

# AugMapNet: Improving Spatial Latent Structure via BEV Grid Augmentation for Enhanced Vectorized Online HD Map Construction

## Supplementary Material

### Overview of Supplementary Material

The supplementary material includes the following parts:

- A. Results on more Perception Ranges
- B. Formulation of Dense Spatial Supervision
- C. Mathematical Definition of Loss Functions
- D. Theoretical Foundation for Gradient Stop
- E. Non-functional Changes with AugMapNet Method
- F. Results on Original nuScenes Split
- G. Results on Argoverse2 for AugMapNet-SQD and Larger Range
- H. Ablation of Kernel Size for Latent BEV Grid Processing CNNs
- I. Additional Qualitative Results
- J. Visualization of Principal Components

### A. Results on more Perception Ranges

Tab. 7 shows the results on various perception ranges. Beyond the reported 13.3 % improvement on perception range  $60\text{ m} \times 30\text{ m}$ , we get 19.6 % on  $80\text{ m} \times 40\text{ m}$ , 24.6 % on  $100\text{ m} \times 50\text{ m}$ , 23.4 % on  $120\text{ m} \times 60\text{ m}$ , and 41.4 % on  $150\text{ m} \times 75\text{ m}$ . We find that the relative improvement of AugMapNet increases with larger perception ranges, which suggests a stronger benefit of our augmentation method on wider ranges.

### B. Formulation of Dense Spatial Supervision

The loss is calculated over all elements of the output representation  $\hat{\mathcal{M}}$ . Supervision is done by optimizing a learnable set of weights  $W$  based on the gradients of the loss:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \hat{\mathcal{M}}} \frac{\partial \hat{\mathcal{M}}}{\partial W}. \quad (4)$$

The total amount of spatial supervision therefore is related to the number of elements in  $\hat{\mathcal{M}}$ .

In the case of a raster map decoder, the predicted output is a rasterized representation:  $\hat{\mathcal{M}} = \hat{\mathcal{M}}_{\text{raster}}$ . The total gradient signal is calculated from the loss over all pixels. For a raster with spatial resolution  $H = 100$  and  $W = 50$ , this gives a spatial supervision for  $|\hat{\mathcal{M}}_{\text{raster}}| = H \times W = 100 \times 50 = 5000$  elements.

In the case of a vector map decoder, the predicted output is a vectorized representation:  $\hat{\mathcal{M}} = \hat{\mathcal{M}}_{\text{vector}}$ . The total gradient signal is calculated from the loss over all points of all polylines. Our model outputs  $N = 100$  polylines, with  $N_p = 20$  points each, which gives spatial supervision

Range	Variant	AP <sub>ped</sub>	AP <sub>div</sub>	AP <sub>bound</sub>	mAP	Impr.
$60\text{ m} \times 30\text{ m}^\ddagger$	StreamMapNet [42]	31.2	27.3	42.9	33.8	13.3 %
	AugMapNet (ours)	39.4	30.3	45.3	38.3	
$80\text{ m} \times 40\text{ m}^\ddagger$	StreamMapNet [42]	19.4	18.9	24.1	20.8	19.6 %
	AugMapNet (ours)	25.9	21.1	27.6	24.9	
$100\text{ m} \times 50\text{ m}^*$	StreamMapNet [42]	25.5	19.3	24.7	23.2	24.6 %
	AugMapNet (ours)	35.5	22.8	28.4	28.9	
$120\text{ m} \times 60\text{ m}^*$	StreamMapNet [42]	19.29	10.95	12.79	14.34	23.4 %
	AugMapNet (ours)	24.13	12.72	16.26	17.70	
$150\text{ m} \times 75\text{ m}^*$	StreamMapNet [42]	10.5	6.4	3.9	6.9	41.4 %
	AugMapNet (ours)	15.2	7.5	6.7	9.8	

Table 7. Results on nuScenes dataset [1] on more perception ranges. AP thresholds  $^\ddagger$ : {0.5, 1.0, 1.5},  $^*$ : {1.0, 1.5, 2.0}.

for at most  $|\hat{\mathcal{M}}_{\text{vector}}| = N \times N_p = 100 \times 20 = 2000$  elements. Note that, in DETR-style decoders, elements that are correctly predicted as empty class do not induce spatial supervision. Hence, this number is the upper bound and the actual supervision is typically even sparser.

We believe that this difference in the number of elements is why integrating dense spatial supervision into vectorized map decoding is so effective.

### C. Mathematical Definition of Loss Functions

The formulation of the vector map decoder loss,  $\mathcal{L}_{\text{vector}}$ , is taken from StreamMapNet [42] and consists of multiple components. As the first step of the polyline matching loss,  $\mathcal{L}_{\text{line}}$ , bipartite matching is performed between predicted and GT polylines. After matching, the smooth L1 loss is calculated for each of the  $N_p$  points  $P_j$  of the matched polyline  $P$ . The best one in the permutation group,  $\Gamma$ , as introduced in MapTR [19], is used:

$$\mathcal{L}_{\text{line}}(\hat{P}, P) = \min_{\gamma \in \Gamma} \frac{1}{N_p} \sum_{j=1}^{N_p} \mathcal{L}_{\text{SmoothL1}}(\hat{P}_j, P_{\gamma(j)}). \quad (5)$$

The classification loss,  $\mathcal{L}_{\text{class}}$ , calculates the loss between the ground truth class vector,  $c$ , and predicted class vector,  $\hat{c}$ , for each polyline. The loss function is the Focal loss:

$$\mathcal{L}_{\text{class}}(\hat{c}, c) = \mathcal{L}_{\text{Focal}}(\hat{c}, c). \quad (6)$$

An auxiliary transformation loss,  $\mathcal{L}_{\text{trans}}$ , is used to match the ego-motion transformation in latent space. Given a standard  $4 \times 4$  transformation matrix,  $T$ , between the coordinate frames of  $t - 1$  and  $t$ , the polyline of the vector map at time  $t$  is expressed as  $P = T \cdot \text{homogeneous}(P')_{:,0:2}$ , where  $P'$  is the polyline of the vector map at time  $t - 1$ . For a

query in the vector map decoder at time  $t$ ,  $Q$ , an auxiliary prediction is made with  $\hat{P}^{\text{aux}} = \text{Reg}(Q)$ . The auxiliary transformation loss is then defined as:

$$\mathcal{L}_{\text{trans}}(\hat{P}^{\text{aux}}, P) = \sum_{j=1}^{N_p} \mathcal{L}_{\text{SmoothL1}}(\hat{P}_j^{\text{aux}}, P_j). \quad (7)$$

The final loss is a weighted sum of the above loss terms with the factors  $\lambda_1 = 50.0$ ,  $\lambda_2 = 5.0$ , and  $\lambda_3 = 0.1$  over all predicted polylines in  $\hat{\mathcal{M}}_{\text{vector}}$ :

$$\mathcal{L}_{\text{vector}} = \sum_{P \in \hat{\mathcal{M}}_{\text{vector}}} (\lambda_1 \mathcal{L}_{\text{line}} + \lambda_2 \mathcal{L}_{\text{class}} + \lambda_3 \mathcal{L}_{\text{trans}}). \quad (8)$$

The raster map decoder loss  $\mathcal{L}_{\text{raster}}$  is the Dice loss [25]:

$$\mathcal{L}_{\text{raster}}(\hat{\mathcal{M}}_{\text{seg}}, \mathcal{M}_{\text{seg}}) = \mathcal{L}_{\text{Dice}}(\hat{\mathcal{M}}_{\text{seg}}, \mathcal{M}_{\text{seg}}). \quad (9)$$

## D. Theoretical Foundation for Gradient Stop

We provide more analysis on gradient stopping to provide a theoretical foundation for the benefits that we observed empirically. We start our reasoning from the results of the ‘‘Oracle’’ experiment (see details of experiment in Sec. 4.7).

The result of 91.4 % mAP shows that given a perfect raster map, the injection of dense spatial features in our AugMapNet method can yield a close-to-perfect result on the vectorized map construction. We denote using this perfect raster map, that reaches the *empirical upper bound* in the ‘‘Oracle’’ experiment, as  $\hat{\mathcal{M}}_{\text{raster}} = \mathcal{M}_{\text{raster}}^*$ . In practice, the predicted raster map includes residual error  $\eta$ , such that  $\hat{\mathcal{M}}_{\text{raster}} = \mathcal{M}_{\text{raster}}^* + \eta$ . Without gradient stopping  $\nabla e_{\text{raster}}$  is active, backpropagating gradients from the vector loss  $\mathcal{L}_{\text{vector}}$  to  $d_{\text{raster}}$ . Therefore, instead of purely optimizing for a perfect raster map,  $\frac{\partial \mathcal{L}_{\text{vector}}}{\partial W_{d_{\text{raster}}}}$  biases  $d_{\text{raster}}$  toward vector-specific features, increasing  $\|\eta\|$ .

## E. Non-functional Changes with AugMapNet Method

This section provides more details on the non-functional changes when applying our AugMapNet method to StreamMapNet. Our AugMapNet method has only a small effect on the model size in VRAM. We observe  $\approx 1.8$  GB for either of them. During training on batch size 1, we see a VRAM usage of  $\approx 7.7$  GB, a slight increase of 1.6 %. Train time also increases slightly from 14.5 h to 15.4 h (6 %). Inference takes 78 ms vs. 70 ms, an 11 % increase. AugMapNet still meets real-time requirements with 11.9 FPS.

## F. Results on Original nuScenes Split

The original nuScenes Split is still commonly used to train and evaluate approaches for online map construction. For completeness, the results on the original split are shown

Method	AP <sub>ped</sub>	AP <sub>div</sub>	AP <sub>bound</sub>	mAP	Impr.
VectorMapNet [21]	36.1	47.3	39.3	40.9	
MapTR [18]	46.3	51.5	53.1	50.3	
MapTRv2 [19]	59.8	62.4	62.4	61.5	
MapVR [43]	47.7	54.5	51.4	51.2	
MGMap [20]	57.4	63.5	63.3	61.4	
StreamMapNet [42]	60.2	65.1	61.1	62.1	
AugMapNet (ours, $\nabla e_{\text{raster}}$ off)	60.8	65.7	61.7	62.7	1.0 %
AugMapNet (ours, $\nabla e_{\text{raster}}$ on)	61.9	65.4	63.6	63.6	2.4 %
StreamMapNet 100x50 [42]	63.5	64.9	57.1	61.8	
AugMapNet 100x50 (ours)	66.1	64.1	60.0	63.4	2.5 %
SQD-MapNet [39]	62.2	67.0	65.4	64.9	
AugMapNet-SQD (ours)	64.1	65.0	67.8	65.6	1.1 %
SQD-MapNet 100x50 [39]	62.9	65.8	61.4	63.3	
AugMapNet-SQD 100x50 (ours)	67.3	68.6	63.7	66.5	3.5 %

Table 8. Results on original nuScenes [1] split with geospatial overlap. Results from baselines are taken from the respective papers. Values are much higher compared to geospatially disjoint split due to overfitting.

in Tab. 8. Comparing the absolute values to the results on the geospatially disjoint split (*cf.* Tab. 1) reveals severe overfitting due to geospatial overlap: StreamMapNet result increases from 33.8 % mAP to 62.1 % mAP, an 84 % jump due to overfitting. AugMapNet overfits less with only 64 % higher mAP (62.7 % mAP on old vs. 38.3 % mAP on geospatially disjoint splits). Thanks to less overfitting, AugMapNet improvements are smaller on the old split, though still substantial. A primary reason is gradient stopping: turning off  $\nabla e_{\text{raster}}$  improves performance on the geospatially disjoint split (*cf.* Sec. 4.7) but not on the old split (62.7 % mAP without  $\nabla e_{\text{raster}}$  vs. 63.6 % mAP without  $\nabla e_{\text{raster}}$ ). Consistent with the results on the geospatially disjoint split, AugMapNet shows an even stronger improvement on larger perception ranges with 2.5 % when evaluated on 100 m  $\times$  50 m range on the original split (*cf.* Tab. 3 and Tab. 8).

We follow best practices in ML to only perform comparisons on truly unseen evaluation data (*i.e.*, the geospatially disjoint split). For completeness, other baselines are added to Tab. 8. We note that comparison of absolute values mainly depends on the choice of baseline. Once code for the latest state-of-the-art models is available, AugMapNet can easily be integrated to further advance them.

## G. Results on Argoverse2 for AugMapNet-SQD and Larger Range

This section extends upon the results of AugMapNet on Argoverse2 (see Tab. 2). To assess how well our method generalizes across datasets and sensor configurations on different ranges, we extend Tab. 2 by adding 100 m  $\times$  50 m range. Beyond the 3.0 % improvement on 60 m  $\times$  30 m range, we observe 3.4 % improvement on 100 m  $\times$  50 m range.

To assess how well our method generalizes not only across datasets and sensor configurations, but also across

Range	Method	AP <sub>ped</sub>	AP <sub>div</sub>	AP <sub>bound</sub>	mAP	Impr.
60 m × 30 m <sup>‡</sup>	StreamMapNet [42]	56.0	54.4	61.0	57.1	3.0 %
	AugMapNet (ours)	57.4	57.4	61.6	58.8	
100 m × 50 m <sup>*</sup>	StreamMapNet [42]	57.9	44.4	47.5	49.9	3.4 %
	AugMapNet (ours)	60.6	44.8	49.4	51.6	
60 m × 30 m <sup>‡</sup>	SQD-MapNet [39]	58.3	54.7	62.2	58.4	4.3 %
	AugSQD (ours)	60.0	58.0	64.6	60.9	
100 m × 50 m <sup>*</sup>	SQD-MapNet [39]	59.2	45.9	49.7	51.6	6.2 %
	AugSQD (ours)	63.3	49.7	51.4	54.8	

Table 9. Results on different perception ranges for AugMapNet and application of our method to SQD-MapNet [39] denoted as AugSQD. Results are on Argoverse2 split without geospatial overlap [1]. AP thresholds <sup>‡</sup>: {0.5, 1.0, 1.5}, <sup>\*</sup>: {1.0, 1.5, 2.0}.

Index	Kernel	# Layers	AP <sub>ped</sub>	AP <sub>div</sub>	AP <sub>bound</sub>	mAP
a)	-	0	31.2	27.3	42.9	33.8
b)	1	1	36.6	30.3	41.4	36.1
c)	3	1	36.4	30.0	43.7	<b>36.7</b>
d)	5	1	33.9	30.3	43.8	36.0

Table 10. Ablation of kernel size of BEV processing CNNs. a) is StreamMapNet [42].

models without any changes, we train AugMapNet-SQD (“AugSQD”) on the Argoverse2 dataset [40] and show the results in Tab. 9. AugMapNet-SQD reaches 60.9 % mAP on 60 m × 30 m range, a 4.3 % improvement over SQD-MapNet. On 100 m × 50 m range, AugMapNet-SQD reaches 54.8 % mAP, an even higher improvement of 6.2 %. The improvements are substantial and even higher than the 3.0 % improvement of AugMapNet over StreamMapNet on Argoverse2 (*cf.* Tab. 2), confirming the broader applicability of our method. Furthermore, the values show an increase in relative improvement at larger perception ranges a fifth time (*cf.* Tab. 3 and Tab. 8).

## H. Ablation of Kernel Size for Latent BEV Grid Processing CNNs

In this study, shown in Tab. 10, we investigate the effect of kernel size by comparing sizes of 1, 3, and 5 (*b*, *c*, *d*, resp.) on nuScenes. Similar to Sec. 4.8, here we do not use our BEV augmentation. Size 3 is used for further experiments since it gives the best result with 36.7 % mAP.

## I. Additional Qualitative Results

We provide further qualitative results to highlight the benefits of AugMapNet.

A separate video file is included with this submission to show the performance of AugMapNet on a full scene. It overlays the predicted vector map polylines in each camera view to visualize the spatial accuracy.

Fig. 6 shows a scene where StreamMapNet misses the road divider on the left side of the ego vehicle. AugMapNet predicts the road divider to include a roadway turnout,

which is reasonable given the large driveway that is occupied by a truck. Fig. 7 shows a scene where a traffic island and a crosswalk to the front-left of the ego vehicle are missed by StreamMapNet, but correctly predicted by AugMapNet. Fig. 8 visualizes a rainy scene to show model performance under challenging weather conditions, including a rain droplet on the front-facing camera lens that limits visibility. StreamMapNet misses a crosswalk in the area that AugMapNet predicts correctly. Fig. 9 shows a scene at night with limited illumination. AugMapNet predicts the left road boundary fairly accurately given the visibility, whereas StreamMapNet misses it completely.

## J. Visualization of Principal Components

As an extension to the top 3 Principal Components (PCs) visualized in Fig. 4, we visualize the top 16 PCs for example scene 1 in Fig. 10 and for example scene 2 in Fig. 11. A general observation is that the PCs tend to have more radial artifacts in StreamMapNet. The PCs of AugMapNet have better spatial structure and correspondence with the GT map due to dense spatial supervision.

Finally, Fig. 12 gives more details on example scene 1, where StreamMapNet misses a pedestrian crossing that is correctly predicted by AugMapNet. Specifically, we highlight the PC that has the highest visual correspondence to the pedestrian crossing GT out of the top 16 PCs visualized in Fig. 10 for each StreamMapNet and AugMapNet. It is PC 14 for StreamMapNet and PC 8 for AugMapNet. The smaller number for AugMapNet indicates that AugMapNet has stronger latent features for pedestrian crossings, which likely helped with its correct prediction. This also matches the better visual correspondence between the AugMapNet PC and the pedestrian crossing GT compared to StreamMapNet.



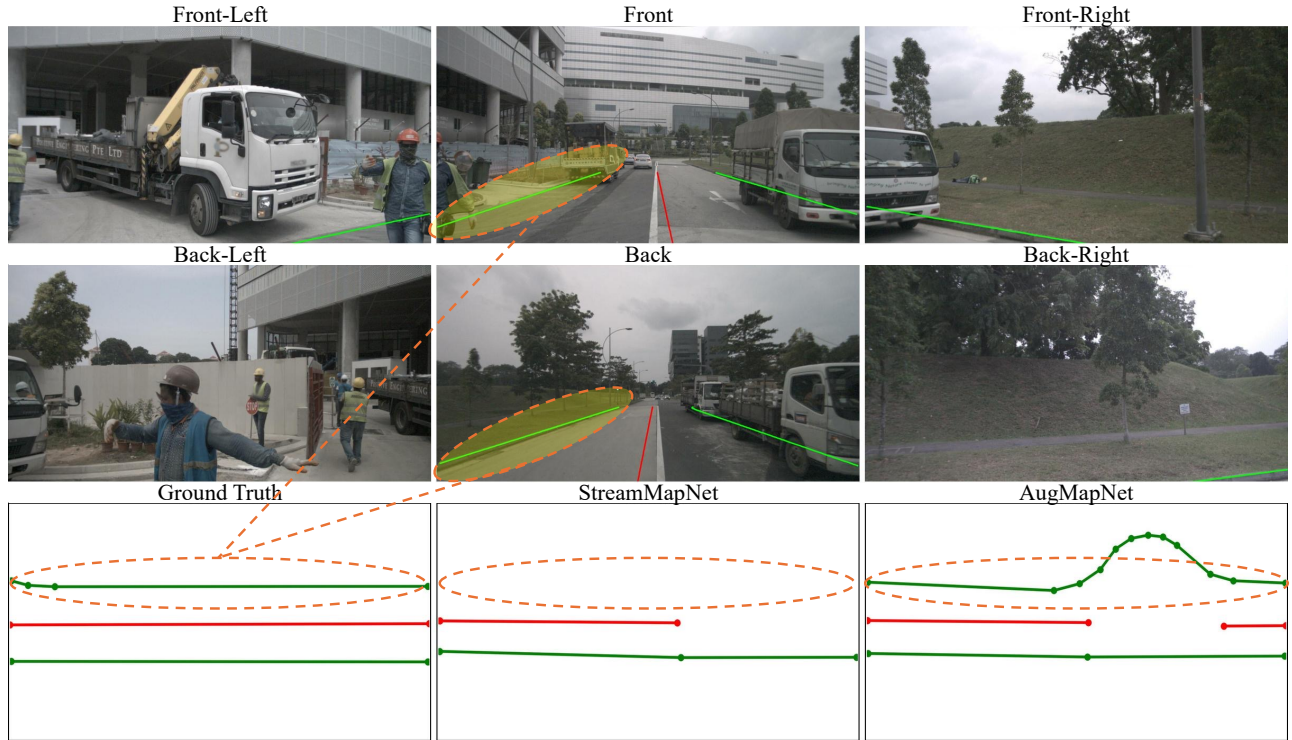


Figure 6. Qualitative results for example scene 3 with an extensive driveway on the left side. Lane dividers are red, road boundaries are green, and pedestrian crossings are blue.

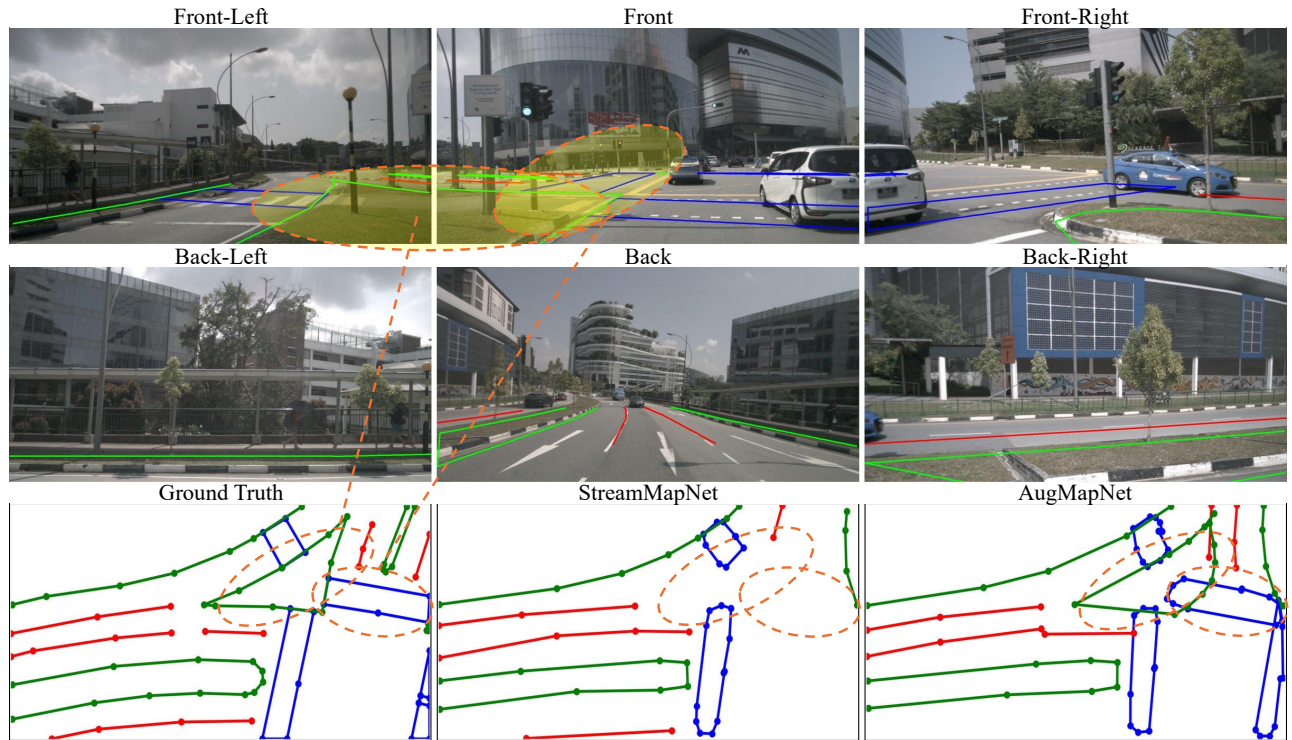


Figure 7. Qualitative results for example scene 4 with traffic island and pedestrian crossing on the left. Lane dividers are red, road boundaries are green, and pedestrian crossings are blue.

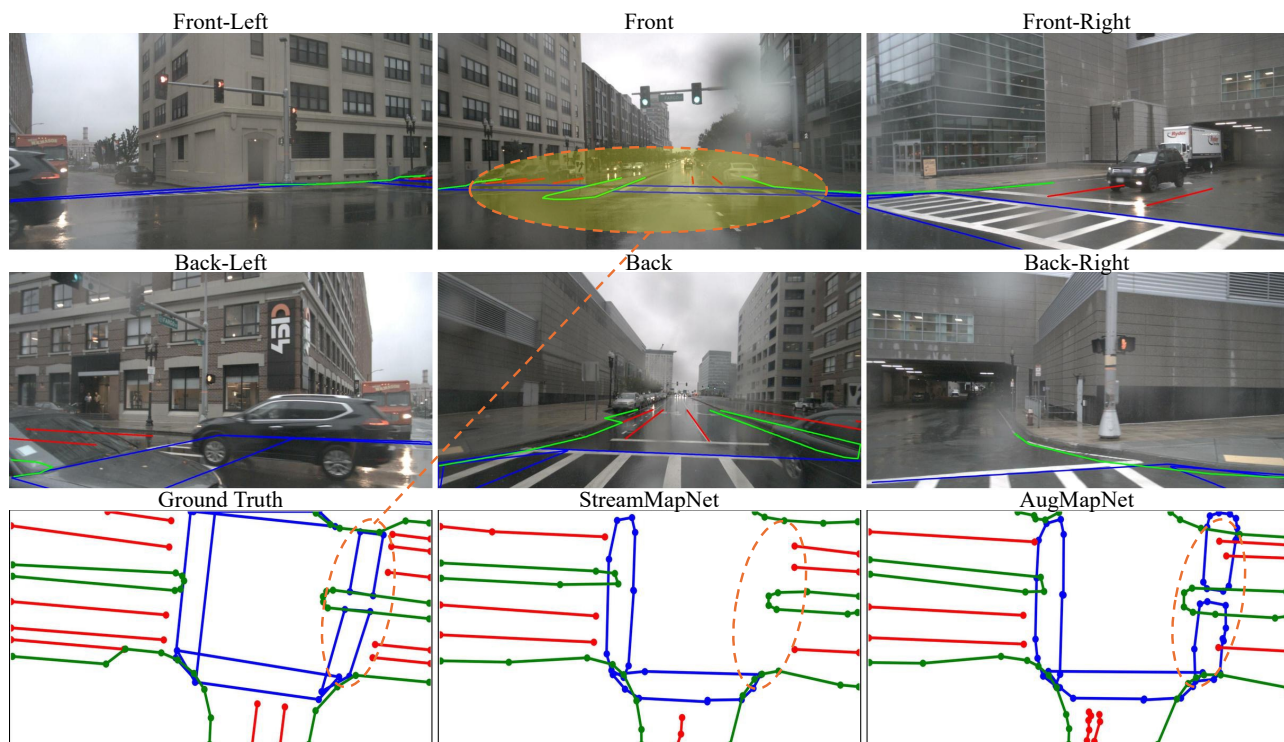


Figure 8. Qualitative results for example scene 5 under rainy conditions. Lane dividers are red, road boundaries are green, and pedestrian crossings are blue.

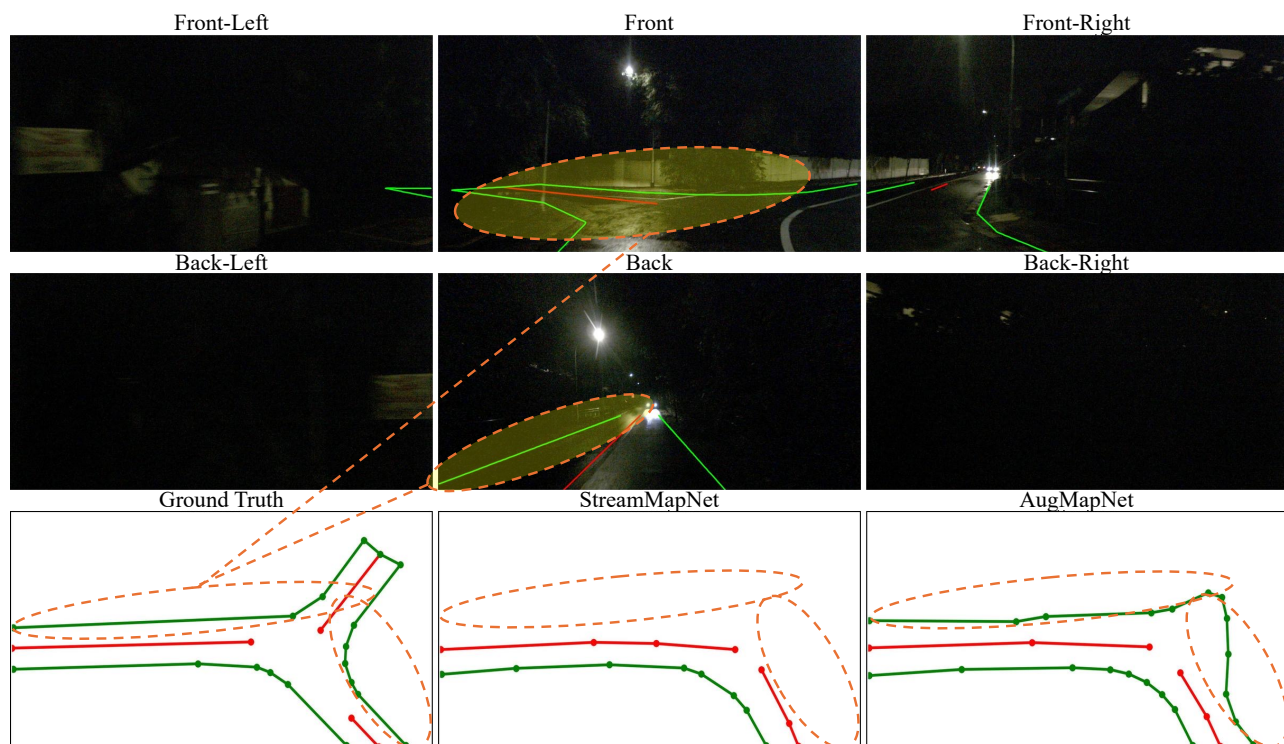
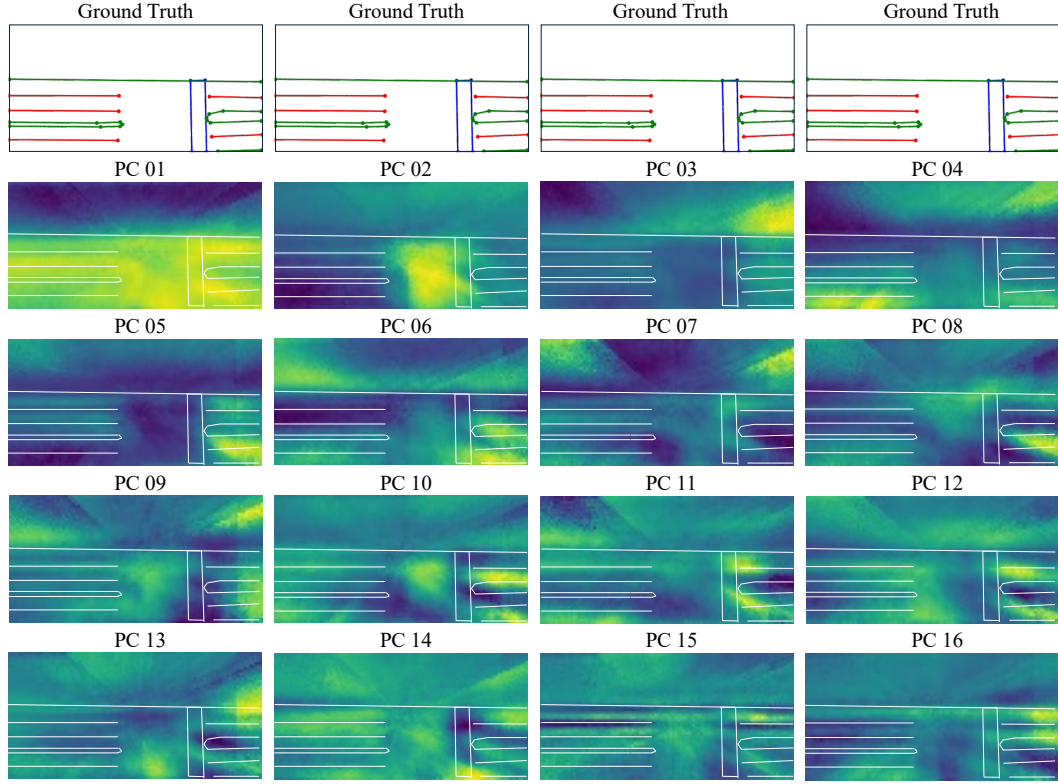
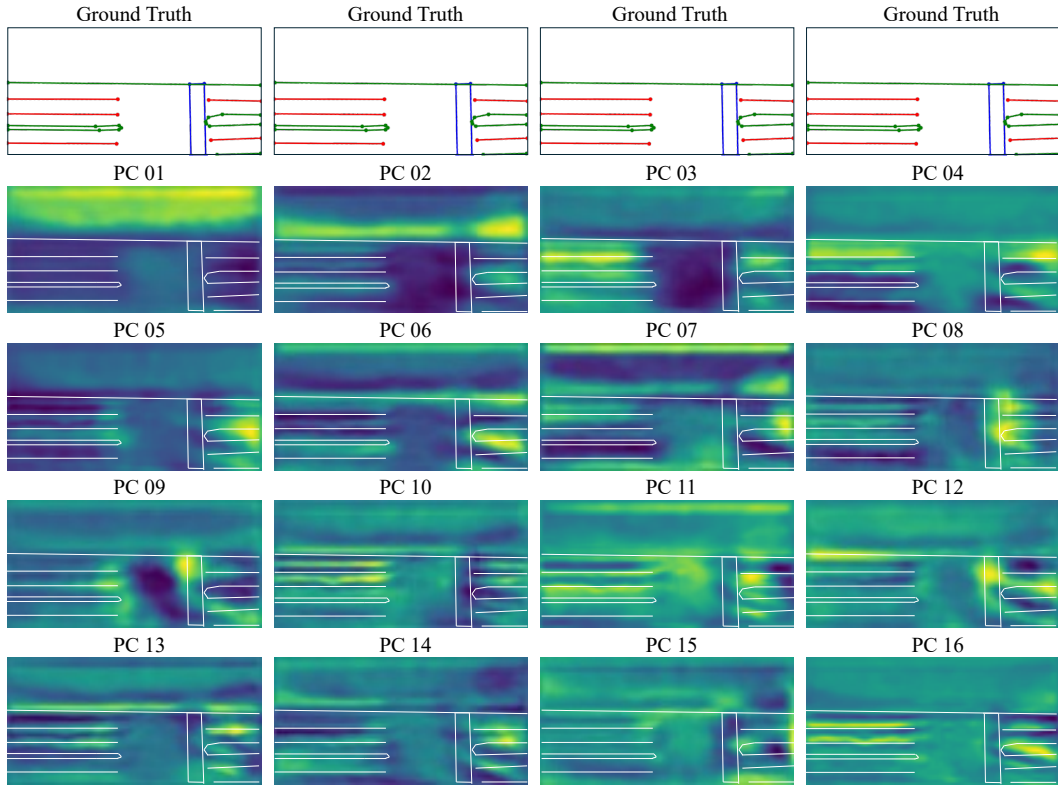


Figure 9. Qualitative results for example scene 6 with limited illumination. Lane dividers are red, road boundaries are green, and pedestrian crossings are blue.



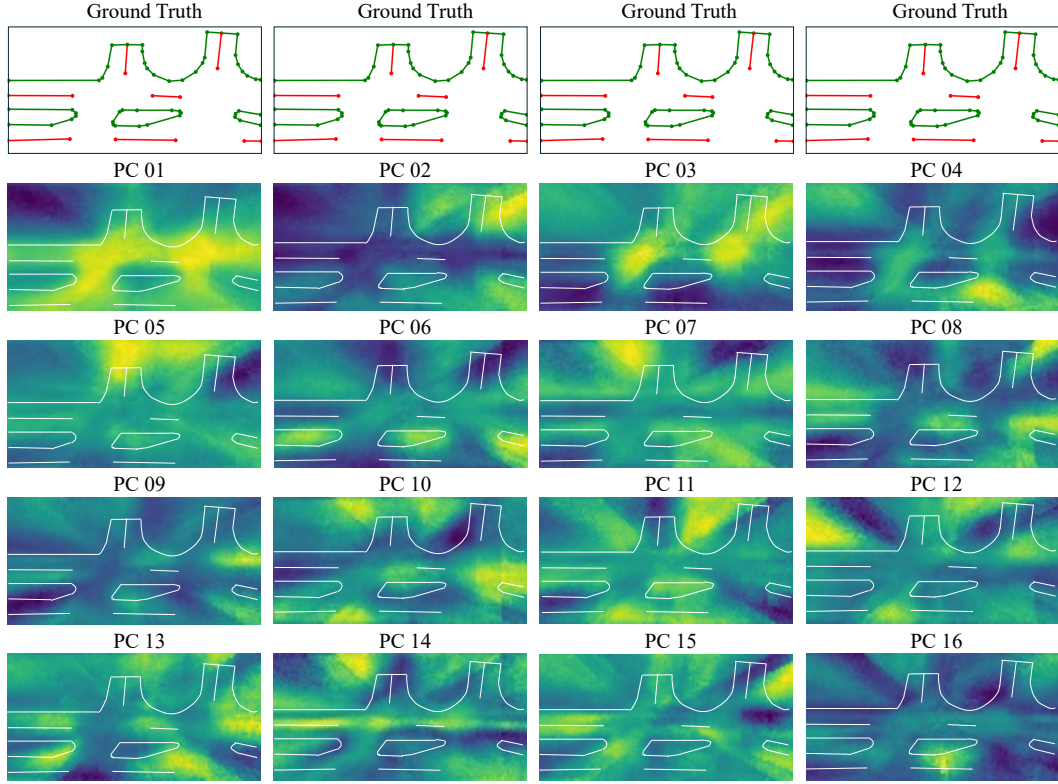


(a) StreamMapNet

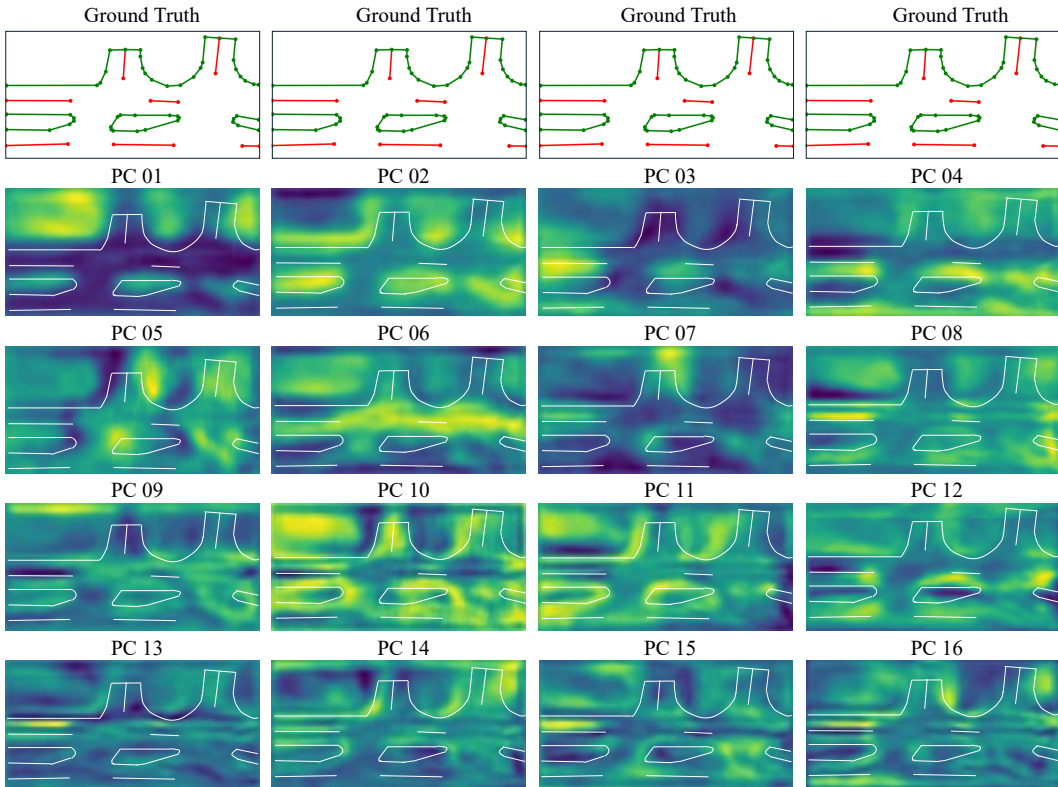


(b) AugMapNet

Figure 10. Example scene 1: Top 16 Principal Components of latent BEV grid of (a) StreamMapNet and (b) AugMapNet. Vector map ground truth is visualized at the top of each column as well as overlaid as white lines to ease the assessment of spatial accuracy. Notice that StreamMapNet tends to have radial artifacts.



(a) StreamMapNet



(b) AugMapNet

Figure 11. Example scene 2: Top 16 Principal Components of latent BEV grid of (a) StreamMapNet and (b) AugMapNet. Vector map ground truth is visualized at the top of each column as well as overlaid as white lines to ease the assessment of spatial accuracy. Notice that StreamMapNet tends to have radial artifacts.

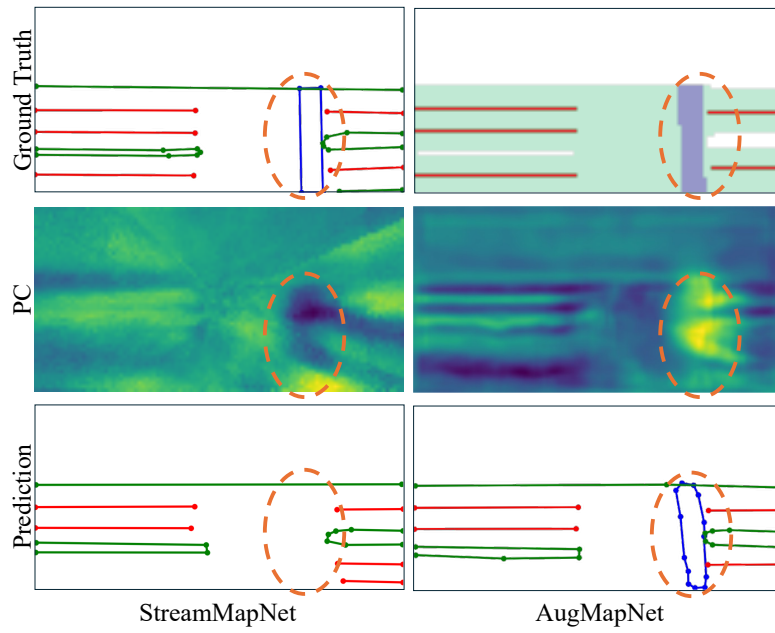


Figure 12. Extension of Fig. 4, visualizing example scene 1 where pedestrian crossing is missed by StreamMapNet but predicted by AugMapNet. PC is the Principal Component of the BEV latent grid that has the highest visual correspondence to the pedestrian crossing out of the top 16 Principal Components visualized in Fig. 10. It is PC 14 for StreamMapNet and PC 8 for AugMapNet, suggesting that this “crossing” PC is stronger in AugMapNet than StreamMapNet.