TD-Road: Top-Down Road Network Extraction with Holistic Graph Construction

Yang He Ravi Garg Amber Roy Chowdhury

Amazon Last Mile
{yanhea,ravigarg,amberch}@amazon.com

Abstract. Graph-based approaches have been becoming increasingly popular in road network extraction, in addition to segmentation-based methods. Road networks are represented as graph structures, being able to explicitly define the topology structures and avoid the ambiguity of segmentation masks, such as between a real junction area and multiple separate roads in different heights. In contrast to the bottom-up graph-based approaches, which rely on orientation information, we propose a novel top-down approach to generate road network graphs with a holistic model, namely TD-Road. We decompose road extraction as two subtasks: key point prediction and connectedness prediction. We directly apply graph structures (i.e., locations of node and connections between them) as training supervisions for neural networks and generate road graph outputs in inference, instead of learning some intermediate properties of a graph structure (e.g., orientations or distances for the next move). Our network integrates a relation inference module with key point prediction, to capture connections between neighboring points and outputs the final road graphs with no post-processing steps required. Extensive experiments are conducted on challenging datasets, including *City-Scale* and *SpaceNet* to show the effectiveness and simplicity of our method, that the proposed method achieves remarkable results compared with previous state-of-the-art methods.

Keywords: Road Network Extraction; Relation Inference; End-to-end Appoach; Remote Sensing

1 Introduction

Road network extraction from satellite imagery is a fundamental component for automatically constructing rich and accurate maps, and enabling further route planning and navigation applications. High quality maps require several good properties, including road connectivity, precise localization on junctions and multiple interactive roads, and large coverage of the physical world. To resolve the above challenges, a large variety of methods have been proposed, which are typically categorized into segmentation-based [28,2] and graph-based methods [1,23,10]. While segmentation based methods are good at modeling contextual dependencies, segmentation masks are vague in representing complex structures [10] and require various post-processing heuristics to convert road

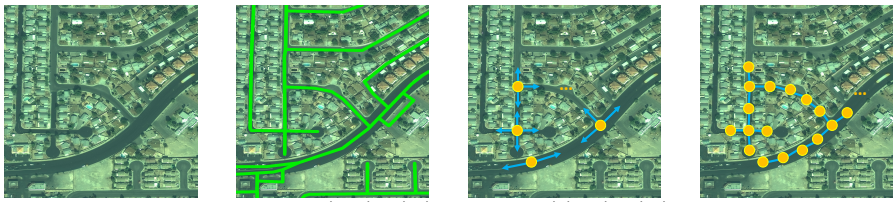

Fig. 1. Supervision signals for various road extraction methods. Different from prior work, we directly leverage graph structures to supervise the training of our network, and then our network produces graphs in a straightforward manner during inference.

masks into road networks. In this work, we focus on the second category and propose a novel top-down approach for road graph construction.

Previous graph-based methods make use of the orientation clue to construct a road graph iteratively [1,23,3] or simultaneously [10]. In each location of a road, they estimate the orientation to explore and move forward, and add the next location to the graph, which are bottom-up approaches and gradually extend a road graph. In the end, road graph extraction is completed when no orientation can be found from all the locations of the current graph. Furthermore, a recent work [10] improves the iterative graph construction scheme by encoding key point locations and orientations to extend as the outputs of neural networks. Similar to segmentation, it performs graph construction using a dense prediction network, integrating more context dependencies and avoiding expensive iterative scanning of satellite images. However, this method still relies on orientations to establish edges for graph construction.

In spite of the success of previous methods in building road graphs, we question if orientation-based methods are the best way to generate road graphs from satellite images? This question is from the observation that orientation prediction in these methods is quantized and can cause imperfection in geometries and node localization in the resultant road graph. Besides, orientation is not a direct matching, hence a post-processing is indispensable to convert the intermediate results into final graphs, which might introduce further errors and cause mismatch between different locations. In this paper, we propose a simple alternative approach to generate road graphs, where we aim to learn the connectedness between different points and output graphs directly. The proposed method of predicting the connectedness not only helps us to emit road graph structures, but also allows our network to train using graph supervision (i.e., location of nodes and connected edges between them), which is completely different to other approaches, as shown in Fig. 1.

Relation reasoning has attracted much attention in learning the relation between multiple instances, achieving broad applications such as answering complicated questions from images [19], learning non-maximum suppression in object detection [12]. In our work, we introduce a relation reasoning module into road network extraction, and apply it to learn the connectedness between two locations from the key point prediction component of our network. Finally, the whole

network can produce a set of points on a road as well as their connections with a holistic scheme.

This paper introduces a novel holistic graph construction method using neural networks. We highlight the key novelties of our work below:

- We propose a new road network extraction framework, TD-Road, which regards road network extraction as key point prediction and connectedness prediction subtasks. Our model learns to generate a road graph end-to-end using graph supervision, as compared to intermediate information used to generate graphs in previous graph-based methods. Our method is extremely simple that it outputs a graph structure without any further post-processing.
- We introduce relation inference into road network extraction, which shows appealing capability to model the relations between different locations of a map. In our work, we leverage a relation reasoning module to learn the connectedness between two points on a road. Further, we propose a neighbor-guided relation reasoning module to boost our framework.
- Extensive experiments and comparisons show the effectiveness and advantages of our new scheme for road extraction. We demonstrate that the proposed method localizes crucial graph nodes precisely and performs better in dealing with ambiguous regions.

2 Related Work

2.1 Road Network Extraction

Graph-based approaches As an early study of graph-based model, RoadTracer [1] formulates road network extraction as a graph growing procedure, which starts from initial seeds and extracts roads iteratively by predicting the orientations to extend the graph. Further, VecRoad [23] aims to overcome the imprecise graph exploration with a fixed step size, and boost the iterative graph construction by using a flexible step size and segmentation cues. Besides, These graph-growing approaches also suffer from inefficiencies since they need to feed-forward an image patch to CNN to obtain the orientations in each step. To overcome the low-efficiency of single prediction, Sat2Graph [10] represents the graph as tensor coding, which encodes the key points and orientations at the same time with a dense prediction network, which directly produces the prediction over a large area. Besides, Sat2Graph shows inspiring results in handling ambiguous regions, such as multiple parallel roads and challenging highways with bridge interactions at different layers. Further, graph convolution networks have been exploited in similar tasks to learn the attribute for each road segments [13] or locations [11].

Segmentation-based approaches In addition to graph-based methods, other approaches consider the road extraction as a segmentation task to output road masks. These methods can model global context, but find it hard to represent complex structures well using a simple road mask. Many previous works focus on designing network architectures for road segmentation [24,6,20,16,9,28], which

can capture long and narrow shapes, as well as large variances of layout structures. In particular, DlinkNet [28] is a successful architecture designed for road segmentation, which leverages skip connection over different stages and dilated convolutions in the bottleneck. Deep layer aggregation has also been shown to be an effective architecture for this task [10]. To improve connectivity, joint orientation learning [2] has been combined with segmentation, and orientations are demonstrated as a crucial clue for road segmentation.

Observing the prior work, orientation is important and exploited by most methods, either graph-based or segmentation-based approaches. However, very few works directly models the connections between adjacent locations. Besides, graph-based methods still require a post-processing step to convert the outputs of networks into graphs, based on the orientations or moves for the next points. Different from others [1,23,10], our network can output a graph structure in a holistic way and allows optimization using graph structures (i.e., nodes and edges) directly.

2.2 Relation Network

Relation network is a neural network component, aiming to infer the relationship among multiple instances, which could be objects [19], images [7,21] etc. For example, it is proposed to answer complex questions regarding multiple objects in visual question answering [19]. By incorporating a relation reasoning module, it is able to answer more complicated questions, for example "What is the color of the object behind the blue cube?". Relation network is also utilized to learn the similarity scores between images for few-shot learning [22], which shows strong capability to learn complex manifolds. Besides, relation inference has been employed in object detection for learning non-maximum suppression [12], which picks the best bounding box from all the region proposals and allows for training end-to-end, integrated with other components of object detection.

Inspired by relation reasoning in prior work, we design a new model for road network extraction by representing roads as graphs. Differently, we learn pixelwise and pairwise relations from a dense feature map, for constructing graph structures from image inputs. As far as we know, there is no previous work on road network extraction modeling the relation between different locations of a map and output a road network graph directly from a network.

3 A Holistic Model for Direct Graph Construction

3.1 Overview

We depict our model for holistic road graph construction from satellite images. Our approach leverages graphs to supervise the training of our network. We decompose the graph construction into two sub-tasks: **key point prediction** and **connectedness prediction** between key points. Fig. 2 draws the overview of our model, and we discuss the details of the two components as follows. Particularly, we highlight that our graph generation is trained end-to-end from input

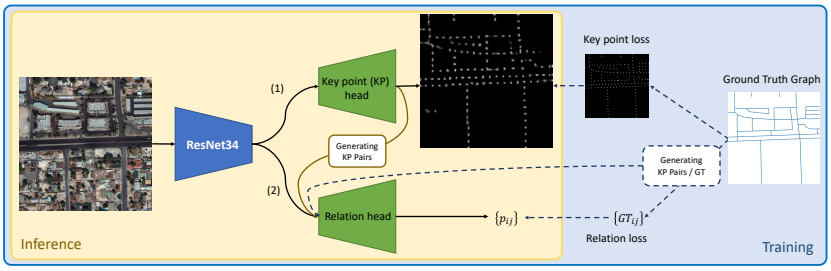

Fig. 2. Overview of TD-Road model. The model outputs a set of key points as well as their connections as the extracted road graph. Our network has two branches, consisting of key point prediction and relation reasoning over key points. For each key point, the relation reasoning module computes the probabilities between each key point and its neighboring pairs, which takes a dense feature map and neighbor information of key points as inputs. In training, the ground truth graph provides the pairs regarding the connectedness between key points. In inference, our network uses the predicted key points and corresponding generated neighbors to perform relation reasoning, and then at the end outputs a road graph.

satellite imagery to generated graphs. To build our model, ResNet is applied as our backbone to extract dense features and different heads for individual subtasks are designed. First, the key point prediction head outputs a 2-d heatmap indicating the location of key points of roads, which can be trained using a segmentation-like loss and will be discussed with more details in Sec 3.3. Second, we generate key point pairs if two points are close enough, and classify each pair as connected or not based on our relation reasoning module, as followed in Sec 3.2. In model training, we create positive/negative pairs using the key point locations as well as their connectedness from ground truth graphs. In inference, we first run the key point branch and apply the outputs from key point prediction as the inputs of relation branch to predict the connectedness between predicted key points.

3.2 Relation Reasoning for Graph Edges

As the main contribution of this work, we first introduce our relation reasoning module for connecting key points and constructing a graph. To build a relation reasoning module for establishing edges between graph nodes, we create a separate decoder with a shared encoder to key point prediction, as illustrated in Fig. 2. The decoder first outputs a dense feature, and feature extraction of key points over the dense features is processed for classifying the key point pair. Let $F_R \in \mathbf{R}^{C\times H_0\times W_0} = \mathbf{\Theta}(\mathbf{E}(I))$ be the output of a decoder $\mathbf{\Theta}$ followed with a feature extractor $\mathbf{E}$ performed on the input image I. $\{(x_i, y_i)\}_{i=1}^{K}$ are the predicted key points, and then our goal is to know if there is an edge between points (x_i, y_i) and (x_j, y_j).

Naive relation reasoning: To reach the above goal, we check the edges between a candidate point and all of its neighbors, relying on a distance threshold, as shown in Fig. 3. During training, graph annotation provides us the correct edges,

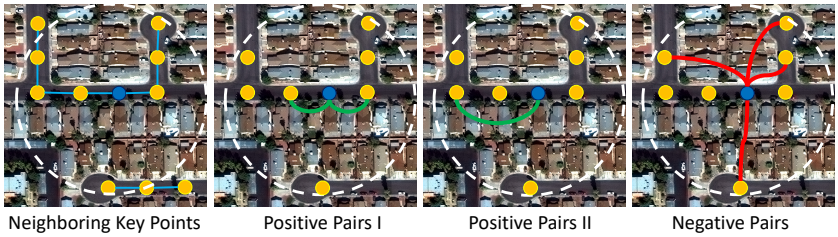

Fig. 3. Training examples for relation reasoning module. We perform binary classification for each pair between a candidate (blue) and its neighbors (orange). Positive pairs are regarded as the key point pairs, which are connected through a straight line. Otherwise, it will be regarded as negative pairs.

therefore, we know the connectedness of each key point pair. And thus we can easily build a binary classifier to predict the connectedness between two key points. Specifically, we consider all the neighbors of a candidate point as potential connections, and assign labels to them. Clearly, a direct edge between two points is a valid positive example, as shown in the second plot of Fig. 3. Further, even two points are not directly connected, but they are routable in the same direction through the intermediate connecting point, we also consider this is a positive pair, as shown in the third plot of Fig. 3. Last, the negative pairs are the points not traversed, or not in a same direction, as shown in the last plot of Fig. 3.

To learn the relation between two points, we extract pixelwise features from F_R at the locations (x_i, y_i) and (x_j, y_j). Since the feature map might be in a low resolution format, we apply bilinear interpolation operations to extract features, where the operations are differentiable, resulting in features $F_i^* = \text{Interpolation}(F_R, x_i, y_i)$ and $F_j^* = \text{Interpolation}(F_R, x_j, y_j)$. And then, we apply a linear projection on the concatenation of F_i^* and F_j^* (i.e., $\text{Cat}(F_i^*, F_j^*)$)to produce the connectedness score. To augment the training and boost the inference, we predict the probability by switching the order of two points. Finally, the probability to establish the edge can be calculated as

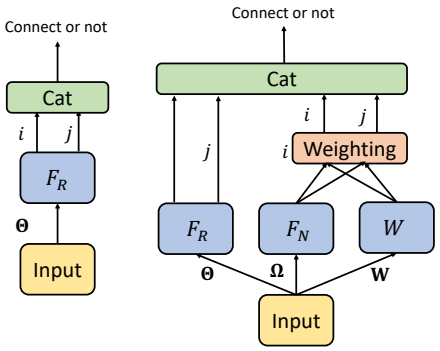

Fig. 4. Illustration of the naive relation reasoning module (left) and the neighbor-enhanced relation module (right).

$$P = (\text{Linear}(2C, 1)(\text{Cat}(F_i^*, F_j^*)) + \text{Linear}(2C, 1)(\text{Cat}(F_j^*, F_i^*)))/2, \quad (1)$$

and the model parameters $\mathbf{\Theta}$, $\mathbf{E}$ and the above linear classifier can be learned by using binary cross entropy loss, which is denoted as $\mathcal{L}_R$ for the relation module.

Neighbor-enhanced relation reasoning: Observing the relation module mentioned above, we realize that feature extractions for the key points are separately accomplished. Hence, it fails to model context information in relation reasoning. Accordingly, it is interesting to know if contexts, in particular other key points, help relation reasoning and determining connectedness. To aggregate the context information over key points, we learn additional projection $\mathbf{\Omega}$ and weighting function $\mathbf{W}$, outputting features of C and 1 channels, respectively. Therefore, we project the feature map from the encoder and obtain $F_N = \mathbf{\Omega}(\mathbf{E}(I))$ and $W = \mathbf{W}(\mathbf{E}(I))$, where $F_N \in \mathbf{R}^{C \times H_0 \times W_0}$ and $W \in \mathbf{R}^{H_0 \times W_0}$. The context of neighbors for a key point at (x_i, y_i) can be weighted by F_N and W, which is formulated as

$$\hat{F}_i = \frac{\sum_{(x_j,y_j) \in S_{x_i,y_i}} F_N(:, x_j, y_j) * W(x_j, y_j)}{\sum_{(x_j,y_j) \in S_{x_i,y_i}} W(x_j, y_j)}, \tag{2}$$

where S_{x_i,y_i} are the locations which are the neighbors of the point at (x_i, y_i). Similar to Eq.(1), the final prediction can be computed by concatenating them: $P = (\text{Linear}(4C, 1)(\text{Cat}(F_i^*, F_j^*, \hat{F}_i, \hat{F}_j)) + \text{Linear}(4C, 1)(\text{Cat}(F_j^*, F_i^*, \hat{F}_j, \hat{F}_i)))/2$.

3.3 Key Point Prediction for Graph Nodes

To predict the locations of key points, we create a dense prediction model based on an encoder-decoder architecture and carefully design a suitable loss based on road mask and key points. In the following, we present our decoder used in this work, which is inspired by deep layer aggregation architectures [27] and All-MLP decoder [26].

DLA-MLP decoder as task heads: Key point prediction is a task focusing on local information and semantics. Key points usually appear in the junction areas or abrupt blending locations and interpolated locations between others. Therefore, propagating very wide context is likely unreasonable for this task, therefore, we present a simple decoder based on Multi-Layer Perceptron (MLP) which performs prediction over individual pixels. Since the encoder already compresses long range context information into feature maps, we do not aggregate further contextual semantics in the decoder. Further, recent state-of-the-art segmentation model [26] shows that competitive results with MLP decoder can be achieved when encoder is able to capture representative features, which further motivates us to design our decoder with MLP to infer individual road key points.

Deep layer aggregation (DLA) [27] is a network scheme, which summarizes CNN features across layers and augments a base model to model what and where better. With fewer parameters, it shows better accuracy with iterative and hierarchical feature fusion, compared to skip connections with concatenation. In this work, we present our MLP-based decoder with DLA to effectively aggregate low-level localizable features as well as high-level contextual semantics.

By combining MLP decoder [26] and DLA architectures [27], our DLA-MLP decoder fuses the hierarchical features gradually with MLP components. Let $\{F_{i,0}\}_{i=1}^{N}$ be the multiscale hierarchical features from the encoder, where we

have N stages in total. For each feature map $F_{i,0}$, it has a size of $C_i \times H_i \times W_i$. Formally, we formulate the steps of our DLA-MLP decoder with C embedding dimensions, and the decoder outputs a feature map with size of $C \times H_0 \times W_0$ as follows.

$$F_{i,j}^{'} = \text{Linear}(C_{i,j}, C)(F_{i,j}), \tag{3}$$

$$F_{i,j}^{'} = \text{Upsampling}(H_{i-1}, W_{i-1})(F_{i,j}^{'}), \tag{4}$$

$$F_{i,j+1} = \text{Linear}(C_{i-1,j} + C, C)(\text{Cat}(F_{i-1,j}, F_{i,j}^{'})), \tag{5}$$

where $j = [0, ..., N-2]$, $i = [N-j-1, ..., N]$, and finally a feature map $F_{N,N-1} \in \mathbf{R}^{C \times H_0 \times W_0}$ is obtained as the fused representation. And then we apply a 1×1 convolution to project the representations to a heatmap of one channel for key point prediction. Furthermore, we apply our DLA-MLP decoder as a component ($\mathbf{\Omega}$ in Sec 3.2) of our relation head to aggregate features similarly, in addition to key point prediction. In this work, we apply the same embedding dimension for key point prediction and relation reasoning module with $C = 128$.

3.4 Loss Function

As a binary classification problem, key point prediction can be also trained using binary cross entropy (BCE) loss. However, it needs to address the extreme imbalanced training example for key points. When vanilla BCE loss is applied, we observe poor results in that very few key points are detected and the performance of our overall model is restricted.

The reasons are two aspects. First, the background non-road pixels are a lot, therefore, the training is likely dominated by the negative pixels. Second, the key point ground truths are conceptually contradictory, in that many road pixels are labeled negative, even though they share similar visual patterns with the positively labeled pixels, such as colors, shapes, contexts, etc.

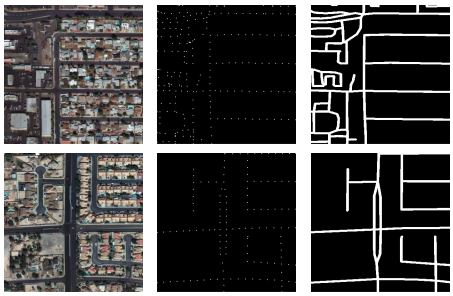

Fig. 5. Ground truth examples in computing loss function. We leverage road mask to help more effective key point prediction.

Since our approach relies on the key points to construct a graph, we need to deal with the challenges mentioned above. In this work, we weight the different pixels for key point predictions, based on BCE loss. We consider there are three types of pixels: key point pixels; road pixels but not key points; background pixels. Specifically, we set different loss weights for those pixels. Formally, given the ground truth label y_t for key points and prediction p_t over location t, our road mask driven BCE loss for key point prediction can be computed by

$$\mathcal{L}_{KP} = -\sum_{t} w_t \cdot (y_t \cdot \log(p_t) + (1 - y_t) \cdot \log(1 - p_t)), \tag{6}$$

where $w_t \in \{w_{kp}, w_r, w_b\}$ are the loss weights for individual pixels depending on their types, where Fig. 5 shows an example of ground truth used for loss function computation.

To further exploit the road information, we learn an additional decoder to predict binary road masks and orientations of neighboring key points, which are widely adopted in prior work for road extraction. In our work, we highlight that our approach does not reply to segmentation masks or orientations to construct our road graph, but we still observe this information is useful and compatible with our framework. Therefore, we regard this additional task as a regularization term $\mathcal{L}_{reg}$ to train our full model.

Finally, the loss for overall system is in a combination between key point, relation and regularization heads:

$$\mathcal{L} = \mathcal{L}_{KP} + \mathcal{L}_R + \mathcal{L}_{reg}. \tag{7}$$

Our method is embarrassingly simple in generating road graphs from satellite images using neural networks, that we only need two hyperparameters to filter incorrect key points and connections out. It is allowed to tune the thresholds to achieve higher precision or recall.

4 Experiments and Results

4.1 Datasets & Evaluation Settings

***City-Scale*:** This dataset [10] provides satellite imagery focusing on 20 US cities. The dataset covers downtown areas with complex structures such as the overlaid highways and bridges. Each image has a resolution of 2048×2048, and every pixel corresponds to 1 meter in the real world. Therefore, this is a challenging scenario, where multiple parallel roads or complicated structures appear in the images. In addition, the dataset also provides the key points used to train a graph-based model. We follow previous work [10] to use the same key points and connections to learn our model. Finally, we use 144 images to train our model, and evaluate different approaches on 27 examples.

***SpaceNet*:** This dataset [25] contains 2780 satellite images. We follow the data split of [10] and experimental setup to resize the images to 1 meter per pixel. This split contains 2040, 358, 382 examples for training, validation and testing. The dataset provides the ground truths in the format of line strings, indicating the center of roads. To train our method, we first convert the line strings into the graph format with key points (nodes) and connections (edges). In particular, we linearly interpolate the key points with 20 pixels in case the original connecting key points in the dataset are very far from each other.

Evaluation metrics: First, we adopt the widely used APLS (i.e., average path length similarity), introduced by [25], to evaluate the performance of each model.

The APLS metric computes the shortest path length between road network pairs, which captures the overall performance of extracted roads. Second, we report the TOPO scores [4] of precision (P), recall (R) and F1-score of extracted roads in topology. We utilize the implmentation of TOPO scores from [10] to compare our approach with others, which is very strict to penalize the details of extracted roads.

Comparison methods: We compare our approach with previous popular methods and recent state-of-the-art models including graph-based model RoadTracer [1], Sat2Graph [10] as well as segmentation-based model UNet [18], DlinkNet [28], DeepRoadTracer [17] and orientation-learning based method [2].

4.2 Implementation Details

In this work, all the models are trained with 1024×1024 cropped image patches for both datasets, and we use whole images to infer road networks. During training, we crop image patches at random locations and then apply flipping operations vertically and horizontally at 50% probability. Further, we rotate image/graph pairs in the range of $(-15^{\circ}, 15^{\circ})$ and a photometric distortion [5] is leveraged including brightness, contrast, saturation changes of an image. Specifically, we implement our models using PyTorch and mmseg package [8], and train the model with 8 Tesla V100 GPUs. AdamW [15] is applied to optimize the training, where initial learning rate is 1×10^{-3}, and betas is $(0.9, 0.999)$, and weight decay is 0.01. Further, we set the learning rate of key point prediction and relation inference heads as $10\times$ of the backbone. We use 1×10^{-6} to warm up the training with 200 iterations. Following many previous work [1,28,2], all of our models are implemented with ResNet-34 as the backbone, and the model weights are initialized using the ImageNet-1k pretrained representation. 20k and 60k training iterations are applied for *City-Scale* and *SpaceNet* datasets, and learning rates are adjusted using cosine-based schedulers. We set $\{w_{kp}, w_r, w_b\}$ as $\{200, 20, 1\}$ for key point prediction loss in both datasets.

4.3 Comparison Results

In Table 1 and Table 2, comparison results with other state-of-the-art methods are listed. From the tables, we observe the proposed graph-based road extraction method performs comparable with previous state-of-the-art method, Sat2Graph [10], whereas Sat2Graph adopt a stronger backbone DLA [27]. In Table 1, our best model obtains 65.74 APLS, which outperforms all the other methods. Further, our best model also obtains higher precision than other methods, which indicates the effectiveness of relation reasoning which builds connections properly. Regarding recall, the proposed model is slightly worse than DLA [27] and Sat2Graph [10], but clearly better than all the other segmentation-based methods and graph-growing-based method RoadTracer [1]. In particular, we notice that the segmentation-based method using DLA already provides 73.89 recall, which indicates the capability of DLA to detect more roads than ResNet-34 or similar architectures. Similar to *City-Scale*, we achieve more favorable results in *SpaceNet* when the same backbone is applied compared with other methods.

Table 1. Comparison results on *City-Scale* dataset.

Method	Backbone	Type	Topo			APLS
			P	R	F-1	
UNet [18]	CNN	Seg.	78.00	57.44	66.16	57.29
DeepRoadMapper [17]	ResNet-50		75.34	65.99	70.36	52.50
Orientation [2]	ResNet-34		75.83	68.90	72.20	55.34
DLinkNet [28]	ResNet-34		78.63	48.07	57.42	54.08
DLA [27,10]	DLA		75.59	72.26	73.89	57.22
RoadTracer [1]	CNN	Graph	78.00	57.44	66.16	57.29
Sat2Graph [10]	DLA		80.70	**72.28**	76.26	63.14
Ours (Naive)	ResNet-34		77.82	68.44	72.54	62.17
Ours (Neighbors)			**81.94**	71.63	**76.27**	**65.74**

We also highlight that our full model achieves competing results with previous state-of-the-art model Sat2Graph [10] with a DLA backbone.

Further, we also compare our models using different relation reasoning modules with other approaches. By using naive relation reasoning mentioned in Sec 3.2, our method already outperforms other segmentation-based methods in APLS and recall of TOPO evaluation. Obviously, we can see the effectiveness of incorporating neighbor information into each key points, that all the evaluation metrics are consistently improved by using neighbors-based relation reasoning in both datasets.

In addition to quantitative comparison, we also visualize the extracted roads from different approaches in Fig. 6. For the graph-based methods, we first convert the results of graph formats into binary road masks. From this plot, we would like to highlight several points. First, we clearly observe the advantages of our method over DLinkNet, which suffers from the connectivity issues in some areas with rich vegetation or low-contrast appearances, as shown in the 1^{-st}, 2^{-nd}, 5^{-th}, 6^{-th} rows of the plot. Second, comparing our method with Sat2Graph, we can see our method handles junctions better, because our method directly considers two key points are supposed to connect or not. However, Sat2Graph relies on the orientations between different key points, which are not accurate. And the heuristic post-processing might also introduce errors. In contrast, we link different key points by learning a binary classifier, which is a straightforward solution. Besides, our neighbors-based relation reasoning module can help to aggregate useful context information in constructing graph edges, and achieves more precise results than our naive reasoning.

4.4 Ablation Studies and Analysis

How important is the loss function for key point prediction? Key point prediction plays an important role, which provides inputs to the relation reason module. Hence, we show the results of using different loss function for key point prediction. To avoid the negative impact of imbalanced pixels, focal loss [14] is

Table 2. Comparison results on *SpaceNet* dataset.

Method	Backbone	Type	Topo			APLS
			P	R	F-1	
UNet [18]	CNN	Seg.	68.96	66.32	67.61	53.77
DeepRoadMapper [17]	ResNet-50		82.79	72.56	77.34	62.26
Orientation [2]	ResNet-34		81.56	71.38	76.13	58.82
DLinkNet [28]	ResNet-34		**88.42**	60.06	68.80	56.93
DLA [27,10]	DLA		78.99	69.80	74.11	56.36
RoadTracer [1]	CNN	Graph	78.61	62.45	69.60	56.03
Sat2Graph [10]	DLA		85.93	76.55	80.97	64.43
Ours (Naive)	ResNet-34		82.45	73.54	77.74	61.91
Ours (Neighbors)			84.81	**77.80**	**81.15**	**65.15**

Table 3. Comparison results on *City-Scale* dataset in association with different loss functions for key point prediction.

Loss	Topo			APLS
	P	R	F-1	
BCE	75.71	68.81	71.89	60.32
Focal [14]	80.82	62.16	69.79	47.95
Ours	77.82	68.44	72.54	62.17

another option. Therefore, we compare our road mask weighting strategy with standard BCE loss and focal loss. For BCE loss, we set loss weight for positive and negative examples are 100 and 2. For focal loss, we set the loss weights as 100 and 5, we set γ in focal loss as 1. In Table 3, we list the comparison results on *City-Scale* dataset, which is trained using naive relation module. From this table, it is apparent that mask-driven loss is beneficial to reach higher performance, owing to the more accurate key point localization. Even though focal loss is widely used in handling imbalanced data distribution, we cannot observe a successful application in our case. We can see focal loss achieves 80.82 precision, but performs not so well in road coverages and capturing road structures, that recall and APLS are significantly worse than other losses.

Further, we show an example of predicted key points from standard BCE loss and our version in Fig. 7. From this figure, we can see standard BCE and our version capture similar structures of roads, and both loss functions help isolate key points successfully, which are not very close to other points. However, our version can distinguish the key points better when many of them fall into a small region. as highlighted by the red bounding box in Fig. 7.

Is the model sensitive to the thresholds for determining graph nodes and edges? We perform road extraction by using different thresholds for key point prediction and connectedness prediction, varying from 0.3 to 0.5 with step 0.1. Fig. 8 shows the results by fixing each threshold and changing another. From this figure, we can see the model is less sensitive to the connectedness prediction.

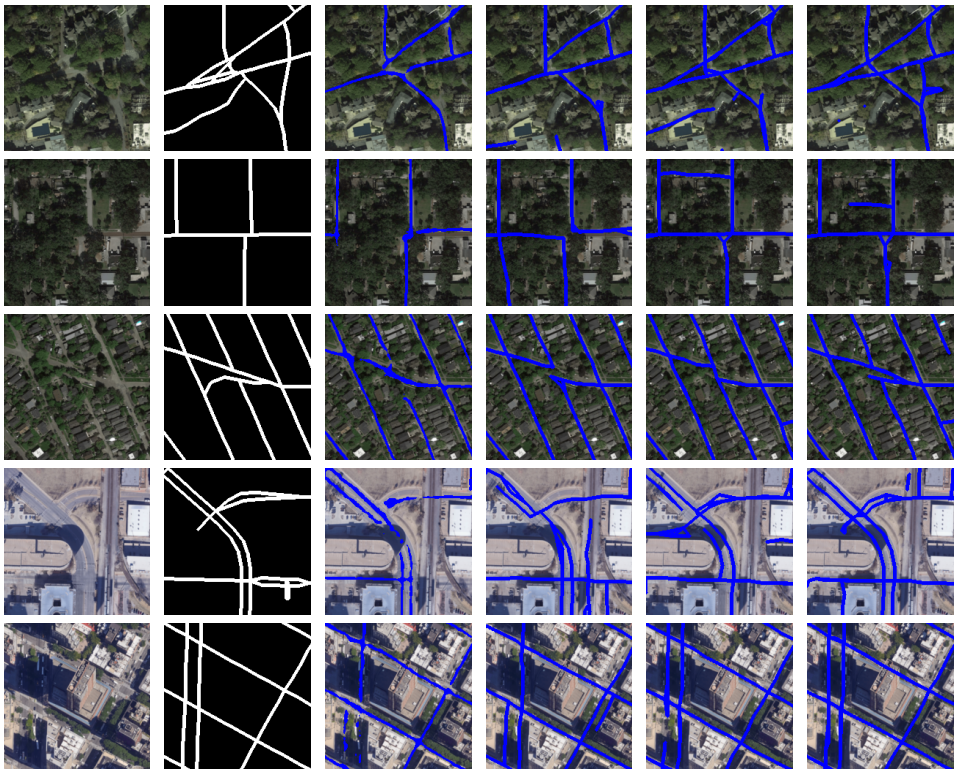

Fig. 6. Qualitative results on the extracted roads in *City-Scale* dataset. We compare our models with naive and neighbors-based relation reasoning modules with DLinkNet [28] and Sat2Graph [10].

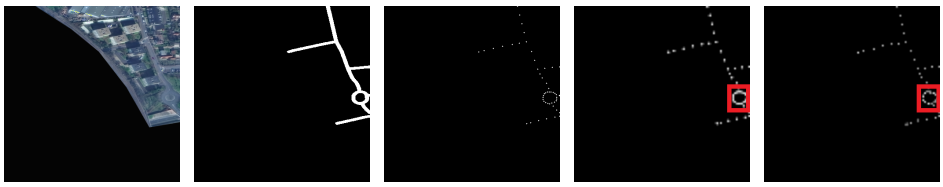

Fig. 7. Example of key point prediction using different losses. (a) Image. (b) Mask for the loss. (c) Key point GT. (d) Prediction w/o mask-based loss. (e) Prediction with our mask-driven loss.

In other words, it gives us similar predictions once a relation reasoning module is learned. In contrast, the key point prediction is crucial in providing an initial point set for the relation module, and drastically affects the performance, which

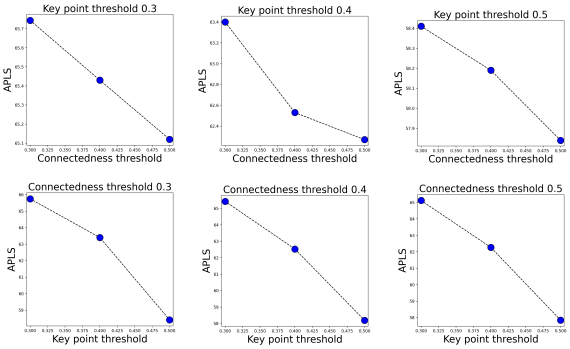

Fig. 8. Results of applying different thresholds on *City-Scale*.

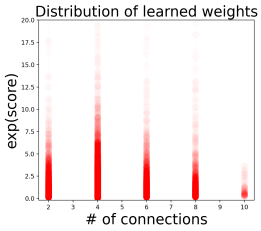

Fig. 9. Scatter plot of learned weights in neighbors-based relation reasoning module.

sheds the light for future directions, that we need to improve the key point prediction with a stronger encoder and specific decoder for key points.

What does the neighbors-based relation learn? In our neighbor-enhanced relation reasoning module, we learn to weight individual key points. To understand the learned relation, we show the statistics about the key points on *City-Scale*. Fig. 9 shows the scatter plot, where the x-axis is the number of connections for key points. For example, a point with 4 connections is a junction and the one with 2 connections is a common key point. From this plot, we can see the weights for 4 connections are larger than others, therefore, the information from junctions could be propagated to other locations and affect the relation reasoning more than other points. In contrast, the naive relation reasoning cannot leverage junction information to determine connectedness, and obtain less accurate results.

5 Conclusions

In this paper, we present TD-Road, a simple-yet-novel graph-based method for road network extraction. Different from most previous work, we directly learn and emit graph structures with neural networks, instead of producing intermediate representations such as orientations, next moves, etc. Our method is extremely simple, in that we regard the graph generation as key point prediction and connectedness learning problems. By integrating a pixel-level relation module into a dense prediction network, our approach is able to produce graph structures in a holistic way. We also present an effective relation reasoning module with neighbors for each detected key point, and the overall model achieves more favorable results than other methods using the same network backbone.

Acknowledgments

The authors sincerely thank Dr. Songtao He for providing the ground truth of SpaceNet3 dataset used in [10].

References

1. Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., DeWitt, D.: Roadtracer: Automatic extraction of road networks from aerial images. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4720–4728 (2018). https://doi.org/10.1109/CVPR.2018.00496
2. Batra, A., Singh, S., Pang, G., Basu, S., Jawahar, C., Paluri, M.: Improved road connectivity by joint learning of orientation and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
3. Belli, D., Kipf, T.: Image-conditioned graph generation for road network extraction. arXiv preprint arXiv:1910.14388 (2019)
4. Biagioni, J., Eriksson, J.: Inferring road maps from global positioning system traces: Survey and comparative evaluation. Transportation research record **2291** (2012)
5. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
6. Chaurasia, A., Culurciello, E.: Linknet: Exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP). pp. 1–4. IEEE (2017)
7. Chen, N., Zhou, Q.Y., Prasanna, V.: Understanding web images by object relation network. In: Proceedings of the 21st international conference on World Wide Web. pp. 291–300 (2012)
8. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation` (2020)
9. Ding, L., Bruzzone, L.: Diresnet: Direction-aware residual network for road extraction in vhr remote sensing images. IEEE Transactions on Geoscience and Remote Sensing (2020)
10. He, S., Bastani, F., Jagwani, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Elshrif, M.M., Madden, S., Sadeghi, M.A.: Sat2graph: Road graph extraction through graph-tensor encoding. In: European Conference on Computer Vision. pp. 51–67 (2020)
11. He, S., Bastani, F., Jagwani, S., Park, E., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., Sadeghi, M.A.: Roadtagger: Robust road attribute inference with graph neural networks. In: AAAI (2020)
12. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: CVPR (2017)
13. Jepsen, T.S., Jensen, C.S., Nielsen, T.D.: Graph convolutional networks for road networks. In: ACM SIGSPATIAL (2019)
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
15. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
16. Lu, X., Zhong, Y., Zhao, J.: Multi-scale enhanced deep network for road detection. In: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium (2019)
17. Máttyus, G., Luo, W., Urtasun, R.: Deeproadmapper: Extracting road topology from aerial images. In: Proceedings of the IEEE international conference on computer vision (2017)

18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241 (2015)
19. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. Advances in neural information processing systems (2017)
20. Shamsolmoali, P., Zareapoor, M., Zhou, H., Wang, R., Yang, J.: Road segmentation for remote sensing images using adversarial spatial pyramid networks. IEEE Transactions on Geoscience and Remote Sensing (2020)
21. Sun, C., Shrivastava, A., Vondrick, C., Murphy, K., Sukthankar, R., Schmid, C.: Actor-centric relation network. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 318–334 (2018)
22. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR (2018)
23. Tan, Y.Q., Gao, S.H., Li, X.Y., Cheng, M.M., Ren, B.: Vecroad: Point-based iterative graph exploration for road graphs extraction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
24. Tao, C., Qi, J., Li, Y., Wang, H., Li, H.: Spatial information inference net: Road extraction using road-specific contextual information. ISPRS Journal of Photogrammetry and Remote Sensing **158**, 155–166 (2019)
25. Van Etten, A., Lindenbaum, D., Bacastow, T.M.: Spacenet: A remote sensing dataset and challenge series. arXiv preprint arXiv:1807.01232 (2018)
26. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34** (2021)
27. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: CVPR (2018)
28. Zhou, L., Zhang, C., Wu, M.: D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2018)