Appendix

A. Further Implementation Details

In this section, we further elaborate on the implementation details of our LCPS. Section 3.1 in the main paper explains that the LiDAR branch consists of point and voxel streams. We employ three MLP layers for the point stream to extract point-level features with an output dimension of 64, 128, and 256 channels. Following processing by ACPA and SARA, the fused point features are compressed to 16 channels to match the voxel features. As for the voxel stream, the cylindrical encoder maps original point features (XYZ-axis, remission, reflections, etc) to 16 dimensions. After PVP module, features from the voxel stream and point stream are merged into cylinders and then fed into Cylinder3D [42] backbone network. In Cylinder3D, four layers of down-sampling 3D convolutions with BatchNorm and ReLU activation are applied to the fused voxel features, transforming the channel numbers of voxel features to 32, 64, 128, and 256, respectively. Finally, the voxel feature dimension is compressed back to 128 after pooling layers and remains at this dimension in the subsequent four up-sampling layers of Cylinder3D.

Regarding the image branch, images from multiple cameras are concatenated and scaled to 0.4 of the original size. The SwiftNet-18[37] network comprises four pairs of down-sampling and up-sampling layers. The four down-sampling layers transform features maps to $64 \times 320 \times 180$, $128 \times 160 \times 90$, $256 \times 80 \times 45$, and $512 \times 40 \times 23$, respectively. Then, a multi-scale spatial pooling module [10] is utilized to compress the four feature maps to 128 channels. Eventually, the other 4 up-sampling layers symmetrically up-sample the feature map to the input size for following geometric-consistent and semantic-aware alignment and fusion.

B. Further Discussion

Visual Ablations on Asynchronous Compensation. Here we further provide qualitative comparisons of asynchronous compensation in Figure 7. The first and the second lines are visual results without or with asynchronous compensation respectively. The leftmost three columns demonstrate the effectiveness, especially for foreground objects of various sizes and distinct geometric shapes. For instance, the most apparent improvement is exhibited in the second leftmost column, where few points can be mapped to distant and marginal trucks, greatly enhancing the robustness of LiDAR-Camera fusion. We also demonstrate a typical failed case here to illustrate the limitation. When the ego-vehicle slows down, or objects come at the front or back view, it is possible that the asynchronous compensation almost makes no difference because the time gap or the changes of view angles is small.

Discussions on Time and Memory Cost. We compare the time and memory cost in Table 5 with other SOTA approaches if their projects are open-source or if such information is provided in their papers. Our LCPS full model is slightly slower than LiDAR-only methods (including our LiDAR-only baseline). Interestingly, adding image branches does not essentially drop the FPS since we choose lightweight ResNet-18 as the image backbone. The parameter number of our LiDAR-only baseline is high since we replace the backbone BEV U-Net in Panoptic-Polarnet [41] with Cylinder3D [42]. Similarly, the parameter number of DS-Net [11] is also above 50M since it adopts Cylinder3D as the backbone network too.

	FPS(Hz)	Params(M)
DS-Net [11]	3.2	56.5
Panoptic-PolarNet [41]	11.6	13.8
Panoptic-PHNet [19]	11.0	-
LCPS(LiDAR-Only)	8.6	65.9
LCPS(Full)	8.3	77.7

Table 5. Results of FPS and parameter scales on SemanticKITTI.

More In-Depth Analysis on Metrics. We provide further in-depth analysis of our experimental results. It appears that some *Stuff* metrics, e.g., PQ^{st} and mIoU, are slightly lower than Panoptic-PHNet (0.3%-0.7%) on NuScenes val. and test set. However, compared to the LiDAR-only baseline, our approach boosts the performance in terms of PQ^{st}, SQ^{st}, RQ^{st}, and mIoU consistently. This phenomenon reflects that the improvement on *Stuff* is not as noticeable as *Thing* objects and further indicates that the LiDAR-Image fusion has more benefits on *Thing* objects. The possible reason is that *Thing* objects often have fewer points than *Stuff*; thus, images may provide more crucial information for the former.

Our experimental results do not obviously surpass SOTA methods since our baseline is relatively weak. Our baseline project is built on the current highest open-source benchmark, Panoptic-Polarnet [41], while other SOTA methods have not released their codes yet. We reproduce the Panoptic-Polarnet and get 67.7% PQ on NuScenes, and 55.7% PQ on SemanticKITTI. Then we further improve the NuScenes baseline to 72.9% using data augmentations as stated in Section 4.2. Based on this baseline, our fusion strategy obtains +6.9%, +6.7%, and +3.3% PQ improvement on NuScenes validation, NuScenes test, and SemanticKITTI validation set, reported in the main content. After submission, we improve our baseline on SemanticKITTI by using rare *Stuff* copy-paste augmentation, demonstrating higher overall performances, as shown in Table 7. We provide the improved version here for additional reference and analysis. The fusion improvement on SemanticKITTI is lower than NuScenes since SemanticKITTI has only two front-view cameras; thus, the number of matched points is lower than NuScenes, as illustrated in

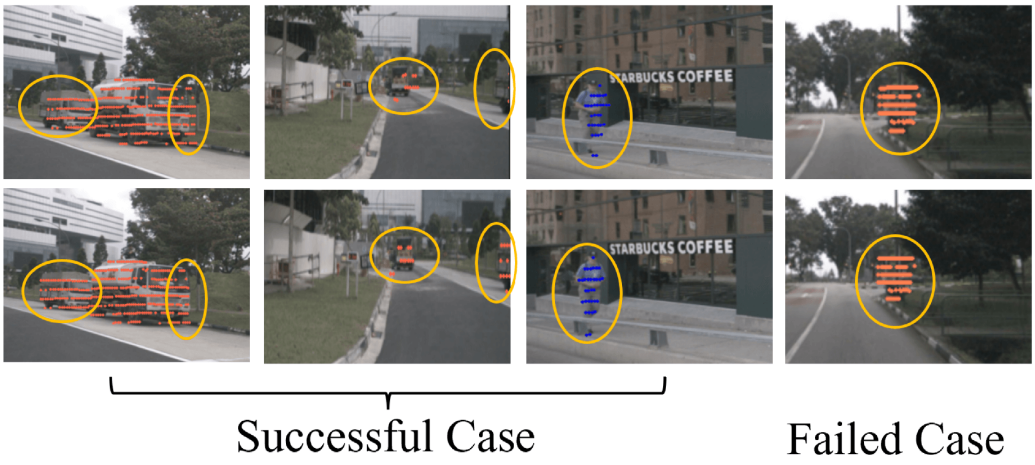

Figure 7. Visual comparisons of asynchronous compensation. The first and the second lines are visualizations without or with asynchronous compensation respectively. The leftmost three columns demonstrate the effectiveness, especially for foreground objects of various sizes and geometric shapes. The last column specifies that the asynchronous compensation almost makes no difference when the time gap is small or at the front view.

Table 6. Additionally, the improvement on a higher SemanticKITTI baseline is lower than the original one because heavy data augmentation (we add extra rare stuff augmentation in order to achieve a higher baseline) may diminish the benefits of LiDAR-Camera fusion, which is also reported in previous research on detection tasks like PointAugmenting [35].

As for mIoU, it is mainly evaluated for semantic segmentation, and we include it following previous research [19, 41]. Better PQ simultaneously needs better semantic segmentation ability (mIoU) and instance segmentation quality. Therefore, a model of high mIoU may perform worse in terms of PQ. Our method can achieve comparable mIoU performance with SOTA Panoptic-PHNet in NuScenes, while worse in SemanticKitti due to the weak baseline issue. Besides, our fusion strategy consistently improves PQ and mIoU in both NuScenes and SemanticKitti datasets compared to the LiDAR-only baseline.

	NuScenes	SemanticKITTI
Matched Points	17182	39780
Total Points	34720	120387
Percentage	52.2 %	33.2 %

Table 6. Statistics on the averaged number of points matched to images per frame.

Methods	Improved (Val.)		Improved (Test.)	
	PQ	mIoU	PQ	mIoU
LCPS(LiDAR-Only)	60.6	66.8	57.8	62.0
LCPS(Full)	61.4	67.5	58.8	62.8

Table 7. Results of the fusion methods on the improved baseline of SemanticKITTI.

Ablation Study on Perception Distance. As our backbone network adopts a cylindrical voxel representation, we need to set the perception distance of the scene volume, which is defined as the radial distance from the LiDAR sensor to objects or points. Setting the perception distance too close or too far is sub-optimal for training because a close distance setting may miss some small objects far away and diminish PQ performance, while a far distance setting may involve more noise points and disturb training stability.

In our experimentation on the NuScenes validation set (as shown in Table 8), we find that as the perception distance increases, the performance initially improves and then declines. The result shows that ± 100 meters and ± 120 meters yield the highest PQ scores, while ± 80 meters produce the best mIoU. Intuitively, ± 80 meters can be the valid distance at which the LiDAR sensor is able to accurately detect objects in NuScenes, while approximately ± 150 meters is the farthest perception distance. Based on these findings, we ultimately choose ± 100 meters as the moderate perception distance for NuScenes.

Correction Ability. When we review the visualization results, one interesting observation is that our model appears to correct segmentation errors in the ground-truth labels. Due to the utilization of bounding boxes and semantic labels in rule-based scripts for automatically generating panoptic labels, errors in ground-truth labels are commonly observed. Hence, it is essential for our network to boost more generalizability and avoid overfitting with ground-truth labels during the training process.

As illustrated in Figure 9, our LCPS network is capable of correcting ground-truth errors by preserving the intact shape of an instance. Figure 9 (a) demonstrates that the predicted truck segmentation retains the top edge area since it is spatially and geometrically proximate to the truck segments below. Similarly, in a sequence of frames in Fig-

Distance (±, [meters])	50	60	70	80	90	100	110	120	130	NFV
PQ [%]	70.6	72.2	72.5	72.8	72.8	**72.9**	72.2	**72.9**	72.3	70.5
mIoU [%]	74.3	74.6	74.0	**75.2**	74.5	75.1	74.4	74.5	74.4	73.7

Table 8. The ablation of perception distance on NuScenes validation set. The experiment is tuned on our LiDAR-only baseline network. NFV represents No Fixed Volume, which means we select the farthest point (usually $\geq$ 170m) as the distance for each LiDAR scan rather than a fixed distance.

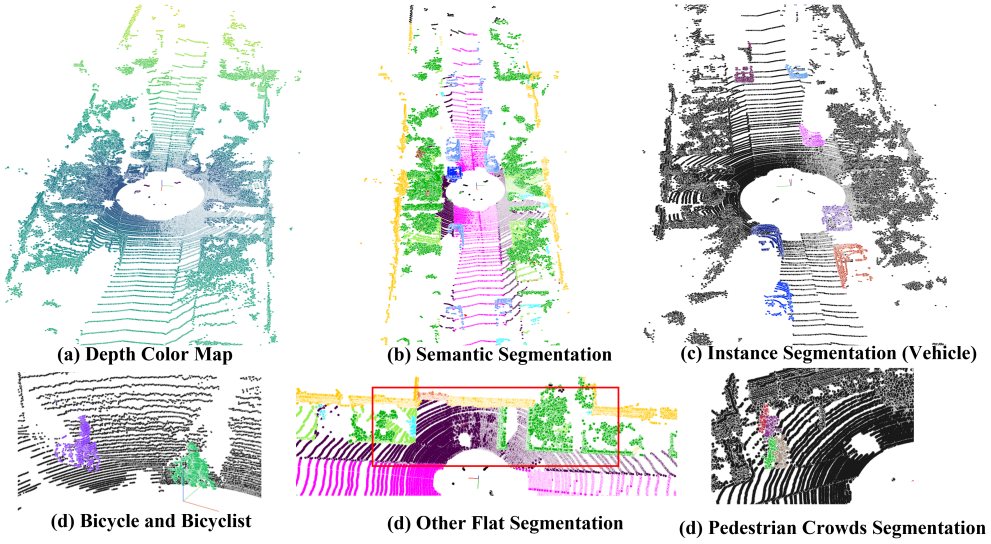

Figure 8. Visualization results of SemanticKITTI validation set.

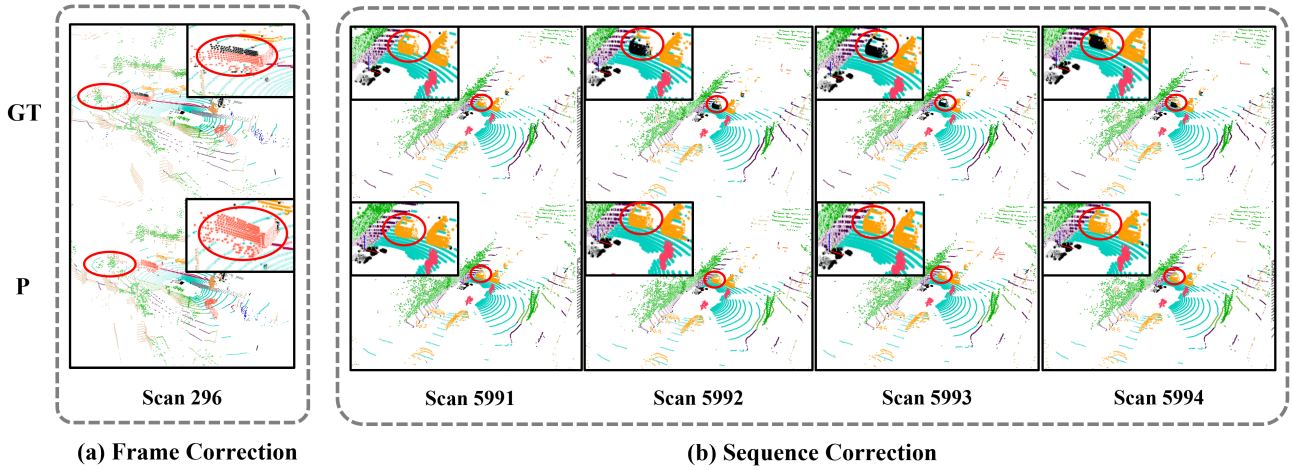

Figure 9. The visualization of correction ability between ground-truth (GT, first line) and our model predictions (P, second line). The results are from the NuScenes validation set, where the scan number below represents the sample index and red circles highlight notable differences. (a) shows that our model can segment an intact segmentation of truck, even though the ground-truth labels overlook the top area. (b) demonstrates that our model can consistently segment an entire vehicle during a sequence of frames over time, while the ground-truth labels miss the back surface when the ego-car advances at a high speed.

ure 9 (b), the rule-based ground-truth label generation results in the absence of the back surface of the vehicle, as the

ego-car advances at high speed. Nevertheless, our network still maintains consistency in predicted segmentation over time, which serves as compelling evidence that our network obtains robust feature representation of objects by leveraging LiDAR and image features, enabling it to correct false ground-truth labels.

Further Qualitative Results. We provide further visualization results on the validation set of NuScenes (Figure 10 and Figure 11) and SemanticKITTI (Figure 8) dataset to detailedly demonstrate the panoptic segmentation ability of our network. In Figure 10, we display the objects whose perspective projections are within the single image and compare ground-truth labels and predictions in 3D and perspective view. Our network effectively recognizes small objects (such as *bicycle* and *motorcycle*) and rare objects (such as *trailers* and *construction vehicle*). Especially in the right-construction vehicle sub-figure, our segmentation quality is slightly better than ground-truth labels at the position of robotic arms. Regarding Figure 11, we compare the visualization results of objects across multiple images. We primarily select challenging scenarios such as crowding (pedestrian and car) and severe occlusion (truck). For example, the truck segmentation is largely occluded by walls, which severely damages the geometric structure in LiDAR scenes and feature completeness in images. Under such conditions, our network can correctly segment most of the truck instances while missing one truck only (which is occluded by the orange construction vehicles). Moreover, for pedestrian segmentation, our network additionally segments two more occluded figures in the middle image column, although it wrongly recognizes two tiny figures as one person in the leftmost image column.

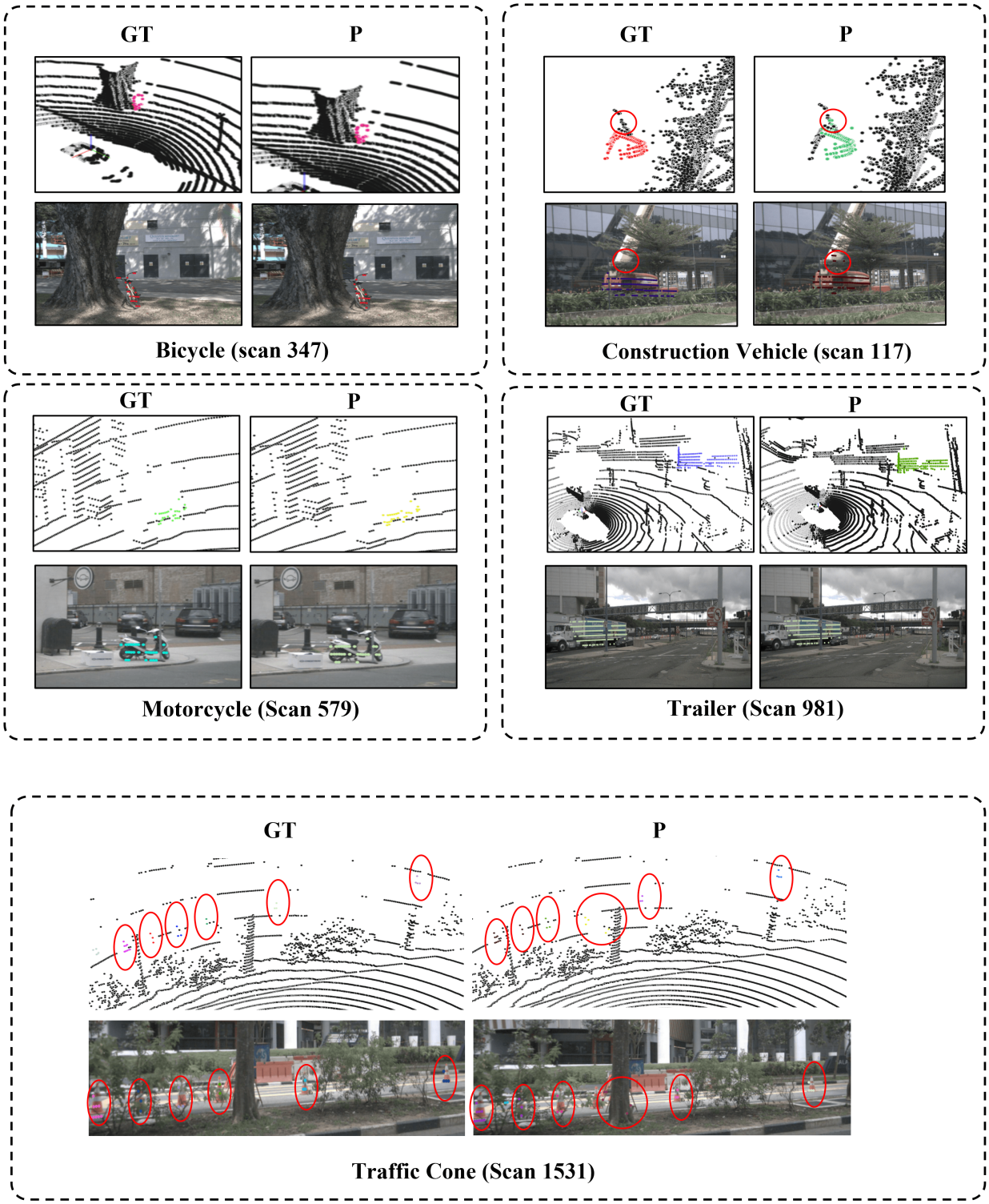

Figure 10. Visualization results of foreground objects. GT represents ground-truth labels, while P represents predictions of our LCPS. Text such as "Back" and "FRONT_LEFT" refers to the specific camera sensor. In this figure, the perspective projections of object segmentation are within the same image. Generally, our network achieves accurate segmentation results over these small and distant objects, such as *bicycle* and *motorcycle*, or rare objects like *construction vehicle*.

Figure 11. Visualization results of foreground objects. GT represents ground-truth labels, while P represents predictions of our LCPS. Texts like "Back" and "FRONT_LEFT" refer to the specific camera sensor. This figure shows that most objects of diverse types and spatial locations across images can be consistently identified.