DLWM: Dual Latent World Models enable Holistic Gaussian-centric Pre-training in Autonomous Driving

Supplementary Material

A. Evaluation Metrics

3D Occupancy Perception. The primary metrics for 3D occupancy perception are the mean Intersection over Union (mIoU) and the Intersection over Union (IoU):

$$\text{mIoU} = \frac{1}{|\mathcal{C}'|} \sum_{i \in \mathcal{C}'} \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{1}$$

$$\text{IoU} = \frac{TP_{\neq c_0}}{TP_{\neq c_0} + FP_{\neq c_0} + FN_{\neq c_0}}, \tag{2}$$

where $\mathcal{C}'$ is the set of non-empty classes, c_0 is the empty class, and TP_i, FP_i, and FN_i are the number of true positives, false positives, and false negatives for class i, respectively. $TP_{\neq c_0}$, $FP_{\neq c_0}$, and $FN_{\neq c_0}$ are the aggregation of these values across all non-empty classes.

4D Occupancy Forecasting. For the forecasting task, we assess the model's ability to predict scene evolution over a future horizon of 3 seconds, corresponding to 6 frames (at 2Hz). The final metrics are reported as the average of the mIoU and IoU scores computed across these 6 future timestamps:

$$\text{mIoU}_{4D} = \frac{1}{T} \sum_{t=1}^{T} \text{mIoU}_t, \tag{3}$$

$$\text{IoU}_{4D} = \frac{1}{T} \sum_{t=1}^{T} \text{IoU}_t, \tag{4}$$

where T is the total future time steps, mIoU_t and IoU_t denote the metrics calculated at the t-th future frame.

Motion Planning. For the motion planning task, we evaluate the quality of the predicted ego-vehicle trajectories using the L2 error and collision rate.

- **L2 Error.** The average Euclidean distance between the predicted waypoints $\hat{T}_t$ and the ground-truth waypoints T_t over the future horizon N_f:

$$\text{L2} = \frac{1}{N_f} \sum_{t=1}^{N_f} ||\hat{T}_t - T_t||_2 \tag{5}$$

- **Collision Rate.** A collision occurs if the ego-vehicle's bounding box, at future timestep N_f intersects with the set of all ground-truth obstacle voxels. The final metric is the percentage of scenes where a collision is detected.

B. Implementation Details

Details of Gaussian Perception Module. The Gaussian perception module is formulated to construct a compact and high-fidelity 3D scene representation $\mathcal{G}$ parameterized by a set of 3D Gaussians from multi-view image observations. The architecture is composed of an image encoder and a Gaussian transformer decoder.

The image encoder $\mathcal{E}$ takes the multi-view images $\mathcal{I} = \{\mathbf{I}_i \in \mathbb{R}^{3\times H\times W} | i = 1, ..., N\}$ as input. For each view, a potent backbone (e.g., ResNet-101) first encodes the images into multi-level features F'. A Feature Pyramid Network (FPN) then refines F' to generate the final multi-scale image features F, which capture both semantic and spatial details.

Central to the Gaussian perception module is a Gaussian transformer decoder. The 3D scene is represented by K distinct, learnable Gaussian queries. Each query k is initialized with a learnable Gaussian anchor $g_k \in \mathbb{R}^{K\times C}$, which constitutes the learnable 3D geometric primitive, and its associated query features $q_k \in \mathbb{R}^{K\times D}$, initially set as zero vectors. C and D are the dimension of Gaussian primitives and query features respectively. These query features q_k are guided by their corresponding anchors g_k to interact with each other via a self-encoding block, and interact with the image features F via deformable cross-attention to predict the Gaussian attributes.

Specifically, the self-encoding block applies 4D sparse convolution to capture the relations of Gaussians within the current frame, and propagates temporal Gaussian queries from the previous frame in a stream manner. The deformable cross-attention aggregates the current image information from 2D image to 3D Gaussian queries via camera intrinsics $\mathcal{K}$ and extrinsics $\mathcal{T}$. Finally, an MLP-based Gaussian head is applied to each Gaussian query, and Gaussian parameters are iteratively refined across decoder layers including position μ, scale S, rotation R, opacity α, and semantics c.

Details of Latent World Model with Gaussian Flow. To further illustrate the latent world model guided by Gaussian flow, we provide a pseudo code in Alg. 1. Given the current Gaussian G_t, we aim to predict latent BEV $\hat{B}_{t+1}$ at the next time step. Firstly, we estimate the Gaussian flow $\Delta\mu^t$ by feeding the current Gaussian query features q_t to the flow head. Then in the next frame, we obtain the mean μ^{t+1} of Gaussians by propagating dynamic flow $\Delta\mu^t$ to the next frame via the ego motion alignment $T_{ego}^{t\to t+1}$: $T_{ego}^{t\to t+1}(\mu_k^t + \Delta\mu_k^t)$. To predict the Gaussian set $\hat{G}_{t+1}$, we update the

Algorithm 1 Latent World Model with Gaussian Flow.

Input: $G_t = \{\mu^t, \Sigma^t, \alpha^t, c^t, q^t\}, T_{ego}^{t \to t+1}$
Output: $\hat{B}_{t+1}$
 # predict the absolute flow for each Gaussian
$\Delta\mu^t \leftarrow$ **FlowHead**(q^t)
 # apply flow and align ego motion
$\mu^{t+1} \leftarrow T_{ego}^{t \to t+1}(\mu^t + \Delta\mu^t)$
 # update Gaussian set
$\hat{G}_{t+1} \leftarrow \{\mu^{t+1}, \Sigma^t, \alpha^t, c^t, q^t\}$
 # predict latent BEV by rasterization
$\hat{B}_{t+1} \leftarrow$ **Rasterize**$(\hat{G}_{t+1})$
$\hat{B}_{t+1}$

Algorithm 2 Latent World Model with Ego Planning.

Input: G_t, Q_{wp}
Output: $\hat{B}_{t+1}$
 # obtain current latent BEV by rasterization
$B_t \leftarrow$ **Rasterize**(G_t)
 # extract scene queries from BEV
$Q_{scene}^t \leftarrow$ **ExtractQueries**(B_t)
$Q_{scene}^t \leftarrow$ **SelfAttn**(Q_{scene}^t)
 # predict ego trajectory
$Q'_{wp} \leftarrow$ **CrossAttn**(Q_{wp}, Q_{scene}^t)
$\hat{T} \leftarrow$ **MLP**(Q'_{wp})
 # motion-aware layer norm conditioned on $\hat{T}$
$Q_{norm} \leftarrow$ **MLN**$(Q_{scene}^t, \hat{T})$
 # predict next scene queries
$Q_{scene}^{t+1} \leftarrow$ **SelfAttn**(Q_{norm})
 # reconstruct future BEV
$\hat{B}_{t+1} \leftarrow$ **Fuse**(Q_{scene}^{t+1}, B_t)
$\hat{B}_{t+1}$

current set G_t with the estimated mean μ^{t+1} while all the others remain the same. Finally, to obtain the next latent BEV $\hat{B}_{t+1}$, we project the predicted $\hat{G}_{t+1}$ to 2D BEV plane by query feature rasterization [2].

Details of Latent World Model with Ego Planning. Regarding the latent world model guided by ego planning, we list the pseudo code step by step in Alg. 2. Our target is to predict the next latent BEV $\hat{B}_{t+1}$ based on the current Gaussian G_t coupled with ego planning. Firstly, we rasterize current Gaussian G_t to latent BEV B_t [2]. To improve computational efficiency, we extract scene queries Q_{scene}^t from the current BEV B_t motivated by [1]. The following self-attention module is used to capture global context. To predict ego trajectory, the randomly initialized waypoint queries Q_{wp} interact with Q_{scene}^t through cross attention, followed by a MLP head for predicting ego trajectory $\hat{T}$. Then we apply a Motion-Aware Layer Normalization (MLN) [3] layer for conditioning scene queries Q_{scene}^t on the predicted trajectory $\hat{T}$. The obtained Q_{norm} predicts the next scene queries Q_{scene}^{t+1} by self attention. Finally, we predict future BEV $\hat{B}_{t+1}$ by fusing Q_{scene}^{t+1} with the current BEV B_t following [1].

Table 1. **Ablation study on "Dual" and "Unified".**

Model	mIoU (%) ↑	L2 (m) ↓	Col. (%) ↓
Unified	18.9	0.58	0.22
Dual	**19.3**	**0.46**	**0.19**

Table 2. **Ablation study on the number of 3D Gaussians.**

Nums	mIoU (%) ↑	IoU (%) ↑
7500	18.35	28.83
25600	19.30	**30.56**
51200	**19.43**	29.79

Table 3. **Ablation study on the number of future frames.**

Nums	mIoU (%) ↑	IoU (%) ↑
0	18.83	30.12
1	**19.30**	**30.56**
2	19.23	30.52
4	19.06	29.20

Table 4. **The computation cost analysis of all three tasks.**

Task	Memory	Latency
Perception	4.4GB	327ms
Forecasting	6.8GB	684ms
Planning	4.2GB	274ms

C. Additional Experimental Results

Comparing Dual Models with Unified Model. The motivation of using two world models is that decoupling the training of Gaussian flow and ego planning is imperative for reducing learning complexity.

As illustrated in Eq.5 (main paper), the predicted Gaussian mean is propagated through a combination of Gaussian flow and ego-motion alignment. For the Gaussian-flow-guided model, leveraging GT ego motion reduces the learning complexity of Gaussian flow. Conversely, explicitly integrating Gaussian flow learning into the motion planning model would increase the latter's learning burden. This is because self-supervised Gaussian flow lacking explicit supervision introduces unnecessary complexity to the planning, since the two items are coupled as shown in Eq.5 (main paper).

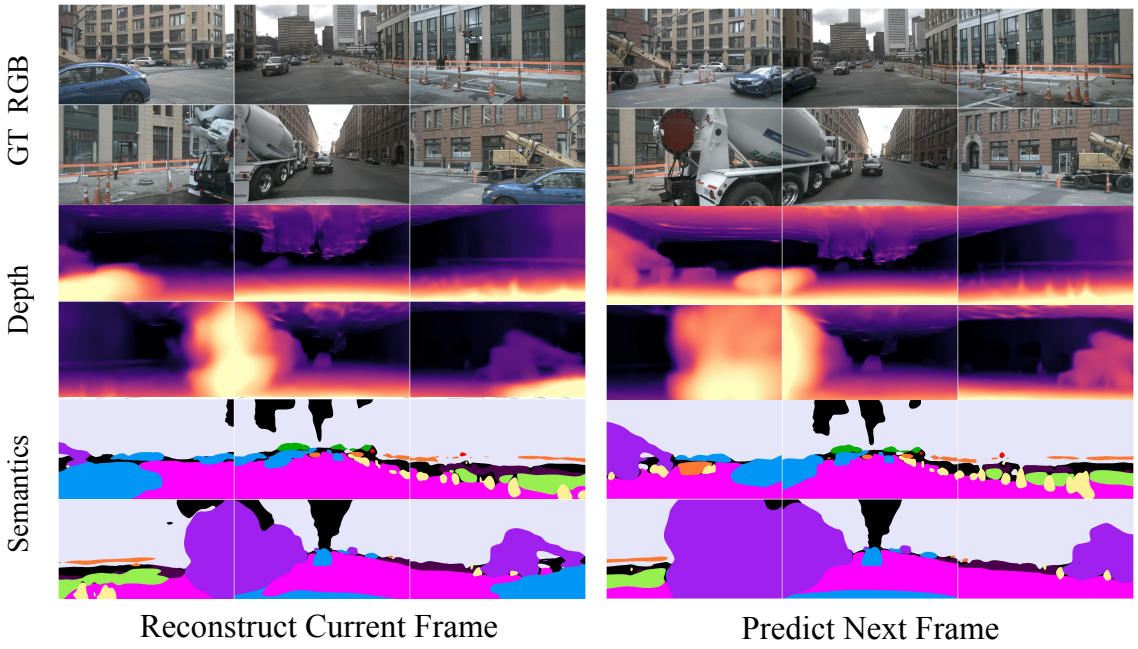

Figure 1. **Qualitative results of depth and semantic reconstruction and forecasting during the pre-training.** The left column shows the depth and semantic reconstruction at the current frame while the right column illustrates the depth and semantic images rendered by predicted Gaussians at the next frame compared with the GT RGB images.

To verify this, we supplement an ablation experiment comparing the "Dual" models with the "Unified" model, where the predicted ego motion is utilized for motion transformation in place of GT in Eq. **??** for unifying learning of both flow and planning. The results in Tab. 1 confirm that the decoupled design achieves superior performance in perception and planning, justifying its necessity.

Number of Gaussians. We conduct an ablation study (see Tab. 2) to evaluate the influence of the number of 3D Gaussian queries on the 3D occupancy perception performance. Testing configurations of 7,500, 25,600, and 51,200 Gaussians, the results reveal a trade-off: The mIoU (19.43) peaks at 51,200 Gaussians, indicating superior semantic granularity and expressive power. The IoU (30.56) achieves its best geometric accuracy with 25,600 Gaussians, suggesting this configuration strikes the optimal balance between density and reconstruction efficiency. We prioritize the configuration with 25,600 Gaussians for our final model.

Number of Future Frames. We investigate the influence of the predicted number of future frames for pre-training stage 2 regarding Gaussian-flow-guided latent world model. 3D occupancy perception is selected for an ablation study. As shown in Tab. 3, we predict future Gaussians with four settings including "0", "1", "2", and "4" future frames. The setting "0" denotes that we only reconstruct images without future prediction. By incorporating the prediction of the future frame $N = 1$, the performance peaks at 19.30 mIoU and 30.56 IoU, demonstrating the beneficial impact of the Gaussian-flow-guided latent world model. The inclusion of additional future frames may slightly impair performance, potentially attributable to the greater scene changes caused by more distant future frames, which complicates accurate future prediction.

Inference Latency and Memory Analysis. The dual world models are only used during the pre-training to facilitate representation learning in Gaussian perception module. For the inference stage, only the perception module and each task-specific head are used. Thus, the pre-training does not introduce additional inference latency. We measure inference latency and memory for each task on one GPU with batch size one. The evaluated results in Tab. 4 show lightweight memory and latency cost.

D. BEV Supervision Discussion

BEV rasterization is used only for temporal supervision, not as the representation itself. Crucially: 1. All geometric reasoning happens in 3D Gaussian space during both pre-training and inference. 2. BEV features are derived from 3D Gaussians via differentiable rasterization that preserves height information through vertical stacking (following GaussianLSS [2]).

E. Visualization Results

Depth and Semantic Reconstruction and Forecasting. To validate the effectiveness of our holistic Gaussian-centric

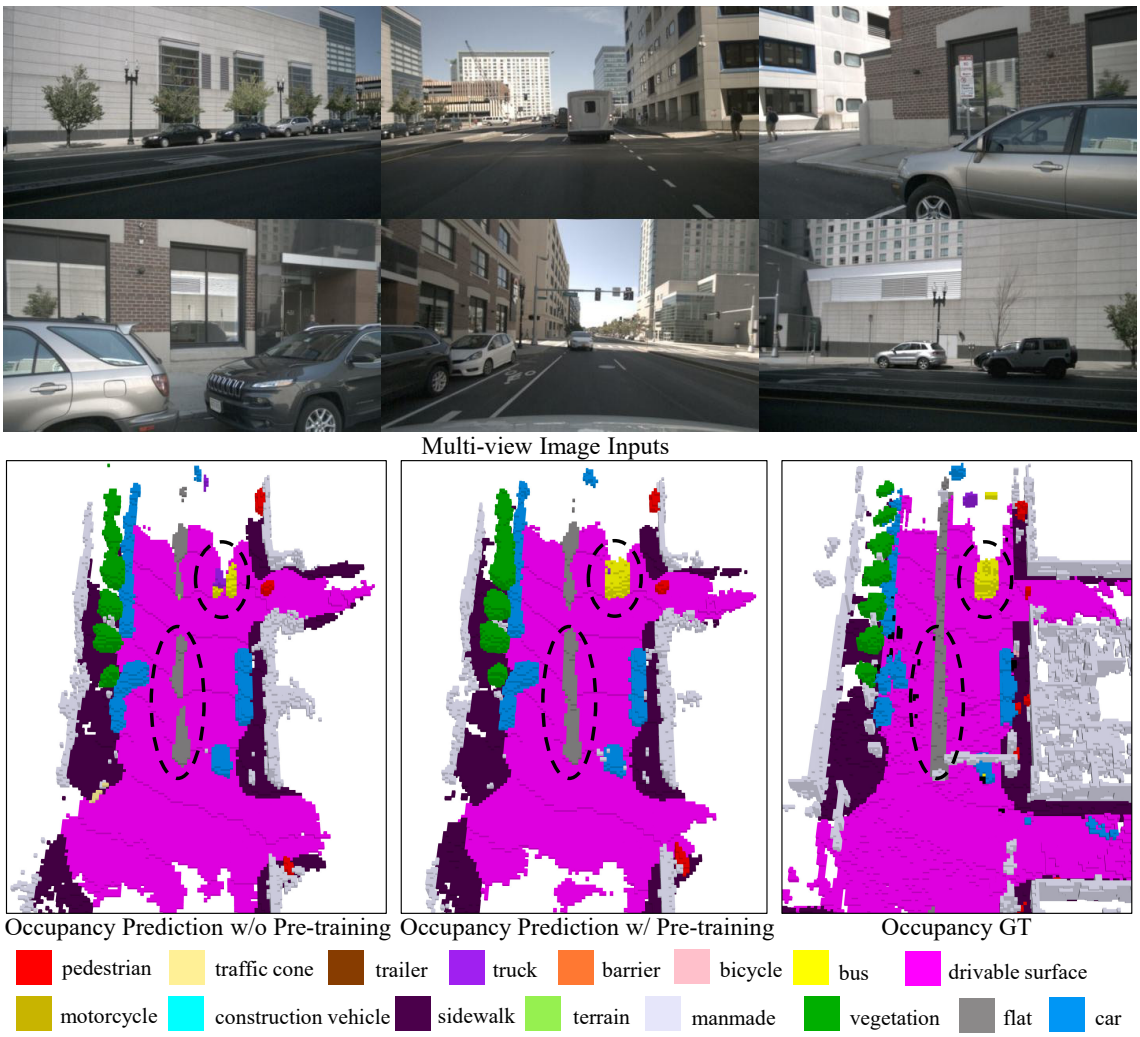

Figure 2. **Qualitative comparison of 3D Occupancy Perception over GT, baseline without pre-training and ours with pre-training.**

pre-training, we present qualitative visualization results for depth and semantic image rendering in Fig. 1. During the pre-training stage, the model is tasked with reconstructing and forecasting the 3D scene geometry and semantics under self-supervision. The rendered depth maps (shown in the middle rows) exhibit sharp boundaries and accurate depth stratification, showing the ability for recovering detailed geometric structures. The rendered semantic maps (shown in the bottom rows) demonstrate high consistency with the semantic priors. Based on the excellent reconstruction performance in the left column of Fig. 1, the model exhibits accurate depth and semantic prediction at the next frame in the right column, meaning that the pre-training paradigm is able to model the temporal evolution for the driving scene.

3D Occupancy Perception. We provide additional visualization results for the 3D occupancy perception task in Fig. 2. DLWM significantly outperforms the baseline without pre-training in both geometric completeness and semantic consistency. While the baseline suffers from fragmented structures and boundary misclassifications due to sparse observations, DLWM generates significantly more complete geometric structures and more accurate semantic classifications (e.g., for vehicles and vegetation) compared to the baseline.

4D Occupancy Forecasting. We display the forecasted occupancy scenes at future time steps (e.g., 1s, 2s, and 3s) in Fig. 3. Compared to the baseline without pre-training, DLWM generates temporally consistent predictions with sharper geometric details. In contrast to the baseline, which produces blurred or static predictions, DLWM is pre-trained with Gaussian-flow-guided latent world model to explicitly predict Gaussian displacement, ensuring accurate trajectory forecasting for dynamic agents.

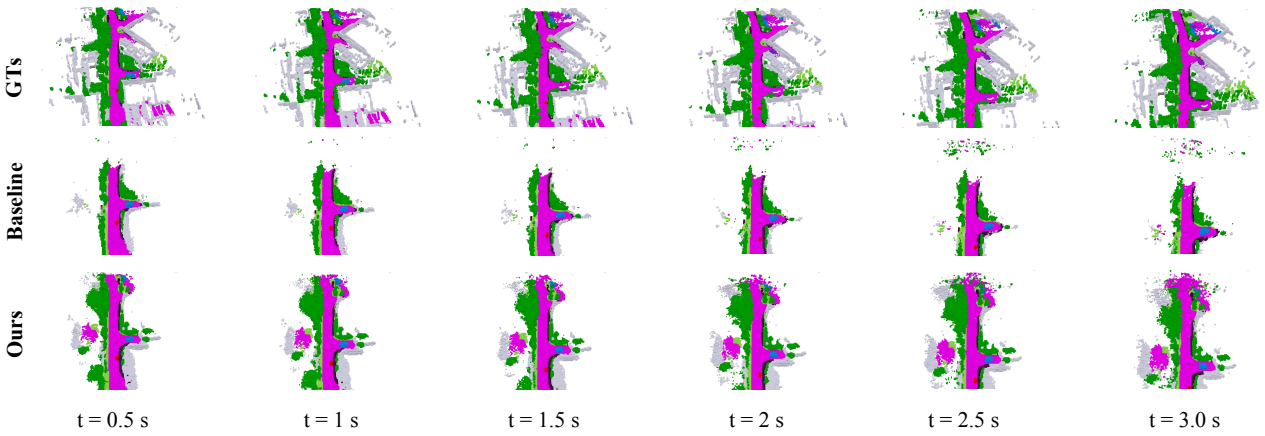

Figure 3. **Qualitative comparison of 4D Occupancy Forecasting over GTs, baseline without pre-training and ours with pre-training.**

References

[1] Peidong Li and Dixiao Cui. Navigation-guided sparse scene representation for end-to-end autonomous driving. *arXiv preprint arXiv:2409.18341*, 2024. 2

[2] Shu-Wei Lu, Yi-Hsuan Tsai, and Yi-Ting Chen. Toward real-world bev perception: Depth uncertainty estimation via gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17124–17133, 2025. 2, 3

[3] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3631, 2023. 2