Supplementary Material
Towards Viewpoint Robustness in Bird's Eye View Segmentation

1. Synthetic Datasets

One of the contributions of our paper is the release of two datasets. The datasets are made available through our project page.

Dataset #1: The first dataset, rendered in CARLA [2] can be used to analyze the impact of camera viewpoint changes without any other domain gaps. This dataset includes 36 train and test datasets: 1 source rig dataset (using the nuScenes [1] rig), 10 yaw datasets, 10 pitch datasets, 10 height datasets, and 5 pitch and height together datasets. We randomize the weather conditions in each scene, rendering 100 images from 2,500 scenes per dataset. The total number of images is 1.8 million (half are train images and half are test images). While we provide 25,000 images in each test dataset, we only tested with 5,000 in our experiments. The datasets contain 3D bounding box labels that can be used for tasks, such as BEV segmentation or 3D object detection. More example images are shown in Fig. 1.

Dataset #2: The second dataset, rendered with NVIDIA DRIVE Sim [4], can be used to evaluate the viewpoint robustness of BEV segmentation models trained on real data. The reason we create a separate dataset for this is because the domain gap between real to CARLA is much more significant than the domain gap between real data to DRIVE Sim data, despite it still being synthetic. We note that the DRIVE Sim data comes from an F-theta lens camera and we rectify it prior to evaluation. Because the real data we use to train our model comes from a source rig that may be different than what others will use in the future, we provide test datasets that cover a range of pitches and heights, such that there is sufficient diversity for other researchers to evaluate the viewpoint robustness, as long as their source rig does not deviate significantly from ours. Our source rig is from a standard sized sedan vehicle, which is typical across the AV industry.

The dataset includes the following test subsets: source rig, +0.2 m height, +0.4 m height, +0.6 m height, +0.8 m height, -5 pitch°, -10 pitch°, +5 pitch°, +10 pitch°, +0.6 m height and -10 pitch° together, and +1.5 m depth. Each test subset contains at least 1,800 images and includes 3D bounding box labels for BEV segmentation or 3D object detection. Fig. 2 provides more examples of images from this dataset.

2. Method Details

2.1. Novel View Synthesis Training

We provide training details of our adapted Worldsheet method. Our total loss is defined as the combination of image L1 loss L_{im}^{l1}, image SSIM loss L_{im}^{ssim}, direct lidar depth loss L_{D}^{direct}, re-rendered lidar depth loss $L_{D}^{rerendered}$, and the regularization term L_{reg}:

$$L = \lambda_1 L_{im}^{l1} + \lambda_2 L_{im}^{ssim} + \lambda_3 L_{D}^{direct} + \lambda_4 L_{D}^{rerendered} + \lambda_5 L_{reg} \quad (1)$$

where we set the weights(5 λ) to be 1.5, 8.5, 10, 10, 0.01, respectively. We apply the same Laplacian regulation terms as Worldsheet [3], but use smaller weights since drivng scenes are more complex. The model is trained on a 16GB cloud computing GPU card and it converges in 20K iterations.

2.2. Baseline: Extrinsic Augmentations

Our two baselines are (1) passing in train extrinsics at test time, and (2) extrinsic augmentations. In this section, we describe the extrinsic augmentations in more detail and provide intuition. The key idea is that, rather than improve the robustness of the BEV segmentation encoder and decoder to different viewpoints, improve the robustness of only the decoder. In BEV segmentation, typically, the encoder maps images to features, the features are transformed based on the extrinsic and intrinsic parameters of the camera, and the decoder maps the transformed features to BEV segmentation maps. While it is challenging to warp images from one view to another and maintain photorealism, augmenting the extrinsics is straightforward and can improve model robustness on its own by exposing the decoder to different transformations of the features. To train this baseline, we use the training images

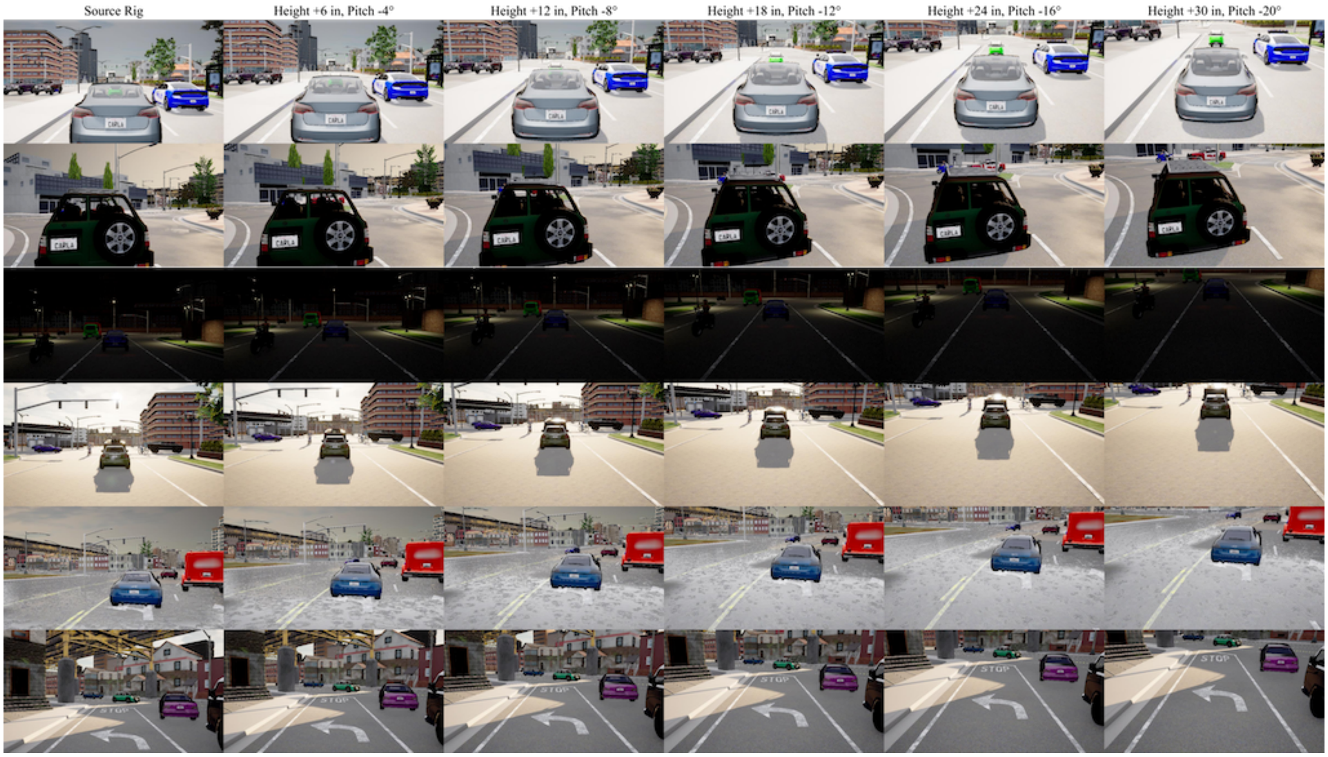
Source Rig
Height +6 in, Pitch -4°
Height +12 in, Pitch -8°
Height +18 in, Pitch -12°
Height +24 in, Pitch -16°
Height +30 in, Pitch -20°

Figure 1: Dataset # 1 - Example Images from CARLA: We render data across camera viewpoints using CARLA [2]. The data can be used to train and evaluate models for viewpoint robustness in isolation of other domain gaps.

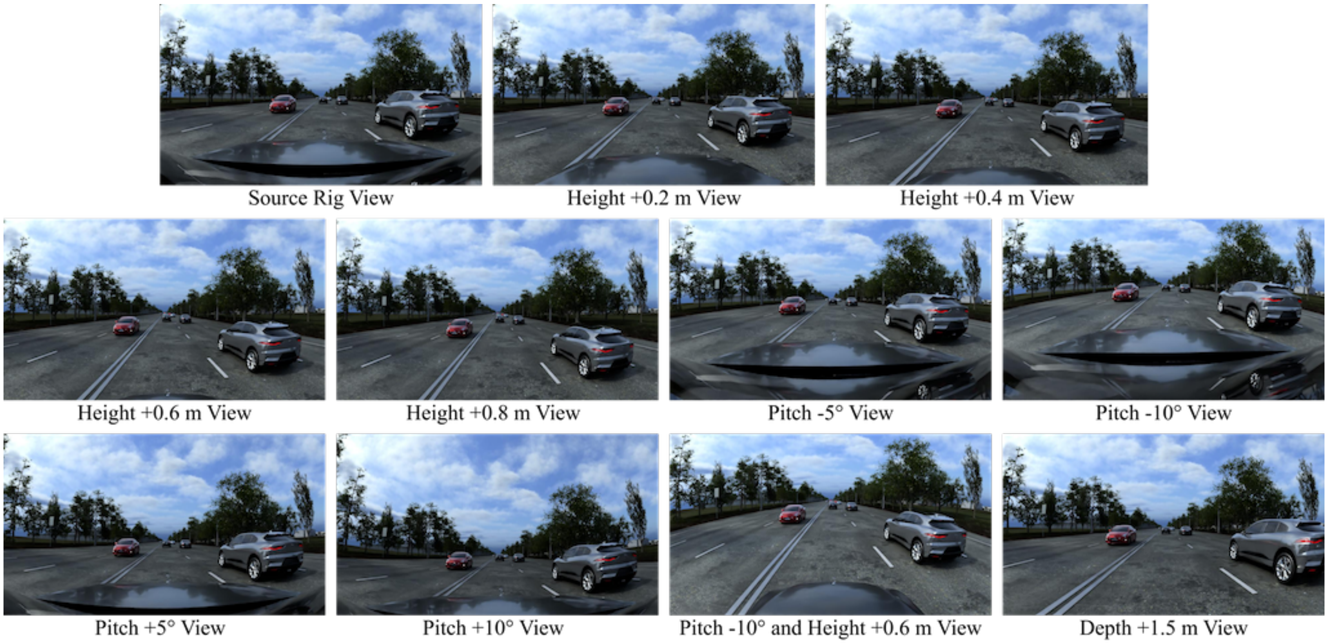

Figure 2: Dataset # 2 - Example Images from NVIDIA DRIVE Sim: We render data across camera viewpoints using NVIDIA DRIVE Sim [4]. The data can be used to evaluate models trained on real-world data for viewpoint robustness.

from the source view, but change the extrinsic parameters that are used to transform the features. In Cross

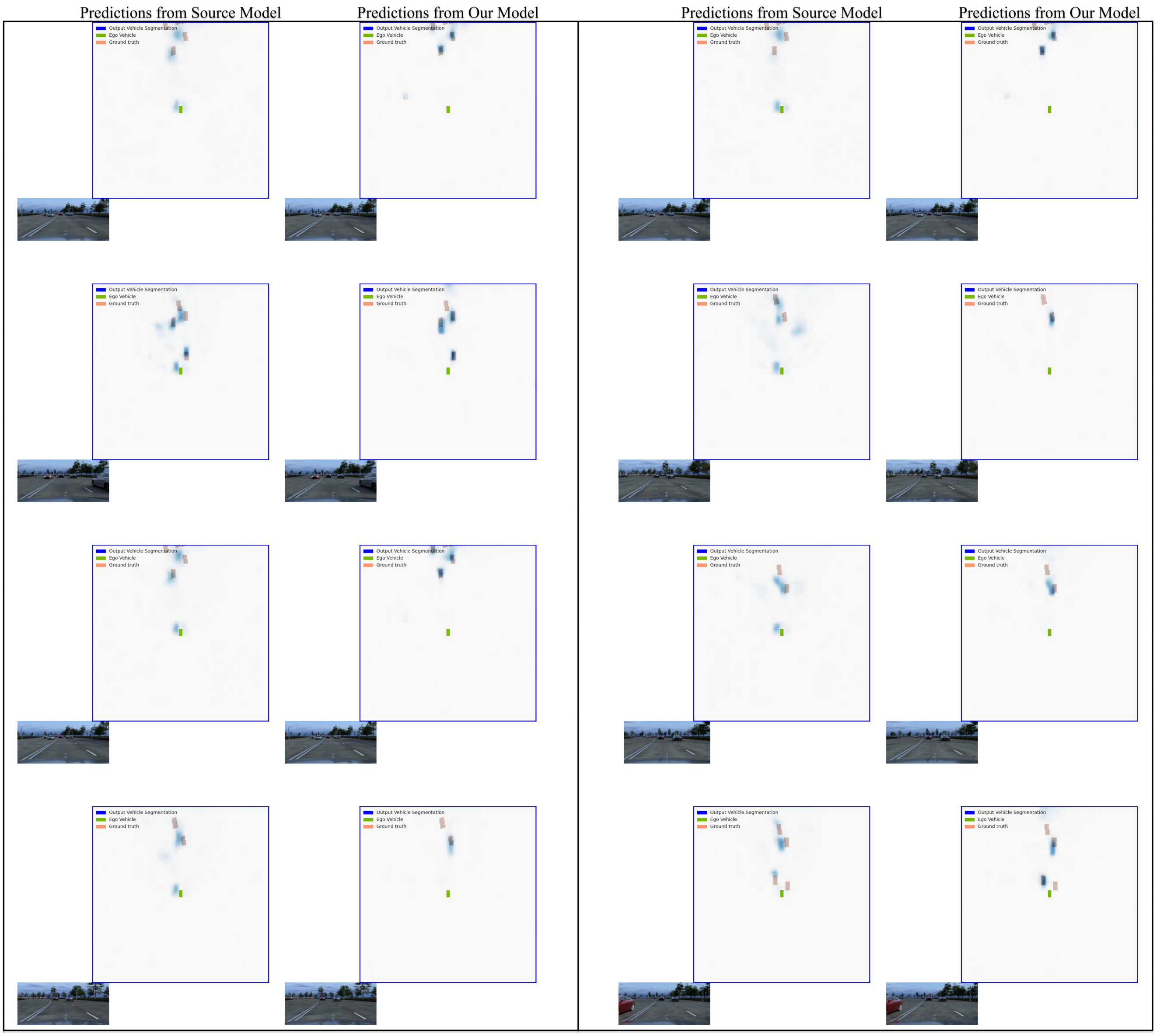

Figure 3: Qualitative Comparison of Model Predictions: We compare the predictions of the source model (CVT trained on data from the source rig) and our model (trained on a mixture of data from the source rig and data transformed to the viewpoint of the target rig). Both models are evaluated on a target rig with pitch of -5°.

View Transformers (CVT) [5], this transformation is an affine transformation between the image features and an embedding created from the extrinsic parameters. We randomly perturb the extrinsic parameters within the range of the camera on the source rig and the target rig during training. In addition, to ensure that the extrinsics and ground truth are in the same coordinate frame, we also transform the ground truth 3D bounding box labels accordingly. As a result, during training, the extrinsics vary within a range, exposing the decoder to many different extrinsic parameters. We observe these extrinsic augmentations signicantly improve the ability for the BEV segmentation model to then generalize to the target rig, but do not work as well as our proposed approach, which augments the encoder and decoder using novel view synthesis. This observation indicates that the drop in performance when BEV segmentation models are exposed to different viewpoints is a result in a lack of generalization in both the image encoder and feature decoder.

	IoU (Real)	IoU (Sim)
CVT	0.245	0.170

Table 1: Real-to-Sim Gap: In our experiments, we train on real data and evaluate on simulated data (see Fig. 2). Shown above is the domain gap. Because our goal is relative performance with vs. without our method, not absolute performance, the domain gap introduced by evaluating with synthetic data is acceptable.

Train Pitch	Test Pitch	E/I?	IoU
0,-10	-5	I	0.149
0,-5	-10	E	0.153
0,5	-5	E	0.144
0,5	-10	E	0.147

Table 2: Interpolation & Extrapolation: We test models trained on two different viewpoints (the source rig viewpoint and data transformed to a target rig viewpoint) on viewpoints in between the two (interpolation) or beyond the two (extrapolation). Shown above are the results for pitch.

3. Results

3.1. Visualizations

In the main text, we provide quantitative comparisons of a CVT model trained on data from only the source rig and evaluated on many target rigs compared to models trained with our approach and evaluated on the corresponding target rig. In Fig. 3, we provide qualitative comparisons of the predictions between these two models. In general, we see that the model trained with our approach (of incorporating data transformed into the viewpoint of the target rig into the training dataset) have more true positive pixels and fewer false positive. A consistent trend we notice across models trained only on data from the source rig is that, when evaluated on data from other rigs, they tend to hallucinate vehicles near the ego-vehicle, which are not there, as seen in several of the examples in Fig. 3. We also notice higher confidence predictions (indicated by the intensity of the blue in Fig. 3) with our approach.

3.2. Sim-to-Real Gap

In our experiments, we train on real data and evaluate on simulated data (see Fig. 2). Table 1 shows the domain gap of 7.5%. However, we note that the average number of ground truth objects between the real and sim test datasets differ, with the real test dataset containing 3 more objects per image on average. We note that this difference can skew the IoU because there is more opportunity to have true positive predictions when evaluating on a dataset with more positive pixels. When evaluated on a real dataset with the same average number of objects, we found the performance was 0.19%, dropping the domain gap to just 2%. Thus, we note two observations: (1) the low BEV segmentation accuracy in our experiments, even in the oracle (0.17%), is a result of very few and far away objects in most images of our test datasets, thus making BEV segmentation very challenging, and (2) the domain gap is between 2-7%. Because our goal is relative performance with vs. without our method, not absolute performance, the domain gap introduced by evaluating with synthetic data is acceptable.

3.3. Interpolation and Extrapolation

As discussed in the main text, the primary focus of our paper is introducing a method to create target rig specific BEV segmentation models without additional data collection or labeling costs. However, we also study whether our training strategy, which involves training with both real data from the source rig and transformed data from the viewpoint of the target rig, allows models to better generalize across target rigs. Initial results are promising, as shown in Table 2. We see that the performance of the model when tested on viewpoints between the two training viewpoints (interpolation) and beyond the two viewpoints (extrapolation) is significantly better than the model only trained on data from the source rig. In this ablation, we only test on changes to pitch. Results are summarized in the main text and more details are provided in Table 2, where I/E specifies whether it is interpolation or extrapolation, respectively.

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/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[3] Ronghang Hu, Nikhila Ravi, Alexander C Berg, and Deepak Pathak. Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12528–12537, 2021.

[4] NVIDIA. Nvidia drive sim. `https://developer.nvidia.com/drive/simulation`, 2021.

[5] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13760–13769, 2022.