Visual Point Cloud Forecasting enables Scalable Autonomous Driving
Supplementary Material

Appendix

A. Discussions

Towards better understanding of our work, we supplement intuitive questions that one might raise.

Q1: *What is the relationship between ViDAR and world models?*

In general, ViDAR could be deemed as a world model - predicting the future world conditioned on observations and actions. However, it distinguishes itself from existing world models for autonomous driving [1, 3, 6, 7]. Unlike these models, which operate within the same modality for both inputs and outputs (*e.g.*, image-in & image-out or LiDAR-in & LiDAR-out), ViDAR for the first time bridges different modalities. It leverages historical visual sequences as inputs to predict future point clouds. By utilizing ViDAR, it becomes possible to generate various future point clouds, conditioned on different future ego-vehicle motions, using visual inputs. This holds significant potential in training vision autonomy in the 3D space.

Q2: *Why use point clouds (LiDAR) as outputs, instead of future images?*

Compared to images, point clouds offer a highly precise depiction of the 3D environment, effectively capturing scene structure, object positions, and geometric properties. This detailed representation proves advantages in various 3D tasks, including perception, reconstruction, and rendering. Consequently, in ViDAR, we opt to employ point clouds as the prediction target. This choice enables the model to extract 3D geometry from visual inputs, thereby empowering downstream models.

Q3: *What are potential applications and future directions of ViDAR?*

In our work, we have demonstrated the effectiveness of ViDAR as a pre-training approach for enhancing downstream end-to-end autonomous driving models. Additionally, considering its capability as a world model, there is significant promise in utilizing ViDAR as a simulator for model-based reinforcement learning, thereby bolstering the decision-making abilities of vision-based autonomous agents. The application of ViDAR in this context opens up avenues for future research. For instance, to facilitate its scalability, investigations into the utilization of Image-LiDAR sequences from diverse datasets are necessary. Furthermore, exploring the combination of ViDAR with other single-modality world models to create more favorable data simulations presents intriguing prospects for future inquiry.

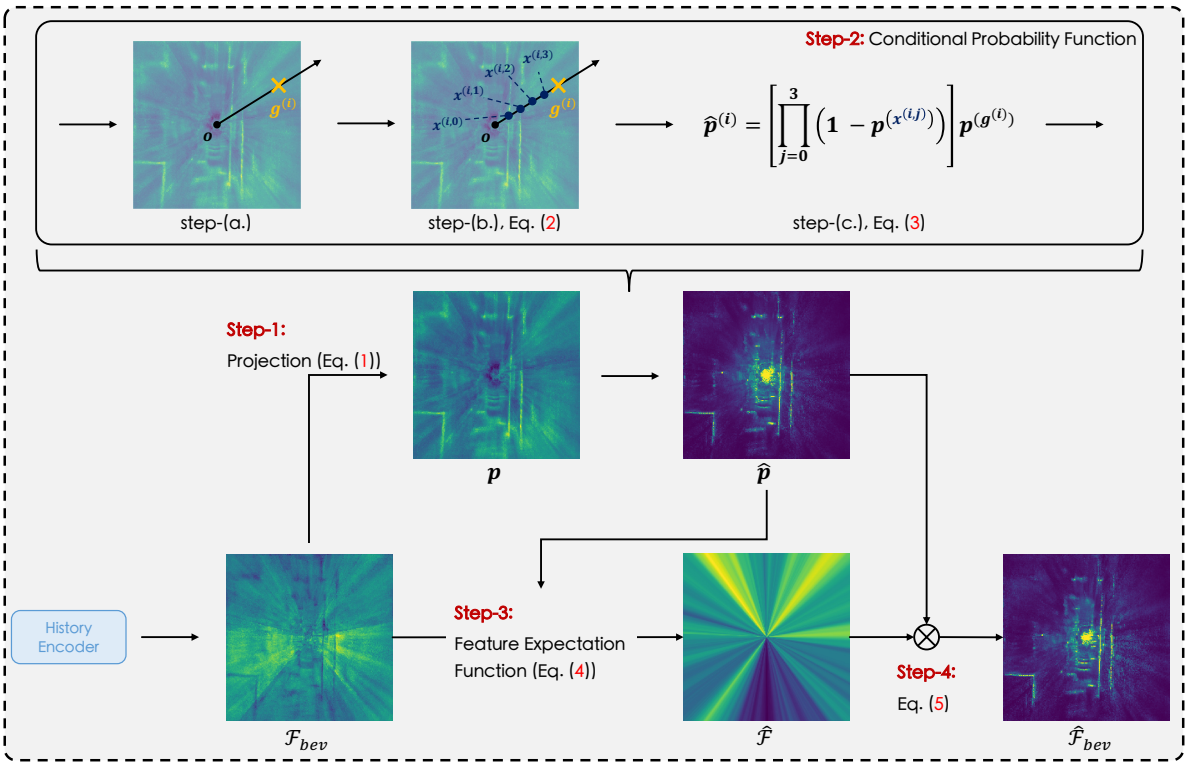

Figure 1. **Detailed architecture of the Latent Rendering operator.** We only show the case of single-group for simplicity. Given the visual BEV embedding $\mathcal{F}_{\text{bev}}$ from the History Encoder, the Latent Rendering operator performs a series of steps. First, it employs a projection layer to estimate the independent probability map, $\mathbf{p}$. Subsequently, the conditional probability function generates the conditional probability of each BEV grid, denoted as $\hat{\mathbf{p}}^{(i)}$. Then, we compute ray-wise features using a feature expectation function, and finally, the resulting ray-wise features are multiplied with the conditional probability map to yield the geometric feature space.

B. Implementation Details

Latent Rendering. We now delve into the specific implementation details of the Latent Rendering operator. As shown in Figure 1, the Latent Rendering operator comprises a total of 4 steps. In the first step, given the BEV embeddings from the Latent Rendering, we use a projection layer with output channels of G to estimate independent probability maps:

$$\mathbf{p} = \texttt{Projection}(\mathcal{F}_{\text{bev}}), \tag{1}$$

where G is the group number for multi-group Latent Rendering. To simplify, we focus on where $G = 1$. In cases of multiple groups, we can easily divide $\mathcal{F}_{\text{bev}}$ into G parts along the channels, and apply Latent Rendering on each part accordingly.

In the second step, we compute the conditional probability of each BEV grid along its respective ray. As shown in step-(a.) of Figure 1, for the i-th BEV grid with coordinates $\mathbf{g}^{(i)} = \{x_i, y_i\}$, we begin by casting a ray from the origin point $\mathbf{o}$, typically the center of BEV feature maps, towards the target BEV grid. The direction of the ray is determined as: $\mathbf{d}^{(i)} = (\mathbf{g}^{(i)} - \mathbf{o})/(||\mathbf{g}^{(i)} - \mathbf{o}||_2)$. Subsequently, we collect a set of prior waypoints along the ray, which are closer to the origin point in distance compared to BEV grid i. These waypoints are selected at uniform distance intervals, termed as λ, as:

$$\mathbf{x}^{(i,j)} = \mathbf{o} + j\lambda\mathbf{d}^{(i)}, \text{ where } j\lambda < ||\mathbf{g}^{(i)} - \mathbf{o}||_2. \tag{2}$$

Here, j represents the index of different waypoints, and the condition $j\lambda < ||\mathbf{g}^{(i)} - \mathbf{o}||_2$ ensures that the distance of these waypoints from the origin point is smaller than that of the corresponding BEV grid. An example is depicted in step-(b.) of Figure 1, where we collect 4 prior waypoints with coordinates as $\mathbf{x}^{(i,j)}$ where $j \in \{0, 1, 2, 3\}$ for the i-th BEV grid. Moving on to step-(c.) of Figure 1, we compute the conditional probability of the target BEV grid based on the independent probability of its prior waypoints. In general, the conditional probability function is defined as:

$$\hat{\mathbf{p}}^{(i)} = \left[\prod_j (1 - \mathbf{p}^{(\mathbf{x}^{(i,j)})})\right] \mathbf{p}^{(\mathbf{g}^{(i)})}, \tag{3}$$

where $\mathbf{p}^{(:)}$ is the bilinear interpolation function to compute the corresponding probability of the position $(:)$ from $\mathbf{p}$.

In the third step, we compute the ray-wise features according to the BEV embedding and conditional probability map:

$$\hat{\mathcal{F}}^{(i)} = \sum_{k} \hat{\mathbf{p}}^{(i,k)} \mathcal{F}_{\text{bev}}^{(k)}, \text{ where } \mathbf{d}^{(k)} = \mathbf{d}^{(i)}, \tag{4}$$

where k indexes those BEV grids in the same direction as the i-th BEV grid.

As illustrated in Figure 1, after the feature expectation function, all BEV grids lying on the same ray share the same global feature embeddings. Therefore, finally, in order to highlight those BEV grids with higher conditional probability, we multiply the ray-wise features and the conditional probability map to obtain the geometric feature responses:

$$\hat{\mathcal{F}}_{\text{bev}} = \hat{\mathbf{p}} \cdot \hat{\mathcal{F}}. \tag{5}$$

In our concrete implementation, we set G and λ as 16 and 1, respectively.

C. Additional Ablative Studies

C.1. Structure of Latent Rendering

In Table 1, we investigate the effectiveness of the conditional probability function (Eq. (3), Step-2 in Figure 1) and the feature expectation function (Eq. (4), Step-3 in Figure 1). For context, we use BEVFormer-small [5] with ImageNet-cls pre-training as the baseline. We evaluate different functions by first pre-training the baseline using ViDAR under various structures and then comparing the downstream 3D detection performance. The baseline result, 40.20% NDS, is obtained by pre-training the model using the straightforward pipeline, mentioned in Section 3.2 of the main paper, which comprises the History Encoder, Future Decoder, and differentiable ray-casting.

Cond. Prob. Func., Eq. (3)	Feat. Exp. Func., Eq. (4)	NDS (%)
-	-	40.20
√	-	47.34
√	√	**47.58**

Table 1. **Ablation on each component of Latent Rendering design.** "Cond. Prob. Func." and "Feat. Exp. Func." represent the conditional probability function and the feature expectation function, respectively.

To evaluate the effectiveness of the conditional probability function solely, we directly multiply the BEV embeddings $\mathcal{F}_{\text{bev}}$ with the aggregated conditional probability map $\hat{\mathbf{p}}$ to obtain the features for future point cloud prediction. As illustrated in the second row of Table 1, the conditional probability function greatly alleviates the ray-shaped feature issue, and brings a 7.14% NDS improvement compared to the simple baseline. Then, by introducing the feature expectation function, the Latent Rendering operator integrates the entire ray-wise feature, which further brings downstream performance improvements.

C.2. Future Predictions

In this study, we aim to showcase the benefits of future prediction in downstream tasks that involve temporal modeling. We conduct ablation studies on the task of Multi-Object Tracking, as it necessitates the model to associate 3D objects across different frames, thereby reflecting its temporal modeling capabilities. For our experiments, we use the first stage of UniAD [2] as the tracking model, and assess its performance on downstream tracking after being pre-trained with ViDAR using varying frames of future point cloud supervision.

No. of Future Supervision	Detection NDS (%) ↑	Tracking AMOTA (%) ↑
0	54.04	42.12
1	53.73	43.79
3	53.60	44.74
6	53.81	**45.10**

Table 2. **Ablation on future supervision.** More future supervisions during the pre-training stage bring consistent improvements in tracking performance.

As shown in Table 2, we evaluate four settings which utilize "0", "1", "3", and "6" future frames to supervise ViDAR when visual point cloud forecasting pre-training. The setting labeled as "0 future supervision" implies that we solely utilize ViDAR to reconstruct LiDAR point clouds for the current frame, without any future prediction and supervision. As illustrated, we observe consistent improvements in tracking performance when using more future frames for supervision. This observation highlights the beneficial impact of visual point cloud forecasting on the temporal modeling capabilities of downstream models.

C.3. Pre-train Structure

In Table 3, we investigate the effect of ViDAR pre-training on different components of downstream BEV encoders. We achieve this by loading different sets of pre-trained model parameters during the downstream fine-tuning process. For context, in this ablation study, we use BEVFormer-small as the downstream model for 3D detection. The baseline performance is 44.11% NDS.

2D Backbone	FPN-neck	View-Transform	NDS (%)
-	-	-	44.11
√	√	-	44.74
√	√	√	47.58

Table 3. **Ablation on pre-training components.** ViDAR mostly benefits the view-transformation part of visual BEV encoders.

As illustrated, the improvements primarily stem from the pre-training of the view-transformation component, which is responsible for extracting BEV features from perspective multi-view image features. It is reasonable considering that the view-transformation module plays a crucial role in correlating 2D features with 3D geometries and scene structures, as discussed in previous works [4]. Furthermore, this highlights the compatibility of ViDAR pre-training with any advanced 2D image pre-training techniques, which could lead to consistent performance improvements when combined with more advanced image backbones, as demonstrated in Table 3 and Table 4 of the main paper.

D. Qualitative Results

D.1. Latent Rendering

Figure 2 presents the effectiveness of the Latent Rendering operator in formulating geometric features from visual sequence inputs. Each pair of images in Figure 2 displays the ground-truth point cloud on the left and visualizes the features, denoted as $\hat{\mathcal{F}}_{\text{bev}}$, on the right. As depicted, ViDAR successfully captures the 3D geometry from multi-view visual sequences and effectively extracts geometric features that accurately represent the underlying 3D world in the latent space.

Next, in Figure 3, we compare the BEV features, $\mathcal{F}_{\text{bev}}$, extracted by BEV encoders pre-trained using the differentiable ray-casting baseline (depicted in the middle) and our ViDAR (depicted on the right). As illustrated, the ray-casting baseline encounters the issue of ray-shaped features after pre-training. In contrast, our ViDAR, thanks to the inclusion of Latent Rendering during the pre-training procedure, effectively highlights the responses of BEV grids preserving geometric information. Consequently, it extracts discriminative features after pre-training, which, in turn, benefits downstream fine-tuning.

D.2. Visual Point Cloud Forecasting

In Figure 4, we provide visual examples of ViDAR for predicting future point clouds based on historical visual images. The historical visual inputs captured within a 1-second timeframe are displayed in the upper portion, while the corresponding predicted future point clouds spanning 3 seconds are shown in the lower portion.

In the first row, we present an example of the ego-vehicle executing a left turn. As observed, ViDAR adeptly captures the related position and orientation between the ego-vehicle and the parked blue bus in its future predictions. Moving on to the second and third rows, we illustrate instances where ViDAR successfully captures the relative motions between the ego-vehicle and other moving objects, such as the yellow bus in the second row and the white car in the third row. By analyzing the LiDAR outputs, ViDAR accurately understands that moving objects exhibit faster movement compared to the ego-vehicle. Consequently, it estimates their positions with increasing relative distances as time progresses. Then, the fourth row showcases an example of the ego-vehicle executing a right turn. This case demonstrates ViDAR's effective modeling of the road map based on the historical visual sequences. It is important to note that all LiDAR point cloud visualizations are presented within the coordinate space of the ego-vehicle, with the ego-vehicle situated at the center of the 3D space.

Additionally, in Figure 5, we demonstrate the capability of ViDAR to simulate various future point clouds based on specific ego-vehicle motions, such as turning left, going straight, and turning right. This showcases the potential of ViDAR as a visual world model for autonomous driving, where it utilizes visual image inputs to generate simulated future point clouds. Such simulations can be valuable for model-based reinforcement learning for training vision autonomy in an unsupervised manner.

D.3. End-to-end Autonomous Driving

Lastly, in Figure 6, we present a comparison between UniAD with and without ViDAR pre-training for end-to-end autonomous driving. As illustrated, the inclusion of ViDAR pre-training enables UniAD to generate more precise future trajectories for other moving objects (highlighted in **red circles**). This improved prediction accuracy plays a crucial role in enhancing the planning process for safety-critical end-to-end autonomous driving scenarios.

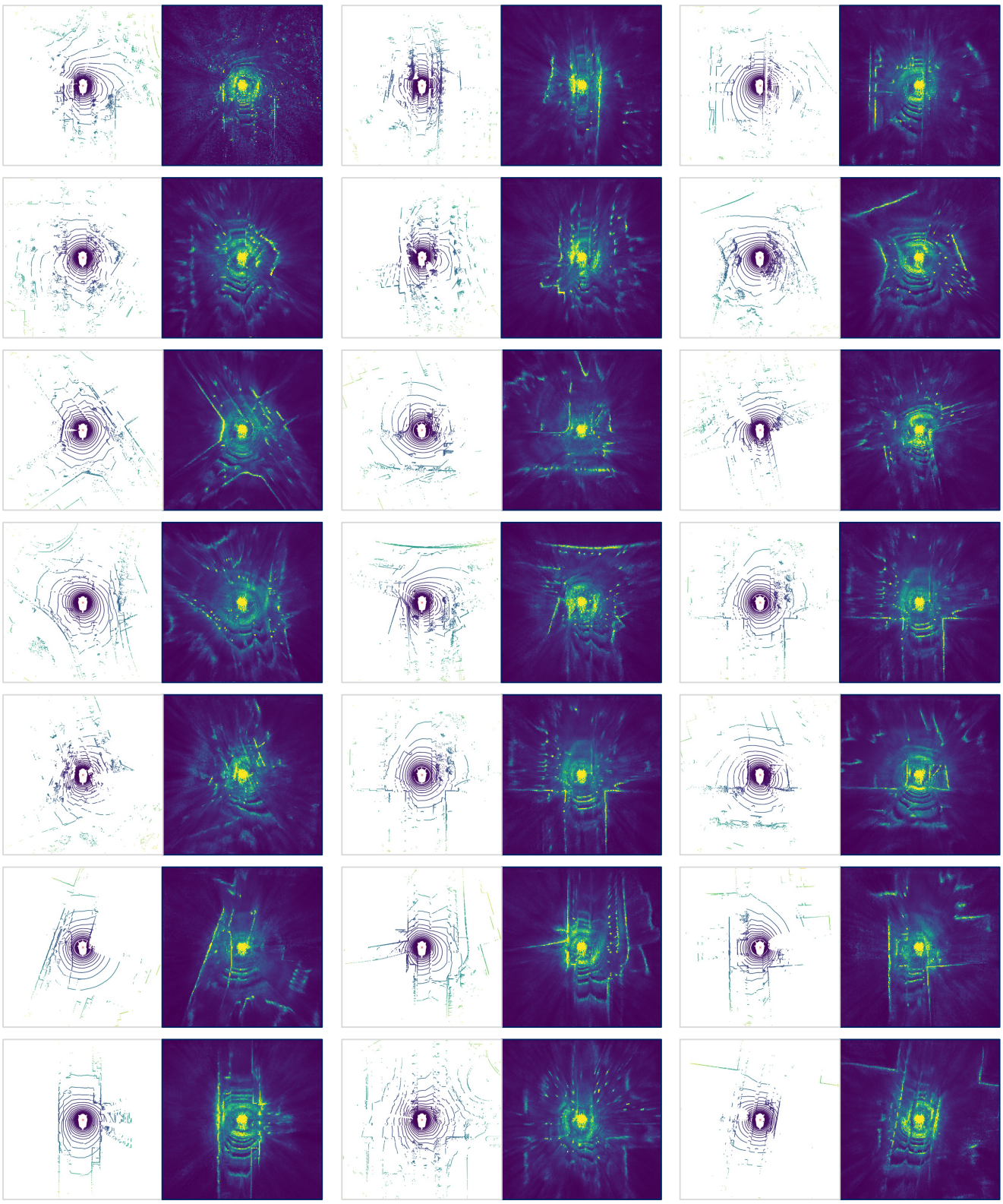

Figure 2. **Visualization of geometric features from the Latent Rendering operator.** In each pair, we present the ground-truth LiDAR point clouds on the left and the BEV features produced by ViDAR from multi-view image inputs on the right. It is evident that, utilizing the Latent Rendering operator, ViDAR successfully captures the underlying 3D geometry in the latent space, resulting in precise descriptions of the 3D world through feature responses.

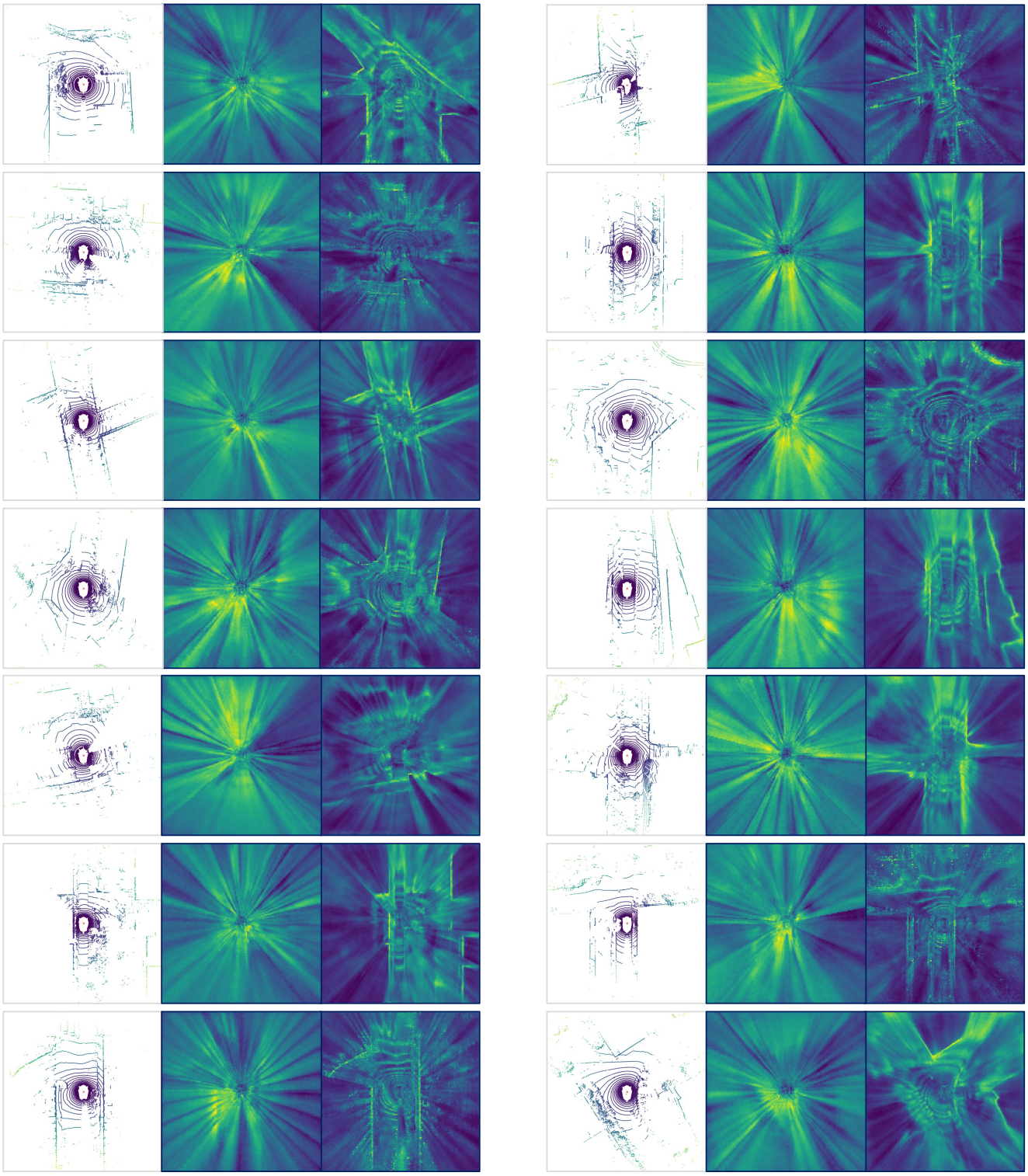

Figure 3. **Visualization of BEV features from visual BEV encoder pre-trained by the differentiable ray-casting baseline and ViDAR with Latent Rendering operator.** In each triplet, we present the ground-truth point cloud on the left, the features pre-trained by the differentiable ray-casting baseline in the middle, and the features pre-trained by ViDAR on the right. As illustrated, the integration of the Latent Rendering operator in ViDAR proves advantageous as it successfully mitigates the occurrence of ray-shaped features during visual point cloud forecasting pre-training. Consequently, ViDAR empowers the BEV encoders to extract informative and discriminative BEV features from visual sequence inputs.

Figure 4. **Qualitative results of ViDAR for visual point cloud forecasting on nuScenes validation set.** Top: historical visual inputs in 1 second; bottom: future point cloud predictions in 3 seconds. The first, second, and third rows provide examples of modeling relative motions between the ego-vehicle and other parked or moving objects (highlighted in **red circles**); the fourth row shows the future estimation where the ego-vehicle is turning right. As illustrated, ViDAR effectively captures the information of 3D geometry and temporal dynamics, and thus correctly estimates future point clouds from visual sequence inputs. All point cloud visualizations are under the ego coordinates.

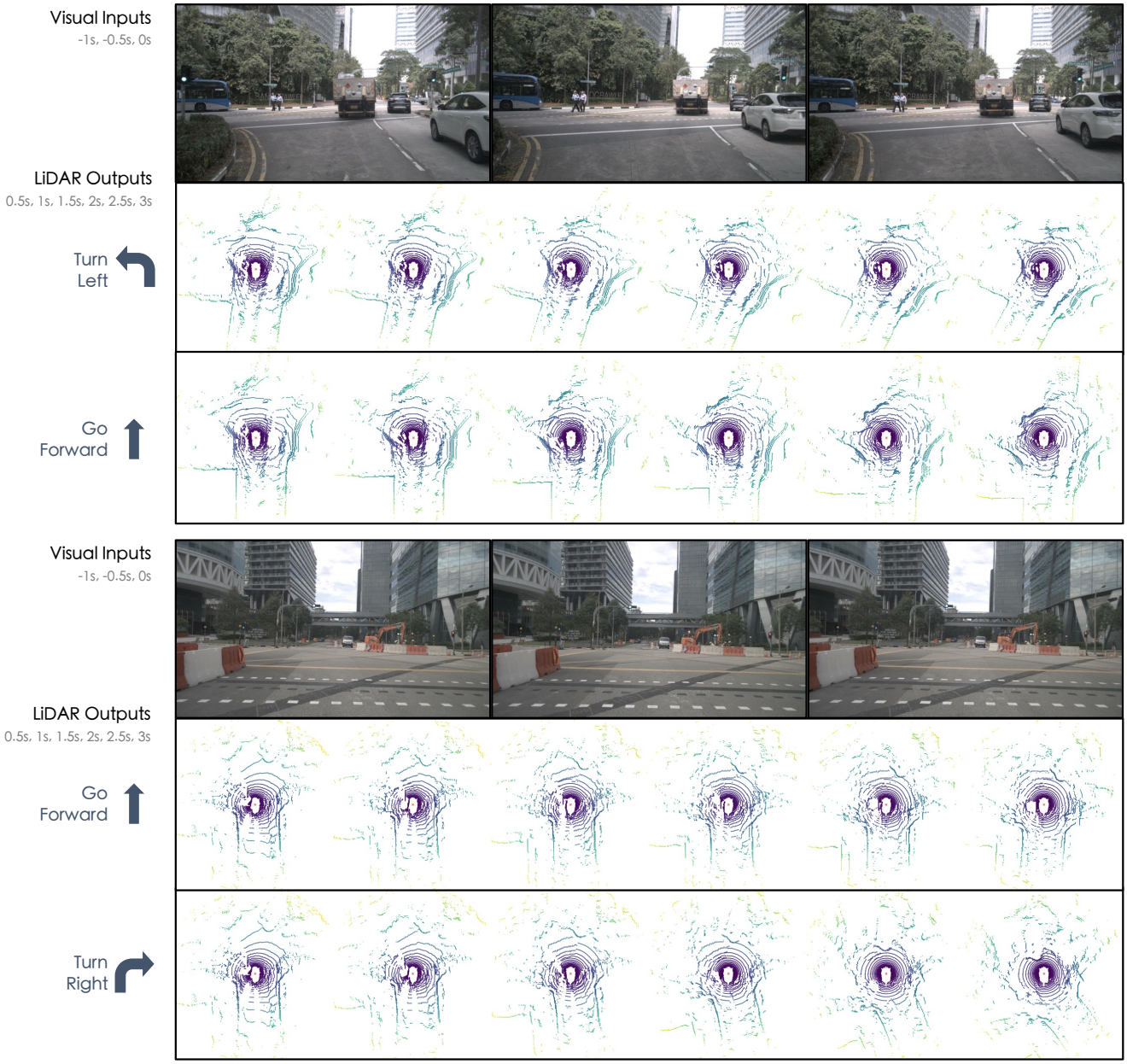

Figure 5. **Qualitative results of ViDAR conditioned on different future ego-vehicle motion conditions.** ViDAR can simulate different future point clouds based on the specific future ego-vehicle conditions. All point cloud visualizations are under the ego coordinates.

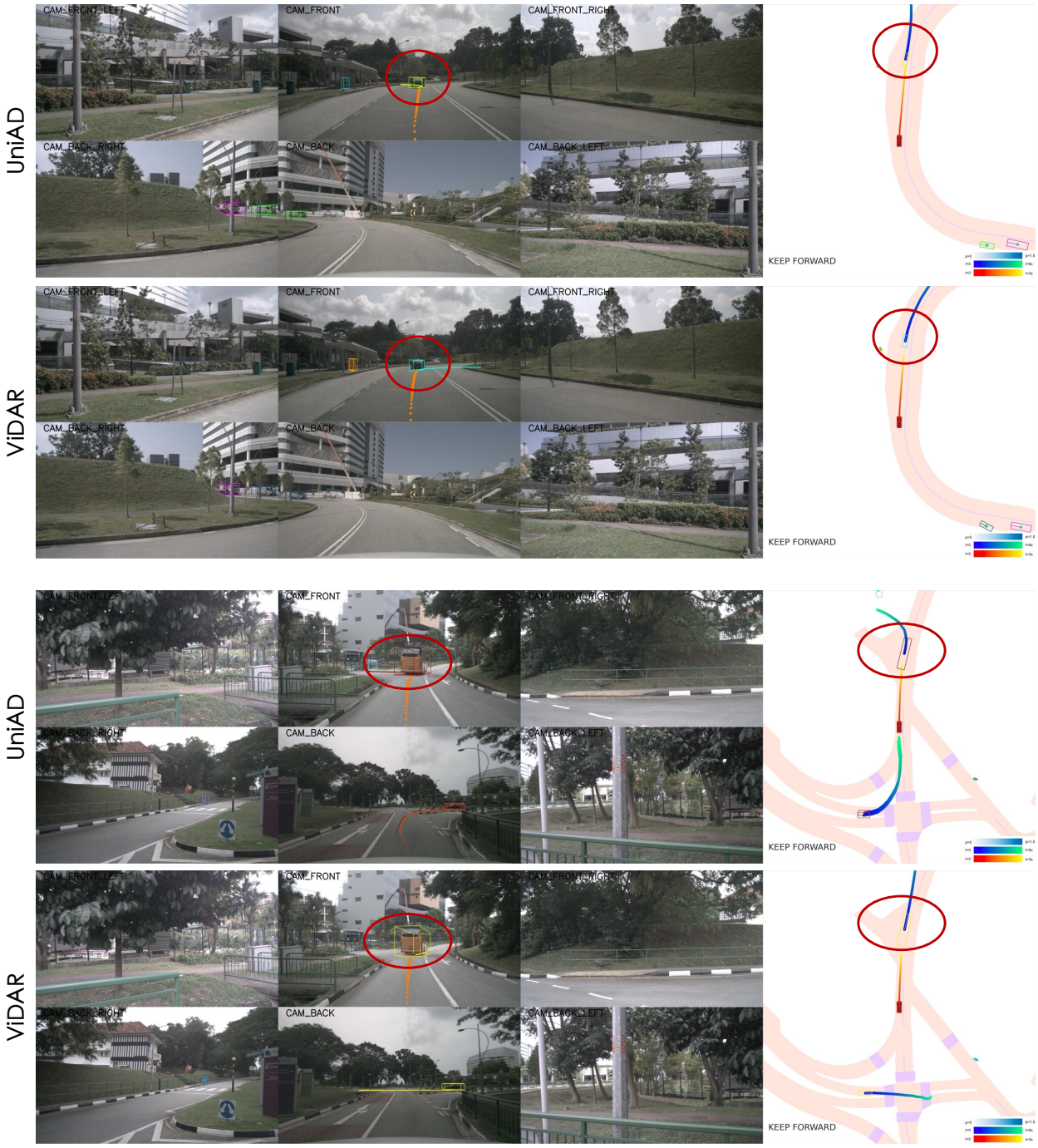

Figure 6. **Qualitative comparisons between UniAD with and without ViDAR pre-training.** ViDAR effectively models the temporal dynamics by estimating future points during pre-training, which in turn improves UniAD in predicting the future motion of moving objects.

References

[1] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving. *arXiv preprint arXiv:2309.17080*, 2023. 1

[2] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented Autonomous Driving. In *CVPR*, 2023. 3

[3] Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. DrivingDiffusion: Layout-Guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*, 2023. 1

[4] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. BEVDepth: Acquisition of Reliable Depth for Multi-view 3D Object Detection. In *AAAI*, 2023. 4

[5] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. In *ECCV*, 2022. 3

[6] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. DriveDreamer: Towards Real-world-driven World Models for Autonomous Driving. *arXiv preprint arXiv:2309.09777*, 2023. 1

[7] Lunjun Zhang, Yuwen Xiong, Ze Yang, Sergio Casas, Rui Hu, and Raquel Urtasun. Learning unsupervised world models for autonomous driving via discrete diffusion. *arXiv preprint arXiv:2311.01017*, 2023. 1