Translating Images to Road Network: A Non-Autoregressive Sequence-to-Sequence Approach

Jiachen Lu* Fudan University

Renyuan Peng⋆ Fudan University

Xinyue Cai Huawei Noah's Ark Lab

Hang Xu Huawei Noah's Ark Lab

Hongyang Li Shanghai AI Lab

Feng Wen Huawei Noah's Ark Lab

Wei Zhang Huawei Noah's Ark Lab

Li Zhang† Fudan University

1. RoadNet Sequence & Semi-Autoregressive RoadNet Sequence

Details of sequence construction The discretization of v_x, v_y is simply truncating the integer part. Integer representation of v_c is `Ancestor`: 0, `Lineal`: 1, `Offshoot`: 2, `Clone`: 3. Discretizing e_{px} and e_{py} can be challenging since the Bezier control points may exceed the Bird's Eye View (BEV) range, and their values may become negative. As a solution, we discretize e_{px} and e_{py} by applying the `int` function to $(e_{px} + 10)$ and $(e_{py} + 10)$, respectively, to avoid negative values. Figure 3 shows a example of both RoadNet Sequence and Semi-Autoregressive RoadNet Sequence.

2. Input and target sequence construction

Sequence embedding Each vertex-edge pair is represented by 6 integers. To prevent embedding conflicts between the 6 integers, we divide them into separate ranges which is shown in Table 1. As a default, we set the embedding size to 576, which is sufficient to accommodate all the integer ranges.

Synthetic noise objects technique The input sequence of RoadNet Sequence starts with a `start` token and the target sequence ends with an `EOS` token. The `EOS` token makes the model know where the sequence terminates, but the experiments have shown that it tends to cause the model to stop predicting early without getting the complete sequence. Inspired by [2], we use a similar sequence augmentation technique to alleviate the problem called the *synthetic noise objects technique* [2]. The technique composes of *sequence augmentation* and *sequence noise padding*. The sequence augmentation adds noise to the position of landmarks and the coefficient of centerlines in input sequence. Whereas, sequence noise padding is a padding technique. For input sequences, we generate synthetic noise vertices and append them at the end of the real vertices sequence. Each noise vertex includes random locations(v_x, v_y), category(v_c), index of parent(v_d) and Bezier curve coefficient(e_{px}, e_{py}). As for the target sequence, the `EOS` token is added to the end of the real vertices sequence. We set the target category(v_c) of each noise vertex to a specific noise class(different from any of the ground-truth labels), and the remaining components(v_x, v_y, v_d, e_{px}, e_{py}) of the noise vertex to the "n/a" class, whose loss is not calculated in the back-propagation.

Item	Range
v_x, v_y	$0 \sim 199$
v_c	$200 \sim 249$
v_d	$250 \sim 349$
e_{px}, e_{py}	$350 \sim 569$
`noise category`	570
`EOS`	571
`Start`	572
`n/a`	573

Table 1. Embedding range of different integers.

However, we only use sequence noise padding as sequence augmentation has been shown to cause a decrease in performance. The introduced modifications of the synthetic noise objects technique are illustrated in Figure 1

The padding rules of Semi-Autoregressive RoadNet Sequence are much the same as auto-regressive RoadNet Sequence. As mentioned in the main submission, we pad the 2-dimensional Semi-Autoregressive RoadNet Sequence to $[[y_{1,1}, y_{1,2}, \cdots, y_{1,L}], \cdots, [y_{M,1}, y_{M,2}, \cdots, y_{M,L}]]$, where L is the maximum length of each sub-sequence and M is the number of sub-sequences. The valid sub-sequences be-

[*] Equal contribution

[†] Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University.

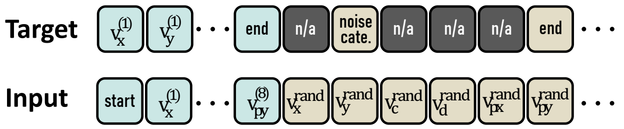


Figure 1. An illustration of synthetic noise objects technique [2] on RoadNet Sequence. Loss weight for `n/a` tokens are set to zero. `Noise cate.` stands for noise category.

BEV Aug	Sequence Aug	Sequence Noise	L-F	R-F
✓	✗	✗	58.6	64.3
✗	✗	✓	57.5	62.7
✓	✗	✓	60.2	66.0
✓	✓	✓	59.1	65.2

Table 2. Ablation studies on BEV augmentation and synthetic noise objects [2] (including sequence augmentation and sequence noise padding). NAR-RNTR with VoVNetV2 [3] pretrained on extra data is applied. The row with gray color is our final choice.

Embedding size	class weight	L-F	R-F
576	1.0	60.1	65.5
576	0.5	60.1	**66.1**
576	0.1	60.2	66.0
576	0.2	**60.2**	66.0
1000	0.2	60.1	65.8
2000	0.2	59.7	65.5

Table 3. Ablation studies on sequence embedding size and class weight. NAR-RNTR with VoVNetV2 [3] pretrained on extra data is applied. The row with gray color is our final choice.

gin with a key-point. For each valid sub-sequence, we follow the same padding rules of RoadNet Sequence, except there isn't a `start` token in an input sub-sequence because of the Key-point Prompt. We set the other sub-sequences to the "n/a" class making the loss of these sub-sequences without a key-point not calculated.

Thresholds for Reachability Precision-Recall are chosen from $[0.5, 1.0, 1.5, 2.0, 2.5]m$.

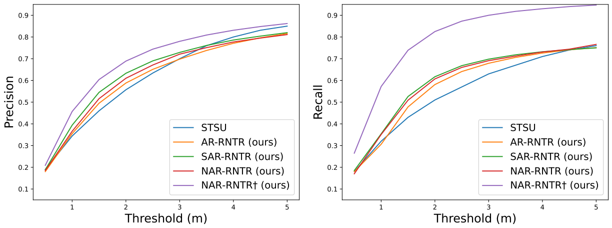


Figure 2. Mean Precision/Recall v.s. thresholds. Data of STSU [1] is recorded from Figure 7 of [1]. "†" use VoVNetV2 [3] pretrained on extra data as backbone. Thresholds are from $[0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0]m$.

3. Additional experiments

3.1. Precision-Recall curve

In addition to our overall advantage in mean Precision-Recall (as presented in Table 1 of the main submission), Figure 2 displays the precision/recall versus thresholds curve. Our models outperform others in terms of precision and recall for smaller thresholds, highlighting our accuracy advantage.

3.2. Ablation studies

Non-unique Sorting We show the difference between the random ordering strategy and an ordering based on coordinates in Table

BEV augmentation The first column of Table 2 shows that the BEV augmentation provides a significant 2.7/3.3 improvement on both Landmark and Reachability scores.

Synthetic noise objects The second column of Table 2 shows that the sequence augmentation of Synthetic noise objects technique [2], however, leads to a drop in performance. Whereas, the third column shows that the sequence noise padding 1.6/1.7 improved on both Landmark and Reachability scores. But the sequence noise padding is less effective than BEV augmentation.

Class weight We exam the class weight for MLE loss, *i.e.*, w for

$$\max \sum_{i=1}^{L} w_i \log P(\hat{y}_i | y_{<i}, \mathcal{F}), \tag{1}$$

$$\max \sum_{i=1}^{M} \sum_{j=1}^{L} w_j P(y_{i,j} \mid \hat{y}, \mathcal{F}, \mathcal{V}_{kp}), \tag{2}$$

Due to the high frequency of `Lineal` for v_c and the default index for v_d, we assign a lower weight to these categories. Although the second column of Table 3 does not indicate a clear relationship between class weight and performance, using a lower weight for the loss results in more stable performance.

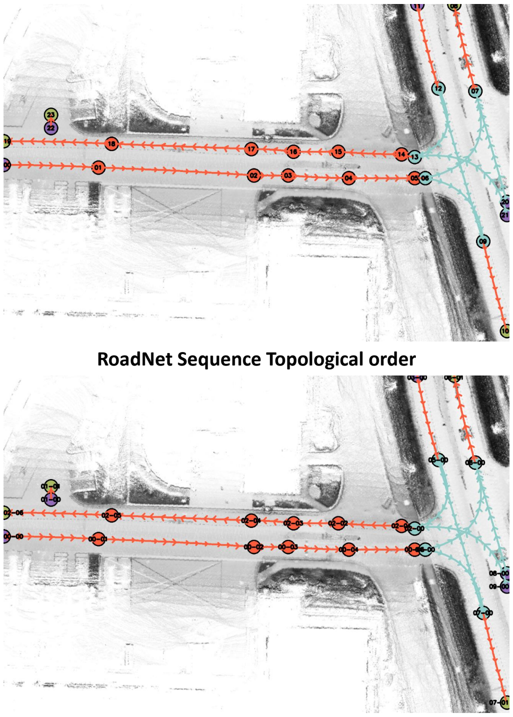
Embedding size If we extend the embedding size from 576 to 1000 or 2000, useless embeddings clearly harm the performance.

References

[1] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *CVPR*, 2021. 2

[2] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In *ICLR*, 2021. 1, 2

[3] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020. 2



RoadNet Sequence Topological order

Semi-Autoregressive RoadNet Sequence Topological order

RoadNet Sequence:

0, 61, 0, 0, 0, 0, 36, 62, 1, 0, 32, 107, 95, 65, 1, 0, 80, 109, 108, 65, 1, 0, 115, 111, 131, 66, 1, 0, 134, 112, 156, 66, 1, 0, 157, 112, 160, 66, 1, 0, 172, 112, 179, 33, 1, 0, 202, 106, 171, 0, 1, 0, 188, 62, 182, 90, 2, 7, 194, 115, 191, 124, 1, 0, 200, 153, 157, 0, 0, 0, 0, 0, 165, 32, 1, 0, 174, 62, 165, 32, 3, 10, 187, 107, 156, 58, 1, 0, 189, 98, 151, 57, 1, 0, 168, 104, 127, 56, 1, 0, 153, 103, 110, 56, 1, 0, 132, 102, 94, 55, 1, 0, 116, 101, 41, 53, 1, 0, 81, 100, 0, 52, 1, 0, 34, 98, 191, 75, 0, 0, 0, 0, 191, 75, 3, 8, 199, 100, 191, 80, 0, 0, 0, 0, 191, 80, 3, 14, 199, 100, 18, 47, 0, 0, 0, 0, 18, 42, 1, 0, 32, 91

Semi-Autoregressive RoadNet Sequence:

Key-point (0, 61): 36, 62, 1, 0, 32, 107, 95, 65, 1, 0, 80, 109, 108, 65, 1, 0, 115, 111, 131, 66, 1, 0, 134, 112, 156, 66, 1, 0, 157, 112, 156, 66, 3, 5, 172, 112

Key-point (18, 47): 18, 42, 1, 0, 32, 91

Key-point (156, 58): 151, 57, 1, 0, 168, 104, 127, 56, 1, 0, 153, 103, 110, 56, 1, 0, 132, 102, 94, 55, 1, 0, 116, 101, 41, 53, 1, 0, 81, 100, 0, 52, 1, 0, 34, 98

Key-point (157, 0): 157, 0, 3, 6, 174, 62

Key-point (160, 66): 160, 66, 3, 7, 202, 106, 160, 66, 3, 8, 194, 115

Key-point (165, 32): 165, 32, 3, 3, 189, 98, 165, 32, 3, 8, 187, 107

Key-point (179, 33): 171, 0, 1, 0, 188, 62

Key-point (182, 90): 191, 124, 1, 0, 200, 153

Key-point (191, 75): 191, 75, 3, 7, 199, 100

Key-point (191, 80): 191, 80, 3, 3, 199, 100

Figure 3. *Left* shows topological order of RoadNet Sequence and Semi-Autoregressive RoadNet Sequence. *Right* shows original RoadNet Sequence and Semi-Autoregressive RoadNet Sequence without input/target processing.