

RNGDet++: Road Network Graph Detection by Transformer With Instance Segmentation and Multi-Scale Features Enhancement

Zhenhua Xu¹, Graduate Student Member, IEEE, Yuxuan Liu², Graduate Student Member, IEEE, Yuxiang Sun¹, Member, IEEE, Ming Liu³, Senior Member, IEEE, and Lujia Wang¹, Member, IEEE

Abstract—The road network graph is a critical component for downstream tasks in autonomous driving, such as global route planning and navigation. In the past years, road network graphs are usually annotated by human experts manually, which is time-consuming and labor-intensive. To annotate road network graphs effectively and efficiently, automatic algorithms for road network graph detection are demanded. Most existing methods either adopt a post-processing step on semantic segmentation maps to produce road network graphs, or propose graph-based algorithms to directly predict the graphs. However, these works suffer from hard-coded algorithms and inferior performance. To enhance the previous state-of-the-art (SOTA) method RNGDet, we add an instance segmentation head to better supervise the training, and enable the network to leverage multi-scale features of the backbone. Since the new proposed approach is improved from RNGDet, we name it RNGDet++. Experimental results show that our RNGDet++ outperforms baseline methods in terms of almost all evaluation metrics on two large-scale public datasets.

Index Terms—Road network graph detection, imitation learning, autonomous driving, robotics.

I. INTRODUCTION

THE vector map of road elements, including high-definition (HD) maps and standard-definition (SD) maps, is important for autonomous vehicles. The road network graph is a

kind of SD map that records reliable position and topology information of drivable roads. It is a fundamental component for downstream tasks of autonomous driving, such as global route planning and navigation [1], [2]. Autonomous vehicles can query prior road network graphs to find an optimal road path to reach a destination, especially in complicated urban scenarios. In addition, road network graphs can be applied in navigation tasks, such as using Google maps to navigate ourselves in daily life. Usually, a road network graph consists of vertices and edges, where vertices represent key points of the road network (e.g., road ends and road intersections) and edges represent road segments. Since road network graphs can cover a large area, such as a city or even a country, manually annotating it is time-consuming and labor-intensive, which severely increases the cost and hinders the wide applications of autonomous vehicles. Therefore, methods that can automatically detect the road network graph are of great interest to the research community.

Since road network graphs only contain road-level information, the detection for them does not require images with very high resolution. So, aerial images obtained by unmanned aerial vehicles (UAVs) or satellites [3] are sufficient for this task. They are much cheaper and easier to access than the images/point clouds collected by vehicle-mounted sensors. In this letter, the road network graph is detected from large aerial images at the resolution of 1 m/pixel. There are many existing works in this area, which could be classified into three categories. The first category is based on semantic segmentation [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. These works first predict the semantic segmentation mask of road networks and then extract the graphs by post-processing. The second category of works has two stages and can directly obtain the road network graph without complicated post-processing [14], [15], [16]. These works first calculate graph vertices by predicting the vertex heatmap, and then predict graph edges by connecting obtained graph vertices. The last category treats the graph detection task as a Markov decision process (MDP) problem and proposes an iterative decision-making algorithm to detect graphs [17], [18], [19], [20]. Starting from predicted initial candidates, these works train an agent network that can detect the road network graph by iterations, which behaves in a similar way to human experts. Among them, Road Network Graph Detector (RNGDet) [19] proposes and trains a DETR-like (Detection by Transformer) [21] transformer network to track and detect the

Manuscript received 20 September 2022; accepted 28 February 2023. Date of publication 5 April 2023; date of current version 12 April 2023. This letter was recommended for publication by Associate Editor M. Garratt and Editor P. Pounds upon evaluation of the reviewers' comments. This work was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grants 2021B1515120032 and 2022A1515010116, in part by Foshan-HKUST Project under Grant FSUST20-SHCIRI06C, and in part by Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project under Grant HZQB-KCZYB-2020083. (Corresponding author: Lujia Wang.)

Zhenhua Xu and Yuxuan Liu are with the Hong Kong University of Science and Technology, Guangzhou, Guangdong 511400, China (e-mail: zxubg@connect.ust.hk; yliuhb@connect.ust.hk).

Yuxiang Sun is with the Department of Mechanical Engineering, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: yx.sun@polyu.edu.hk).

Ming Liu is with the Hong Kong University of Science and Technology, Guangzhou, Guangdong 511400, China, also with the Hong Kong University of Science and Technology, Hong Kong, SAR, China, and also with the HKUST Shenzhen, Hong Kong Collaborative Innovation Research Institute, Shenzhen 518000, China (e-mail: liu.ming.prc@gmail.com).

Lujia Wang is with the Hong Kong University of Science and Technology, Hong Kong, SAR, China, and also with the Clear Water Bay Institute of Autonomous Driving (Shenzhen), Shenzhen 518000, China (e-mail: rugga.wang@gmail.com).

Our code and supplementary materials are available at <https://tonyxuqaq.github.io/projects/RNGDetPlusPlus/>.

Digital Object Identifier 10.1109/LRA.2023.3264723

road network graph, which presents the state-of-the-art (SOTA) performance so far. However, RNGDet does not fully make use of multi-scale features extracted by the CNN backbone, which restricts the further improvement of this network.

In this letter, we propose RNGDet++, a novel approach that directly detects road networks in the graph format. Compared with RNGDet, RNGDet++ can better use the multi-scale features of the backbone and presents superior results in our task. Besides, we add an instance segmentation head into the network to better supervise the training, which enhances our robustness and performance. RNGDet++ is trained by imitation learning. We conduct comparative experiments and ablation studies on the city-scale dataset released by Sat2Graph [15], and the SpaceNet dataset [22]. Our contributions are listed as follows:

- We propose RNGDet++, a novel approach that can make full use of the multi-scale features to effectively detect road networks in the graph format.
- We add an instance segmentation head to the network, which can better supervise the training of the network and improve the performance.
- We evaluate our RNGDet++ and all the baseline methods on two large-scale public datasets. Our RNGDet++ presents superior results.
- We open-source our code and release the data at <https://tonyxuqaq.github.io/projects/RNGDetPlusPlus/>.

II. RELATED WORKS

A. Graph Detection of Simple Road Elements From Bird's-Eye View

Simple road elements refer to those with simpler topology, such as road boundaries [14], [23], [24], road curbs [25], [26] and road lanelines [27], [28], [29]. Under common circumstances, these road elements do not have complicated merges, splits, or intersections, thus the detection of the graph of these road elements is less difficult. The input data of past works is bird's-eye view (BEV) images which are either aerial images [14], [24], [25] or BEV images of the pre-built map [23], [27], [28], [30]. Most previous works utilize a decision-making network to iteratively detect the graph of target objects. Liang et al. [23] proposed a CNN-based decision network to detect the road boundary in BEV images obtained from the pre-built point cloud map. Li et al. [28] further modified the algorithm to handle simple topology changes of lanelines (e.g., split and merge) on highways. Although the aforementioned approaches could achieve satisfactory results in their specific detection tasks, they cannot be adapted to the detection task of the road network graph, since road network tends to have more complicated topology structure, such as road intersections and road overlapping (e.g., overpasses). Therefore, more powerful and robust algorithms are demanded.

B. Road Network Detection From Aerial Images

With the fast development of aerial imaging techniques, high-resolution aerial images from all over the globe could be easily accessed nowadays. Thus, most past works on road network

detection take aerial images as input [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [16], [17], [18], [19], [20]. They could be classified into three categories: (1) Segmentation-based approaches [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. This category of approaches first predict the semantic segmentation map of road networks, and then conduct post-processing algorithms (e.g., skeletonization and binarization) to extract the graph. However, they usually have inferior topology correctness, especially when road intersections or road overlappings are encountered; (2) Two-stage-graph-based approaches [15], [16]. He et al. [15] proposed a two-stage algorithm Sat2Graph to directly predict the graph of road networks without complicated hard-code post-processing. The authors first predicted the heatmap of road network graph vertices and extracted vertex coordinates by processing algorithms. Then, based on predicted graph vertices, they designed an encoding scheme to demonstrate graph edges by projecting the input aerial image into an 18-D tensor. A deep neural network is trained to predict the 18-D encoding tensor of the input image, and the graph edges could be calculated by decoding the predicted encoding tensor. Sat2Graph presents quite promising results, but it is not end-to-end trainable, which degrades its final performance. Moreover, the isomorphic encoding issue [15] also restricts Sat2Graph from having better evaluation scores; (3) Iterative-graph-based approaches [17], [18], [19], [20]. These approaches convert the road network detection task to an MDP problem, in which an agent is trained to detect the road network graph vertex by vertex iteratively. It is believed that RoadTracer proposed by Bastani et al. [17] is the first work belonging to this category of approaches. RoadTracer trains a CNN-based decision network to control an agent to explore the road network by iterations. At each step, the network predicts the moving direction of the next step, and the agent moves in the predicted direction by a fixed distance. Inspired by RoadTracer, Xu et al. [19] proposed a DETR-like network RNGDet to detect the road network graph. RNGDet achieves the SOTA performance. At each step, RNGDet directly predicts the coordinates of vertices in the next step, so that RNGDet can have an adjustable step length and handle road intersections with arbitrary numbers of incident roads. However, RNGDet only utilizes the feature of the deepest layer of the backbone, leaving multi-scale features not fully used, which prevents further improvement.

C. Detection by Transformer

Compared with CNN, transformer [31] can better handle variant-length input, and capture global relationships between patches of the input image. Transformer-based detection framework Detection by Transformer (DETR) is first proposed by Carion et al. [21]. Compared with previous detection works, DETR is more simple, effective, and end-to-end trainable. Taken as input images, DETR directly outputs a fixed-length vector encoding certain information about each candidate object. By modifying the definition of the output vector as well as the transformer network, DETR is adopted to handle various different detection tasks, such as line segment detection [32], road centerline detection [33], and road network graph detection [19].

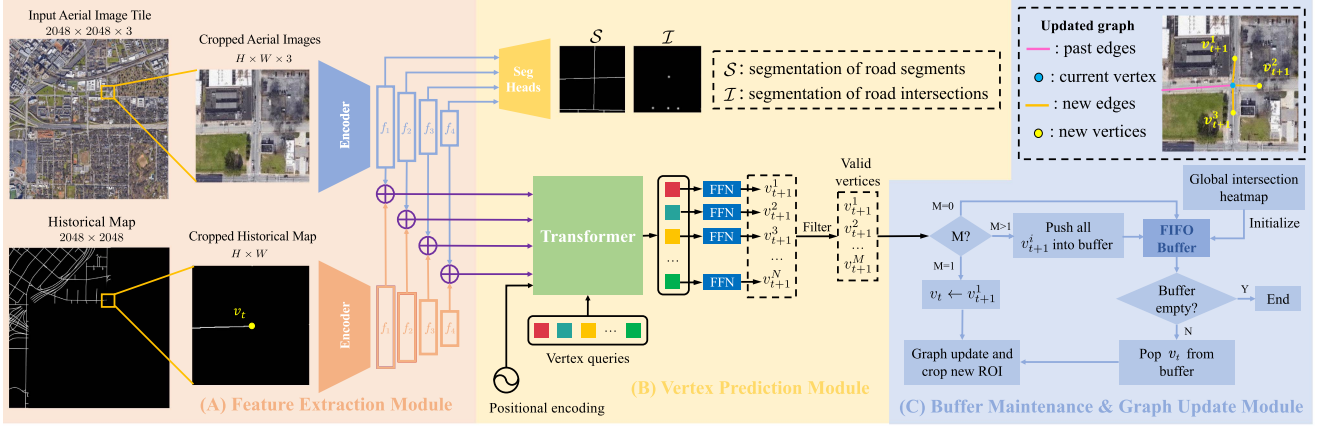


Fig. 1. System diagram of RNGDet++. In this diagram, RNGDet++ conducts single-step processing at time t . RNGDet++ mainly consists of three modules: (A) Feature extraction module. With the RGB aerial image and the binary historical map as input, this module first crops ROIs centering at v_t , and then extracts multi-scale deep features of ROIs by two backbones. (B) Vertex prediction module. Based on extracted multi-scale features, this module predicts two semantic segmentation maps S, \mathcal{I} , vertices in the next step $\{v_{t+1}^i\}_{i=1}^M$, as well as instance segmentation maps of all valid vertices $\{(S_{\mathcal{I}}^i)_{t+1}^i\}_{i=1}^M$. (C) Buffer maintenance & graph update module. After obtaining vertices in the next step, RNGDet++ updates the graph G , and controls the agent to make corresponding actions. The agent keeps repeating the above steps until the buffer is empty. When the buffer is empty, RNGDet++ stops and outputs the predicted road network graph G . This figure is best viewed in color. Please zoom in for details.

Even if most DETR-like approaches present satisfactory results for a specific task, they only utilize a single feature layer obtained by the backbone, while the multi-scale features are not fully used. In this letter, RNGDet++ is proposed to conquer this problem by using multi-scale features for both training and inference to further enhance the final performance.

III. METHODOLOGY

A. Overview

In this letter, we propose RNGDet++ to detect the graph of road networks for downstream autonomous driving applications. Compared with RNGDet, RNGDet++ makes full use of multi-scale backbone features and adds an instance segmentation head to better supervise the training phase. Suppose the road network graph is $G = (V, E)$, where V is a set of key points of the road network as vertices and E is a set of road segments as edges. Taken as input large aerial image tiles, the task of RNGDet++ is to predict the road network graph G . The system diagram of RNGDet++ is displayed in Fig. 1.

RNGDet++ controls an agent to iteratively detect the road network graph, whose current coordinates are denoted by v_t , where t is the current time stamp. Since the input aerial image tile (i.e., I) is usually very large, such as 2048×2048 or 4096×4096 , considering limited computation resources, RNGDet++ processes a 128×128 region of interest (ROI) at one time. To provide the agent with historical information, the rasterized graph detected by RNGDet++ so far is recorded as the historical map H . H is represented by a binary image whose size is the same as that of I . H is obtained by rasterizing the vector format historical graph into an image so that it can be processed by the CNN backbone together with I . Centering at v_t , the image ROI (i.e., I_R) and historical map ROI (i.e., H_R) are cropped on I and H , respectively. Taken as input I_R and H_R , RNGDet++ extracts the multi-scale deep features by two

ResNet [34] backbones. The i -th feature layer is denoted by f_i , and larger i indicates a deeper feature layer.

With a feature pyramid network (FPN) [35], RNGDet++ predicts the segmentation of road segments (i.e., S) and the segmentation of road intersection points (i.e., \mathcal{I}). The DETR-like transformer network makes full use of multi-scale backbone features, and predicts coordinates (i.e., $\{v_{t+1}^i\}_{i=1}^N$) and valid probability (i.e., $\{p_{t+1}^i\}_{i=1}^N$) of N vertices in the next step. These predicted vertices are then filtered by removing those with low valid probability p_{t+1}^i and RNGDet++ finally obtains M valid vertices in the next step.

RNGDet++ maintains a first-in-first-out (FIFO) buffer saving initial candidates, which are initial vertices to initialize the iteration of the agent. These initial candidates may come from local peaks of the global segmentation heatmap of road intersections, or from breakpoints of the agent iteration. Based on the number of valid vertices M in the next step, the agent takes different actions to update the graph. If $M = 1$, there is only one vertex v_{t+1}^1 in the next step, and the agent directly moves to v_{t+1}^1 ; if $M = 0$, the agent pops a new initial candidate from the buffer; if $M > 1$, it indicates that road intersections are met, so the agent pushes all $\{v_{t+1}^i\}_{i=1}^M$ into the buffer and pops a new initial candidate from it. After this, the agent crops new ROIs, predicts new vertices, and repeats the aforementioned steps until the buffer is empty.

B. Feature Extraction

Centering at the current coordinate of the agent (i.e., v_t), RNGDet++ crops I_R and H_R on the input aerial image and the historical map. I_R and H_R provide the agent with visual information and historical information, respectively. RNGDet++ extracts multi-scale deep features of these two ROIs by two ResNet backbones. Each ResNet backbone can obtain four layers of features denoted as $f_i, i \in (1, 2, 3, 4)$. A larger i indicates

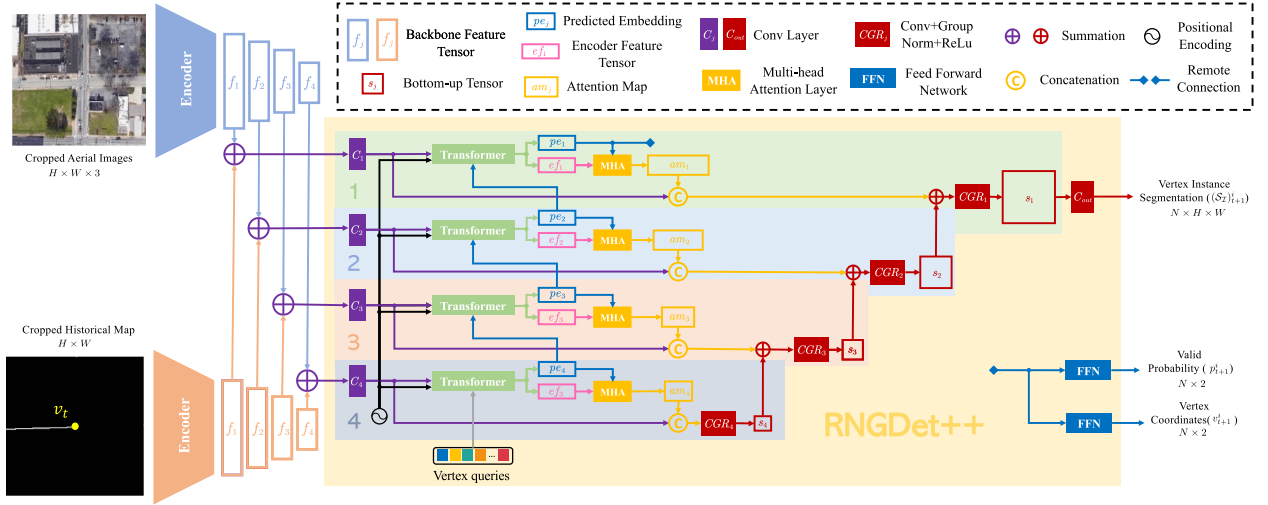


Fig. 2. Network structure of RNGDet++. RNGDet++ can make use of all four levels of features while RNGDet only utilizes the 4-th feature layer. At each layer, the transformer predicts an embedding tensor pe_i (blue box) of each input vertex query, and outputs the feature tensor extracted by the transformer encoder ef_i (pink box). pe_1 is fed to feed-forward networks (FFNs) to predict the coordinates and valid probability. A multi-head attention layer (orange rectangle) is used to predict the attention map am_i (orange box) based on pe_i and ef_i . The predicted attention map is then sent to an FPN (red elements) to predict the instance segmentation map. This figure is best viewed in color. Please zoom in for details.

a deeper feature layer. The extracted features obtained by two backbones are added for feature fusion.

C. Multi-Scale Feature Fusion

The main difference between RNGDet++ and the previous RNGDet is that RNGDet++ can make use of the multi-scale backbone features while RNGDet only utilizes a single feature layer. Suppose the ResNet backbone obtains four levels of features denoted by $\{f_i\}_{i=1}^4$, and larger i indicates a deeper level feature tensor. RNGDet only uses the deepest layer f_4 for vertex prediction while tensors at other levels are ignored, which causes information loss. Inspired by UNet [36] and FPN [35], the proposed RNGDet++ makes predictions based on features extracted at all four levels, which can better capture the deep feature of the input images to predict the vertex in the next step. Please refer to our supplementary document for a detailed comparison between RNGDet and RNGDet++.

Transformers are trained to process multi-scale features. Taken as input f_i and a vertex query, a shared transformer predicts an embedding tensor pe_i . pe_1 is sent to a feed-forward network (FFN) for the prediction of one vertex in the next step. Together with the output of transformer encoder ef_i , pe_i is used to calculate the attention map am_i . Both $\{am_i\}_{i=1}^4$ and $\{f_i\}_{i=1}^4$ are fed to an FPN segmentation head for instance segmentation of road segments ahead. The network structure of RNGDet++ is shown in Fig. 2.

D. Vertex Prediction

With the extracted multi-scale deep features as input, RNGDet++ outputs several predictions by different heads.

1) *Semantic Segmentation*: RNGDet++ predicts the semantic segmentation maps of road segments (i.e., \mathcal{S}) and road intersection points (i.e., \mathcal{I}) with an FPN segmentation head. \mathcal{S} helps the network to learn the feature of the road network, while

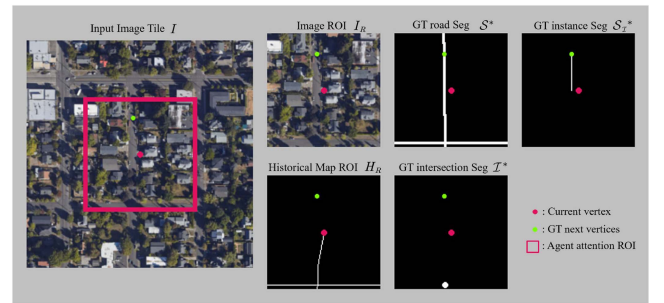


Fig. 3. Visualization of the ground-truth instance segmentation mask label (top right mask). When the agent v_t (pink point) is away from the right track, the instance segmentation head still supervises the agent to capture the correct road information, which improves the final performance.

\mathcal{I} enables the network to be better aware of road intersections, which improves the performance of RNGDet++ to detect road networks with complicated intersections. The semantic segmentation head only takes deep features of the input aerial image as input and ignores that of the historical map.

2) *Next Step Vertices*: Besides multi-scale deep features, the DETR-like transformer also takes N vertex queries as input. The vertex query is a fixed-length trainable tensor, which could be treated as a slot that the transformer can utilize to make predictions. For each vertex query, the transformer predicts the x-y coordinate of the vertex in the next step v_{t+1}^i , a valid probability p_{t+1}^i , and an instance segmentation map of the ahead road $(\mathcal{S}_T)_{t+1}^i$. If p_{t+1}^i is larger than a threshold, the predicted vertex is classified as valid and will be used to update the graph. Suppose v^* is the vertex projected from a vertex v onto the ground-truth road network. $(\mathcal{S}_T)_{t+1}^i$ is the road segment ahead of v_t that connects v_t^* and $(v_{t+1}^*)^*$, which could better supervise the training phase of the network and improve the reasoning ability of the agent. An example is visualized in Fig. 3 to show how we define instance segmentation labels.

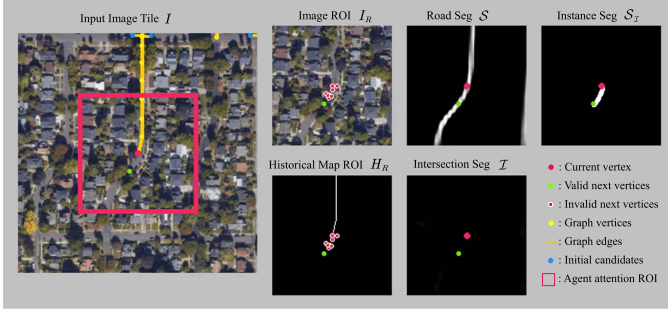


Fig. 4. Visualization demonstrations for the output of RNGDet++. With ROIs I_R and H_R as input, RNGDet++ predicts two semantic segmentation maps S and \mathcal{I} , as well as N vertices in the next step. The prediction of each vertex contains its coordinates, valid probability, and instance segmentation of the road ahead. Vertex with high valid probability is classified as valid vertex (green node), otherwise it is filtered out as invalid (red node with white margin). This figure is best viewed in color. Please zoom in for details.

Since there are N input vertex queries, the transformer predicts N vertices in the next step. After filtering those vertices with low valid probability, we finally have M valid vertices in the next step. The visualization of RNGDet++ outputs is shown in Fig. 4.

E. Buffer Maintenance & Graph Update

During the iteration of RNGDet++, we maintain a FIFO buffer to save initial candidates. An initial candidate is one initial vertex that the agent starts the iteration. At the very beginning, we predict the global segmentation heatmap of road intersections by merging \mathcal{I} of ROIs that cover the whole I , and push all local peak points of the segmentation map into the buffer as initial candidates. Then, we pop one initial candidate from the buffer, crop ROIs centering at the newly popped initial candidate and predict vertices in the next step. Based on the number of valid predicted vertices M , we take different actions to maintain the buffer and update the graph:

- $M = 0$. No road is ahead. Thus, the agent stops processing the current road instance, and pops a new initial candidate from the buffer to work on.
- $M = 1$. A single road is ahead. The agent directly moves to v_{t+1}^i ($i = 1$) and continues the iteration without operating on the buffer.
- $M > 1$. Multiple roads are met (e.g., road split or road intersections). The agent pushes all v_{t+1}^i into the buffer as new initial candidates, and pops a new vertex from the buffer to work on.

RNGDet++ keeps repeating the aforementioned steps until the buffer is empty. If the buffer is empty, RNGDet++ completes the detection task of the current input aerial image and outputs the predicted G . The working pipeline of RNGDet++ is visualized in module C of Fig. 1.

F. Training Data Sampling

The training of RNGDet++ relies on imitation learning. We create an expert by using the breadth-first search (BFS) graph traversal algorithm proposed in [19]. The expert can annotate the

road network graph vertex by vertex in the same way as human experts. For better robustness, we add even-distributed noise to expert trajectories.

Then the task of RNGDet++ is to mimic the behavior of the expert and try to learn its policy. In our experiment, considering the training efficiency, we use the behavior-cloning imitation learning algorithm [37] to collect training samples and train RNGDet++.

G. Loss Functions

For semantic segmentation maps S and \mathcal{I} , binary cross entropy loss \mathcal{L}_{seg} is utilized. To alleviate the imbalance of foreground pixels and background pixels, a larger training weight is applied for foreground pixels.

Similar to DETR [21], for vertices in the next step, RNGDet++ first matches predictions with the ground truth by conducting the Hungarian algorithm. After matching, the vertex coordinate is trained by L1 loss (i.e., \mathcal{L}_{coord}), and valid probability is trained by cross-entropy loss (i.e., \mathcal{L}_{prob}).

For each valid predicted vertex, we calculate the binary cross entropy segmentation loss (i.e., \mathcal{L}_{ins}) to train the instance segmentation head.

Finally, we have the training loss $\mathcal{L} = \mathcal{L}_{seg} + \alpha\mathcal{L}_{coord} + \beta\mathcal{L}_{prob} + \gamma\mathcal{L}_{ins}$.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Dataset

In this letter, all experiments are conducted on the city-scale dataset released in [15] and the SpaceNet dataset [22]. The city-scale dataset contains 180 RGB aerial images at 2048×2048 (pixels) captured from different cities across the world. This dataset provides the ground truth for road networks in the format of vector graphs. Following Sat2Graph [15], we split the dataset into train/valid/test with 144/9/27 tiles, respectively. The SpaceNet dataset has 2551 RGB aerial images at 400×400 (pixels). Compared with the city-scale dataset, the SpaceNet dataset focuses on smaller regions. The dataset is split into train/valid/test with 2042/127/382 images, respectively. Images of both datasets have a 1 m/pixel resolution. These two datasets are large and have aerial images collected from various scenarios, making them sufficient for training road network graph detection approaches and conducting evaluations comprehensively.

B. Baselines and Evaluation Metrics

We compare our RNGDet++ with previous SOTA approaches, including four segmentation-based approaches and three graph-based approaches.

- Segmentation-based baselines [8], [9], [15], [36]. Unet [36] is a widely used classic semantic segmentation network, and we adopt it to our task. Deep Road Mapper (DRM) [9] and ImprovedRoad [8] conduct post-processing steps to refine the road network segmentation results, which achieves better performance. Deep Layer Aggregation (DLA) [38]

TABLE I
THE QUANTITATIVE COMPARISON RESULTS

Methods	City-scale Dataset				SpaceNet Dataset			
	Prec. ↑	Rec. ↑	F1↑	APLS ↑	Prec. ↑	Rec. ↑	F1↑	APLS ↑
Seg-UNet [36]	75.34	65.99	70.36	52.50	68.96	66.32	67.61	53.77
Seg-DRM [9]	76.54	71.25	73.80	54.32	82.79	72.56	77.34	62.26
Seg-Improved [8]	75.83	68.90	72.20	55.34	81.56	71.38	76.13	58.82
Seg-DLA [38]	75.59	72.26	73.89	57.22	78.99	69.80	74.11	56.36
RoadTracer [17]	78.00	57.44	66.16	57.29	78.61	62.45	69.90	56.03
Sat2Graph [15]	80.70	72.28	76.26	63.14	85.93	76.55	80.97	64.43
RNGDet [19]	85.97	69.78	76.87	65.75	90.91	73.25	81.13	65.61
RNGDet++	85.65	72.58	78.44	67.76	91.34	75.24	82.51	67.73

The best results are highlighted in bold font. For all the metrics, larger values indicate better performance.

has a more powerful backbone, which is also utilized by Sat2Graph [15] for feature extraction.

- Graph-base baselines [15], [17], [19]. RoadTracer [17] is believed to be the first graph-based approach for the road network detection task. Sat2Graph and RNGDet are two SOTA approaches that can directly output the graph of road networks.

All the approaches are evaluated with the metrics used in [15], including TOPO [39] and average path length similarity (APLS) [22]. Within the input aerial image, TOPO first randomly samples seed vertices on the ground truth graph and the predicted one, then compares the similarity of sub-graphs that seed vertices can reach. This metric uses precision, recall, and F1 to measure the average sub-graph similarity. APLS randomly samples a vertex pair (v_1, v_2) on the ground truth graph and projects them to the predicted graph as (\hat{v}_1, \hat{v}_2) . Then APLS compares the shortest distance between (v_1, v_2) and (\hat{v}_1, \hat{v}_2) . Smaller distance difference means better graph similarity. For both metrics, larger scores indicate better performance.

C. Implementation Details

In our experiments, all ROIs are 128×128 -sized. We run the sampling algorithm to collect the training data, and finally obtain around 400 K samples to train RNGDet++. During training, the loss weights α , β and γ are set to 5, 1 and 1, respectively. Considering the road topology, we set the number of vertex queries as 10, which is sufficient to handle common road networks. RNGDet++ is trained for 50 epochs, with 10^{-4} initial learning rate. All the experiments are conducted on 4 RTX-3090 GPUs.

In our experiments, all approaches are tuned on the validation set of the corresponding dataset. We aim to maximize the TOPO-F1 score of the validation set by trying different parameter settings of the evaluated approach and then use the tuned network to infer the test set. APLS is not considered during parameter tuning.

D. Comparative Experiments

Our RNGDet++ is compared with seven baseline approaches, including four segmentation-based approaches and three graph-based baselines. The quantitative results are shown in Table I. Qualitative demonstrations on the city-scale dataset are visualized in Fig. 5.

TABLE II
THE QUANTITATIVE RESULTS FOR THE ABLATION STUDY

I	M	City-scale Dataset				SpaceNet Dataset			
		Prec. ↑	Rec. ↑	F1↑	APLS ↑	Prec. ↑	Rec. ↑	F1↑	APLS ↑
	✓	86.04	70.65	77.94	66.36	90.74	75.10	82.18	67.21
✓		85.62	71.88	78.01	66.94	91.46	75.11	82.48	67.01
✓	✓	85.65	72.58	78.44	67.76	91.34	75.24	82.51	67.73

The best results are highlighted in bold font. For all the metrics, larger values indicate better performance. We assess the instance segmentation head (I) and the multi-scale features (M).

Segmentation-based approaches first predict the pixel-level semantic segmentation map of road networks, and then conduct post-processing algorithms to extract and refine the graph of road networks. From Table I, we can see that they tend to have relatively good TOPO scores since TOPO mainly measures the performance of the sub-graph detection, which focuses on the locality. Since segmentation-based approaches directly optimize pixels, they can achieve satisfactory results on pixel-level or local topology-level metrics. However, they have a degraded APLS score mainly because their global topology is not good enough, which could be caused by incorrect detection of road intersections or overlapped overpasses. Therefore, we claim that segmentation-based approaches can detect the road network graph with satisfactory local performance, but cannot present good results on the global scale.

Different from the aforementioned segmentation-based approaches, graph-based approaches directly output and optimize the graph of road networks. Thus, they usually have better topology-level performance. Sat2Graph and RNGDet present the SOTA performance among the baselines, and present superior results not only in the TOPO metrics but also in the APLS metric. RNGDet++ is enhanced from RNGDet by utilizing multi-scale deep features, and it has the best evaluation scores in terms of both APLS and TOPO metrics. So, the superiority and effectiveness of our RNGDet++ are demonstrated and verified.

E. Ablation Studies

We conduct ablation studies to verify the rationality of the design of RNGDet++, including the instance segmentation head and the multi-layer features. The quantitative results of ablation studies are shown in Table II.

First, the instance segmentation head is removed from RNGDet++. The instance segmentation head is used to predict road segments ahead of v_t , which can better supervise the training of RNGDet++, making it better capture the spatial and topology information of the road network graph. Based on the evaluation scores, RNGDet++ without the instance segmentation head presents inferior results. So, the importance of the instance segmentation head is well demonstrated.

Then, to learn how multi-layer features affect the performance of RNGDet++, we train RNGDet++ by only utilizing the deepest feature layer (i.e., f_4). From the evaluation results, we find that RNGDet++ without utilizing multi-layer features cannot reach as good results as the original RNGDet++. Thus, the necessity of using multi-scale features is verified.



Fig. 5. Qualitative visualization on the city-scale dataset. (a) Ground truth road networks (Cyan lines). (b) Road network graph detected by Sat2Graph. (c) Road network graph detected by RNGDet. (d) Road network graph detected by RNGDet++. For (b)–(d), yellow points represent graph vertices and orange lines represent graph edges. For the visualization, it is found that RNGDet++ can output road network graphs with more accurate structure and correctness compared with previous works. This figure is best viewed in color. Please zoom in for details. (a) Ground truth. (b) Sat2Graph [15]. (c) RNGDet [19]. (d) RNGDet++.

TABLE III
THE INFERENCE TIME COST. WE REPORT THE TIME COST (HOURS) USED TO
INFER ALL TESTING IMAGES

	Sat2Graph	RNGDet	RNGDet++
City-scale Dataset	2.51h	2.68h	3.85h
SpaceNet Dataset	1.15h	1.22h	1.88h

F. Limitations

1) *Lower Efficiency*: Since at each layer, the transformer makes one prediction based on fused features, the time used for the inference of RNGDet++ is relatively longer than RNGDet. The inference time cost is reported in Table III. However, it should be noted that our task (i.e., road network graph detection) is an offline task, which is not sensitive to efficiency. Thus, considering the superior effectiveness performance of RNGDet++, we think the relatively lower efficiency is acceptable at this stage. We plan to further simplify the network and replace the network modules with lighter structures.

2) *Too Complicated Intersection and Overpass*: Even though RNGDet++ outperforms previous works and presents the best results, it still cannot handle too complicated road intersections or overpasses very well. Moreover, since RNGDet++ is trained by imitation learning, incorrect predictions in these scenarios may affect the subsequent behaviors of the agent. We plan to further optimize the training strategy, and use more powerful backbones to improve the reasoning ability of RNGDet++.

V. CONCLUSION

In this letter, we enhanced the previous road network graph detection approach RNGDet by adding an instance segmentation head, and enabling it to leverage multi-scale features of the backbone. The new novel approach is named RNGDet++. The instance segmentation head could better supervise the network training and improve the reasoning ability of RNGDet++. Besides, RNGDet++ could utilize all layers of features obtained by the backbone, which enables the network to capture multi-scale

information of the road network, so that RNGDet++ presents superior results. RNGDet++ is trained by the behavior-cloning imitation learning algorithm. RNGDet++ and all the baselines are fairly evaluated on two large publicly available datasets. Compared with all baselines, RNGDet++ achieves better evaluation scores, not only at the pixel level but also at the topology level. In the future, we plan to further simplify the network structure for efficiency, and apply more powerful backbones to better capture the features of road networks.

REFERENCES

- [1] T. Liu et al., "The role of the hercules autonomous vehicle during the COVID-19 pandemic: An autonomous logistic vehicle for contactless goods transportation," *IEEE Robot. Automat. Mag.*, vol. 28, no. 1, pp. 48–58, Mar. 2021.
- [2] H. Christensen et al., "Autonomous vehicles for micro-mobility," *Auton. Intell. Syst.*, vol. 1, no. 1, pp. 1–35, 2021.
- [3] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 210–223.
- [4] X. Hu, Y. Li, J. Shan, J. Zhang, and Y. Zhang, "Road centerline extraction in complex urban scenes from LiDAR data based on multiple features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7448–7456, Nov. 2014.
- [5] W. Shi, Z. Miao, Q. Wang, and H. Zhang, "Spectral-spatial classification and shape features for urban road centerline extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 4, pp. 788–792, Apr. 2013.
- [6] C. Unsalan and B. Sirmacek, "Road network detection using probabilistic and graph theoretical methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4441–4453, Nov. 2012.
- [7] G. Cheng, F. Zhu, S. Xiang, and C. Pan, "Road centerline extraction via semisupervised segmentation and multidirection nonmaximum suppression," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 545–549, Apr. 2016.
- [8] A. Batra, S. Singh, G. Pang, S. Basu, C. Jawahar, and M. Paluri, "Improved road connectivity by joint learning of orientation and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10385–10393.
- [9] G. Mátyus, W. Luo, and R. Urtasun, "DeepRoadMapper: Extracting road topology from aerial images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3438–3446.
- [10] A. V. Etten, "City-scale road extraction from satellite imagery V2: Road speeds and travel times," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1786–1795.
- [11] W. G. C. Bandara, J. M. J. Valanarasu, and V. M. Patel, "Spin road mapper: Extracting roads from aerial images via spatial and interaction space graph reasoning for autonomous driving," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 343–350.
- [12] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017.
- [13] G. Zhou, W. Chen, Q. Gui, X. Li, and L. Wang, "Split depth-wise separable graph-convolution network for road extraction in complex environments from high-resolution remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5614115.
- [14] Z. Xu et al., "csBoundary: City-scale road-boundary detection in aerial images for high-definition maps," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5063–5070, Apr. 2022.
- [15] S. He et al., "Sat2Graph: Road graph extraction through graph-tensor encoding," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 51–67.
- [16] G. Bahl, M. Bahri, and F. Lafarge, "Single-shot end-to-end road graph extraction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1403–1412.
- [17] F. Bastani et al., "Roadtracer: Automatic extraction of road networks from aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4720–4728.
- [18] Y.-Q. Tan, S.-H. Gao, X.-Y. Li, M.-M. Cheng, and B. Ren, "Vecroad: Point-based iterative graph exploration for road graphs extraction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8910–8918.
- [19] Z. Xu et al., "RNGDet: Road network graph detection by transformer in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4707612.
- [20] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1715–1724.
- [21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
- [22] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "SpaceNet: A remote sensing dataset and challenge series," 2018, *arXiv:1807.01232*.
- [23] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, "Convolutional recurrent network for road boundary extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9512–9521.
- [24] Z. Xu, Y. Sun, and M. Liu, "Topo-boundary: A benchmark dataset on topological road-boundary detection using aerial images for autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7248–7255, Oct. 2021.
- [25] Z. Xu, Y. Sun, and M. Liu, "iCurb: Imitation learning-based detection of road curbs using aerial images for autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1097–1104, Apr. 2021.
- [26] Z. Xu, Y. Sun, L. Wang, and M. Liu, "CP-loss: Connectivity-preserving loss for road curb detection in autonomous driving with aerial images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1117–1123.
- [27] N. Homayounfar, W.-C. Ma, S. Kowshika Lakshmikanth, and R. Urtasun, "Hierarchical recurrent attention networks for structured online maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3417–3426.
- [28] N. Homayounfar et al., "DAGMapper: Learning to map by discovering lane topology," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2911–2920.
- [29] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "HdmapNet: An online HD map construction and evaluation framework," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 4628–4634.
- [30] Z. Xu, Y. Liu, Y. Sun, M. Liu, and L. Wang, "CenterLineDet: Road lane centerline graph detection with vehicle-mounted sensors by transformer for high-definition map creation," 2022, *arXiv:2209.07734*.
- [31] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
- [32] Y. Xu, W. Xu, D. Cheung, and Z. Tu, "Line segment detection using transformers without edges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4257–4266.
- [33] Y. B. Can, A. Liniger, D. P. Paudel, and L. Van Gool, "Structured bird's-eye-view traffic scene understanding from onboard images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15661–15670.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
- [36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.
- [37] T. Osa et al., "An algorithmic perspective on imitation learning," *Found. Trends Robot.*, vol. 7, no. 1/2, pp. 1–179, 2018.
- [38] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.
- [39] J. Biagioni and J. Eriksson, "Inferring road maps from global positioning system traces: Survey and comparative evaluation," *Transp. Res. Rec.*, vol. 2291, no. 1, pp. 61–71, 2012.