Supplementary Material

In this supplementary material, we provide additional details which we could not include in the main paper due to space limitations, including more experimental analysis and visualization details that help us develop further insights to the proposed ***BeMapNet***. We discuss:

- Additional details of architecture design.
- More experimental results for map construction.
- The statistical analysis of benchmark extension.
- Qualitative visualization results of our approach.

A. Additional Details of Architecture Design

A.1. The Motivation of IPM-PE

According to the principle of perspective geometry, a pair of corresponding points in the camera space and *BEV* space is theoretically the projection result from the same point in the unified world coordinate system. Based on this prior, on the one hand, we perform IPM to map multi-view camera coordinate grids into the world coordinate system, on the other hand, the *BEV* feature grid is also transformed into the same world space through a scale coefficient. This process is illustrated in detail in Fig.1. Compared with the conventional positional encoding method that establishes the correspondence within each single-view, the proposed IPM-PE models the position relationship of ***multi-view*** and ***multi-space*** simultaneously, which is more conducive to perspective transformation. Table 1 shows the effectiveness of the proposed IPM-PE module.

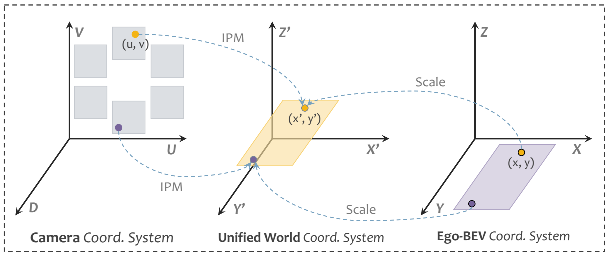

Figure 1. Illustration of the motivation of IPM-PE.

PE Type	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
Learned PE	40.2	32.2	33.8	35.4
IPM-PE	**46.4**	**39.2**	**37.8**	**41.1**

Table 1. The impact of different number of *BEV* queries.

A.2. Alignment Method of IPM-PE

We illustrate four align methods with common *FC* layer in Fig.2, namely camera-only, *BEV*-only, *exclusive* camera-*BEV*, and *shared* camera-*BEV*. Table 2 summarizes the experimental results in detail. The comparison of row 1 and $2 \sim 4$ indicates that the single or exclusive alignment instead causes performance decline. We conjecture that none of these three align methods share any information between the two branches, and the additional *FC* instead arouse positional embedding from IPM-PE to lose the original accurate geometric prior. Interestingly, with adopting the setting of shared camera-*BEV*, the model gains performance improvement with 2.6 AP. We argue that this is due to these shared parameters bridge the two groups of positional embeddings (*i.e.* f_c^{pe} and f_b^{pe}) with mutual alignment, which neutralizes the unreasonable assumptions in IPM to a certain extent.

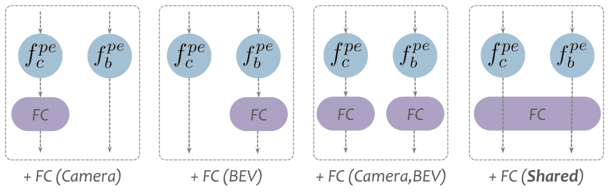

Figure 2. Illustration of different align methods of IPM-PE.

Align Type	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
IPM-PE	46.4	39.2	37.8	41.1
+ FC (Camera)	42.1	34.1	36.3	37.5
+ FC (*BEV*)	46.9	38.6	35.9	40.5
+ FC (Camera,*BEV*)	38.6	33.3	33.9	35.3
+ FC (Shared)	**49.1**	**42.2**	**39.9**	**43.7**

Table 2. The impact of different align methods of IPM-PE.

A.3. Coordinate Regression Head Design

Inspired by the popular dynamic conv [2], we propose the Split Coordinate Regression Head in the main paper and illustrate its detailed framework in the Fig.3 (left). In fact, there is a more direct and common point regression method, which directly predicts the coordinate value through conducting a *FC* projection on the splitted Bézier descriptor, as shown in the Fig.3 (right). We compared the performance of these two approaches in Table 3, which demonstrates that the proposed dynamic way is more advantageous.

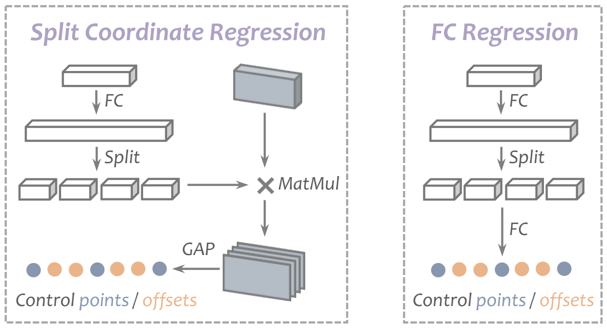

Figure 3. Different design of coordinate regression head.

Shape	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
FC Reg.	48.7	38.9	38.7	42.1
Split Coord. Reg.	**49.1**	**42.2**	**39.9**	**43.7**

Table 3. The impact of different methods of regression head.

B. More Experimental Results

B.1. The running time and model size.

The Table 4 below provides more detailed performance comparison between *HDMapNet* and ***BeMapNet***, including inference speed and parameter quantity. Note all numbers are obtained on one RTX 2080Ti GPU with $bs = 1$. Under the same backbone setting of EfficientNet-B0 [3], our approach achieves better results with $0.4\times$ fewer parameters ($69.8 \rightarrow 27.6$) and $6.9\times$ higher speed ($0.7 \rightarrow 4.8$). Note that the inference time is the total time of network forwarding and post-processing, excluding the data loading time. The *FPS* is obtained averagely on the whole validation set.

Method	Backbone	mAP ↑	FPS ↑	Params (MB) ↓
HDMapNet	Eff-B0	22.7	0.7	69.8
BeMapNet	Eff-B0	40.7	**4.8**	**27.6**
BeMapNet	SwinT	**43.7**	3.7	55.3

Table 4. The comparison of running time and model size.

B.2. The Number of Transformer Layers.

Our ***BeMapNet*** is a DETR-like architecture and we explore the influence of the number of transformer layers in Table 5. As the number of layers increases, the overall AP improves gradually, but the performance tends to be saturated when the number reaches 6. Note that when we reduce the number of layers to 1 for all three modules, the performance can still reach 30.9 *AP*.

$(N^{bev}_{enc}, N^{bev}_{dec}, N^{ins}_{dec})$	AP$_{divider}$	AP$_{ped}$	AP$_{boundary}$	mAP
[1, 1, 1]	35.4	27.8	29.5	30.9
[2, 2, 2]	44.1	36.0	36.5	38.9
[2, 2, 6]	45.9	38.4	36.1	40.1
[2, 4, 6]	49.1	42.2	39.9	43.7
[2, 6, 6]	50.8	43.0	41.8	45.2
[6, 4, 6]	**52.1**	**44.2**	**43.0**	**46.4**
[6, 6, 6]	51.9	42.7	42.9	45.8

Table 5. The impact of different number of transformer layers.

B.3. The Number of *BEV* Queries.

According to the main paper, each query models the feature of a specific region on *BEV*. In other words, the number of queries represents the resolution of *BEV* feature, which is closely related to the final AP performance. Table 6 verifies the above conjecture.

# *BEV* Queries	AP$_{divider}$	AP$_{ped}$	AP$_{boundary}$	mAP
8×16	39.3	31.9	33.3	34.8
16×32	44.6	39.3	37.3	40.4
32×64	49.1	42.2	39.9	43.7
64×128	**51.3**	**43.5**	**42.1**	**45.6**
80×160	51.1	43.2	39.6	44.6

Table 6. The impact of different number of *BEV* queries.

B.4. The Number of Bézier Queries.

The instance Bézier decoder infers curve predictions with a fixed-size set, which is usually larger than the typical number of map elements. Table 7 shows the impact with different number of queries for *lane-divider*, *ped-crossing*, *road-boundary*.

# Bézier Queries	AP$_{divider}$	AP$_{ped}$	AP$_{boundary}$	mAP
[10, 12, 8]	47.7	39.5	39.1	42.1
[20, 25, 15]	49.1	**42.2**	39.9	43.7
[30, 36, 24]	**50.2**	41.4	**40.0**	**43.8**

Table 7. The impact of different number of Bézier queries.

B.5. The Impact of Different Bézier Setup

For piecewise Bézier curve, various $\boldsymbol{n}$ and $\boldsymbol{k}$ determine different fitting capabilities. Taking *road-boundary* as an example, we explore the impact in different Bézier setup on model performance from three aspects as follows,
1) fixed $\boldsymbol{k}$. The curve with larger n has stronger curve fitting ability, but it will also increase the learning difficulty of the model. The results in Table 8 confirm this conjecture.
2) fixed $\boldsymbol{n}$. As shown in Table 9, as the piece number gradually increases, the model performance gradually improves and tends to be saturated when k reaches a certain value.
3) fixed the number of control points $\boldsymbol{nk+1}$. With exactly same expressive ability, Table 10 proves that too many pieces and too large degree both degrade the performance.

$\langle \boldsymbol{k}, \boldsymbol{n} \rangle$	AP$_{boundary}$
$k = 8,\ n = 1$	33.4
$k = 8,\ n = 2$	39.2
$k = 8,\ n = 3$	**39.6**
$k = 8,\ n = 4$	39.5

Table 8. The impact of different number of degree n ($k = 8$).

$\langle \boldsymbol{k}, \boldsymbol{n} \rangle$	AP$_{boundary}$
$k = 2,\ n = 3$	32.6
$k = 4,\ n = 3$	38.4
$k = 6,\ n = 3$	39.1
$k = 7,\ n = 3$	**39.9**
$k = 8,\ n = 3$	39.6
$k = 10,\ n = 3$	39.4
$k = 12,\ n = 3$	39.6

Table 9. The impact of different number of piece k ($n = 3$).

$\langle \boldsymbol{k}, \boldsymbol{n} \rangle$	AP$_{boundary}$
$k = 24,\ n = 1$	33.8
$k = 12,\ n = 2$	37.7
$k = 8,\ n = 3$	**39.6**
$k = 6,\ n = 4$	39.1

Table 10. The impact of different number of n and k ($nk = 24$).

B.6. The Impact of Input Resolution

Comparing the 1-*st* and 4-*th* row in Table 11, the resolution is increased by $8.4\times$, and the performance is improved by 4.5 AP. We adopt resolution 896×512 in all experiments.

Input Size	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
512×320	46.2	37.5	38.7	40.8
640×384	47.9	38.9	38.5	41.8
896×512	49.1	42.2	39.9	43.7
1536×896	**50.4**	**43.3**	**42.4**	**45.3**

Table 11. The impact of different input resolution.

B.7. The Impact of Different FPN-Output Shape

We directly interpolate the multi-scale features from FPN to a fixed size and then concat them together as the input of the subsequent *BEV* decoder. As the upsample shape increases, the overall AP improves gradually, but the performance tends to be saturated when the shape reaches 30×70.

UpSample Shape	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
12×28	47.8	38.4	39.2	41.8
21×49	49.1	42.2	39.9	43.7
30×70	**50.3**	**43.3**	**42.0**	**45.2**
36×84	50.2	42.7	41.6	44.8

Table 12. The impact of different output shape of FPN.

B.8. The Impact of Different Curve Length

According to the matrix form of the Bézier definition, $P = B \times C$, where $B \in \mathbb{R}^{m\times n}$ and m is the point number of the curve, *i.e*. curve length, which is closely related to the curve supervision part in *PCR-Loss*. Table 13 summarizes the performance with different curve length.

Curve Length	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
25	46.2	40.2	37.7	41.4
50	48.0	39.8	39.3	42.4
100	**49.1**	**42.2**	**39.9**	**43.7**
200	47.8	40.0	38.4	42.0

Table 13. The impact of different length of restored Bézier curve.

B.9. More details of the proposed *GenGT*

Line-1&2&3: initialize the Bernstein coefficient matrix and its pseudo-inverse with given n, m through math definition.
Line-4: keep the loop condition: *start index* $<$ *end index*.
Line-5: interpolate the (s, e) sub-curve to length m(get $P^{\dagger}$).
Line-6: get control points $\mathbb{C}^{\dagger}$ of $P^{\dagger}$ by least squares fitting.
Line-7: restore its Bézier curve $P^{\ddagger}$ accurately by above $\mathbb{C}^{\dagger}$.
Line-8: compute fitting error (*CD-dist*) between $P^{\dagger}$ and $P^{\ddagger}$.
Line-9&10&11: update start/end index based on fit status.

B.10. Verification of the Generated Bézier GTs

To verify the reliability of the ground truth produced by Algorithm ***GTGen*** in the main paper, we first restore the generated control point sequences to their corresponding Bézier curves, and then treat these curves as predictions and the original annotations as ground-truths. By conducting the exact same evaluation protocol, we calculate the AP performance between them, see Table 14 for details. When the Chamfer Distance threshold of true positive (termed as τ in Table 14) is set to $0.2m$, no matter which degree of the curve is adopted, the overall mAP can reach more than 99.94. Moreover, under the more tight setting of $\tau = 0.1m$, each mAP can still achieve $97.7 \sim 98.7$.

Degree	τ	$AP_{divider}$	AP_{ped}	$AP_{boundary}$	mAP
1	0.1	97.082	99.485	96.599	97.722
2	0.1	98.978	99.485	96.949	98.471
3	0.1	99.401	99.485	96.221	98.369
4	0.1	99.803	99.485	96.726	98.671
1	0.2	99.955	99.892	99.998	99.948
2	0.2	99.955	99.892	99.999	99.949
3	0.2	99.955	99.892	99.995	99.947
4	0.2	99.955	99.892	99.999	99.949

Table 14. The reliability verification of generated Bézier GTs.

C. Statistical Analysis of Benchmark

C.1. Data Split by Different Conditions

In order to explore the effectiveness of ***BeMapNet*** under different lighting and weather conditions, we further divide *NuScenes* [1] into five kinds of scene, *i.e. day*, *night*, *sunny*, *cloudy*, and *rainy*. Fig.4∼5 provide the data distribution and scenario example for different subsets respectively.

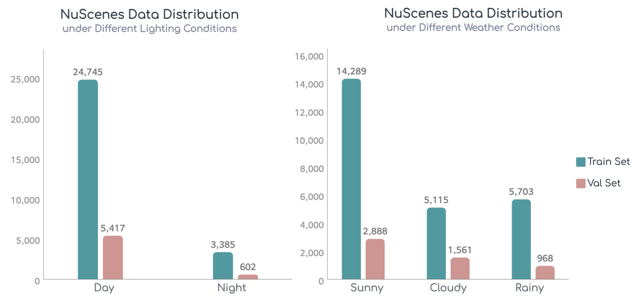

Figure 4. Data distribution under different conditions.

C.2. More Compact Map Element Expression

As for data labeling and downstream applications, the expression form of map elements is crucial, which affects the efficiency of data storage and transmission. Fig.6 compares the *NuScenes* original point-GT and Generated Bézier curve-GT from the perspective of the number of required

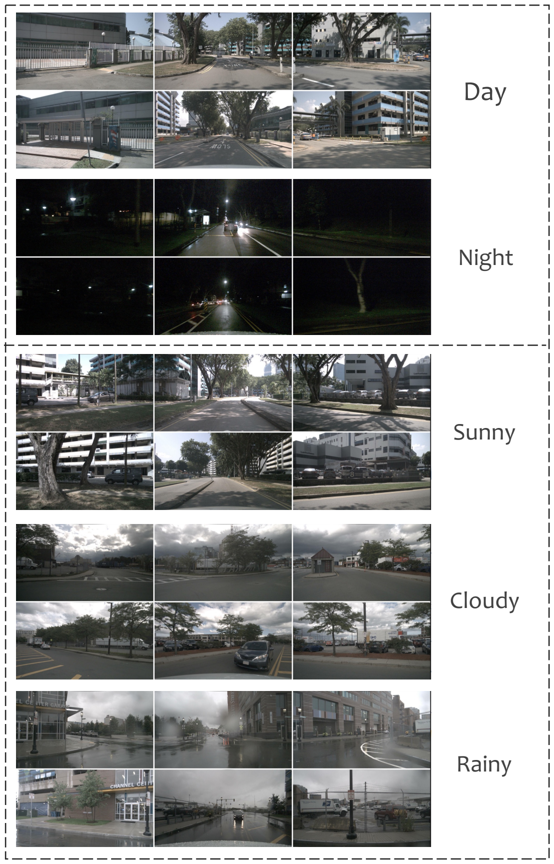

Figure 5. Scenario examples under different conditions.

annotated points. We notice that the latter form is more than $\mathbf{81}\%$ more effective than the former at least, indicating that Bézier curve is a more compact expression pathway.

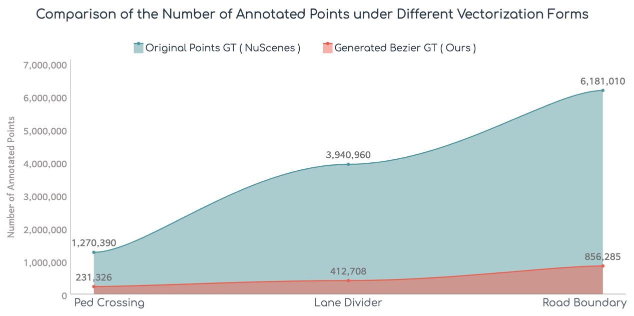

Figure 6. The compactness of different annotated forms.

C.3. How many pieces are reasonable?

when using piecewise Bézier curve to express *HD-map*, the number of pieces for different object is various with a given degree. Fig.7 counts the number of segments required for different map instances in *NuScenes* under the prerequisite of being fully represented. In the main paper, we utilize the deployment of $\langle \mathbf{3}, \mathbf{2} \rangle$, $\langle \mathbf{1}, \mathbf{1} \rangle$, $\langle \mathbf{7}, \mathbf{3} \rangle$ for $\mathit{lane\text{-}divider}$, $\mathit{ped\text{-}crossing}$ and $\mathit{road\text{-}boundary}$ respectively, which is a reasonable setting according to the Fig.7.

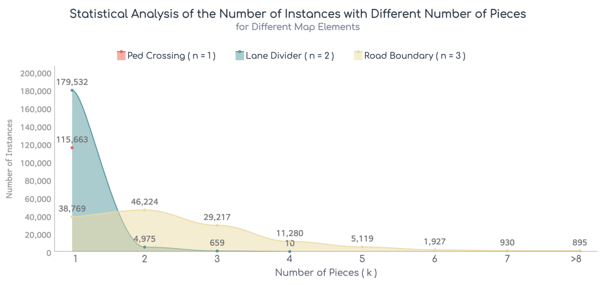

Figure 7. The statistics of instance number under different pieces.

C.4. How many degree are reasonable?

Taking $\mathit{road\text{-}boundary}$ as an example, Fig.8 shows the piece-number & instance-number distribution at different degree. Note the higher the degree, the fewer the number of pieces required and the curve is more difficult to model.

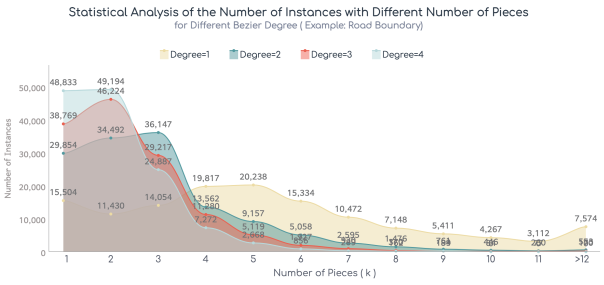

Figure 8. The statistics of *ins.*-number under different degree.

D. Qualitative Visualization

- Fig.10∼11: results under different lighting conditions.
- Fig.12∼14: results under different weather conditions.
- Fig.15: more results under difficult road scenarios.
- Fig.16: some badcases for future improvement.

D.1. The reason for rounded corner arising

The rounded corner issue is mainly caused by the slightly inaccurate prediction of some ***key*** control points. For example, a right-angle case in Fig. 9 can be simply formulated by $\langle \mathbf{2}, \mathbf{2} \rangle$ with 5 control points in GT. Assuming that the control point prediction at the turn position has only a small offset ($c_2^0 \rightarrow \hat{c}_2^0$), while the other locations are completely accurate, the final restored Bézier curve will naturally produce rounded corner. Further efforts on some ***key*** control points are important future jobs.

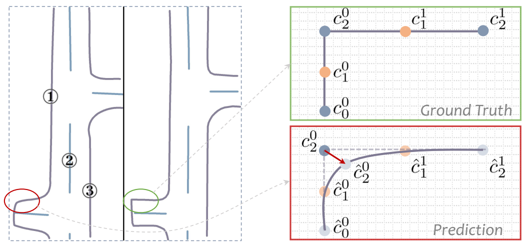

Figure 9. The illustration for rounded corner interpretation.

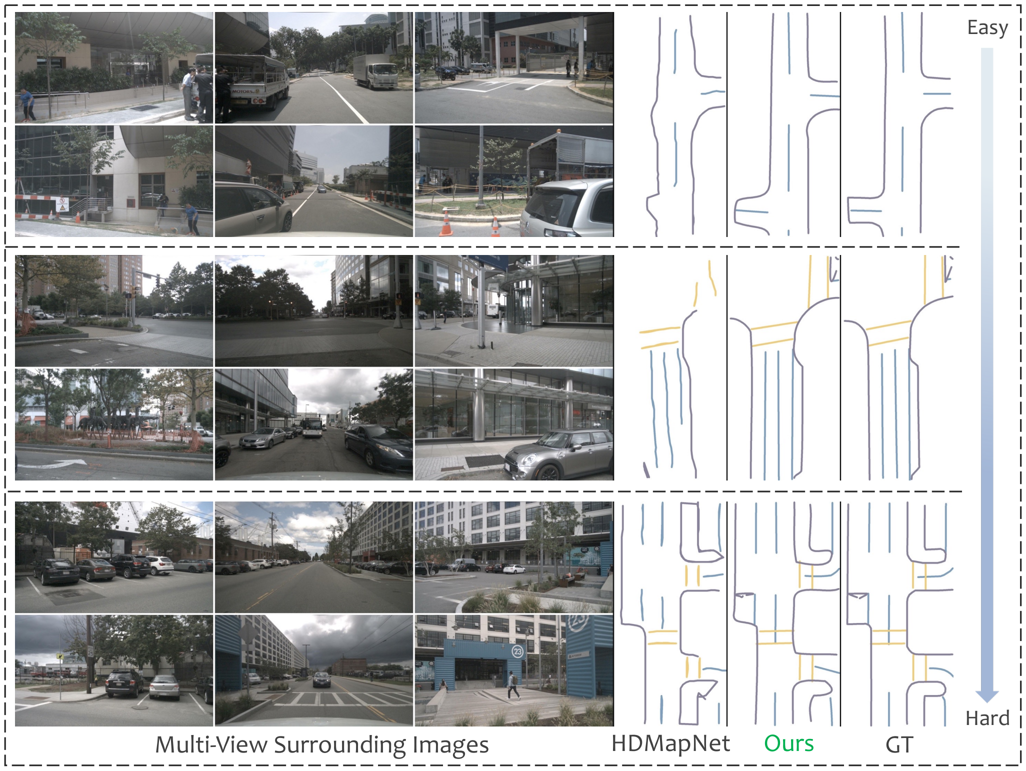

Figure 10. The visualization results under the lighting condition of ***daytime*** (easy $\rightarrow$ hard).

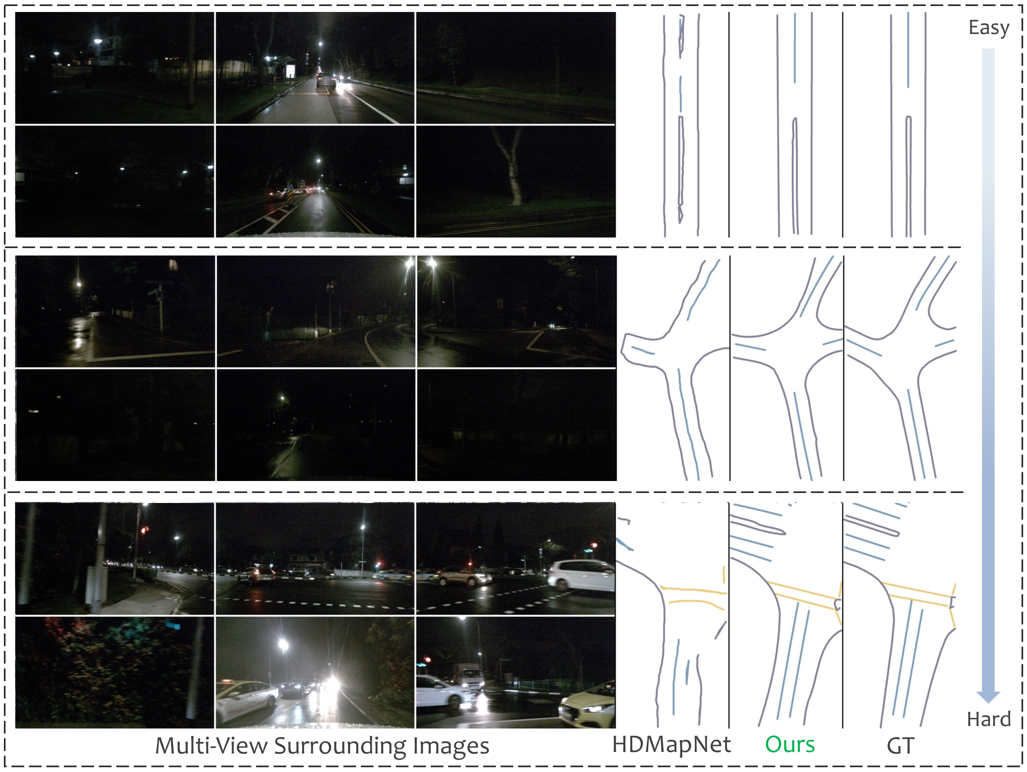

Figure 11. The visualization results under the lighting condition of ***nighttime*** (easy $\rightarrow$ hard).

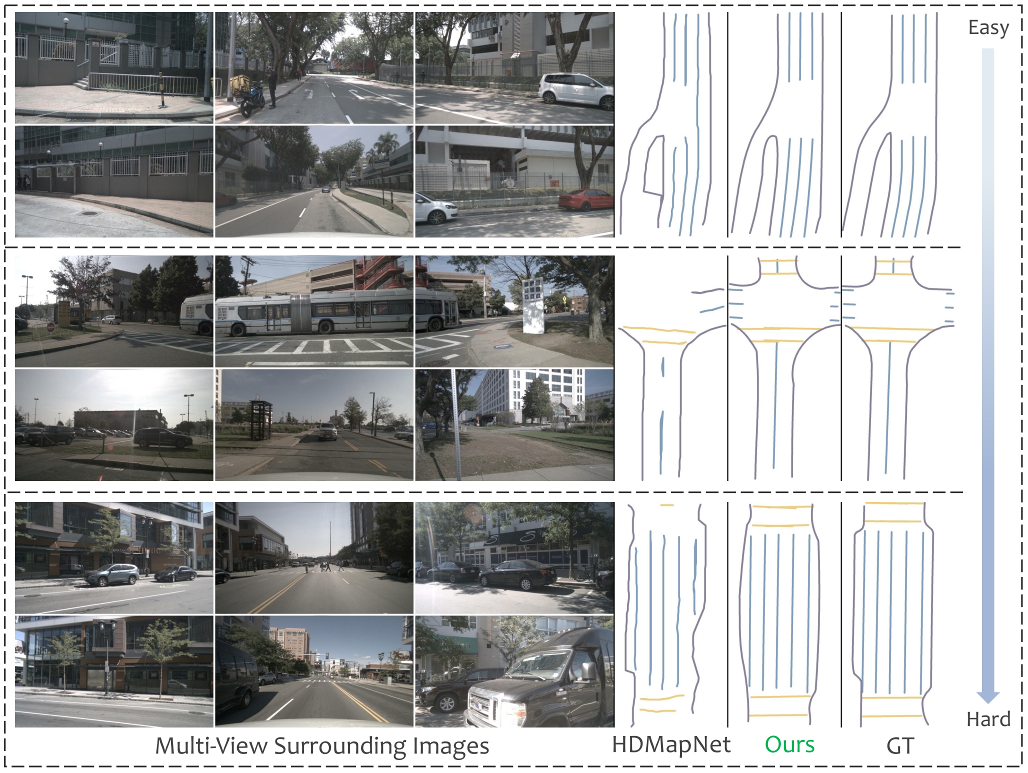

Figure 12. The visualization results under the weather condition of ***sunny*** (easy → hard).

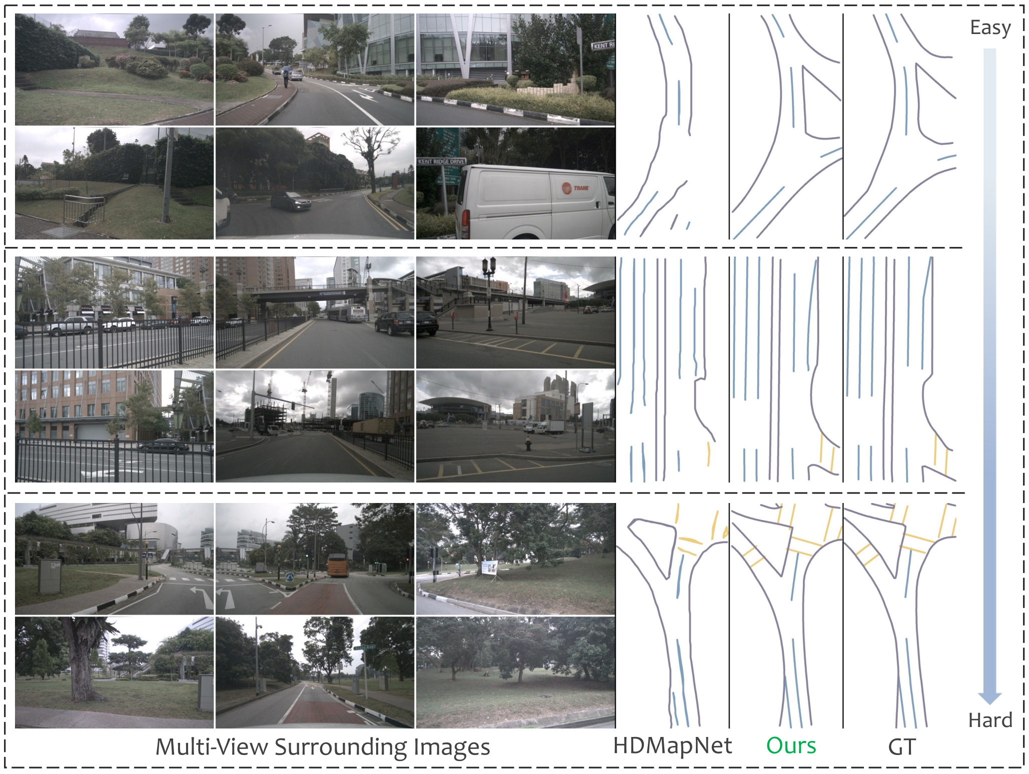

Figure 13. The visualization results under the weather condition of ***cloudy*** (easy → hard).

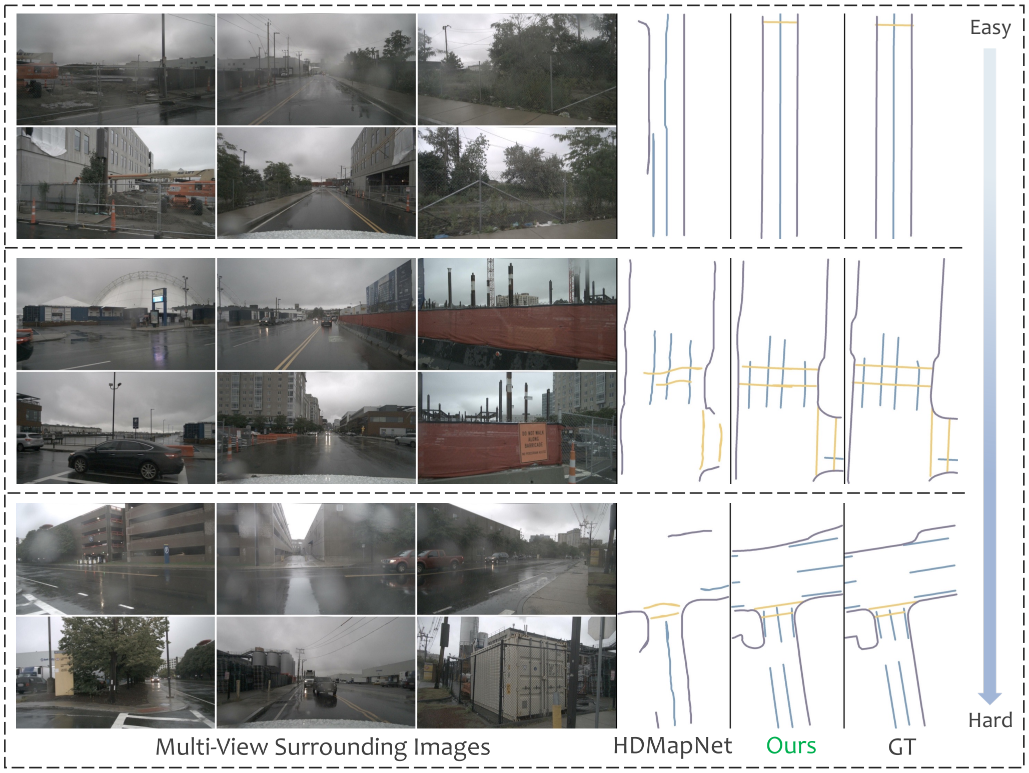

Figure 14. The visualization results under the weather condition of ***rainy*** (easy → hard).

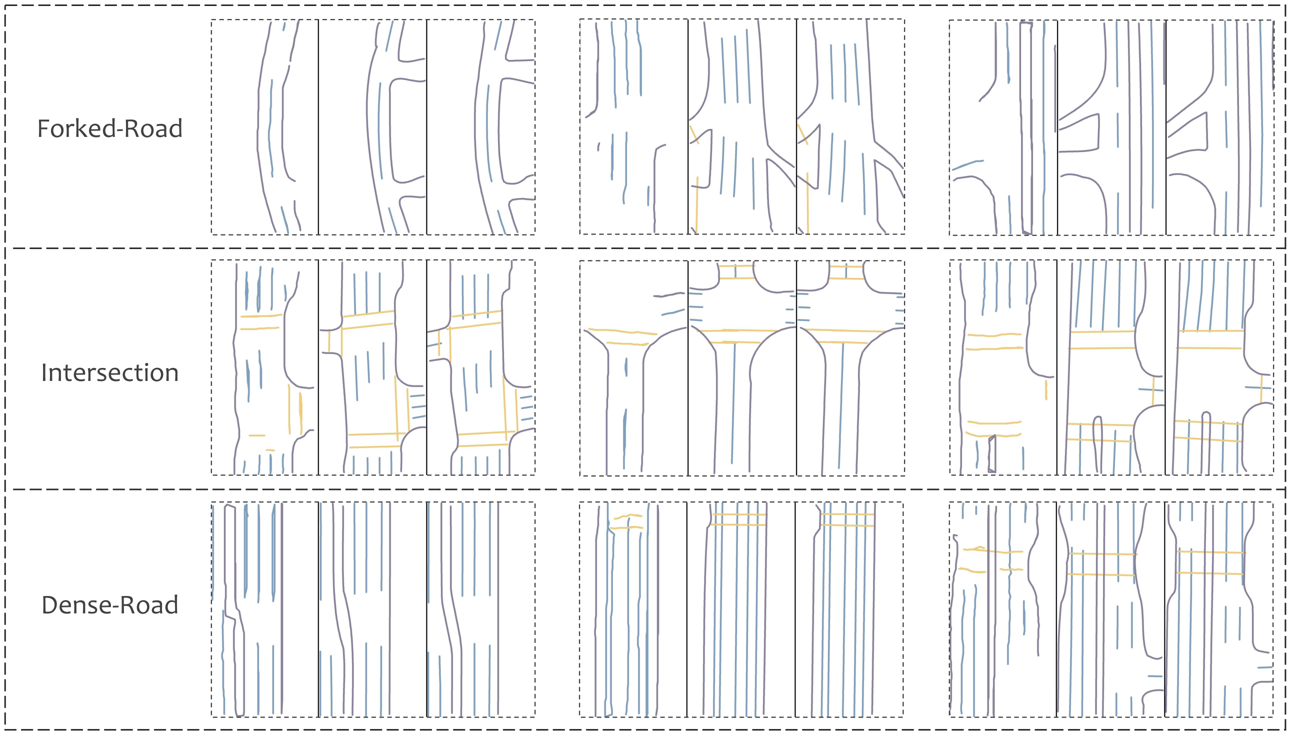

Figure 15. More visualization results under difficult road scenarios (Predictions from *HDMapNet*, ***BeMapNet*** and Ground-Truth).

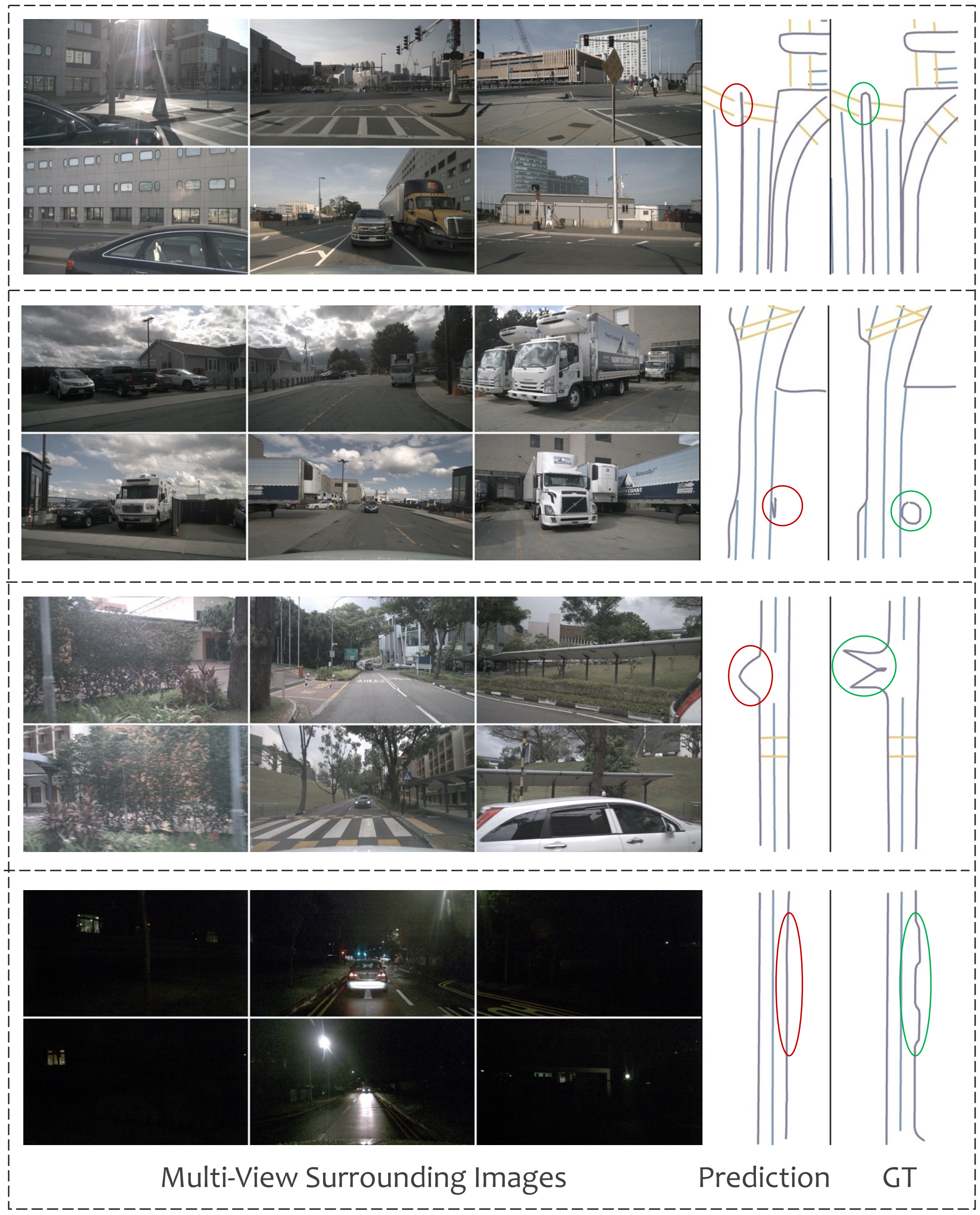

Figure 16. Some badcases in our current model (for future improvement).

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 3

[2] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020. 1

[3] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 2