Appendices

A. Introduction

As part of the supplementary materials for this paper, we present our hyper-parameters and show more visual and quantitative results as an extension to the ones shown in the paper. The supplementary materials contain:

- An ablation study analyzing the performance of X-Align on various backbones.
- An ablation study to measure the adaptability of X-Align to camera noise.
- An ablation study to measure the impact of X-Align on different driving conditions and its adaptability to noise.
- Implementation details and training hyper-parameters for all our experiments.
- Qualitative results on nuScenes as an addition to the example shown in Figure 5 of the paper, and visual comparison with the baseline network.

B. Varying Backbones

Model	*Encoder*	*Modality*	*mIoU*
BEVFusion [9]	Swin-T	Camera, LiDAR	62.7
X-Align$_{\text{view}}$	Swin-T	Camera, LiDAR	64.1
BEVFusion	ConvneXt-T	Camera, LiDAR	62.1
X-Align$_{\text{view}}$	ConvneXt-T	Camera, LiDAR	63.8
BEVFusion	ConvneXt-S	Camera, LiDAR	63.9
X-Align$_{\text{view}}$	ConvneXt-S	Camera, LiDAR	**64.7**

Table B.1: **Quantitative evaluation on nuScenes** in terms of mIoU, varying the camera encoder.

To verify the generalizability of our X-Align method, we present results for three different encoders, *i.e.*, SWin-T, ConvneXt-T, and ConvneXt-S, in Table B.1. For this ablation study, we choose our **X-Align**$_{\text{view}}$ variant because this shows that a given model can be improved without adding additional computational complexity during inference. Compared to the current state-of-the-art result from BEVFusion [9], we can improve their result from 62.7 to 64.1 in terms of mIoU. Similar improvements can be observed for our additionally investigated backbones ConvneXt-T and ConvneXt-S. These results show that our method generalizes well to different backbones without hyperparameter tuning.

C. Adaptability to Noise

Model	*Encoder*	*Modality*	$\sigma = 0.0$	$\sigma = 0.05$	$\sigma = 0.075$	$\sigma = 0.1$
BEVFusion [9]	Swin-T	Camera	56.6	52.7	47.2	40.7
X-Align$_{\text{view}}$	Swin-T	Camera	**58.0**	**55.2**	**51.4**	**45.6**
BEVFusion [9]	Swin-T	Camera, Lidar	62.7	59.2	54.8	48.9
X-Align$_{\text{view}}$	Swin-T	Camera, Lidar	64.1	62.1	58.3	54.1
X-Align$_{\text{all}}$	Swin-T	Camera, Lidar	**65.7**	**64.3**	**62.6**	**59.6**

Table C.1: **Quantitative evaluation on nuScenes**, adding gaussian noise to the input images. We observe that X-Align methods provide more consistent results as they are lesser variant to noise.

Another interesting property of deep neural network-based methods is their susceptibility to input perturbations because it gives insights into their robustness in real environments. Therefore, we add Gaussian noise to the input camera images and observe how the performance degrades in Table C.1. The noise is zero-mean normalized, and σ in Table C.1 is the standard deviation of the Gaussian noise. From the table, we can deduce two main observations: First, the performance of X-Align methods under input perturbations is relatively stable compared to the baseline. We attribute this to the fact that the input

camera noise will make the extracted camera feature less reliable. Due to our robust attention-based cross-modal feature fusion (X-FF), the method can still correct the camera features using LiDAR predictions. Our second interesting observation concerns the comparison to the baseline BEVFusion. We observe that our performance loss at a higher noise-variance is much smaller than for BEVFusion, e.g., the performance of our method **X-Align**$_{\text{all}}$ at $\sigma = 0.1$ drops from 65.7 to 59.6. In contrast, BEVFusion drops from 62.7 to 48.9. This shows superior properties of our method in terms of robustness.

D. Adverse Weather Conditions

Model	*Encoder*	*Modality*	*nuScenes-Night*	*nuScenes-Rain*	*nuScenes*
BEVFusion [9]	Swin-T	Camera	30.8	50.5	56.6
X-Align$_{view}$	Swin-T	Camera	**33.1**	**51.1**	**58.0**
BEVFusion [9]	Swin-T	Camera, LiDAR	43.6	55.9	62.7
X-Align$_{\text{view}}$	Swin-T	Camera, LiDAR	44.5	56.3	64.3
X-Align$_{\text{all}}$	Swin-T	Camera, LiDAR	**46.1**	**57.8**	**65.7**

Table D.1: **Quantitative evaluation on nuScenes-rainy and nuScenes-night**. We observe that X-Align provides consistent improvements in adverse weather conditions.

To further investigate our method's robustness, we show results of how our method performs in adverse weather conditions in Table D.1. For this experiment, we use the metadata provided by nuScenes, which indicates the weather and light conditions of the recorded scene. With this information, we filter the validation set to obtain two splits containing all images recorded at night and all images recorded during rainy conditions. Our results show consistent improvement in these adverse conditions by **X-Align**$_{\text{view}}$ as well as **X-Align**$_{\text{all}}$. This shows that our method improves on comparably easy samples and difficult ones, which is an important property for future deployment of our method.

E. Training Details and Hyperparameter Analysis

In this subsection, we provide the hyper-parameters and training details for all our experiments in the paper. All our models are trained using Pytorch [11] on 4 Nvidia Tesla A100 GPUs.

nuScenes experiments: To reproduce the BEVFusion [9] baseline model, we adapt the code provided by their authors[1] into the mmdetection3d [3] framework. This is because they do not provide training code in the current version. In the camera pipeline, the images are downsampled into a size of 256×704 before being passed through a Swin-T [7] or ConvNext [8] backbone pretrained on ImageNet [13]. The intermediate features extracted from the camera are passed through a set of FPN [6] layers to retain the salient low-level encoder features. These are then passed to View-Transformers based on LSS [12]. The baseline is trained using their reported hyper-parameters [9], with a learning schedule of 20 epochs and a cyclic learning rate, starting for $1e^{-4}$ and performing a single cycle with target ratios $\{10, 1e^{-4}\}$ and a step of 0.4. For our experiments, we implement our proposed blocks:

- **X-FF Modules**: For the X-FF modules described in Section 3.3 of the paper, we switch the naive convolutional fuser in the baseline model. To implement the Self-Attention fuser, we tokenize the inputs into patches of size 3×3 and a stride of 2. This step is followed by a Multi-Head-Self-Attention block as described in [15], containing 8 heads and an embedding dimension of 256. To implement the Spatial-Channel Attention module, we use the Split-depth Transpose Attention (SDTA) block from EdgeNext [10].We use an embedding dimension of 256, 2 scales, and 8 heads. We also use DropPath [5] of 0.1. A deconvolution block follows the SDTA fusion module to output 256 channels and the fused feature resolution. Finally, to implement the pose-driven deformable convolutional operation, we pass the input pose through a 2-layer MLP network, whose output can is interpolated to the feature size. This pose channel is concatenated with the two input modalities, then passes through a convolutional block to obtain 18 offset channels, input into a DCNv2 [17] block of kernel size 3×3 The output of the DCNv2 block is our fused feature dimension.

- **X-FA Loss**: As explained in Section 3.4 of the paper, we use cosine similarity as a loss function to model the X-FA similarity. Using a sparse hyper-parameter search, we set the best value of γ_2 in Equation (2) of the main paper as -0.002. We reduce its negative value because we want to increase the cosine similarity.

[1] https://github.com/mit-han-lab/bevfusion

- **X-SA Module**: As explained in Section 3.5 of the paper, we utilize a perspective decoder to generate 2D perspective view semantic labels. The decoder has two Convolution-BN2D-ReLU sequences with a hidden dimension of 256 channels, generating semantic probabilities at the output. As nuScenes [2] does not contain ground truth segmentation labels; we make use of a state-of-the-art (SOTA) model pre-trained on Cityscapes to generate pseudo labels $\boldsymbol{y}^{\mathrm{PV}}$. This is an HRNet-w48 [16] checkpoint, trained with the InverseForm [1] loss, as made public by the authors in their codebase.[2] We pick γ_3 in Equation (2) of the main paper as 0.1 following a sparse hyper-parameter search. Once we have the 2D perspective features, we share parameters with the LSS Transform from earlier to splat the probability maps to BEV space. We use convolution blocks for channel aggregation at the input and output and interpolate to equalize dimensions. The output is then supervised with BEV GT as shown in Figure 3 of the paper. The loss weight γ_4 is tuned to 0.1 by observing the total loss value and setting the weight to produce $\sim 20\%$ of the total loss.

KITTI360 experiments: To reproduce the PanopticBEV [4] scores, we use the code made public by the authors and their default settings.[3] The model consists of an EfficientDet-d3 [14] camera encoder, which encodes multi-resolution features. Next, a multi-scale transformer converts the perspective view to BEV features. This is followed by decoders for both instance and semantics. Finally, their outputs are combined to generate panoptic predictions. The baseline uploaded by the authors reaches a lower value than what was obtained in their paper. For our experiments with the **X-SA module**, we added a 2D Perspective decoder to the multi-scale encoded features to predict semantic labels in the perspective view. As KITTI360 contains 2d semantic labels, we use them to supervise this decoder. Then, we reuse the view transformer to map predictions to BEV space. We supervise our X-SA branch with loss weights γ_3 set at 0.5 and γ_4 set at 0.1, adding them to the total loss of the PanopticBEV baseline.

F. Qualitative Analysis

In this Section, we present more sample scenes from nuScenes, as an extension to the one shown in Figure 5 of the main paper. Each scene consists of 5 parts: a) six surround camera inputs b) LiDAR scan, c) ground-truth BEV segmentation map, d) baseline BEV segmentation, e) BEV segmentation using **X-Align**$_{\mathrm{view}}$, and d) BEV segmentation **X-Align**$_{\mathrm{all}}$. For each scenario, we observe that the baseline model prediction is highly erroneous in the region highlighted in green. We highlight this region of interest in the input views as well. By using our proposed X-SA losses, **X-Align**$_{\mathrm{view}}$ can already correct substantial errors in the baseline prediction, and the **X-Align**$_{\mathrm{all}}$ model further improves accuracy.

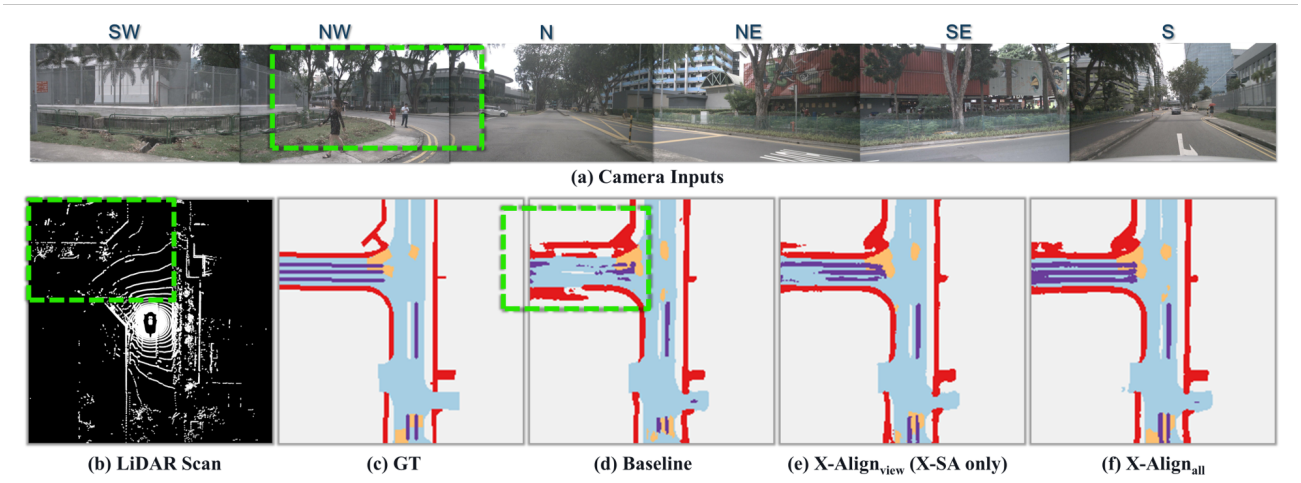

Figure F.1: **Scenario 1 on nuScenes:** We present a scene where multiple road occlusions are apparent in the N and NW camera images, which were caused by pedestrians. The baseline model fails to fill in the gaps properly through the LiDAR-camera fusion, producing an entangled segmentation representation of the intersecting roads. Using the two X-SA losses, **X-Align**$_{\mathrm{view}}$ improves the segmentation map's prediction. By adding all the components, **X-Align**$_{\mathrm{all}}$ can address the occlusions and produce a more refined semantic representation of the scene.

[2]https://github.com/Qualcomm-AI-research/InverseForm
[3]https://github.com/robot-learning-freiburg/PanopticBEV

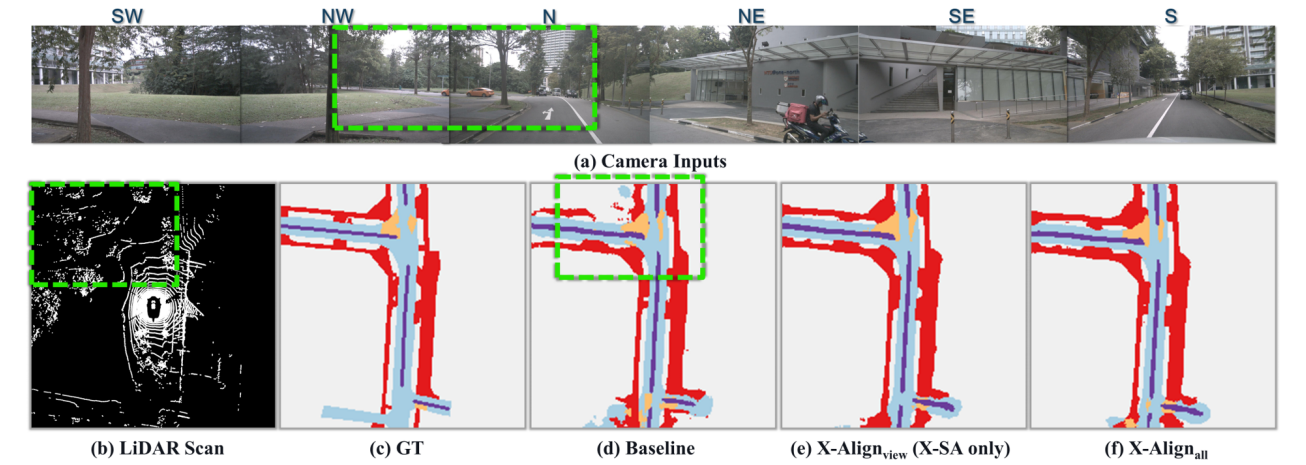

Figure F.2: **Scenario 2 on nuScenes:** In this scene from nuScenes, the road boundaries in the north west direction of the vehicle appear vague in the LiDAR point cloud's green box area. However, in the N and NW camera images, they are apparent. The baseline model's boundary prediction is faulty due to misaligned camera features. Using the two X-SA losses, **X-Align**$_{view}$ rectifies some of the camera features. By adding all the components, **X-Align**$_{all}$ can predict a map where the road boundaries are properly segmented.

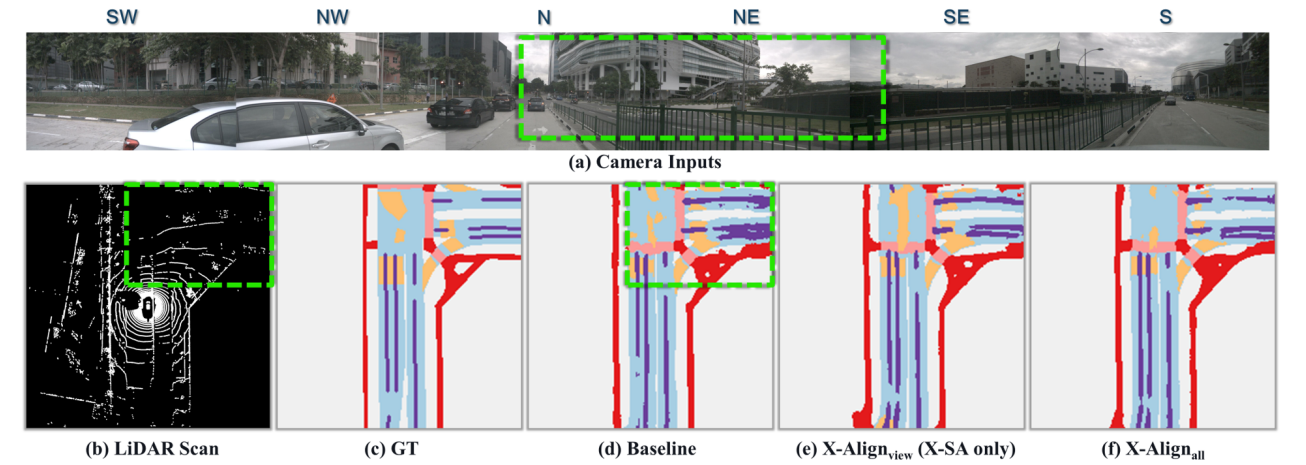

Figure F.3: **Scenario 3 on nuScenes:** We present a scene where patterned occlusions like fences are evident in the N and NW camera images. Due to these repetitive occlusions, the baseline model has difficulty producing a clear segmentation map of the intersection. By using the two X-SA losses, **X-Align**$_{view}$ can extract more BEV-segmentation-oriented features from the images. By adding all the components, **X-Align**$_{all}$ can properly rectify the camera's incomplete view of the road and align it properly with the LiDAR's features, producing a clear segmentation map of the road as shown in (f).

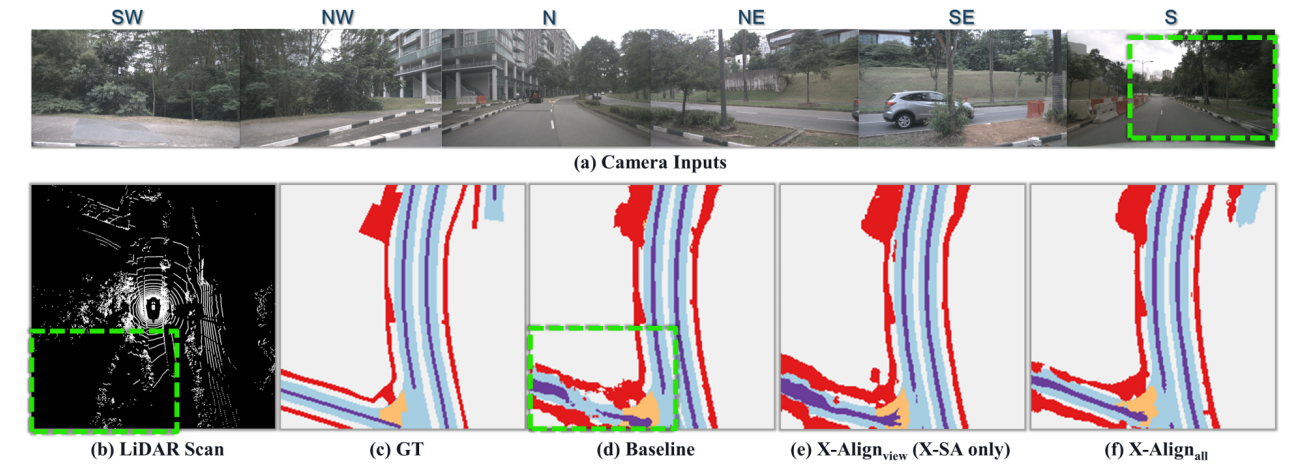

Figure F.4: **Scenario 4 on nuScenes:** We present a scene from nuScenes, where the back intersection is dimly lighted with slightly occluded road boundaries, as shown in the S camera image. The LiDAR point cloud input lacks salient features capturing the road boundaries. We show the baseline's erroneous boundary prediction due to their simple concatenation-based fusion between the two modalities. We also show that by using two X-SA losses, **X-Align**$_{\text{view}}$ can have a more accurate prediction of the road boundaries, and by additionally using X-FF and X-FA, **X-Align**$_{\text{all}}$ can output an accurate segmentation map of the intersection.

References

[1] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proc. of CVPR*, pages 5901–5911, 2021.

[2] 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 *Proc. of CVPR*, pages 11621–11631, 2020.

[3] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020.

[4] Nikhil Gosala and Abhinav Valada. Bird's-Eye-View Panoptic Segmentation Using Monocular Frontal View Images. *IEEE RA-L*, 7(2):1968–1975, 2022.

[5] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. of CVPR*, pages 2117–2125, 2017.

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proc. of ICCV*, pages 10012–10022, 2021.

[8] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proc. of CVPR*, pages 11976–11986, 2022.

[9] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. *arXiv preprint arXiv:2205.13542*, 2022.

[10] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications. *arXiv preprint arXiv:2206.10589*, 2022.

[11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[12] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In *Proc. of ECCV*, pages 194–210, 2020.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet Large Scale Visual Recognition Challenge. *Proc. of IJCV*, 115(3):211–252, 2015.

[14] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proc. of CVPR*, pages 10781–10790, 2020.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proc. of NIPS*, pages 5998–6008, Dec. 2017.

[16] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep High-resolution Representation Learning for Visual Recognition. *Proc. of PAMI*, 43(10):3349–3364, 2020.

[17] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable Convnets V2: More Deformable, Better Results. In *Proc. of CVPR*, pages 9308–9316, 2019.