99	69.80	74.11	56.36
RoadTracer [5]	78.00	57.44	66.16	57.29	78.61	62.45	69.90	56.03
Sat2Graph [23]	80.70	72.28	76.26	63.14	85.93	76.55	80.97	64.43
TD-Road [24]	81.94	71.63	76.43	65.74	84.81	**77.80**	81.15	65.15
RNGDet [56]	85.97	69.78	76.87	65.75	90.91	73.25	81.13	65.61
RNGDet++ [57]	85.65	**72.58**	**78.44**	67.76	91.34	75.24	**82.51**	67.73
SAM-Road	**90.47**	67.69	77.23	**68.37**	**93.03**	70.97	80.52	**71.64**

Table 1. Comparison with existing methods on different datasets. SAM-Road achieved the highest TOPO precision of 90.47% on City-scale and 93.03% on SpaceNet. It also shows the highest APLS metric of on both sets. Overall the graph accuracy is among the very top. SAM-Road leans more towards precision in TOPO metrics, this might be due to the low positive / negative example ratio in its binary classification tasks.

Method	City-scale Dataset	SpaceNet Dataset
Sat2Graph	150.6 min	69.0 min
RNGDet++	231.0 min	112.8 min
SAM-Road	**4.6 min**	**8.2 min**

Table 2. The inference time for the three methods, on both City-scale and SpaceNet datasets. Ours is within 10 minutes while the other two methods take 1-2 hours.

	City-scale Dataset			SpaceNet Dataset		
Setup	Time Cost	F1↑	APLS↑	Time Cost	F1↑	APLS↑
16 × 16	4.6 min	77.23	68.37	8.2 min	80.52	71.64
8 × 8	3.3 min	77.20	67.21	3.1 min	80.84	71.12
4 × 4	2.9 min	77.00	67.03	1.7 min	80.85	70.88

Table 3. The time cost with different stride sizes in sliding-window inference, on both datasets.

Variant	Opt	SAM	TFM	Offset	F-target	Itsc	F1↑	APLS↑
A	✓	✓	✓	✓	✓	✓	77.23	68.37
B	✓		✓	✓	✓	✓	31.79	12.39
C	✓	✓		✓	✓	✓	73.75	59.39
D	✓	✓	✓		✓	✓	77.36	66.67
E	✓	✓	✓	✓		✓	77.42	67.08
F	✓	✓	✓	✓	✓		71.94	64.62
G	✓	✓				✓	69.21	63.32
H		✓	✓	✓	✓	✓	76.05	67.95

Table 4. The SAM-Road variants compared for ablation studies. Opt: using optimized score thresholds. SAM: using pre-trained SAM. TFM: using a transformer for topology prediction. Offset: taking relative offsets in topology decoder. F-target: topology decoder takes target node feature. Itsc: predict intersection masks.

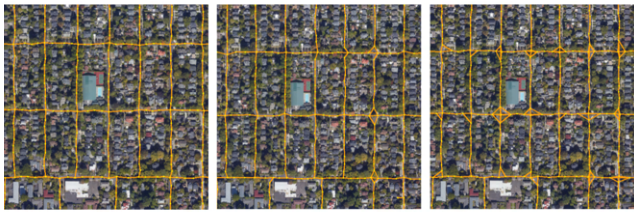


Figure 6. Left: standard SAM-Road. Middle: no intersection mask. The intersections are noticeably noisier. Right: using an A-star algorithm for topology prediction, which induces many false positive connections.

ometrical layout, which may hinder the topology reasoning and cause false-positive connections and discontinuities.

Whether taking the target vertex feature as input. A vs E shows a minor performance drop. Interestingly, not using the target node features doesn't harm performance too much. This might be because ViT-B has a sufficiently large effective field of view due to the transformer architecture, and the source feature alone contains sufficient image context in the region.

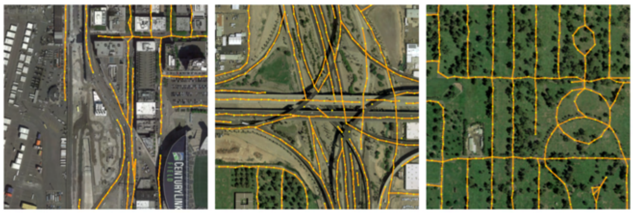


Figure 7. Some error patterns. Left: geometry decoder missed the road segment in the middle. Middle: topology decoder missed connections in a complex interchange. Right: an interesting case where SAM-road predicts the trails in a park which are not part of the label.

Whether using the learning-based topology decoder. A vs G shows that it's critical for SAM-Road's performance. Intuitively, a naive method that might achieve a similar effect is just to run a pathfinding algorithm between a pair of vertices, using the road existence map as the cost field, and see if there's a sufficiently low-cost path between the two without passing through other vertices. We implemented such a variant G using an A-star algorithm. Metrics are much worse, as qualitatively shown in Figure 6. This approach can mess up intersections, overpasses, and close parallel roads.

Whether predicting the intersection vertices. This is answered by A vs F. Predicting intersection points is important for building correct intersection structures as shown in 6. Without it, both metrics drop.

5. Limitations and Future Work

One current limitation of SAM-Road is we have not designed specific approaches to more accurately handle overpasses. There is an ambiguity for the topology decoder at the exact point where overpassing roads intersect, as the correct answer depends on which layer is being asked. This issue is minor though, as most vertices are not at these spots. Future work could improve this by predicting an overpass heatmap to suppress vertex formation at these locations.

In addition, in this work, we only used the smallest Segment Anything model, ViT-B. Larger variants may be explored as a future work, where we hope to explore parameter-efficient tuning methods, such as LoRA [25].

We are also interested in exploring the integration of other state-of-the-art foundational models, such as DINOv2 [42], PaLI [12] and GPT-4V [1] with graph learning.

6. Conclusion

We demonstrate the power of SAM [27], a foundational vision model on a graph learning task. It reaches state-of-the-art accuracy with a simple design while being much more efficient. This indicates a high-capacity model with massive pre-training can be a strong graph representation learner.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 8

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 1

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 3

[4] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020. 1

[5] Favyen Bastani, Songtao He, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Sam Madden, and David DeWitt. Roadtracer: Automatic extraction of road networks from aerial images, 2018. 2, 7

[6] Anil Batra, Suriya Singh, Guan Pang, Saikat Basu, CV Jawahar, and Manohar Paluri. Improved road connectivity by joint learning of orientation and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10385–10393, 2019. 2, 7

[7] James Biagioni and Jakob Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation research record*, 2291(1): 61–71, 2012. 5

[8] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 4

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 2

[10] Hao Chen, Zipeng Qi, and Zhenwei Shi. Remote sensing image change detection with transformers. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 1

[11] Keyan Chen, Chenyang Liu, Hao Chen, Haotian Zhang, Wenyuan Li, Zhengxia Zou, and Zhenwei Shi. Rsprompter: Learning to prompt for remote sensing instance segmentation based on visual foundation model, 2023. 2

[12] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2023. 8

[13] Guangliang Cheng, Ying Wang, Shibiao Xu, Hongzhen Wang, Shiming Xiang, and Chunhong Pan. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3322–3337, 2017. 2

[14] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2, 3

[15] Hetang Congrui, H Qin, S Liu, and J Yan. Impression network for video object detection. *arXiv preprint arXiv:1712.05896*, 2017. 5

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 3

[17] Adam Van Etten, Dave Lindenbaum, and Todd M. Bacastow. Spacenet: A remote sensing dataset and challenge series, 2019. 2, 5

[18] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. 1, 3

[19] Sarthak Garg, Helisa Dhamo, Azade Farshad, Sabrina Musatian, Nassir Navab, and Federico Tombari. Unconditional scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16362–16371, 2021. 3

[20] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive computing*, 7(4):12–18, 2008. 1

[21] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications, 2017. 2

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[23] Songtao He, Favyen Bastani, Satvat Jagwani, Mohammad Alizadeh, H. Balakrishnan, Sanjay Chawla, Mohamed Mokhtar Elshrif, Samuel Madden, and Mohammad Amin Sadeghi. Sat2graph: Road graph extraction through graph-tensor encoding. In *European Conference on Computer Vision*, 2020. 2, 5, 7

[24] Yang He, Ravi Garg, and Amber Roy Chowdhury. Td-road: Top-down road network extraction with holistic graph construction. In *ECCV 2022*, 2022. 7

[25] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 8

[26] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, Sijing Liu, Haozhe Chi, Xindi Hu, Kejuan Yue, Lei Li, Vicente Grau, Deng-Ping Fan, Fajin Dong, and

Dong Ni. Segment anything model for medical images? *Medical Image Analysis*, 92:103061, 2024. 2

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 1, 2, 3, 4, 8

[28] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022. 2, 5

[29] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks, 2018. 3

[30] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection, 2022. 3, 6

[31] Yu-Jhe Li, Xinshuo Weng, Yan Xu, and Kris M Kitani. Visio-temporal attention for multi-camera multi-target association. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9834–9844, 2021. 5

[32] Yu-Jhe Li, Yan Xu, Rawal Khirodkar, Jinhyung Park, and Kris Kitani. Multi-person 3d pose estimation from multi-view uncalibrated depth cameras. *arXiv preprint arXiv:2401.15616*, 2024. 3

[33] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction, 2023. 2

[34] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023. 2

[35] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning, 2023. 2

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 3

[37] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images, 2023. 2

[38] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3438–3446, 2017. 2, 7

[39] Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps, 2021. 2

[40] Volodymyr Mnih and Geoffrey E. Hinton. Learning to detect roads in high-resolution aerial images. In *Computer Vision – ECCV 2010*, pages 210–223, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 2

[41] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 3

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 8

[43] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6134–6144, 2021. 5

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 7

[47] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 2

[48] Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, et al. Relationformer: A unified framework for image-to-graph generation. In *European Conference on Computer Vision*, pages 422–439. Springer, 2022. 2

[49] Guanglu Song, Biao Leng, Yu Liu, Congrui Hetang, and Shaofan Cai. Region-based quality estimation network for large-scale person re-identification. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[50] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3

[52] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989. 4

[53] Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhao-Yang Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. Medical sam adapter: Adapting segment anything model for medical image segmentation. *ArXiv*, abs/2304.12620, 2023. 2

[54] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2018. 3

[55] Yan Xu and Kris Kitani. Multi-view multi-person 3d pose estimation with uncalibrated camera networks. In *BMVC*, page 132, 2022. 3

[56] Zhenhua Xu, Yuxuan Liu, Lu Gan, Yuxiang Sun, Xinyu Wu, Ming Liu, and Lujia Wang. Rngdet: Road network graph detection by transformer in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022. 2, 7

[57] Zhenhua Xu, Yuxuan Liu, Yuxiang Sun, Ming Liu, and Lujia Wang. Rngdet++: Road network graph detection by transformer with instance segmentation and multi-scale features enhancement, 2023. 1, 2, 5, 7

[58] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018. 7

[59] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint anything: Segment anything meets image inpainting, 2023. 2

[60] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. 2

[61] Kaidong Zhang and Dong Liu. Customized segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.13785*, 2023. 2

[62] Tongjie Y Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984. 2

[63] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021. 1