CycleBEV: Regularizing View Transformation Networks via View Cycle Consistency for Bird's-Eye-View Semantic Segmentation

Supplementary Material

6. Implementation Details

In this section, we provide the detailed information about our experiments in Sec. 4. Our implementation code is also provided along with this supplementary material for better understanding. Note that the code will be made public once it is accepted for publication.

6.1. IVT Network

Pseudo Label Generation nuScenes partially provides GT PV segmentation maps for multi-view images, whereas our framework requires GT for all of them. To address this issue, we first train Mask2Former [5] on the available labels. We then use the trained model to generate PV map predictions for all multi-view images in nuScenes, which serve as pseudo labels. Figure 6 illustrates examples of GT PV maps and pseudo labels generated by the trained Mask2Former, along with their corresponding input images. From now on, we refer to the pseudo label as the GT label for simplicity.

Pre-training We train the proposed IVT network on the pairs of GT BEV map and the corresponding GT PV maps using BCE loss. The loss weights for *drivable area*, *vehicle*, and *pedestrian* are empirically set to 0.03, 0.5, and 1, respectively, reflecting the number of pixels in the BEV maps assigned to each category. AdamW [21] is used for the optimization with an initial learning rate of $4 \cdot 10^{-4}$ and batch size of 4 for 24 epochs.

Fine-tuning When training VT models under the regularization of the IVT network, we also fine-tune the pre-trained IVT network using AdamW with an initial learning rate of $4 \cdot 10^{-4}$. Specifically, when it receives GT BEV maps, we add random Gaussian noise to the inputs so that the IVT network learns to better handle noisy predictions from the VT models.

6.2. Proposed Framework

Regularization through IVT Network Given the pre-trained IVT network, we jointly optimize the four baselines—LSS [24], CVT [38], PETRv2 [20], and BEVFormer [16]—together with the IVT network under the proposed framework. Note that to enable the baseline models to jointly predict the three semantic categories (*drivable area*, *vehicle*, and *pedestrian*) and the height map, we modify CVT's decoder so that the overall decoder architecture includes two sub-decoders—one for the semantic categories and the other for the height map—and use this modified decoder for all four models. The loss weights for *drivable area*, *vehicle*, and *pedestrian* are empirically set to 0.03, 0.5, and 1, respectively, reflecting the number of pixels in the BEV maps assigned to each category. The hyperparameters in Eqn. 5 are empirically set to $\lambda_1 = 1.0$, $\lambda_2 = 10^{-3}$, $\lambda_3 = 0.4$, and $\lambda_4 = 1.0$. We train the models using AdamW with a batch size 2 for up to 50 epochs. For parameters related to optimization (e.g., initial learning rate), we strictly follow the original implementations of the four baselines. Unless otherwise stated, the same settings are applied to the later experiments.

Object Height Map Generation The height map $\mathbf{H}$ in Eqn. 3 has the same spatial resolution as $\mathbf{O}$, where each of its pixels has a value in $[0, 1]$, representing the normalized height of the moving object occupying that pixel. Pixels corresponding to road elements are set to 0. We obtain height information for moving objects from their ground-truth 3D bounding boxes and normalize it by dividing by 5, which is considered the maximum height (in meters) of a moving object. If the normalized height of a pixel exceeds 1, we clip it to 1.

6.3. Existing VCC-based Frameworks

To implement the VCC-based learning frameworks proposed by CVTM [34] and FocusBEV [36], we let the VT encoders of four baselines and the proposed IVT encoder serve as **PV2BEV** and **BEV2PV** in Fig. 1, respectively, with minor adjustments to the IVT network. For CVTM, following the original implementation, we introduce the cycle loss $\mathcal{L}_{cycle} = ||\mathbf{X} - \mathbf{X}''||_1$, where $\mathbf{X}$ and $\mathbf{X}''$ respectively denote PV image features and those recovered by **BEV2PV**. In contrast, since FocusBEV does not explicitly enforce cycle consistency via a cycle loss, we optimize it using only the binary cross entropy loss.

6.4. BEV Map Auto-Encoder

The BEV autoencoder (AE) used for Tab. 4 consists of an encoder and a decoder. The encoder takes the BEV map $\mathbf{O}$ and the height map $\mathbf{H}$ as inputs, while the decoder reconstructs these maps from the high-dimensional representations $\{\bar{\mathbf{B}}_s\}_s$ produced by the encoder. The encoder comprises a series of **ResBlock**, each consisting of *Conv*, *BatchNorm*, *MaxPool*, and *ReLU* layers. As the input passes through multiple **ResBlock**s, the spatial resolution decreases. The decoder has a similar architecture to the encoder, except that (1) it includes multiple skip connections from the encoder, and (2) *DeConv* layers replace *MaxPool* layers. AdamW is used with an initial learning rate of

Model	*Driv.*	*Veh.*	*Ped.*	Avg.
Single-branch	82.32	70.48	32.21	61.67
Dual-branch	81.82	68.91	28.27	59.67

Table 7. **PV segmentation performance of the proposed IVT network on nuScenes validation set.**

$5 \cdot 10^{-4}$, a batch size of 4, and a total of 10 epochs. During training, we add randomly generated Gaussian noise to $\bar{\mathbf{B}}_s$ of the smallest resolution to make the AE robust to noise following [37].

6.5. Input Data Augmentation

We randomly rotate (from $-1°$ to $1°$), resize (scaling from 0.8 to 1.2), and crop (up to 20% of the image area) the input images to VT models, and adjust the corresponding camera parameters following [7, 11]. However, we do not modify the corresponding PV segmentation maps accordingly because they are recovered from the BEV maps predicted by the VT models.

6.6. Semantic-Aware Image Backbone Training

Following [35], we jointly optimize the image backbones of the four baselines within the PV segmentation task during training. To this end, we devise a PV segmentation decoder, which has a similar architecture to the decoder of UNet [26]. We further apply soft-thresholding to the backbone image features: each feature is scaled by the corresponding PV segmentation logits (after applying a sigmoid for normalization). The resulting scaled features are then used as keys and values in the self- and cross-attention layers.

7. Further Analysis

7.1. IVT Network Design

Figure 7 shows PV segmentation maps predicted by the proposed dual-branch IVT network. Given that the positions of objects and road shapes in PV space are well reconstructed from the BEV maps, it can be inferred that the IVT network effectively learns the reverse mapping function. However, it fails to recover the fine outlines of *vehicle* and *pedestrian* instances in PV space, as these objects are represented as rectangular regions in the BEV maps.

Table 7 reports the PV segmentation results of the proposed IVT network with the single- and dual-branch designs. The main difference between the two designs is that the former fuses the MR feature maps generated by the CNN encoder progressively in one IVT encoder, whereas the latter first processes high and low resolution feature maps separately through dedicated IVT encoders and then fuses the resulting features in the decoding process. From now on, we refer to the single- and dual-branch designs as *early* and *late* fusion architectures, respectively. The table

Model	VCC	Height	Align	*Driv.*	*Veh.*	*Ped.*	Avg.
CVT				76.80	31.41	10.89	39.70
	✔			77.20	32.84	12.73	40.90
	✔			76.99	32.65	12.00	40.55
	✔	✔		77.11	33.62	13.16	41.30
	✔	✔		77.12	33.76	13.32	41.40
	✔	✔	✔	77.23	34.14	13.65	41.67
	✔	✔	✔	77.40	34.24	13.69	41.78
BEVFormer				78.06	33.23	11.70	41.00
	✔			78.27	33.77	13.26	41.77
	✔			78.20	33.61	13.32	41.71
	✔	✔		78.11	34.35	13.06	41.84
	✔	✔		78.11	34.26	13.04	41.80
	✔	✔	✔	77.76	34.31	13.34	41.80
	✔	✔	✔	78.20	34.46	13.39	42.02

Table 8. **Ablation study on the effectiveness of our contributions.** *VCC*, *Height*, and *Align* refer to view cycle consistency, height prediction, and intermediate BEV feature alignment, respectively. Results highlighted in blue and red are produced using a single- and dual-branch IVT networks, respectively.

shows that the early fusion achieves better PV segmentation accuracy than its late fusion counterpart, leading to stronger regularization on VT models (see Table 8, baselines with VCC only). However, when the proposed auxiliary tasks are incorporated into training, the late-fusion design provides better regularization for the baselines. We speculate that the early fusion helps PV segmentation because the network gets a unified view of spatial and semantic cues. On the other hand, the late fusion keeps high- and low-resolution streams separate for longer. When the auxiliary tasks are added, that separation gives each branch room to specialize—one can focus on fine spatial detail, the other on semantic abstraction—before combining them. This diversity in representation may regularize the learning process better. The fact that the late fusion benefits from the intermediate BEV feature alignment (referred to as *Align* in the table) more than the early fusion does validates our claim.

7.2. Performance Gain under Temporal Setting

We apply the proposed regularization framework to BEVFormer [16] under a temporal setting, where the model is configured to leverage information from previous frames, to see how much performance gain can be achieved by the proposed framework on top of the temporal setting. Specifically, we train the model to take N_p previous frames in addition to the current frame as input. Since the key frames in nuScenes are sampled at 2 Hz, the model observes a $0.5 \times N_p$-second history. Table 9 shows the result. One noticeable finding is that the static model (BEVFormer-S) outperforms the temporal model (BEVFormer-T) on *drivable area*, which can also be observed in the original paper [16]. We speculate that, unlike *vehicle* and *pedestrian*, *drivable area* lacks distinctive characteristics that can be matched to those in previously observed scenes. With respect to *vehicle* and *pedestrian*, BEVFormer-T outperforms BEVFormer-S,

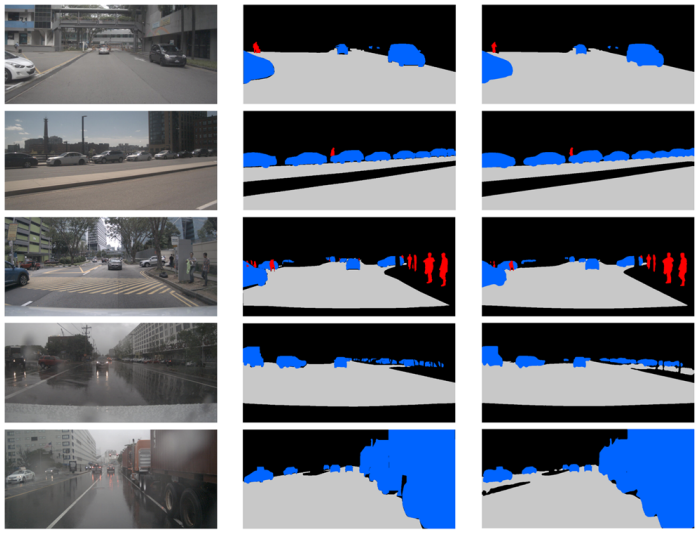

Figure 6. **Visualization of multi-view images (first column), the GT PV segmentation maps (second column), and the pseudo labels generated by Mask2Former (third column).** *Drivable area*, *vehicle*, and *pedestrian* are color-coded with gray, blue, and red, respectively.

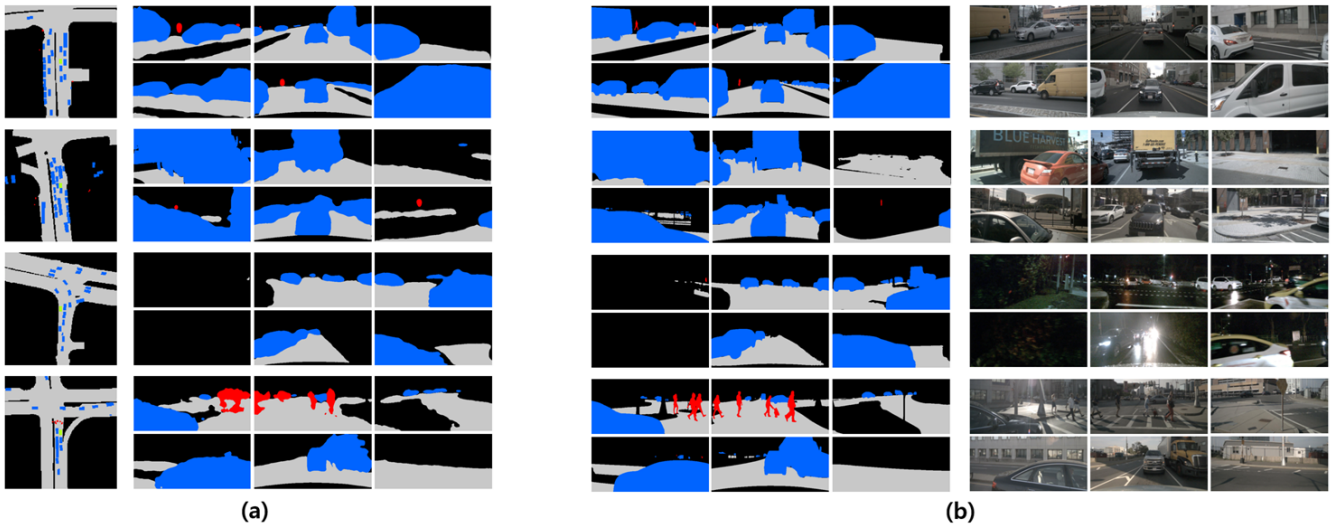

Figure 7. **Examples of PV segmentation map prediction results.** (a) GT BEV maps (left) and PV map predictions from the dual-branch IVT network (right). (b) GT PV maps (left) and their corresponding multi-view images (right). *Drivable area*, *vehicle*, and *pedestrian* are color-coded with gray, blue, and red, respectively.

which is consistent with the findings in the original paper. Finally, the proposed regularization framework consistently improves BEVFormer-T for all N_p values (see Δ_{Ours} in the table).

7.3. Comparison with Temporal Setting

In Table 10, we further compare the static model trained under the proposed regularization framework (BEVFormer-S+Ours) and the temporal model (BEVFormer-T) to show the effectiveness of the proposed framework. The values in red and blue are the performance gain over BEVFormer-S, respectively. It is shown that the performance gain achieved by the proposed framework (values in red) surpasses that achieved by the temporal information (values in blue) even though the proposed framework does not cause an increase in computational complexity and network size, whereas the temporal setting does. Finally, it is worth mentioning that comparing the two methods on *drivable area* is unfair because, as we discovered in Tab. 9, the temporal model performs poorly on the category.

Model	N_p	*Driv.*	*Veh.*	*Ped.*	Avg.
BEVFormer-S	0	78.06	33.23	11.70	41.00
BEVFormer-T	1	76.55	33.50	12.36	40.80
	2	76.55	33.47	12.69	40.90
	3	75.65	33.72	12.41	40.59
Δ_{Ours}	1	0.1	0.95	1.34	0.8
	2	-0.3	0.93	1.36	0.67
	3	0.84	1.18	1.56	1.2

Table 9. **Performance gain on BEVFormer with the temporal setting.** N_p denotes the number of previous frames the models take as input. Δ_{Ours} denotes the performance gain achieved by BEVFormer-T+Ours over BEVFormer-T.

Model	N_p	*Driv.*	*Veh.*	*Ped.*
BEVFormer-S	0	78.06	33.23	11.70
Δ_{Temp}	1	-1.51	0.27	0.66
	2	-1.51	0.24	0.99
	3	-2.41	0.49	0.71
Δ_{Ours}	0	0.14	1.23	1.69

Table 10. **Comparison with Temporal Information Aggregation.** N_p denotes the number of previous frames the models take as input. The values in blue are the gains achieved by BEVFormer-T over BEVFormer-S. The values in red are the gains achieved by BEVFormer-S+Ours over BEVFormer-S.

7.4. Training Cost

The table below summarizes the training-time resource consumption. All models were trained using four RTX 4090 GPUs (24GB) with batch size 2 for 30 epochs. Let L_q, L_k, and L_k^{def} denote the numbers of queries, keys, and sampling points in deformable attention, respectively. The memory complexities of CVT and BEVFormer are $\mathcal{O}(L_q L_k)$ and $\mathcal{O}(L_q L_k^{\text{def}})$, respectively, where generally $L_k^{\text{def}} \ll L_k$.

Model	Sec./Iter	Train. Time(Hrs)	GPU mem./Batch(GB)	# Param.(M)
CVT	0.2	6	6.14	4.39
CVT+IVT-Dual	0.41	12	8.18	26.74
BEVFormer	0.24	7	3.58	31.17
BEVFormer+IVT-Dual	0.43	12.5	5.87	53.93

The proposed framework results in roughly a $2\times$ increase in training time and a $1.6\times$ increase in GPU memory usage. However, it introduces no inference-time overhead and converges within a similar number of epochs. Future work will focus on optimizing the training framework to reduce training time and GPU mem. consumption.

7.5. Pseudo-Label (PL) Quality

Since nuScenes partially provides GT PV seg. labels for multi-view images, we train Mask2Former on the available labels and use it to generate PLs. As shown in Fig. 6 of the supple. material, they are less accurate than GT labels but *remain sufficiently reliable for training the IVT net. for the regularization*, as validated by the consistent improvements reported in Tab. 1&2. To further evaluate the robustness of the proposed framework to PL quality, we vary both the PL generator (Mask2Former vs. UNet) and its training budget.

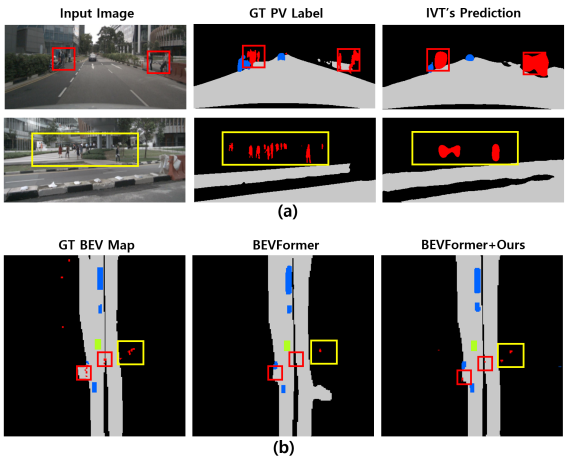

Figure 8. **Prediction examples on a scene with occluded pedestrians.** (a) Input images (the first column), ground-truth PV segmentation maps (the second column), and PV segmentation maps predicted by the proposed IVT network (the third column). (b) Ground-truth BEV map (the first column), BEV map predicted by BEVFormer (the second column), and BEV map predicted by BEVFormer+Ours (the third column). The green boxes indicate the AV. Please zoom in for better visibility.

Exp. ID	PL generator	# Training epochs	PV seg. mIoU
(1)	Mask2Former	10	78.97
(2)	Mask2Former	5	76.34
(3)	UNet	10	67.67

IVT pre-trained on	BEV model	BEV seg. mIoU
-	CVT	39.70
	BEVFormer	41.00
PLs from (1)	CVT+IVT	$41.78_{2.08\uparrow}$
	BEVFormer+Ours	$42.02_{1.02\uparrow}$
PLs from (2)	CVT+IVT	$41.44_{1.74\uparrow}$
	BEVFormer+Ours	$41.94_{0.94\uparrow}$
PLs from (3)	CVT+IVT	$41.56_{1.86\uparrow}$
	BEVFormer+Ours	$41.62_{0.62\uparrow}$

As shown in the top table, Mask2Former trained for 10 epochs (used in the main paper) achieves the highest PV seg. accuracy, while the remaining settings produce progressively lower-quality PLs. We then pre-train the IVT net. using PLs from each setting and apply it to train the two models. The bottom table shows that the proposed framework consistently improves the performance even when the IVT is pre-trained on lower-quality PLs, indicating that the regularization effect is robust to PL noise rather than relying on highly accurate PV supervision. The lack of full GT PV labels remains a dataset limitation, and richer supervision may further improve performance.

8. Additional Prediction Results

Figure 8 shows the prediction results for a scene where pedestrians stand close together in the distance, making them look tiny and partially occluded. While BEVFormer misses nearly all pedestrians around the AV, BEV-

Former+Ours successfully detects several of them, as highlighted by the red and yellow boxes in the figure. Highlighting the presence of the pedestrians in the input images through the learned reverse mapping helps the VT model more effectively recognize and detect the pedestrians on the BEV map. Finally, we further present the prediction results of the four baselines with or without the proposed framework, CVTM [34], FocusBEV [36] in Fig. 9 and 10. As shown in the figure, the proposed framework improves the performance of the four baselines across all categories.

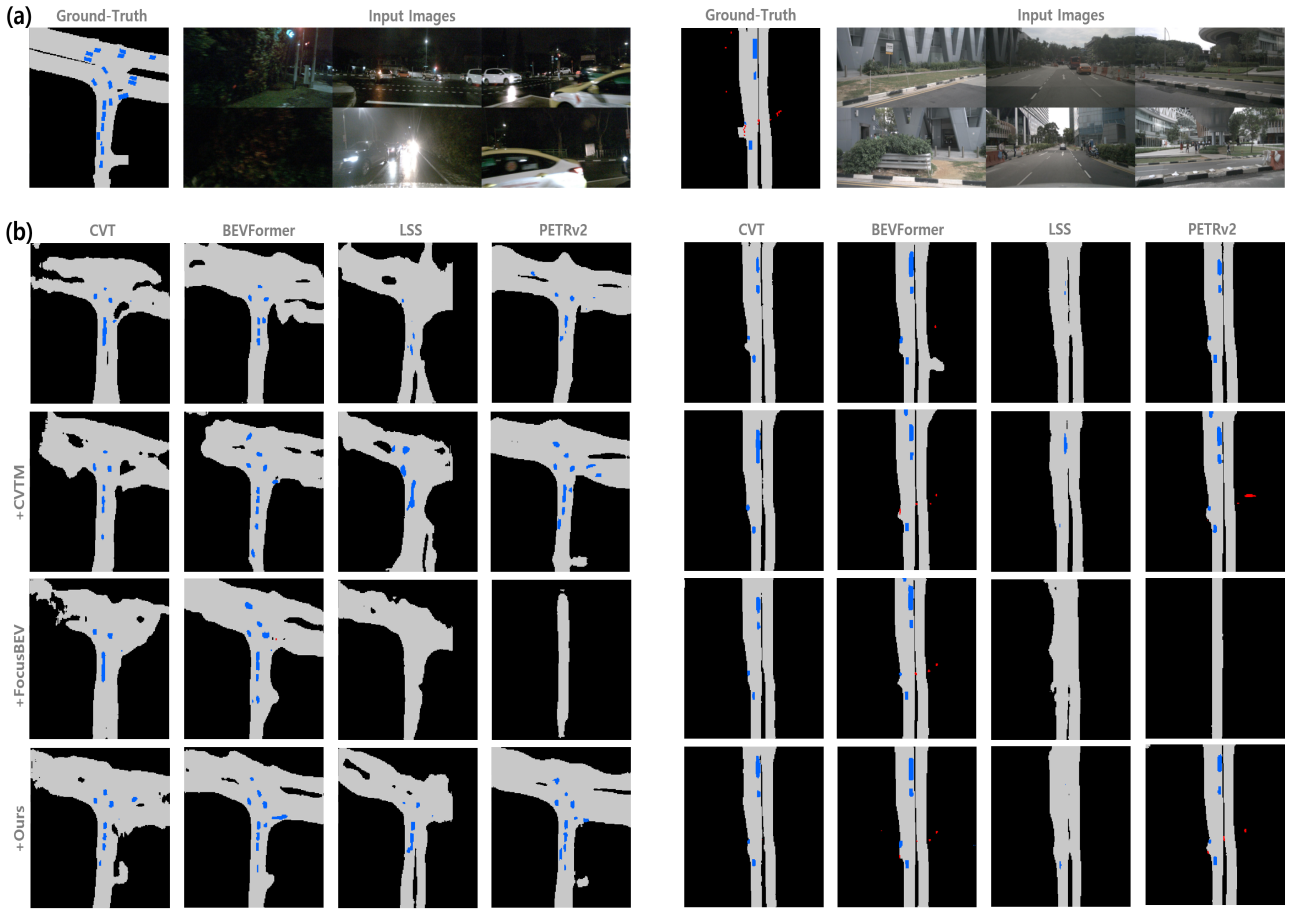

Figure 9. **Prediction results.** (a) Input images and their corresponding ground-truth BEV maps, (b) BEV map prediction results. In (b), the first row shows the predictions from the four baseline models. The second, third, and fourth rows show the results when CVTM [34], FocusBEV [36], and Ours are applied to the baseline models, respectively. *Drivable area*, *vehicle*, and *pedestrian* are color-coded with gray, blue, and red, respectively. Please zoom in for better visibility.

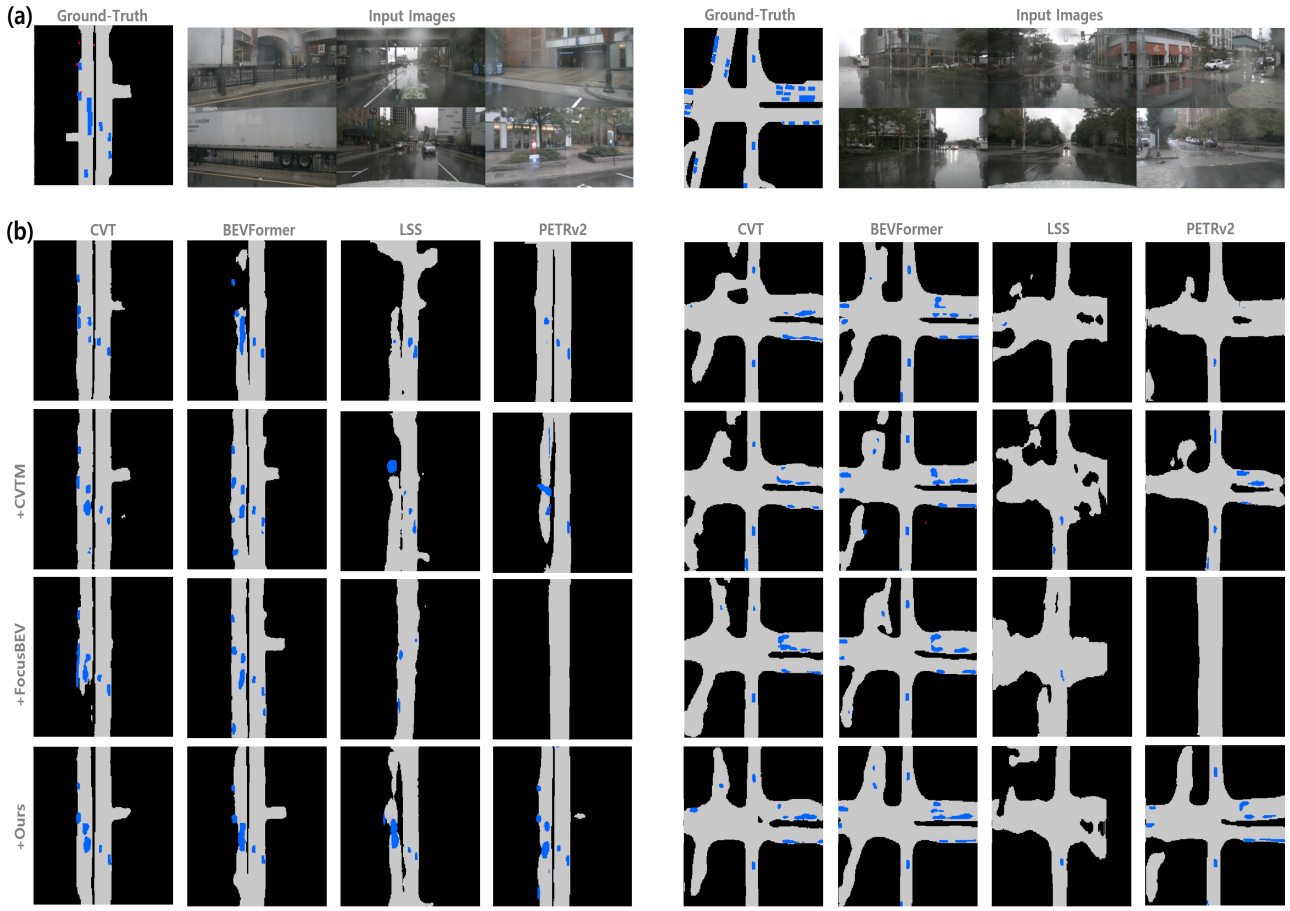

Figure 10. **Prediction results.** (a) Input images and their corresponding ground-truth BEV maps, (b) BEV map prediction results. In (b), the first row shows the predictions from the four baseline models. The second, third, and fourth rows show the results when CVTM [34], FocusBEV [36], and Ours are applied to the baseline models, respectively. *Drivable area*, *vehicle*, and *pedestrian* are color-coded with gray, blue, and red, respectively. Please zoom in for better visibility.