# Synthetic-to-Real Self-supervised Robust Depth Estimation via Learning with Motion and Structure Priors -Supplementary Material-

In the supplementary of the main text in the paper, we illustrate the following contents for better understanding the proposed method: (1) More implementation details for our experiments in Sec.1. (2) The complete differentiable histogram formula and comparison with GAN in Sec.2. (3) More qualitative results on nuScenes, Robotcar and DrivingStereo [3, 7, 12] in Sec.3. (4) More visualization examples of the consistency maps in Sec.4.

## 1. Implementation Details

The experiments on all models of different stages are conducted on the same ResNet architecture [6], so do the baselines. The training for the $\Phi_{day}$ and the $\Phi_{syn}$ are conducted for 20 epochs, while $\Phi_{real}$ is initialized by $\Phi_{syn}$ and trained for 10 epochs. The learning rate is initialized to $1 \times 10^{-4}$ with a decreasing factor of 0.5 every 5 epochs. For the adaptive depth bins in Manydepth [11], we fix them to $3.5-80m$ since [4, 5, 9] all use a weak velocity loss to align the scales. Thanks to [4, 9], we can utilize the synthetic data from their pretrained augmentation models. For the $Dist_{day}$, we obtain a fixed distribution through the whole daytime training dataset, and utilize it for the real adaptation constraint. The training of all the pipeline is conducted on a single NVIDIA A5000 GPU. The synthetic adaptation takes about 14 hours of training, while the real adaptation takes about 7 hours for training.

For the training objective of synthetic adaptation $L_{syn}$, we simply set $\alpha_1 = 1.0, \alpha_2 = 1.0, \alpha_3 = 1.0$; while for the total objective in real adaptation, we set $\alpha_1 = 1.0, \alpha_2 = 0.01, \alpha_3 = 1.0, \alpha_4 = 1.0$. For the consistency-reweighting strategy, the scale factor $\beta$ and the weight bias $\epsilon$ are chosen to be $1.0, 1.0$ respectively.

**Pose Supervision.** Since pose is crucial to construct a cost volume, we also use the daytime pose model to supervise the pose model for synthetic adverse conditions with L2 loss. Specifically, we utilize the rotation matrix represented by the angles, where we have $\theta_{day}$ and $\theta_{syn}$; and the translation vectors: $t_{day}, t_{syn}$. The supervision for pose from daytime to adverse conditions is written as

$$L_\theta = \|\theta_{day} - \theta_{syn}\|_2, \tag{1}$$

$$L_t = \|t_{day} - t_{syn}\|_2, \tag{2}$$

$$L_T = L_\theta + L_t. \tag{3}$$

The pose objective in real adaptation is similar to the above equations.

**The Proportion of Data in Different Conditions.** In the synthetic adaptation, we set the proportion of daytime, nighttime and rain to be $50\%, 25\%, 25\%$. This is because training the model in multi-frame mode and single-frame augmentation are quite important, and this leads us to set more daytime data in training. In the real adaptation, we utilize the data of multiple conditions inside dataset without any specific design for the proportion.

## 2. Differentiable Histogram

The target of differentiable histogram is to using a differentiable function to approximate the step function of the histogram statistical process. We follow the design in [1, 2], using the Kernel Density Estimation (KDE) with a kernel $K$ to estimate the intensity density $f_D$ of a depth map $D_{daytime}$ or $D_{adverse}$:

$$f_D(d) = \frac{1}{hwa} \sum_{x \in \Omega} K\left(\frac{D(x) - d}{a}\right), \tag{4}$$

where $d$ represents the depth value, $\Omega$ is the pixels on a depth map, $a$ is the bandwidth, and $h, w$ refer to the size of map. The kernel $K$ is chosen as the derivative of the sigmoid function:

$$K(x) = \frac{d}{dx}\sigma(x) = \sigma(x)\sigma(-x), \tag{5}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{6}$$

From Equ.5, the kernel is obvious to have non-negative, symmetric characteristics, which is consistent with the requirements of KDE. Then, we compute the depth map of

Table 1. Quantitative comparison between using GAN and using our proposed structure prior. Using structure prior can further decrease the gaps between adverse conditions and daytime, while it is difficult to narrow this fine gap using GAN.

| Method | Test frames | nuScenes-day | | | | nuScenes-night | | | | nuScenes-rain | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AbsRel↓ | SqRel↓ | RMSE↓ | $\delta_1$ ↑ | AbsRel↓ | SqRel↓ | RMSE↓ | $\delta_1$ ↑ | AbsRel↓ | SqRel↓ | RMSE↓ | $\delta_1$ ↑ |
| GAN | 2(-1,0) | 0.1234 | 1.446 | 6.069 | 86.42 | 0.1761 | 1.841 | 7.823 | 72.37 | 0.1332 | 1.614 | 6.766 | 82.20 |
| **Structure Prior** | 2(-1,0) | **0.1170** | **1.371** | **6.023** | **86.66** | **0.1713** | **1.779** | **7.689** | **73.16** | **0.1266** | **1.532** | **6.654** | **82.86** |

range $d_{min}$ $d_{max}$ to a differentiable histogram by dividing the depth into N sub-intervals. The $n^{th}$ interval owns the length of $L = \frac{d_{max}-d_{min}}{N}$ and the center $b_n = d_{min} + L(n + \frac{1}{2})$. Our target is to obtain the probability of one pixel on the depth map belonging to the $n^{th}$ sub-interval, $P(n)$:

$$P(n) = \int_{nL+d_{min}}^{(n+1)L+d_{min}} f_D(x)dx. \tag{7}$$

After solving the integration, $P(n)$ can be rewritten as

$$P(n) = \frac{1}{hw}\sum_{x=1}^{hw}[\sigma(\frac{D(x) - b_n + \frac{L}{2}}{a}) - \sigma(\frac{D(x) - b_n - \frac{L}{2}}{a})], \tag{8}$$

By calculating across every depth sub-interval, the histogram can be formulated as:

$$hist = \{b_n, P(n)\}_{n=1}^{N}, \tag{9}$$

Computing through all the real-world data, we can have $hist_{day}, hist_{night}, hist_{rain}$ for separate conditions. $hist_{day}$ corresponds to $Dist_{day}$ in our main paper, which is utilized as a reference distribution guidance.

In our experiments, $d_{min} = 3.5, d_{max} = 80.0, N = 100, a = \frac{L}{20}$ to approximate the step function. Noted that the reference distribution is calculated through the whole training set of daytime data with a frozen daytime model in our experiments.

**Distribution Prior v.s. GAN.** What's more, comparing with the utility of GAN [10], we notice two most important advantages of this explicit differentiable distribution: (1) Leveraging GAN requires to randomly sample a batch of daytime data in every iteration, which can not guarantee the quality of the sampled data and possible to lead the models to be optimized with randomness. Our design of the explicit distribution is calculated through the dataset of daytime data, which is much more representative and stable for optimization. (2) Using GAN is proper or meaningful when the gaps are quite huge (like optimize a daytime model to directly estimate in nighttime). However, when the gaps have decreased significantly (after training a model in synthetic adverse conditions with pseudo-labels), it is hard for GAN to close the gaps further, due to the rough supervision and optimization mechanism of GAN (as in our experiments, the discriminator is hard to decrease its loss, because the domain gap between current prediction and daytime prediction is much smaller than not doing synthetic adaptation). On the contrary, we find that a fine distribution of statistical depth can further help to decrease the gaps, and the comparison between utility of structure prior and GAN is shown in Tab.1.

## 3. Qualitative Results on Different Datasets

As supplementary to the main paper, we provide more qualitative results on nuScenes [3] (daytime, nighttime, rain), Robotcar [7] (daytime, nighttime) and DrivingStereo [12] (rain, fog), as shown from Fig.1-6.

### 3.1. NuScenes

Our model, as shown in Fig.1, provides comparative daytime predictions as the state-of-the-art, and from Tab.1 in the main paper, ours actually estimates more accurately, which is due to our strategy to transfer motion-structure knowledge for better robust representation learning.

Fig.2 displays more examples in the nighttime condition, which is quite severe due to its extreme darkness and glare. In row 2 to row 3, previous methods all fail to precisely capture depth of the left and parts (where we can see tree on both sides from the ground truth), while our model gives accurate predictions on both left and right trees. In row 4 to row 7, it can be observed from the ground truth that there is a tree on the left of the road and very close to the camera, while [4, 8] detect the objects as far away. However, ours gives much more accurate predictions. In row 8 and 9, previous method suffer from the glare of blinding headlights, giving wrong predictions especially on the right side of the image. On the contrary, ours provides natural and correct depth to the glare parts. In row 10, other methods predict the tree which is far on the left of the view as much closer, and that is highly inconsistent with the ground truth. Ours captures this character and gives reasonable results.

Fig.3 refers to more visualization examples of the rain condition in nuScenes dataset. From row 1 to row 7, all previous methods are significantly affected by the obvious reflections and blur which are common in rain weather. For instance, in row 2 there is a car with severe blur on the left

of the image, where both [4] and [8] both fail and give incorrect results to this part. Ours as seen in row 2, column 5, displays much more robust and precise depth to the challenging region. In row 4, the previous approaches are misguided by the reflections of the white truck to perceive the reflections as obstacles on the road, while our model gives much more smooth depth map of the road. Considering row 9 and 10, we surprisingly found that our model can even less influenced by the moving objects in a scene, compared with [4]. This is due to the consistency-reweighting strategy which can assign smaller supervision weights to the corresponding regions.

### 3.2. Robotcar

In Fig.4, we display more qualitative results in daytime and nighttime conditions. Our model performs similarly in daytime, while significantly provides more robust and accurate estimations in nighttime. In row 3, there is a double-decker bus on the right, where the second deck can not be perceived by [4] due to its darkness, and our method is capable of recognizing this part. In row 4, a traffic light on the right side is affected by both darkness and glare, which make the previous approach unable to predict its depth. However, ours also successfully provide correct depth to the light.

### 3.3. DrivingStereo

From Fig.5 to Fig.6, more examples of DrivingStereo dataset [12] in rain and fog conditions are listed. In Fig.5, md4all [4] and robust-depth [8] are obviously affected by the reflections, such as perceiving the reflections as severe obstacles on the roads (row 3, row 6, row 8-10). In some scenarios the model can not distinguish the objects with its reflections (row 4), leading to wrong predictions. On the contrary, ours is consistently robust to the degradation.

In Fig.6, there are more examples in the fog conditions. In short, previous methods [4, 8] easily suffer from the noises like traces or reflections on the road. For example, in row 6 there are obvious reflections of the roadside sign on the right, which affects other methods' estimation to the road. Ours, however, is not affected by such reflections. Another example can be seen in row 9, where previous methods are influenced by the windows of the bus (left side of row 9), resulting in erroneous predictions of the bus. Compared with those results, our model gives smoother depth to the whole bus.

## 4. Consistency Map Visualization

Our consistency reweighting strategy aims at dynamically assigning importance to different regions of the input based on the agreement between multiple models' predictions, because there is no perfect ground truth. To verify the effectiveness of the proposed consistency-reweighitng strategy, we list more visualization examples in challenging conditions in Fig.7. Considering the consistency maps, the brighter represents the more consistent, and the darker means the more inconsistent. Notably, for night condition, it can be observed the inconsistency is resulted mainly from the extreme dark regions or the parts with severe glare. For the regions with good visibility (such as the near road), it is always consistent. In rain condition, the inconsistency is usually lead by the blur or the reflections. By introducing the consistency maps, we target to provide strong supervision to the reliable regions, such as clear roads with good visibility, and conduct weak supervision to the challenging regions in real-world data. This is important because our earlier trained models can not be guaranteed to provide perfect pseudo-depth labels when facing real world adverse conditions.

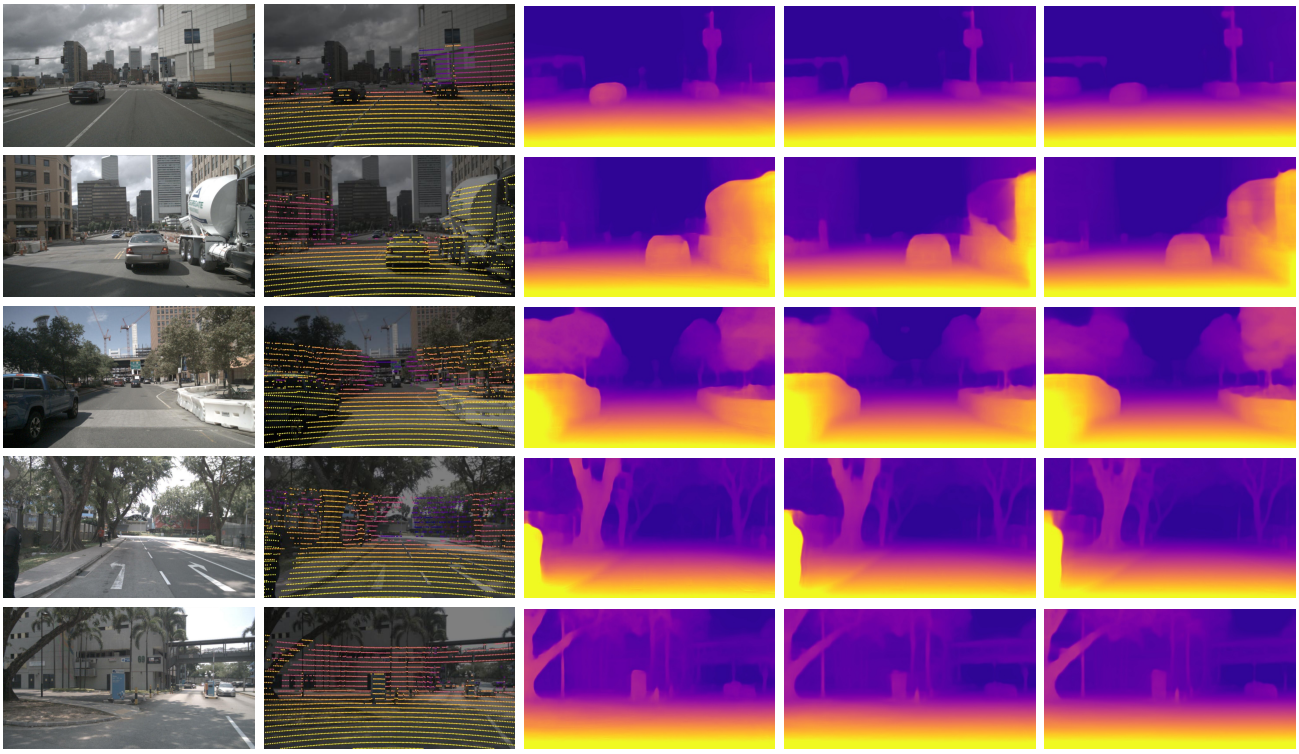


Figure 1. Qualitative examples in nuScenes [3] daytime condition. Ours is comparable with previous state-of-the-art in daytime, while much better in adverse conditons.

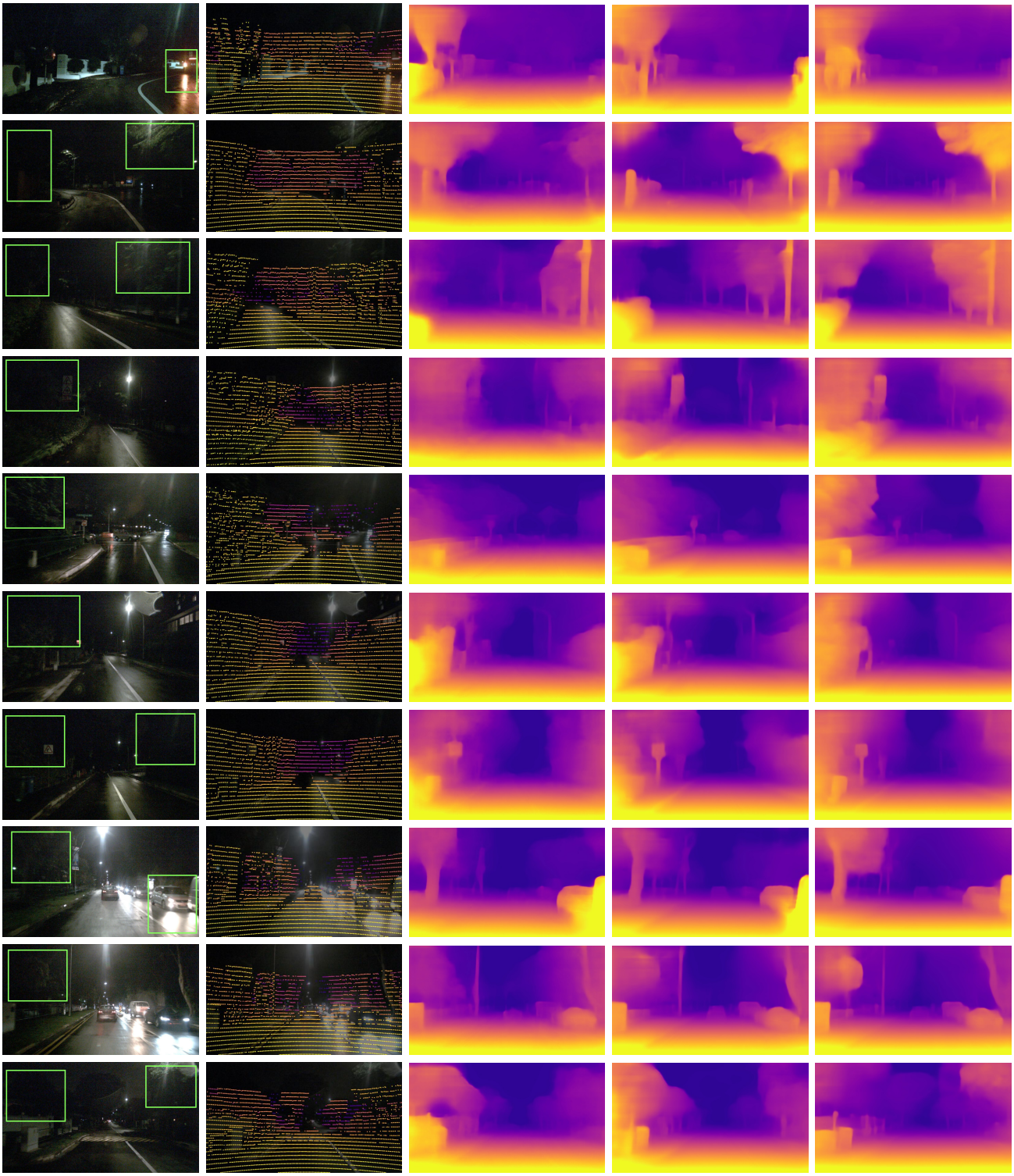


Figure 2. Qualitative examples in nuScenes [3] nighttime condition. The challenging parts that our method is capable of predicting are emphasized by green boxes.

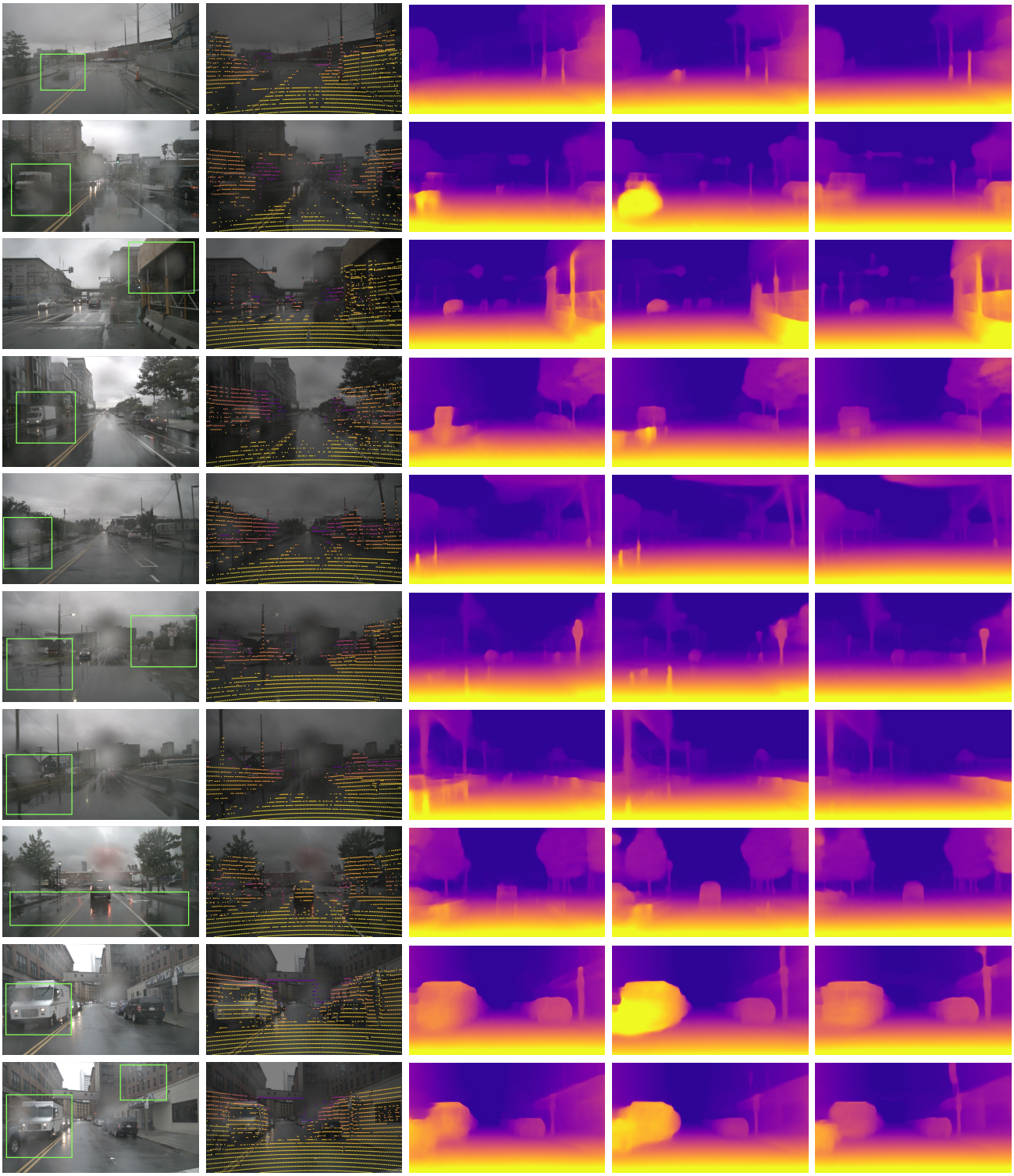


Figure 3. Qualitative examples in nuScenes [3] rain condition. The challenging parts that our method is capable of predicting are emphasized by green boxes.

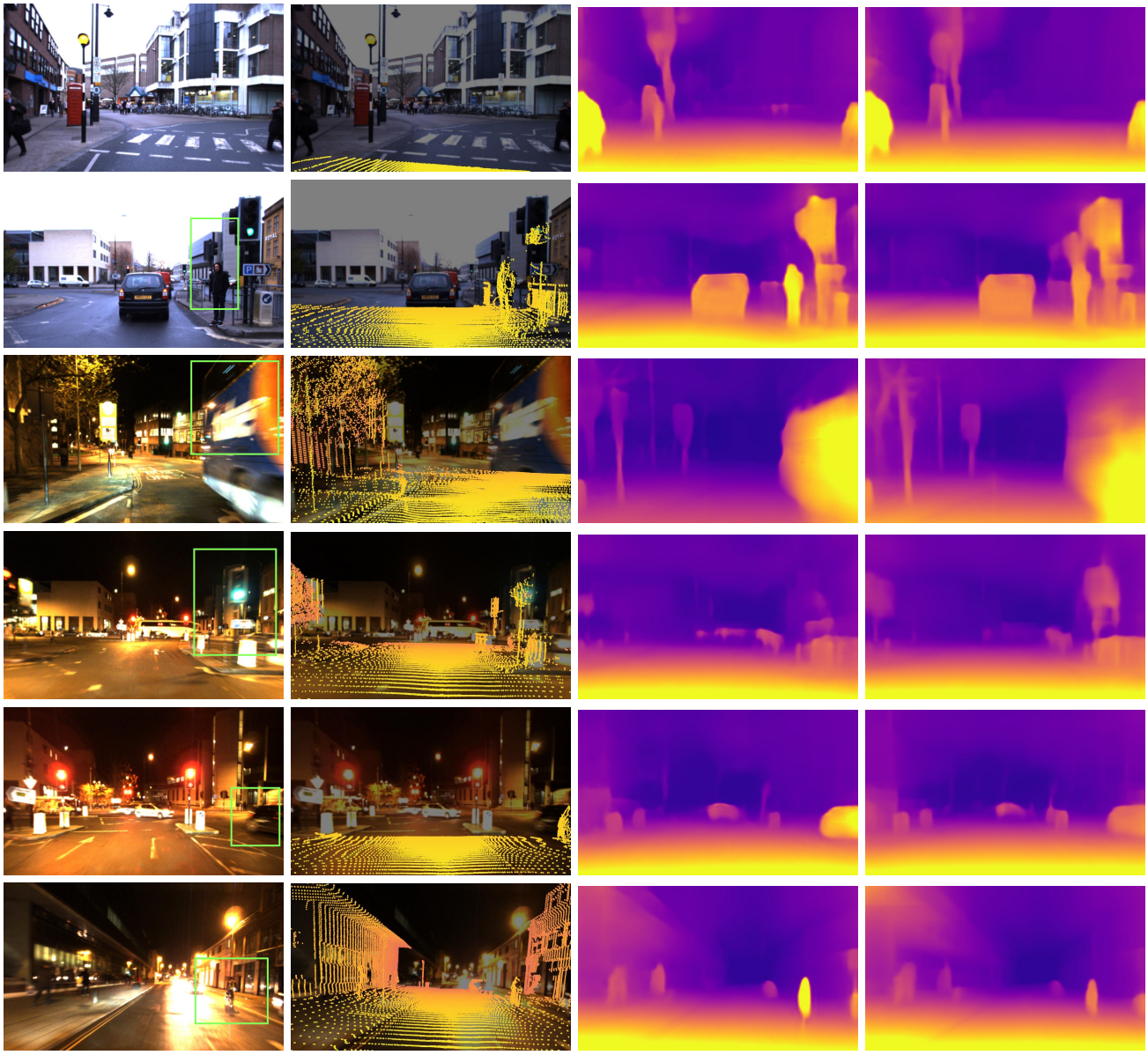


Figure 4. Qualitative examples in Robotcar [7] daytime-nighttime conditions. The challenging parts that our method is capable of predicting are emphasized by green boxes.

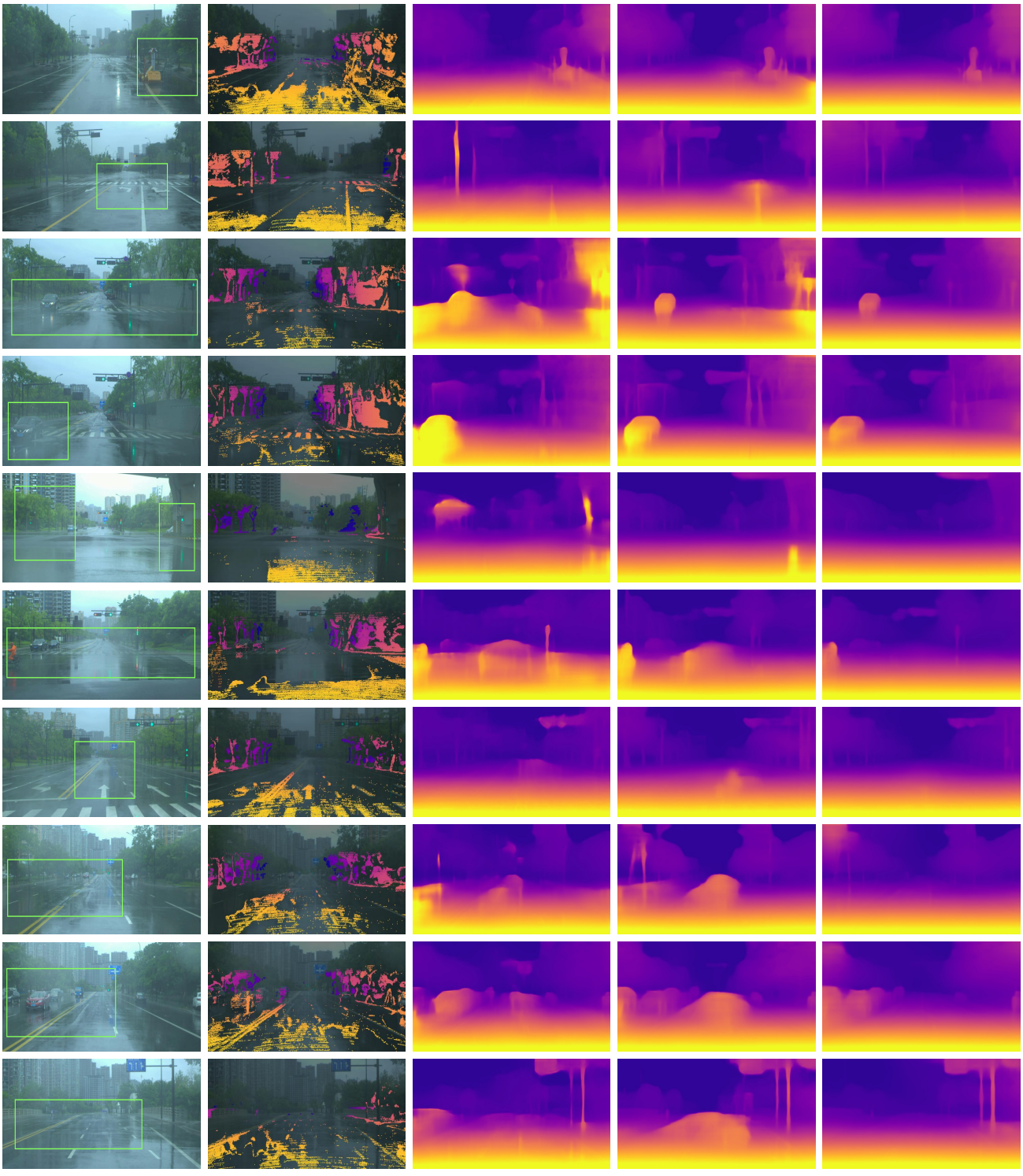


Figure 5. Qualitative examples in DrivingStereo [12] rain condition. The challenging parts that our method is capable of predicting are emphasized by green boxes.

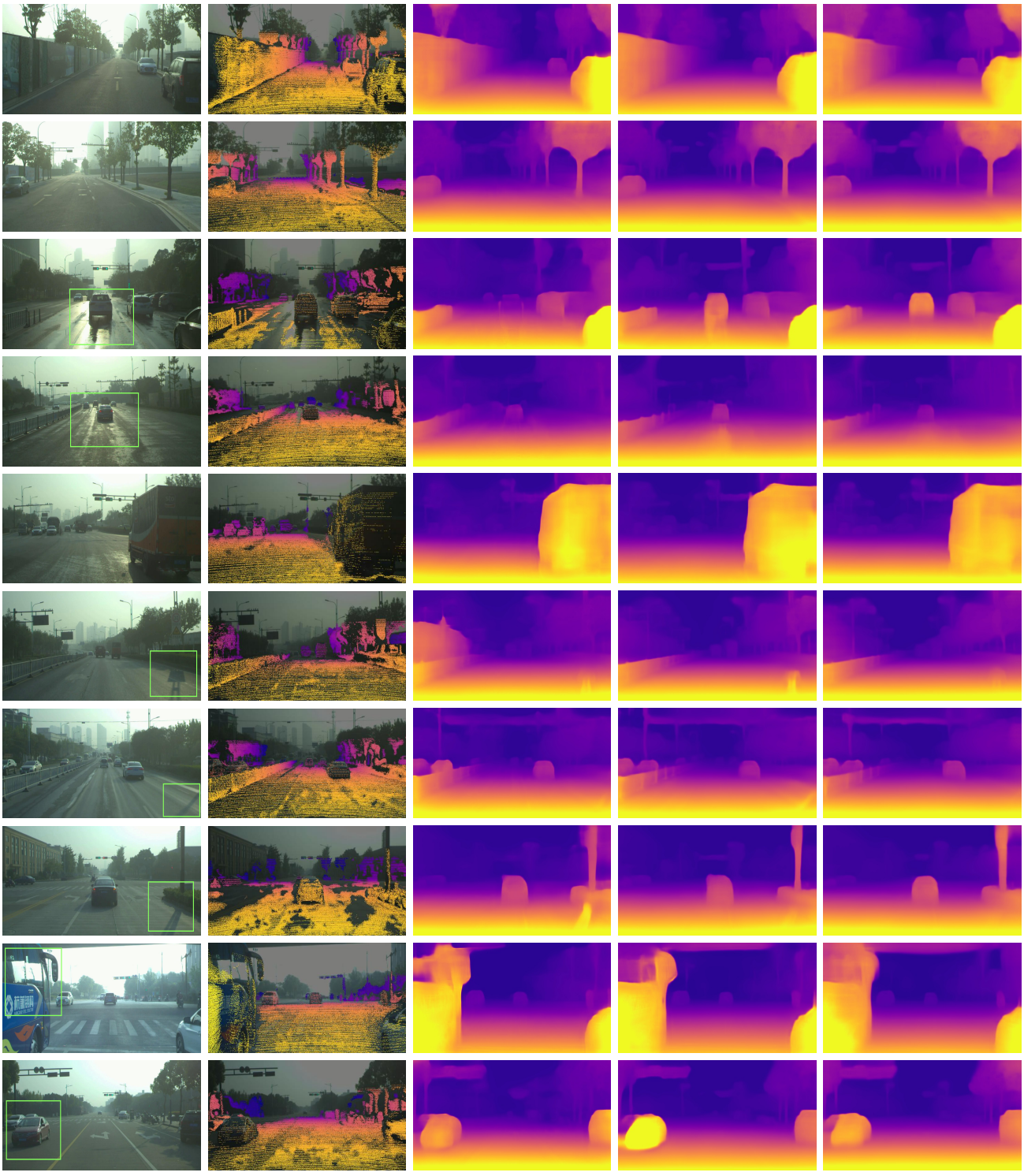


Figure 6. Qualitative examples in DrivingStereo [12] fog condition. The challenging parts that our method is capable of predicting are emphasized by green boxes.

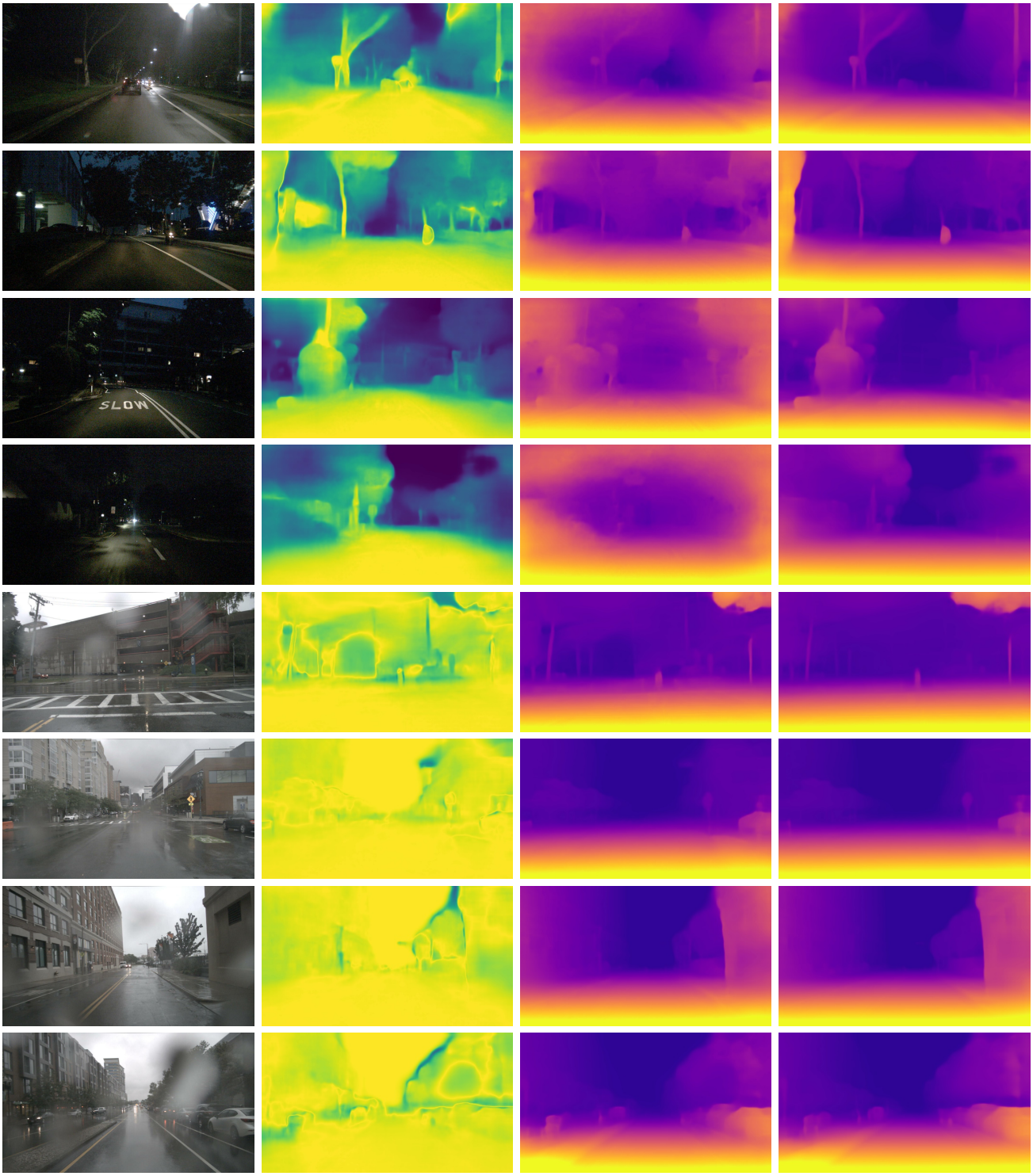


Figure 7. Visualization of the consistency maps in nighttime and rain conditions.

## References


[1] Mor Avi-Aharon, Assaf Arbelle, and Tammy Riklin Raviv. Hue-net: Intensity-based image-to-image translation with differentiable histogram loss functions. *arXiv preprint arXiv:1912.06044*, 2019. 1

[2] Mor Avi-Aharon, Assaf Arbelle, and Tammy Riklin Raviv. Differentiable histogram loss functions for intensity-based image-to-image translation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2023. 1

[3] 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. *arXiv preprint arXiv:1903.11027*, 2019. 1, 2, 4, 5, 6

[4] Stefano Gasperini, Nils Morbitzer, HyunJun Jung, Nassir Navab, and Federico Tombari. Robust monocular depth estimation under challenging conditions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8177–8186, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 9

[5] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[7] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000 km: The oxford robotcar dataset. *International Journal of Robotics Research*, page 0278364916679498, 2016. 1, 2, 7

[8] Kieran Saunders, George Vogiatzis, and Luis J. Manso. Self-supervised monocular depth estimation: Let's talk about the weather. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8907–8917, 2023. 2, 3, 4, 5, 6, 8, 9

[9] Fabio Tosi, Pierluigi Zama Ramirez, and Matteo Poggi. Diffusion models for monocular depth estimation: Overcoming challenging conditions. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[10] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing nighttime weirdness: Efficient self-supervised monocular depth estimation in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16055–16064, 2021. 2

[11] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1174, 2021. 1

[12] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 8, 9