GAUSSIANDIFFUSER: DIFFUSION-BASED 3D GAUSSIAN PRE-TRAINING FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised pre-training has become a fundamental component for advancing 3D scene understanding in autonomous driving. While existing approaches commonly rely on rendering-based reconstruction to enforce geometric consistency, their objectives remain largely confined to camera-observable surface signals, which can make the learned representations overly dependent on directly visible rendering cues. In this paper, we propose GaussianDiffuser, an efficient feedforward framework that integrates diffusion-based denoising into 3D Gaussian Splatting (3DGS) pre-training. Diffusion-based denoising provides a complementary pre-training objective for 3D Gaussian scene representations by regularizing the model to recover clean signals from corrupted latents, encouraging the representation to exploit multi-view context and spatial consistency rather than locally clean rendering cues. To realize this objective, GaussianDiffuser introduces a diffusion-aware Gaussian block for a denoising objective during pre-training, learning enriched spatial representations with minimal architectural modifications. Experimental results on the nuScenes dataset demonstrate that GaussianDiffuser improves depth reconstruction and occupancy prediction, reducing Abs Rel by 17% and improving Occ3D mIoU by 2.4 over previous state-of-the-art methods.

1 INTRODUCTION

Self-supervised pre-training has emerged as a key strategy for vision-centric 3D scene understanding in autonomous driving, reducing the reliance on costly 3D annotations while improving performance on tasks such as 3D object detection and occupancy prediction. Recent self-supervised approaches commonly rely on rendering-based reconstruction objectives. Early approaches (Yang et al., 2024; Huang et al., 2024a; Zhang et al., 2023) based on Neural Radiance Fields (NeRF) (Mildenhall et al., 2021; Wang et al., 2021; Barron et al., 2022; 2023) learn 3D representations by enforcing photometric and geometric consistency through volumetric ray marching, while more recent methods (Zhang et al., 2025a; Xu et al., 2024b; Gan et al., 2025; Zhang et al., 2025b) adopt 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) for faster differentiable rasterization. In parallel, NeRP3D (Jeong et al., 2026) highlights the importance of continuous representations, demonstrating that anchoring primitives to rigid voxel grids introduces discretization artifacts that degrade pre-training quality.

Despite these advances, rendering-based pre-training remains largely confined to camera-observable surface signals. Although these provide strong photometric and geometric constraints, they can make representations overly dependent on directly visible, locally clean cues. Diffusion-based denoising offers a complementary corruption–recovery regularizer: recovering clean signals from corrupted inputs discourages local copying and encourages the use of context and spatial consistency among primitives. Recent studies show that denoising itself can serve as a pretext task for representation learning (Xiang et al., 2023; Chen et al., 2024), while diffusion-based priors also transfer to perception tasks (Ke et al., 2025; Xu et al., 2024a). In autonomous driving, BEVDiffuser (Ye et al., 2025) refines detection-specific BEV features using ground-truth object layouts, leaving denoising for general 3D scene representation pre-training underexplored.

In this paper, we propose GaussianDiffuser, an efficient feedforward framework that integrates diffusion-based denoising into 3D Gaussian representation learning. GaussianDiffuser regularizes the organization of 2D observations into spatially stable 3D Gaussian primitives by corrupting image latents before Gaussian lifting and recovering clean RGB, depth, and feature signals after differen-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

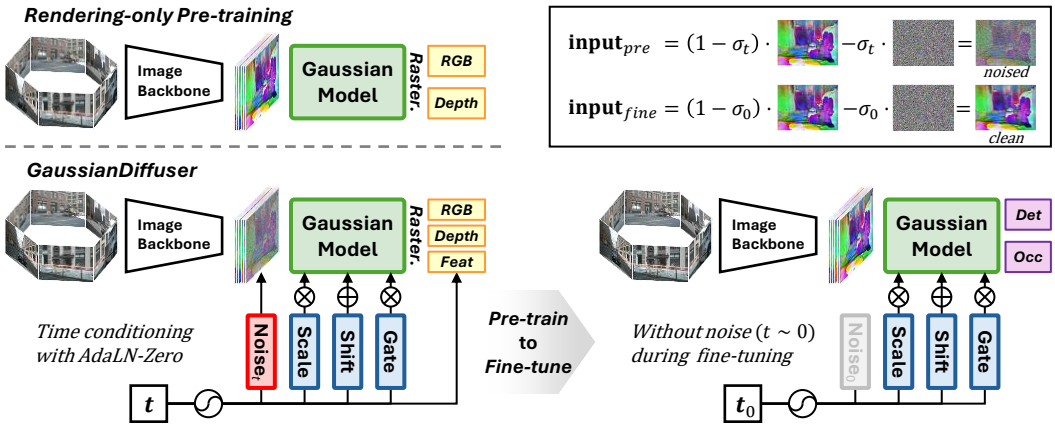

Figure 1: **Comparison of rendering-only pre-training and GaussianDiffuser.** GaussianDiffuser augments rendering-only 3D Gaussian pre-training with noise-corrupted 2D image latents and time conditioning via AdaLN-Zero (Peebles & Xie, 2023). This corruption–recovery objective regularizes Gaussian representation learning during pre-training while preserving the original feedforward pipeline for noise-free fine-tuning.

tiable rasterization. Crucially, GaussianDiffuser treats denoising not as task-specific feature-space refinement, but as a general pre-training objective for 3D Gaussian scene representations. Building on this insight, GaussianDiffuser introduces a diffusion-aware Gaussian block and a denoising objective during pre-training, achieving enriched spatial representations with minimal architectural modifications, as illustrated in Fig. 1. Experimental results on the nuScenes dataset demonstrate that GaussianDiffuser outperforms previous state-of-the-art methods in both scene reconstruction and occupancy prediction.

In summary, our main contributions are as follows:

- We propose GaussianDiffuser, a vision-centric 3DGS pre-training framework built on a 2D latent corruption–3D Gaussian lifting–rasterization-based recovery pipeline.
- We introduce a timestep-conditioned diffusion-aware Gaussian block that refines Gaussian representations derived from corrupted image latents and enables clean RGB, depth, and feature recovery after differentiable rasterization.
- We demonstrate consistent improvements in scene reconstruction and downstream 3D perception with a diffusion-aware Gaussian block and denoising objectives over rendering-only Gaussian pre-training, while requiring minimal architectural modifications.

2 Related Work

2.1 Rendering-based Pre-training and 3D Scene Representations

Self-supervised pre-training through rendering-based reconstruction has become a dominant paradigm for learning 3D representations in autonomous driving. Early methods (Yang et al., 2024; Huang et al., 2024a; Zhang et al., 2023) adopted Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) with volumetric ray marching, while more recent approaches (Xu et al., 2024b; Zhang et al., 2025a; Gan et al., 2025) have shifted toward 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023), which employs differentiable rasterization with explicit attributes for significant efficiency.

Although many rendering-based pre-training methods adopt NeRF or 3DGS, they still tie their representation primitives to discretized structures such as voxel grids (Yang et al., 2024; Huang et al., 2024a; Zhang et al., 2023; 2025a; Gan et al., 2025; Xu et al., 2024b), which can cause a prior mismatch and degrade the quality of learned representations. To resolve this conflict, NeRP3D (Jeong et al., 2026) proposed a point-based detector that learns continuous representations without discretized grids. GaussianFormer (Huang et al., 2024b; 2025) introduced an object-centric 3D Gaussian representation with probabilistic occupancy reasoning decoupled from fixed voxel structures.

SQS (Zhang et al., 2025b) further applied query-based splatting to sparse Gaussian queries for rendering-based pre-training. Although these methods leverage naturally available multi-view observations through photometric or geometric reconstruction, their objectives remain largely driven by reconstructing camera-observable surfaces. Our work complements rendering-based 3DGS pre-training with diffusion-based denoising, encouraging the representation to capture multi-view contextual regularities and spatial consistency across Gaussian primitives.

2.2 Diffusion-based Denoising for Perception

Diffusion models (Ho et al., 2020; Song et al., 2021) learn to capture the underlying data distribution through the denoising process, and the rich prior knowledge acquired during training has proven transferable to a wide range of downstream tasks beyond generation. In 2D perception, Marigold (Ke et al., 2025) and GenPercept (Xu et al., 2024a) demonstrated that diffusion priors trained for image generation retain extensive geometric knowledge transferable to dense prediction tasks such as monocular depth and surface normal estimation. More closely related to our goal, DDAE (Xiang et al., 2023) and *l*-DAE (Chen et al., 2024) have shown that the denoising process itself serves as an effective pretext task for self-supervised representation learning, yielding discriminative intermediate representations and driving representation quality beyond generation-specific mechanisms. These findings motivate our use of diffusion-style denoising as a pre-training regularizer for 3D scene representations.

In autonomous driving, denoising has been explored at different representation levels. DiffBEV (Zou et al., 2024) and DifFUSER (Le et al., 2024) progressively refine task-specific BEV features for perception and multimodal fusion, while BEVDiffuser (Ye et al., 2025) uses ground-truth-guided BEV denoising as a training-time teacher. At the 3D level, S2GO (Park et al., 2026) pre-trains sparse Gaussian queries by perturbing LiDAR-initialized positions and recovering scene geometry, whereas Gau-Occ (Lv et al., 2026) employs LiDAR diffusion to construct geometry-completed Gaussian anchors for multimodal occupancy. Concurrently, HilDA (Wozniak et al., 2026) uses temporal occupancy diffusion as an auxiliary objective for self-supervised LiDAR pre-training. In contrast, GaussianDiffuser corrupts image latents before Gaussian lifting and recovers clean signals after rasterization, using diffusion to regularize a vision-centric 3D Gaussian representation.

3 Method

We present GaussianDiffuser, a pre-training framework that regularizes 3D Gaussian Splatting (3DGS) representations through diffusion-based corruption and denoising for vision-centric autonomous driving. An overview of the proposed framework is illustrated in Fig. 2. In detail, multi-view images are first lifted into 3D Gaussian primitives (Sec. 3.1). During pre-training, noise is injected into the 2D latents before lifting, and a diffusion-aware block denoises the resulting Gaussians conditioned on the timestep (Sec. 3.2). The denoised Gaussians are projected back into the 2D image domain to apply a unified denoising objective (Sec. 3.3). At inference or downstream tasks, the timestep is set to zero and the encoder operates as a standard feedforward model (Sec. 3.4).

3.1 Gaussian Initialization from Multi-view Images

Unlike prior methods (Zhang et al., 2025a; Xu et al., 2024b; Gan et al., 2025) that anchor Gaussian primitives to fixed voxel grids, GaussianDiffuser initializes Gaussians directly using predicted depth distributions from multi-view image features, preserving the continuous nature of 3DGS.

Given multi-view images $\{\mathbf{I}_i\}_{i=1}^{N}$ from N cameras from a single timestep τ with known intrinsics and extrinsics, we first extract per-view feature maps using the image backbone, and apply a lightweight depth network to predict a categorical depth distribution for each latent patch. The latents are lifted to 3D by sampling candidate positions along the camera ray and selecting depths according to the predicted distribution, with a contraction function (Barron et al., 2022) applied to handle unbounded outdoor scenes. The resulting 3D points serve as the initial Gaussian means $\boldsymbol{\mu}_j$, while scale $\mathbf{s}_j$, rotation $\mathbf{q}_j$, opacity α_j, and color $\mathbf{c}_j$ are initialized as Gaussian attributes and subsequently updated by the refinement module. Each Gaussian additionally carries a feature vector $\mathbf{f}_j \in \mathbb{R}^C$, initialized via a positional encoding MLP from the normalized 3D coordinate.

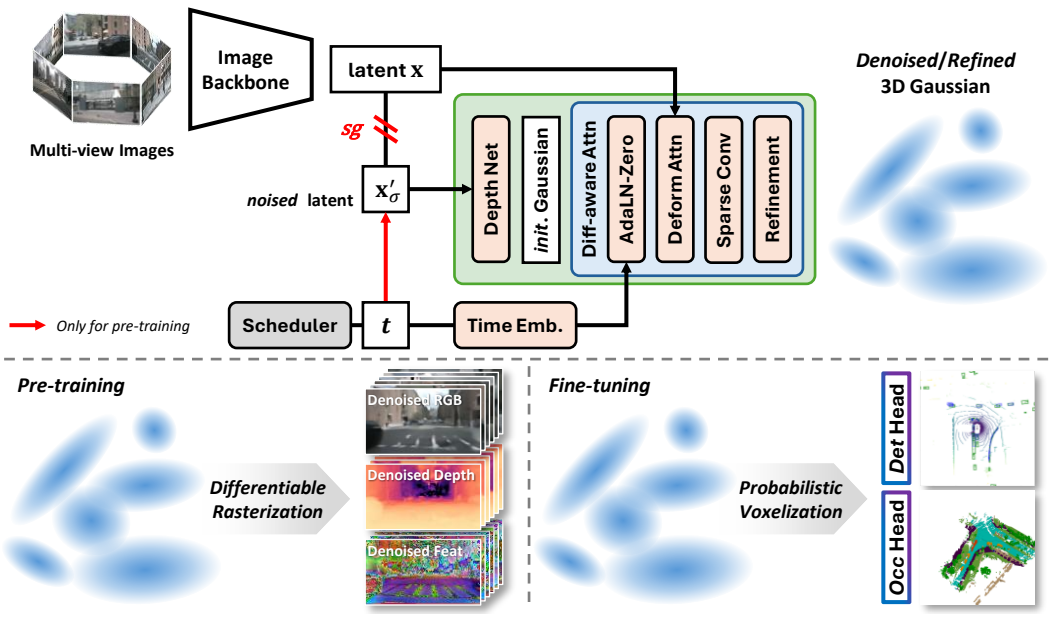

Figure 2: **Overview of GaussianDiffuser.** Pre-training corrupts 2D image latents before Gaussian lifting and refines the resulting Gaussians via a diffusion-aware Gaussian block conditioned on the timestep. Denoised Gaussians are then supervised with 2D photometric, geometric, and feature signals after differentiable rasterization, forming a unified denoising pipeline. Downstream fine-tuning retains the AdaLN-Zero-conditioned encoder with $t = 0$, removes the noise scheduler and projection head, and connects the representation to task-specific heads via probabilistic voxelization.

3.2 Diffusion-based Pre-training via 3D Gaussian Representations

The core contribution of GaussianDiffuser is the integration of a diffusion-based denoising objective that regularizes the 3D Gaussian representation during pre-training. Instead of optimizing only clean-latent rendering reconstruction, GaussianDiffuser corrupts 2D latents before Gaussian lifting and trains the resulting representation to recover clean signals after rasterization, encouraging multi-view context aggregation and spatial consistency across Gaussian primitives.

3.2.1 Forward Process via Flow Matching

We adopt a flow matching formulation (Lipman et al., 2023; Esser et al., 2024) as the corruption process, which defines a continuous probability path between the data distribution and a Gaussian noise distribution. Given clean 2D latent features $\mathbf{x}_0 \in \mathbb{R}^{N\times C\times H\times W}$ extracted from the input, we construct noisy latent features at noise level σ as:

$$\mathbf{x}'_\sigma = (1-\sigma)\,\mathbf{x}_0 + \sigma\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \operatorname{logit}(1-\tilde{\sigma}) \sim \mathcal{N}(\mu_\sigma, s_\sigma^2), \tag{1}$$

where $\tilde{\sigma}$ is mapped to the noise level $\sigma = \alpha\tilde{\sigma}/\big(1+(\alpha-1)\tilde{\sigma}\big)$ via timestep shifting with factor α (Esser et al., 2024), and $\sigma \in (0,1)$ corresponds to the normalized diffusion timestep $t/T = \sigma$, with T diffusion steps. We use $\mu_\sigma = -1$, $s_\sigma = 1$, and $\alpha = 3.0$ by default (Tab. 10).

During training, we share the same σ across all camera views of each sample, while $\boldsymbol{\epsilon}$ is drawn independently for each view. The corrupted latents from Eq. 1 then replace the clean inputs for Gaussian initialization (Sec. 3.1), yielding noisy Gaussian positions and features. These noisy Gaussian representations are subsequently denoised by the diffusion-aware Gaussian block and rasterized to obtain the clean latent prediction $\hat{\mathbf{x}}_0$ for supervision against the original clean latent $\mathbf{x}_0$.

3.2.2 Diffusion-aware Gaussian Block

A key design principle of GaussianDiffuser is the minimal architectural modification required to integrate diffusion into the pre-training pipeline. We achieve this by replacing the standard de-

formable attention normalization with an Adaptive Layer Normalization with Zero initialization (AdaLN-Zero) (Peebles & Xie, 2023), conditioned on the diffusion timestep.

The timestep t is first encoded via a sinusoidal positional embedding followed by a two-layer MLP, producing the timestep embedding $\mathbf{e}_t \in \mathbb{R}^C$. The AdaLN-Zero module then generates per-layer modulation parameters from $\mathbf{e}_t$:

$$\boldsymbol{\beta}_{\text{msa}},\, \boldsymbol{\gamma}_{\text{msa}},\, \boldsymbol{\alpha}_{\text{msa}},\, \boldsymbol{\beta}_{\text{mlp}},\, \boldsymbol{\gamma}_{\text{mlp}},\, \boldsymbol{\alpha}_{\text{mlp}} = \text{AdaLN-Zero}(\mathbf{e}_t), \tag{2}$$

where $\boldsymbol{\alpha}_{\text{msa}}$ and $\boldsymbol{\alpha}_{\text{mlp}}$ are gating vectors initialized to zero. $\boldsymbol{\beta}_{\text{msa}}$, $\boldsymbol{\gamma}_{\text{msa}}$, $\boldsymbol{\beta}_{\text{mlp}}$, and $\boldsymbol{\gamma}_{\text{mlp}}$ provide adaptive affine transformations.

Each encoder block processes the Gaussian features $\mathbf{h}$ through the following sequence of operations. First, the normalized and affined features attend to the multi-view image features via deformable cross-attention (Zhu et al., 2021), with the output gated by α_{msa}:

$$\mathbf{h} \leftarrow \mathbf{h} + (1 + \boldsymbol{\alpha}_{\text{msa}}) \odot \text{DeformAttn}((1 + \boldsymbol{\gamma}_{\text{msa}}) \odot \text{LN}(\mathbf{h}) + \boldsymbol{\beta}_{\text{msa}},\, \mathbf{F}_{\text{img}}), \tag{3}$$

where $\mathbf{F}_{\text{img}}$ denotes the multi-scale image feature maps. Next, a feed-forward network (FFN) with adaptive scale and shift is applied, gated by α_{mlp}:

$$\mathbf{h} \leftarrow \mathbf{h} + (1 + \boldsymbol{\alpha}_{\text{mlp}}) \odot \text{FFN}((1 + \boldsymbol{\gamma}_{\text{mlp}}) \odot \text{LN}(\mathbf{h}) + \boldsymbol{\beta}_{\text{mlp}}). \tag{4}$$

A subsequent sparse 3D convolution (Yan et al., 2018) then propagates features within local neighborhoods of Gaussians under the same AdaLN-Zero conditioning from the shared MLP of Eq. 2, after which a refinement module predicts residual updates to Gaussian attributes, including $\mathbf{f}_j$.

3.2.3 Bridging 3D Representations to 2D Denoising Supervision

The denoising objective is most reliable when clean target signals are naturally observed. In driving datasets, RGB images and LiDAR depth provide natural photometric and geometric targets, whereas dense 3D supervision is often scarce or relies on pseudo-labels. To supervise 3D Gaussian representations through a corruption-recovery objective using clean 2D photometric, geometric, and feature signals, GaussianDiffuser projects continuous 3D Gaussians back to the 2D image domain via differentiable rasterization (Kerbl et al., 2023).

Concretely, given the denoised 3D Gaussians with predicted color $\mathbf{c}_j$, the rendered image $\hat{\mathbf{I}}_i \in \mathbb{R}^{H' \times W' \times 3}$ for camera i can be obtained:

$$\hat{\mathbf{I}}_i(u, v) = \sum_{j \in \mathcal{N}(u,v)} \mathbf{c}_j\, \alpha_j\, G_j(u, v) \prod_{k<j} \left(1 - \alpha_k\, G_k(u, v)\right), \tag{5}$$

where α_j is the opacity of j-th Gaussian, $G_j(u, v)$ is the rasterized 2D Gaussian kernel evaluated at pixel (u, v), and the summation follows a front-to-back ordering. The depth map $\hat{\mathbf{D}}_i \in \mathbb{R}^{H' \times W' \times 1}$ and feature map $\hat{\mathbf{F}}_i \in \mathbb{R}^{H' \times W' \times C}$ are rendered analogously by replacing $\{\mathbf{c}_j\}$ with geometric attributes $\{\mu_j, \mathbf{s}_j, \mathbf{q}_j\}$ and feature $\{\mathbf{f}_j\}$, respectively. For the rasterized feature map $\hat{\mathbf{F}}_i$, a denoising projection head, a lightweight MLP, maps the shared timestep embedding $\mathbf{e}_t$ from Sec. 3.2.2 to scale $\boldsymbol{\gamma}_{\text{out}}$ and shift $\boldsymbol{\beta}_{\text{out}}$ parameters:

$$\hat{\mathbf{x}}_0 = \text{MLP}\Big((1 + \boldsymbol{\gamma}_{\text{out}}) \odot \text{LN}(\hat{\mathbf{F}}_i) + \boldsymbol{\beta}_{\text{out}}\Big). \tag{6}$$

3.3 Training Objectives

All pre-training losses supervise the recovery of clean 2D signals from corrupted image latents after Gaussian lifting and differentiable rasterization. The RGB and depth losses provide complementary photometric and geometric supervision:

$$\mathcal{L}_{\text{rgb}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\mathbf{I}}_i - \mathbf{I}_i \right\|_1, \qquad \mathcal{L}_{\text{depth}} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{V}_i|} \sum_{(u,v) \in \mathcal{V}_i} \left\| \hat{\mathbf{D}}_i(u, v) - \mathbf{D}_i^{\text{gt}}(u, v) \right\|_1, \tag{7}$$

where D_i^{gt} denotes the sparse LiDAR depth map obtained by projecting LiDAR points onto the image plane of camera i, and $\mathcal{V}_i$ is the set of pixels with valid depth values in D_i^{gt}.

For feature denoising, a lightweight timestep-conditioned projection head maps the rasterized Gaussian feature map $\hat{\mathbf{F}}_i$ to the clean-latent prediction $\hat{\mathbf{x}}_0$, supervised using the clean latent $\mathbf{x}_0$. Following JiT (Li & He, 2025), we adopt the *x*-prediction formulation with a *velocity*-space loss. The model directly predicts $\hat{\mathbf{x}}_0$, while the loss is computed on the implied velocity against the target velocity:

$$\mathcal{L}_{\text{feat}} = \mathbb{E}_{\sigma,\boldsymbol{\epsilon}}\left[w(\sigma)\left\|\frac{\mathbf{x}'_\sigma - \hat{\mathbf{x}}_0}{\sigma} - (\boldsymbol{\epsilon} - \mathbf{x}_0)\right\|^2\right], \tag{8}$$

where $w(\sigma)$ is the logit-normal weighting function (Esser et al., 2024) that emphasizes intermediate noise levels. This formulation decouples the prediction target $\mathbf{x}_0$ from the loss space (*velocity*) and stabilizes the training (Li & He, 2025). For numerical stability, we lower-bound the denominator in the *velocity*-space feature loss by $\sigma_{\min}$.

We further employ a temporal photometric reprojection loss (Godard et al., 2019) that warps each pixel into adjacent frames using the rendered depth and known camera poses:

$$\mathcal{L}_{\text{reproj}} = \min_{\tau' \in \{\tau-1, \tau+1\}}\left[(1-\alpha)\left\|\hat{\mathbf{I}}_{\tau\to\tau'} - \mathbf{I}_{\tau'}\right\|_1 + \frac{\alpha}{2}\left(1 - \text{SSIM}(\hat{\mathbf{I}}_{\tau\to\tau'}, \mathbf{I}_{\tau'})\right)\right], \tag{9}$$

where $\hat{\mathbf{I}}_{\tau\to\tau'}$ denotes the current frame warped into the adjacent frame τ', and α controls the balance between L_1 and SSIM (Wang et al., 2004) terms. Adjacent frames are used only as photometric targets for $\mathcal{L}_{reproj}$, not as encoder inputs, and are not used during downstream fine-tuning or inference.

The overall pre-training objective is:

$$\mathcal{L} = \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{feat}}\mathcal{L}_{\text{feat}} + \lambda_{\text{reproj}}\mathcal{L}_{\text{reproj}}, \tag{10}$$

where λ_{rgb}, λ_{depth}, λ_{feat}, and λ_{reproj} are the weights for each pre-training loss.

3.4 Inference and Downstream Transfer

Since the denoising objective is used as a pre-training regularizer rather than an iterative generative sampler, both inference and downstream fine-tuning use clean latents from the fixed timestep $t = 0$ in a single forward pass. To adapt the model for downstream tasks, we retain the AdaLN-Zero-conditioned encoder, while removing the noise scheduler and denoising projection head. We then adopt probabilistic voxelization (Huang et al., 2025) to convert sparse 3D Gaussian representations into the dense voxel feature volume ($\mathbf{V}_G \in \mathbb{R}^{X\times Y\times Z\times C}$) required for downstream tasks like 3D occupancy prediction and 3D object detection. By interpreting each Gaussian as a spatial probability distribution, this voxelization aggregates overlapping Gaussians via probabilistic superposition. Finally, we append task-specific heads and fine-tune the entire model end-to-end. The occupancy head maps $\mathbf{V}_G$ to K-class per-voxel semantic scores via residual linear layers and a sigmoid activation, while the detection head processes the same volume $\mathbf{V}_G$ as detailed in Appendix A.1.

4 Experiments

We evaluate GaussianDiffuser on driving datasets (Caesar et al., 2020; Wilson et al., 2021), comparing against state-of-the-art pre-training approaches. Experiments evaluate both pre-trained representations via scene reconstruction and the downstream transfer to 3D perception after fine-tuning.

4.1 Dataset

nuScenes. The nuScenes dataset (Caesar et al., 2020) contains 700 training, 150 validation, and 150 test scenes, each captured by 6 surrounding cameras with full 360° coverage, a 32-beam LiDAR, and 3D radar. Key samples are annotated at 2 Hz for tasks including 3D object detection and segmentation. Dense 3D semantic occupancy labels are available through Occ3D (Tian et al., 2023) and SurroundOcc (Wei et al., 2023).

Argoverse 2. To assess cross-dataset generalization, we additionally employ the Argoverse 2 (AV2) dataset (Wilson et al., 2021), which provides 1,000 sequences captured by seven ring cameras and two 32-beam LiDARs. The different sensor layout and environmental statistics introduce a substantial domain shift from nuScenes (Caesar et al., 2020). For our experiments, input images are resized to 448×800 and only a 1/4 subset of training data is used for pre-training.

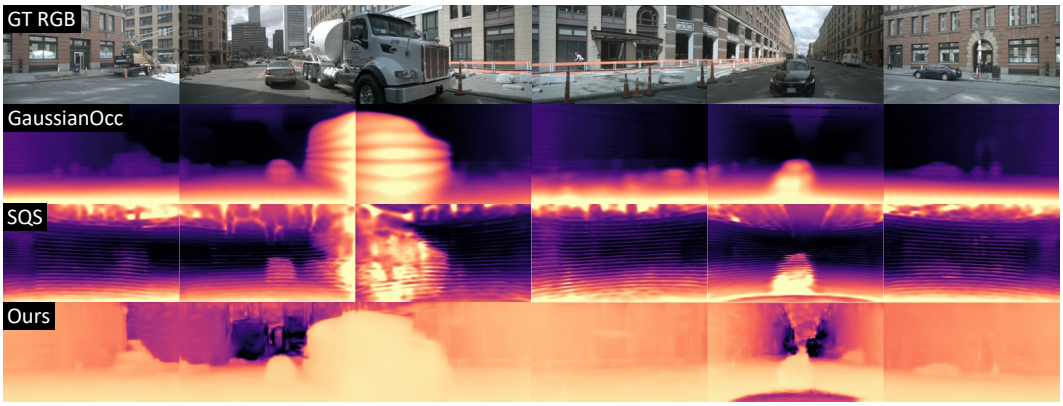

Figure 3: **Qualitative comparison on the rendered depth.** We compare the rendered depth maps of GaussianDiffuser with qualitative results from the recent Gaussian-based methods, GaussianOcc (Gan et al., 2025) and SQS (Zhang et al., 2025b), on the nuScenes `val` set. While the competing methods exhibit horizontally streaked artifacts and indistinct objects, GaussianDiffuser produces sharper object boundaries and cleaner separation. (SQS is included only for qualitative comparison as its rendered depth metrics are not reported in Zhang et al. (2025b).)

Table 1: **Depth estimation on the nuScenes `val` set.** Unless marked otherwise, all methods are evaluated on 112×200 rendered depths from 448×800 image inputs. SelfOcc‡ (Huang et al., 2024a), *depth-optimized* variant, uses 384×800 image inputs for 450×800 rendered depths. GaussianOcc† (Gan et al., 2025) uses 384×640 image inputs for 180×320 rendered depths.

Method	Abs Rel$^{\downarrow}$	Sq Rel$^{\downarrow}$	RMSE$^{\downarrow}$	RMSE log$^{\downarrow}$	$\delta < 1.25^{\uparrow}$	$\delta < 1.25^{2\uparrow}$	$\delta < 1.25^{3\uparrow}$
SelfOcc‡	0.215	2.743	<u>6.706</u>	0.316	0.753	0.875	0.932
UniPAD	0.218	2.512	7.937	0.356	0.763	0.869	0.921
NeRP3D	<u>0.183</u>	2.274	7.884	0.353	<u>0.799</u>	<u>0.883</u>	0.926
GaussianOcc†	0.197	<u>1.846</u>	6.733	<u>0.312</u>	0.746	0.873	<u>0.931</u>
GaussianDiffuser	**0.151**	**1.459**	**5.104**	**0.248**	**0.849**	**0.938**	**0.968**

4.2 Evaluation Metrics

Pretext Task. We compare depth estimation quality by reconstructing depth maps at one-quarter of the input resolution as the pretext output. We report relative errors (AbsRel & SqRel), root mean squared error (RMSE & RMSE log), and accuracy under threshold δ metrics up to $80m$ for depth reconstruction, only for pixels where the 10-sweep LiDAR point cloud is projected.

Downstream Task. We evaluate the performance of occupancy prediction using mean Intersection over Union (mIoU) on two benchmarks based on the nuScenes dataset (Caesar et al., 2020), Occ3D (Tian et al., 2023) and SurroundOcc (Wei et al., 2023). Occ3D defines 0.4 m voxels within $[-40, 40]$ m (X, Y) and $[-1.0, 5.4]$ m (Z) in the ego coordinate system for 17 semantic classes. For SurroundOcc, which follows the LiDAR coordinate system, the voxels are defined as 0.5 m covering $[-50, 50]$ m (X, Y) and $[-5.0, 3.0]$ m (Z), evaluated over 16 classes (excluding 'others').

4.3 Implementation Details

We implement GaussianDiffuser on the MMDetection3D (Contributors, 2020) framework and train on 8 NVIDIA H200 GPUs. Unless marked otherwise, pre-training runs with input resolution 448×800 for 12 epochs with a batch size of 4, using the AdamW optimizer with a learning rate of 2e-4 and a weight decay of 0.01. For fine-tuning, we train for 12 epochs on Occ3D (Tian et al., 2023) and 20 epochs on SurroundOcc (Wei et al., 2023) with a batch size of 2. The loss weights are set to $\lambda_{\text{rgb}} = \lambda_{\text{depth}} = 10$ and $\lambda_{\text{feat}} = \lambda_{\text{reproj}} = 1$. For the image backbone, we use a ConvNeXt-small (Liu et al., 2022) initialized with DINOv3 (Siméoni et al., 2025) to obtain reliable clean feature targets in the denoising objective. To ensure a controlled comparison, all experiments for GaussianFormer-

Table 2: **3D occupancy prediction on the Occ3D `val` set**. Methods without pre-training(top) follow the original Occ3D (Tian et al., 2023) protocol. Pre-training methods(middle) use an input resolution of 448×800, excluding VisionPAD† (Zhang et al., 2025a) (900×1600). $*$(bottom) denotes that methods trained and evaluated on 15 classes, excluding *others* and *other flat*.

Method	mIoU	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
OccFormer	21.9	5.9	30.3	12.3	34.4	39.2	14.4	16.5	17.2	9.3	13.9	26.4	51.0	31.0	34.7	22.7	6.8	7.0
TPVFormer	27.8	7.2	38.9	13.7	40.8	45.9	17.2	20.0	18.9	14.3	26.7	34.2	55.7	35.5	37.6	30.7	19.4	16.8
CTF-Occ	28.5	8.1	39.3	20.6	38.3	42.2	16.9	24.5	22.7	21.1	23.0	31.1	53.3	33.8	38.0	33.2	20.8	18.0
UniPAD	34.1	6.5	39.7	13.0	42.3	45.8	19.4	14.3	20.0	17.7	27.4	33.1	80.0	38.7	49.4	50.6	42.8	38.1
VisionPAD†	35.4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
NeRP3D	35.5	5.5	41.0	15.0	43.9	49.4	19.2	20.0	23.6	16.5	27.9	36.7	81.0	37.4	49.8	53.6	43.9	38.8
GaussianDiffuser	**37.9**	11.0	42.3	24.3	43.5	47.3	21.2	26.3	25.9	24.1	32.8	35.5	81.2	41.2	51.6	56.0	43.2	38.2
GaussianOcc*	38.5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
GaussianDiffuser*	**40.1**	–	42.3	25.8	43.5	48.6	20.6	26.7	26.4	24.9	32.5	36.4	82.0	–	51.9	56.1	44.3	38.9

Table 3: **3D occupancy prediction on the SurroundOcc `val` set**. GaussianDiffuser uses an input resolution of 896×1600 and others use 864×1600, following the GaussianFormer (Huang et al., 2024b; 2025) protocol. Methods are grouped as without(bottom) and with(top) pre-training.

Method	SC IoU	SSC mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
GaussianFormer	29.83	19.10	19.5	11.3	26.1	29.8	10.5	13.8	12.6	8.7	12.7	21.6	39.6	23.3	24.5	23.0	9.6	19.1
GaussianFormer-2	30.56	20.82	21.4	13.4	28.5	30.8	10.9	15.8	13.6	10.5	14.0	22.9	40.6	24.4	26.1	24.3	13.8	22.0
SQS	31.52	20.40	20.0	11.9	28.2	30.7	10.9	15.0	14.3	9.6	14.8	23.0	39.8	23.9	25.5	23.1	14.6	21.3
GaussianDiffuser	**34.05**	**22.19**	21.3	13.2	28.5	31.0	13.3	17.0	14.9	12.2	15.3	23.6	41.5	26.1	27.9	26.8	16.7	25.7

2 (Huang et al., 2025) are reproduced with the same backbone and its weights as GaussianDiffuser, except for Tab. 3. More detailed implementation information is described in Appendix B.

4.4 Main Results

Occupancy Prediction. GaussianDiffuser consistently improves 3D occupancy prediction across both Occ3D (Tian et al., 2023) and SurroundOcc (Wei et al., 2023). On Occ3D (Tian et al., 2023) in Tab. 2, GaussianDiffuser achieves 36.8 mIoU under the 17-class setting, surpassing UniPAD (Yang et al., 2024) and NeRP3D (Jeong et al., 2026) by 3.8 and 2.4 mIoU, respectively. The gains are especially pronounced for fine-grained and sparse object categories such as *bicycle*, *motorcycle*, *pedestrian*, and *traffic cone*, where accurate occupancy prediction requires precise geometric localization rather than only coarse scene layout understanding. Under the 15-class protocol following GaussianOcc (Gan et al., 2025), GaussianDiffuser further reaches 40.1 mIoU, outperforming GaussianOcc by 1.6 mIoU. On SurroundOcc (Wei et al., 2023) in Tab. 3, GaussianDiffuser achieves 34.05 IoU and 22.19 mIoU, improving over SQS (Zhang et al., 2025b) by 2.53 IoU and 1.79 mIoU under the evaluated protocol. In addition to the overall gain, GaussianDiffuser shows clear improvements on several small or sparse object categories similar to the results on Occ3D.

Depth Reconstruction. We assess the geometric fidelity of pre-trained representations through depth rendering on the nuScenes `val` set. As shown in Tab. 1, GaussianDiffuser outperforms

Table 4: **Zero-shot depth estimation (Argoverse 2 → nuScenes).** Cross-dataset generalization is evaluated by zero-shot depth estimation on 112×200 rendered depths from 448×800 images.

Method	Abs Rel$^{\downarrow}$	Sq Rel$^{\downarrow}$	RMSE$^{\downarrow}$	RMSE log$^{\downarrow}$	$\delta < 1.25^{\uparrow}$	$\delta < 1.25^{2\uparrow}$	$\delta < 1.25^{3\uparrow}$
UniPAD	0.985	11.767	14.963	4.390	–	–	–
NeRP3D	<u>0.626</u>	<u>6.251</u>	<u>10.728</u>	<u>0.921</u>	<u>0.167</u>	<u>0.320</u>	<u>0.454</u>
GaussianDiffuser	**0.273**	**2.988**	**9.464**	**0.463**	**0.551**	**0.756**	**0.857**

both NeRF-based baselines (Yang et al., 2024; Huang et al., 2024a; Jeong et al., 2026) and GaussianOcc (Gan et al., 2025) across all error and threshold-accuracy metrics, achieving 0.151 Abs Rel, 1.459 Sq Rel, 5.104 RMSE, and 0.248 RMSE log. The qualitative comparison in Fig. 3 also corroborates these results. Compared with GaussianOcc and SQS (Zhang et al., 2025b), whose depth maps exhibit horizontal streaks and indistinct objects, GaussianDiffuser produces clearer object shapes, clean ground planes, and stronger foreground–background separation. Distant vehicles and street-level structures are also recovered with better contrast, indicating fine-grained 3D representation.

Generalization. We assess transfer under sensor-layout and domain shifts by pre-training on the 1/4 subset of AV2 (Wilson et al., 2021) and evaluating on nuScenes (Caesar et al., 2020). For zero-shot depth estimation without fine-tuning (Tab. 4), GaussianDiffuser consistently outperforms NeRP3D across all depth metrics, achieving lower Abs Rel (0.273 vs. 0.626), Sq Rel (2.988 vs. 6.251), RMSE (9.464 vs. 10.728), and RMSE log (0.463 vs. 0.921). GaussianDiffuser also achieves higher accuracy across all three thresholds compared with NeRP3D, while UniPAD fails to produce meaningful predictions under this domain shift. For 3D object detection, fine-tuning the AV2-pretrained backbone on the 1/2 nuScenes subset for 12 epochs yields **30.53 mAP**, compared with 26.29 for UniPAD and 27.46 for NeRP3D. These results suggest that denoising-regularized Gaussian pre-training improves representation transfer across different datasets and sensor configurations.

4.5 Ablation Studies

Generalization to 3D object detection. To assess transfer beyond occupancy prediction, we fine-tune GaussianDiffuser for 3D object detection on nuScenes. Improvements over the matched from-scratch detector support transfer to object-level perception (Appendix A.1).

Effect of the diffusion objective. We compare against a matched rendering-only baseline to isolate the benefit of denoising beyond rendering supervision, and examine the contributions of individual losses and timestep conditioning (Appendices A.2–A.3). The benefit is more evident in downstream occupancy than in surface-level depth reconstruction.

Corruption domain and backbone initialization. To examine the choice of corruption domain, we compare image-latent corruption before lifting with Gaussian-feature corruption after lifting (Appendix A.4). Moreover, since the backbone supplies clean feature targets, we also test whether the denoising gain is specific to DINOv3 initialization (Appendix A.5).

5 Conclusion

We presented GaussianDiffuser, a feedforward framework integrating diffusion-based denoising into 3D Gaussian pre-training for vision-centric autonomous driving. Its diffusion-aware Gaussian block enables clean RGB, depth, and feature recovery from corrupted latents with minimal architectural modifications, complementing rendering-based reconstruction and encouraging spatially coherent representations. Experiments demonstrate improved depth reconstruction and downstream transfer.

As a representation pre-training method, GaussianDiffuser uses single-pass denoising supervised in the image domain after rasterization, without explicitly modeling the full multi-modal distribution of unobserved 3D space. Its gains over a strong rendering-only Gaussian baseline are smaller on surface-level depth metrics than on downstream tasks. Future work could extend this regularizer to generative 3D scene modeling of occluded regions, long-tailed objects, and volumetric structure.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5470–5479, 2022.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19697–19705, 2023.

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*, pp. 11621–11631, 2020.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.

MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorber, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Wanshui Gan, Fang Liu, Hongbin Xu, Ningkai Mo, and Naoto Yokoya. Gaussianocc: Fully self-supervised and efficient 3d occupancy estimation with gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28980–28990, 2025.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3828–3838, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19946–19956, 2024a.

Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In *European Conference on Computer Vision*, pp. 376–393. Springer, 2024b.

Yuanhui Huang, Amonnut Thammatadatrakoon, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, and Jiwen Lu. Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction. In *Proceedings of the computer vision and pattern recognition conference*, pp. 27477–27486, 2025.

Hyeonjun Jeong, Juyeb Shin, and Dongsuk Kum. To view transform or not to view transform: Nerf-based pre-training perspective. In *The Fourteenth International Conference on Learning Representations*, 2026.

Bingxin Ke, Kevin Qu, Tianfu Wang, Nando Metzger, Shengyu Huang, Bo Li, Anton Obukhov, and Konrad Schindler. Marigold: Affordable adaptation of diffusion-based image generators for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Duy-Tho Le, Hengcan Shi, Jianfei Cai, and Hamid Rezatofighi. Diffusion model for robust multi-sensor fusion in 3d object detection and bev segmentation. In *European Conference on Computer Vision*, pp. 232–249. Springer, 2024.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025.

Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *Advances in Neural Information Processing Systems*, 35:18442–18455, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 18580–18590, 2023.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Chengxin Lv, Yihui Li, Hongyu Yang, and YunHong Wang. Gau-occ: Geometry-completed gaussians for multi-modal 3d occupancy prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2026.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Jinhyung Park, Chensheng Peng, Wenzhao Zheng, Kris Kitani, Wei Zhan, et al. S2go: Streaming sparse gaussian occupancy. In *The Fourteenth International Conference on Learning Representations*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36:64318–64330, 2023.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21729–21740, 2023.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

Maciej Wozniak, Jesper Ericsson, Hariprasath Govindarajan, Truls Nyberg, Thomas Gustafsson, Patric Jensfelt, and Olov Andersson. Hilda: Hierarchical distillation with diffusion for advancing self-supervised lidar pre-training. In *European Conference on Computer Vision (ECCV)*, 2026.

Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15802–15812, 2023.

Guangkai Xu, Yongtao Ge, Mingyu Liu, Chengxiang Fan, Kangyang Xie, Zhiyue Zhao, Hao Chen, and Chunhua Shen. What matters when repurposing diffusion models for general dense perception tasks? *arXiv preprint arXiv:2403.06090*, 2024a.

Shaoqing Xu, Fang Li, Shengyin Jiang, Ziying Song, Li Liu, and Zhi-xin Yang. Gaussianpretrain: A simple unified 3d gaussian representation for visual pre-training in autonomous driving. *arXiv preprint arXiv:2411.12452*, 2024b.

Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

Honghui Yang, Sha Zhang, Di Huang, Xiaoyang Wu, Haoyi Zhu, Tong He, Shixiang Tang, Hengshuang Zhao, Qibo Qiu, Binbin Lin, et al. Unipad: A universal pre-training paradigm for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15238–15250, 2024.

Xin Ye, Burhaneddin Yaman, Sheng Cheng, Feng Tao, Abhirup Mallik, and Liu Ren. Bevdiffuser: Plug-and-play diffusion model for bev denoising with ground-truth guidance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1495–1504, 2025.

Chubin Zhang, Juncheng Yan, Yi Wei, Jiaxin Li, Li Liu, Yansong Tang, Yueqi Duan, and Jiwen Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. *CoRR*, 2023.

Haiming Zhang, Wending Zhou, Yiyao Zhu, Xu Yan, Jiantao Gao, Dongfeng Bai, Yingjie Cai, Bingbing Liu, Shuguang Cui, and Zhen Li. Visionpad: A vision-centric pre-training paradigm for autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17165–17175, 2025a.

Haiming Zhang, Yiyao Zhu, Wending Zhou, Xu Yan, Yingjie Cai, Bingbing Liu, Shuguang Cui, and Zhen Li. Sqs: Enhancing sparse perception models via query-based splatting in autonomous driving. *arXiv preprint arXiv:2509.16588*, 2025b.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

Jiayu Zou, Kun Tian, Zheng Zhu, Yun Ye, and Xingang Wang. Diffbev: Conditional diffusion model for bird's eye view perception. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 7846–7854, 2024.

Table 5: **3D object detection on the nuScenes `val` set.** Each pre-training method is plugged into a different detector baseline, so absolute NDS and mAP are not directly comparable across rows. Superscripts ($^{+\Delta}$) report the gain over each method's corresponding from-scratch detector, which isolates the contribution of pre-training. † denotes results obtained using an input resolution of 256×700 and trained for 36 epochs. ‡ denotes the results trained and evaluated on input resolutions of 512×1408. Other results, including GaussianDiffuser, are based on 448×800 input resolution.

Method	NDS$^{\uparrow}$	mAP$^{\uparrow}$	mATE$^{\downarrow}$	mASE$^{\downarrow}$	mAOE$^{\downarrow}$	mAVE$^{\downarrow}$	mAAE$^{\downarrow}$
UniPAD	37.1	33.7	0.734	0.283	0.603	1.250	0.359
NeRP3D	39.2	35.8	0.719	0.288	0.640	0.977	0.250
SparseBEV †	55.8	44.8	0.581	0.271	0.373	0.247	0.190
SparseBEV † + SQS	56.6$^{+0.8}$	45.2$^{+0.4}$	0.564	0.263	0.362	0.232	0.182
SparseBEV ‡	59.2	50.1	0.562	0.265	0.321	0.243	0.195
SparseBEV ‡ + SQS	60.2$^{+1.0}$	50.9$^{+0.8}$	0.531	0.251	0.318	0.241	0.185
GaussianFormer-2	40.4	34.7	0.768	0.280	0.442	1.003	0.209
+ GaussianDiffuser	41.7$^{+1.3}$	35.7$^{+1.0}$	0.734	0.280	0.456	0.896	0.218

$\mathcal{L}_{depth}$	$\mathcal{L}_{rgb}$	$\mathcal{L}_{reproj}$	$\mathcal{L}_{feat}$	mIoU
✓		✓	✓	34.59
✓	✓		✓	34.27
✓	✓	✓		33.83
✓	✓	✓	✓	34.73

Table 6: **Loss-wise ablation on the Occ3D `val` set.** All variants retain $\mathcal{L}_{depth}$ and the original loss weights, $\lambda_{\text{rgb}} = \lambda_{\text{depth}} = 10$ and $\lambda_{\text{reproj}} = \lambda_{\text{feat}} = 1$. ✓denotes the enabled loss.

A Ablation Studies

A.1 Generalization to 3D Object Detection

A strong pre-training method should transfer broadly across geometry-centric downstream tasks. To assess transfer beyond occupancy prediction, we additionally fine-tune GaussianDiffuser for 3D object detection on the nuScenes (Caesar et al., 2020) `val` set for 24 epochs and compare against representative pre-training methods in Tab. 5.

Protocol. Since detector architectures and evaluation settings differ, the absolute scores in Tab. 5 are provided for reference. We assess pre-training through gains over each method's matched from-scratch detector, denoted by $+\Delta$, where available. This relative gain better isolates the effect of pre-training from architectural differences in the downstream detector. More specifically, UniPAD (Yang et al., 2024) and NeRP3D (Jeong et al., 2026) are evaluated with the UVTR (Li et al., 2022) detection head, predicting 3D bounding boxes and their class scores by deformable cross attention between randomly initialized object queries and values from voxelized feature. On the other hand, SQS (Zhang et al., 2025b) runs its Gaussian decoder alongside a detection model such as SparseBEV (Liu et al., 2023) in parallel for query interaction. For GaussianDiffuser, we follow the UVTR-head setting used by UniPAD and NeRP3D, attaching the UVTR detection head to the voxel features obtained through probabilistic voxelization of our pre-trained 3D Gaussian representation and comparing it with the same detector trained from scratch (Huang et al., 2025).

Results. Under this relative-gain view, GaussianDiffuser improves NDS by $+1.3$ and mAP by $+1.0$ over the matched GaussianFormer-2 (Huang et al., 2025) detector, supporting transfer to object-level 3D perception. For reference, SQS obtains $+0.8$ NDS / $+0.4$ mAP gain over SparseBEV. The larger relative improvement indicates that diffusion-based Gaussian pre-training provides an effective transferable representation for object-level 3D perception. Beyond the detection gain, GaussianDiffuser also maintains computational efficiency during downstream transfer. Compared with approaches that introduce parallel query interaction modules with substantial memory overhead, GaussianDiffuser achieves competitive relative improvements with only negligible additional memory cost, reported in Appendix C.

Table 7: **Depth estimation on nuScenes `val` set.** GaussianFormer-2 (Huang et al., 2025) is the result obtained by rendering-based pre-training using Eq. 7, which corresponds to GaussianDiffuser without the denoising objective and the diffusion-aware Gaussian block. All other components, hyperparameters, and schedules are identical between the two rows.

Method	Abs Rel$^{\downarrow}$	Sq Rel$^{\downarrow}$	RMSE$^{\downarrow}$	RMSE log$^{\downarrow}$	$\delta < 1.25^{\uparrow}$	$\delta < 1.25^{2\uparrow}$	$\delta < 1.25^{3\uparrow}$
GaussianFormer-2	0.159	1.535	5.214	0.253	0.845	0.938	0.967
+ GaussianDiffuser	0.151	1.459	5.104	0.248	0.849	0.938	0.968

Table 8: **3D occupancy prediction on the Occ3D `val` set**. GaussianFormer-2 (Huang et al., 2025) is the fine-tuned result from the rendering-only pre-training, as in Tab. 7. Both rows share the same encoder, hyperparameters, and schedules, except for AdaLN-Zero module in diffusion-aware Gaussian block.

Method	mIoU	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
GaussianFormer-2	35.58	10.3	38.4	21.1	41.3	44.0	17.4	22.9	23.0	20.8	30.7	34.1	79.9	39.6	49.9	53.6	41.1	36.7
+ GaussianDiffuser	**36.56**	11.1	40.4	24.4	42.9	45.5	19.2	25.9	25.2	22.2	27.7	34.5	79.5	40.2	49.8	54.6	41.3	37.2

A.2 Loss-wise Ablation

The pre-training objective combines photometric reconstruction, temporal consistency, feature denoising, and geometric supervision. To quantify the contribution of each additional loss, we conduct a leave-one-out ablation of $\mathcal{L}_{rgb}$, $\mathcal{L}_{reproj}$, and $\mathcal{L}_{feat}$ on Occ3D (Tian et al., 2023), while retaining $\mathcal{L}_{depth}$ and the original loss weights ($\lambda_{\text{rgb}} = \lambda_{\text{depth}} = 10$ and $\lambda_{\text{feat}} = \lambda_{\text{reproj}} = 1$). All variants are pre-trained and fine-tuned for 12 epochs each on the same 1/4 training subset. Tab. 6 shows 34.73 mIoU for training with all losses, and removing $\mathcal{L}_{rgb}$, $\mathcal{L}_{reproj}$, and $\mathcal{L}_{feat}$ reduces mIoU to 34.59, 34.27, and 33.83, respectively. Thus, all three losses contribute, with feature denoising yielding the largest improvement.

A.3 Effect of the Diffusion Objective

To assess the contribution of diffusion-based pre-training beyond rendering-based reconstruction, we compare GaussianDiffuser with a rendering-only GaussianFormer-2 (Huang et al., 2025) baseline. The baseline uses clean latent inputs and omits the diffusion-aware conditioning and feature denoising objective, while retaining the underlying Gaussian representation and rendering pipeline. Both methods use the same image backbone and initialization, input resolution, optimizer, and training schedules unless marked otherwise. We examine the resulting representations through two complementary evaluations: observed-surface depth reconstruction and downstream volumetric semantic occupancy prediction.

Depth reconstruction. Tab. 7 compares the depth reconstruction performance of the pre-trained models. GaussianDiffuser reduces Abs Rel from 0.159 to 0.151 and Sq Rel from 1.535 to 1.459, while the threshold accuracies remain comparable. Depth is evaluated only at sparse LiDAR-projected pixels and therefore measures observed-surface reconstruction rather than directly assessing the scene's volumetric semantic structure. This evaluation is closely aligned with the geometric reconstruction objective used during pre-training, but does not by itself establish how effectively the learned representation transfers to dense 3D prediction. We therefore complement depth reconstruction with downstream occupancy prediction, which evaluates the spatial extent and semantic organization of occupied regions in 3D.

Table 9: **Component ablation of diffusion-based denoising on Occ3D.** "Corruption" denotes noise corruption of 2D latents before Gaussian lifting; "AdaLN-Zero" denotes timestep-conditioned modulation in the diffusion-aware Gaussian block and, where applicable, the feature projection head; "Feat. Loss" denotes the feature denoising loss $\mathcal{L}_{feat}$, applied after differentiable rasterization. The first and last rows correspond to the rendering-only GaussianFormer-2 (Huang et al., 2025) baseline and the full GaussianDiffuser in Tab. 8 but less training data, respectively.

Corruption	AdaLN-Zero	Feat. Loss	mIoU
			33.71
✓			33.53
✓	✓		33.83
✓		✓	34.10
✓	✓	✓	**34.73**

Table 10: **Effect of the noise-level sampling distribution.** We vary the logit-space mean μ_σ in Eq. 1 (with $s_\sigma = 1$ and $\alpha = 3.0$ fixed) and report downstream occupancy mIoU. A smaller μ_σ biases sampling toward higher noise levels.

μ_σ	-2.0	**-1.0**	0.0	1.0
mIoU	33.71	**34.73**	34.33	33.54

Occupancy prediction. Tab. 8 evaluates whether the benefit of diffusion-based pre-training extends from observed-surface reconstruction to volumetric semantic prediction using the $1/2$ subset of the Occ3D training set, 12-epoch pre-training and fine-tuning. GaussianDiffuser improves mIoU on Occ3D (Tian et al., 2023) from 35.58 to 36.56 ($+0.98$) over the rendering-only baseline, with consistent gains on most semantic classes. Notable gains occur for ***long-tailed*** **classes** such as *bicycle* ($+3.3$), *const. vehicle* ($+1.8$), *motorcycle* ($+3.0$), and *pedestrian* ($+2.1$), showing that the improvement extends to object-level semantic occupancy rather than being confined to broad background regions. Overall, these results demonstrate improved transfer to volumetric semantic prediction and are consistent with the hypothesis that corruption–recovery regularization encourages representations useful for 3D spatial and semantic reasoning beyond the surface reconstruction task.

Component-wise analysis. Tab. 9 further examines the contributions of noise corruption, AdaLN-Zero conditioning and the feature denoising loss $\mathcal{L}_{feat}$. All variants are pre-trained and fine-tuned on a $1/4$ training subset for 12 epochs per stage. The first and last rows correspond to the rendering-only GaussianFormer-2 baseline and the full GaussianDiffuser configurations in Tab. 8, but use less training data with the same epoch counts; component effects are therefore assessed within Tab. 9. Noise corruption turns RGB and depth reconstruction into corrupted-to-clean recovery tasks, while AdaLN-Zero enables Gaussian refinement to adapt to the sampled noise level. The feature denoising loss adds explicit clean-latent recovery supervision after differentiable rasterization. Importantly, omitting $\mathcal{L}_{feat}$ does not remove the corrupted-to-clean reconstruction task when noise corruption remains enabled. Combining AdaLN-Zero and $\mathcal{L}_{feat}$ outperforms the variants that omit either component, supporting their complementary roles in diffusion-based pre-training.

Noise-level sampling. Beyond the component ablations, Tab. 10 examines how the noise-level sampling distribution $p_\mu(\sigma)$ in Eq 1 affects the transferability of the pre-trained representations. The parameter μ_σ specifies the logit-space mean of a logit-normal variable used to index the descending noise schedule; smaller μ_σ therefore favors stronger corruption, whereas larger μ_σ favors weaker corruption. All models are pre-trained and fine-tuned on a 1/4 training subset, with 12 epochs per stage. Among the evaluated values, $\mu_\sigma = -1.0$ achieves the highest mIoU of 34.73, compared with 34.33 at $\mu_\sigma = 0.0$, 33.71 at $\mu_\sigma = -2.0$, and 33.54 at $\mu_\sigma = 1.0$. This non-monotonic trend suggests that shifting the sampling distribution toward higher noise levels relative to $\mu_\sigma = 0.0$ can benefit downstream occupancy prediction, but further increasing this bias does not necessarily improve performance.

Feature denoising loss space. We further compare *velocity*-space and *clean-latent*-space feature supervision while retaining the same clean-latent prediction $\hat{\mathbf{x}}_0$ in both variants. Following the distinction between prediction and loss spaces in JiT (Li & He, 2025), we vary the loss space rather than the network's output parameterization. Before denominator clamping, Eq. 1 gives $(\mathbf{x}'_\sigma - \hat{\mathbf{x}}_0)/\sigma - (\boldsymbol{\epsilon} - \mathbf{x}_0) = (\mathbf{x}_0 - \hat{\mathbf{x}}_0)/\sigma$. Using this identity, the *velocity*-space loss in Eq. 1 can be expressed as a reweighted *clean-latent* reconstruction loss:

$$\mathcal{L}^{(v)}_{\text{feat}} = \mathbb{E}_{\sigma,\boldsymbol{\epsilon}}\left[\frac{w(\sigma)}{\sigma^2}\left\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\right\|^2\right], \quad \mathcal{L}^{(x)}_{\text{feat}} = \mathbb{E}_{\sigma,\boldsymbol{\epsilon}}\left[w(\sigma)\left\|\hat{\mathbf{x}}_0 - \mathbf{x}_0\right\|^2\right]. \tag{11}$$

With the same noise sampling distribution and base weighting $w(\sigma)$, the additional $1/\sigma^2$ factor in this unclamped formulation increases the emphasis on lower-noise reconstruction errors relative to the clean-latent-space loss. Both variants are pre-trained and fine-tuned on a $1/4$ training subset for 12 epochs per stage. The *velocity*-space loss achieves 34.73 mIoU on Occ3D, compared with 33.70 for the *clean-latent*-space loss, a gain of 1.03 points. This result supports our choice of *velocity*-space feature supervision for downstream occupancy prediction under the evaluated setting.

A.4 Gaussian-level Noise Corruption.

To examine the effect of corruption domain, we compare the proposed image-latent corruption before lifting with Gaussian-feature corruption after lifting. Since the positional encoder does not explicitly normalize $\mathbf{f}_j$ before corruption, we additionally apply LayerNorm to align its feature statistics with unit-variance Gaussian noise:

$$\mathbf{f}'_{\sigma,j} = (1-\sigma)\text{LN}(\mathbf{f}_j) + \sigma\boldsymbol{\epsilon}_j, \quad \boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{12}$$

The corrupted features are processed by the same diffusion-aware Gaussian block, rasterization, and clean-latent recovery objective, with all other settings unchanged.

Both variants are pre-trained and fine-tuned for 12 epochs each on the same 1/2 Occ3D (Tian et al., 2023) training subset. Gaussian-feature corruption after lifting achieves 35.69 mIoU, while the proposed image-latent corruption before lifting achieves 36.56 mIoU. These results favor image-latent corruption under the evaluated setting. A possible explanation is that it constructs Gaussian features and attributes from a shared corrupted observation, whereas perturbing position-derived features after lifting may disrupt their consistency with the Gaussian attributes.

A.5 Image backbone initialization.

Since the clean-latent target $\mathbf{x}_0$ is produced by the image backbone, its initialization can influence the feature representations used for the denoising supervision. We therefore examine whether the benefit of diffusion-based pre-training persists when DINOv3 (Siméoni et al., 2025) initialization is replaced with ImageNet pre-trained weights, while keeping the ConvNeXt-S (Liu et al., 2022) architecture unchanged. Both the rendering-only baseline (Huang et al., 2025) and GaussianDiffuser use the same ImageNet initialization and are pre-trained and fine-tuned on the same 1/2 Occ3D (Tian et al., 2023) training subset, with 12 epochs per stage. All other training and downstream settings are kept unchanged.

With ImageNet initialization, GaussianDiffuser achieves 34.59 mIoU on Occ3D, compared with 34.05 for the matched rendering-only baseline, yielding an improvement of 0.54 points. Because both methods share the same backbone initialization and training protocol, this gain cannot be attributed to a stronger initial image representation for GaussianDiffuser. Instead, it provides evidence that the proposed diffusion-based pre-training adds value beyond rendering-only reconstruction even when the clean-latent targets are obtained from an ImageNet-initialized backbone. While this comparison does not establish that the magnitude of the gain is invariant to initialization, it shows that the benefit of the proposed corruption–recovery design is not exclusive to DINOv3-initialized features.

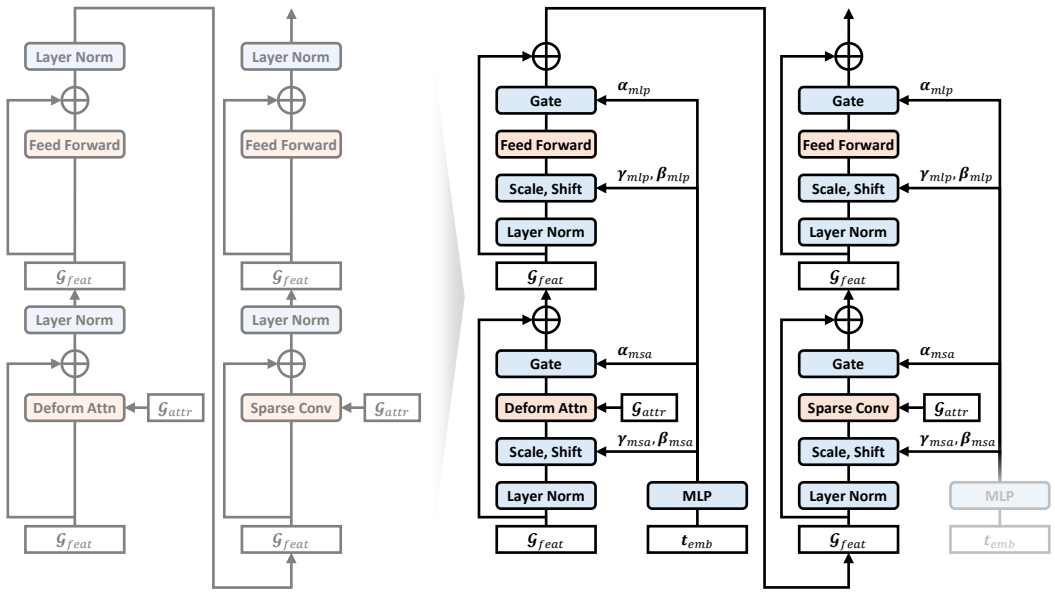

Figure 4: **Diffusion-aware Gaussian block.** The deformable-attention and sparse-convolution sub-blocks of the original GaussianFormer-2 (Huang et al., 2025) encoder (left) are shown alongside the corresponding diffusion-aware sub-blocks used by GaussianDiffuser (right). Gaussian features $\mathcal{G}_{feat}$ and attributes $\mathcal{G}_{attr}$ are retained, while timestep conditioning is introduced through a shared MLP. The diffusion timestep embedding $\mathbf{e}_t$ produces $(\gamma_{msa}, \beta_{msa}, \alpha_{msa})$ for the token-mixing path and $(\gamma_{mlp}, \beta_{mlp}, \alpha_{mlp})$ for the feed-forward path. The MLP is drawn in dashed gray on the sparse-convolution sub-block to indicate that it is shared with the deformable-attention sub-block.

B Implementation Details

B.1 Diffusion-Aware Gaussian Block

GaussianDiffuser adds timestep-conditioned AdaLN-Zero modulation to the deformable-attention and sparse-convolution sub-blocks of the GaussianFormer-2 encoder (Huang et al., 2025). The Gaussian tokens, attribute-to-feature pathway, and token-mixing operations are retained.

Our only addition is a lightweight AdaLN-Zero (Peebles & Xie, 2023) conditioning pathway wrapped around each of these sub-blocks, driven by the diffusion timestep embedding $\mathbf{e}_t$. A single MLP maps $\mathbf{e}_t$ to the modulation parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\alpha})$ and is *shared* across both sub-block types, so the entire encoder becomes timestep-aware through one common conditioning head rather than per-block replicas. Combined with the zero-initialized final projection of this MLP, the encoder at the start of pre-training behaves identically to the original (un-modulated) block, and the diffusion signal is introduced gradually as $(\boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\alpha})$ depart from their zero-valued initialization.

In aggregate, the entire modification amounts to one shared conditioning MLP plus per-sub-block Scale/Shift and Gate projections, leaving the parameter count and compute of the encoder essentially unchanged, as illustrated in Fig. 4.

B.2 Denoising Projection Head

After rasterization, a timestep-conditioned projection head maps the 2D Gaussian feature map to the clean-latent prediction $\hat{x}_0$ (Eq. 6), supervised by $\mathcal{L}_{\text{feat}}$ (Eq. 8). To keep these two sides symmetric and to produce the clean-feature prediction $\hat{\mathbf{x}}_0$ with the same timestep-awareness as the encoder, we place a small AdaLN-Zero projection head on top of the rasterized feature map, following the final-layer pattern used by DiT (Peebles & Xie, 2023) and Stable Diffusion 3 (Esser et al., 2024).

The output $\hat{\mathbf{x}}_0$ is compared against the clean latent via $\mathcal{L}_{\text{feat}}$. The final Linear layer in Eq. 6 is zero-initialized, matching the zero-initialization convention used inside the encoder, so the denoising

Table 11: **Computation analysis on downstream tasks**

(a) **NeRF *vs.* Gaussian comparison on 3D object detection**

Method	GFLOPS
UniPAD	1250.10
NeRP3D	1903.77
GaussianFormer-2	966.21
+ GaussianDiffuser	1079.05

(b) **Transfer overhead comparison on occupancy prediction**

Method	Memory
GaussianFormer	6100 MB
GaussianFormer+SQS	10880 MB
GaussianFormer-2	3880 MB
+ GaussianDiffuser	3988 MB

prediction starts from an identity-like behavior and the diffusion signal is introduced gradually rather than through a random projection at step zero. The head is used only for feature denoising; RGB and depth are rendered without this output modulation.

C Computation Analysis

Protocol. Tab. 11 reports downstream detection GFLOPS and occupancy peak GPU memory. We omit FPS and latency because the baselines require incompatible PyTorch/CUDA environments. Comparisons across methods use the reported downstream settings, while additional transfer costs are assessed relative to each method's corresponding baseline. The comparison with NeRP3D (Jeong et al., 2026) highlights the computational benefit of replacing NeRF-style implicit pre-training with an explicit Gaussian representation, while the comparison with SQS (Zhang et al., 2025b) focuses on the additional overhead introduced by the pre-training module during downstream transfer.

Representation efficiency: NeRF vs. Gaussian. In the downstream detection settings of Tab. 11, GaussianDiffuser requires 966.60 GFLOPS, compared with 1903.77 GFLOPS for NeRP3D, a reduction of 49.2%. Its overhead over GaussianFormer-2 (966.21 GFLOPS) is 0.39 GFLOPS. This indicates that GaussianDiffuser inherits the efficiency of the explicit Gaussian representation, while adding negligible computation over the GaussianFormer-2 (Huang et al., 2025) downstream baseline.

Transfer overhead. The comparison with SQS addresses a different question, how much additional computation is introduced when the pre-training method is attached to the downstream model. SQS increases peak memory from 6100MB to 10880MB, corresponding to a $+4780$MB ($+78.4\%$) overhead over GaussianFormer (Huang et al., 2024b). This increase comes from its query interaction design, where a Gaussian decoder and a detector are executed in parallel during downstream inference. In contrast, GaussianDiffuser does not require an additional decoder branch or a parallel detector interaction module. It only adds lightweight timestep-conditioning components, namely the time embedding and AdaLN-Zero layers, on top of the GaussianFormer-2 encoder. As a result, SQS increases peak memory over GaussianFormer from 6100 to 10880 MB (+4780 MB, +78.4%), while GaussianDiffuser increases peak memory over GaussianFormer-2 from 3880 to 3988 MB (+108 MB, +2.8%), retaining timestep conditioning without an additional parallel decoder branch.

D The Use of Large Language Models and AI tools

During the preparation of this paper, we utilized publicly available large language models (LLMs) or AI tools only to aid in polishing the writing. The model's role was strictly limited to improving grammar, refining sentence structure, and enhancing the overall clarity and readability of the text. All scientific contributions, including the core ideas, experimental design, and analysis of results, are exclusively our work. We carefully reviewed and edited all model-generated suggestions and retain full responsibility for the final content of this paper.

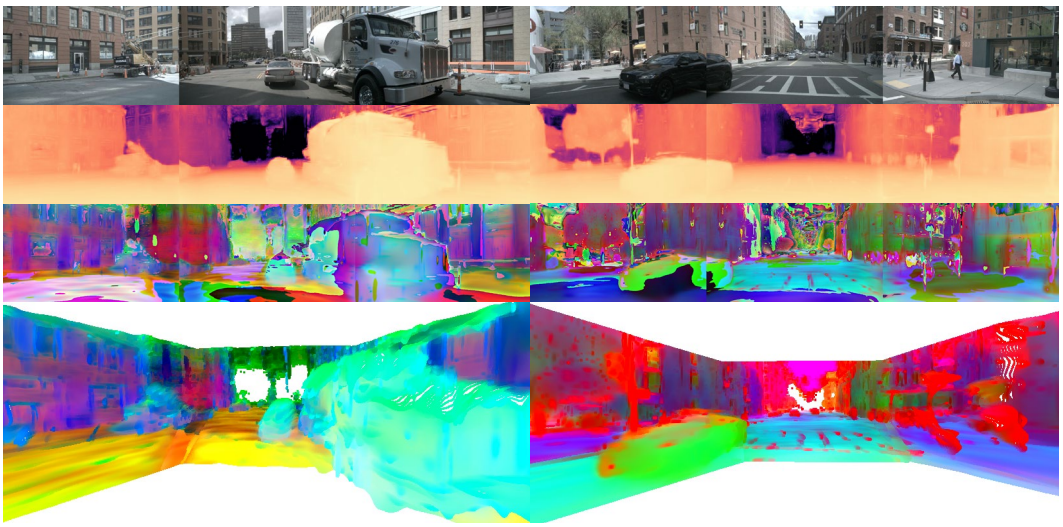

Figure 5: **3D visualization of pre-trained features with predicted depth.** Given multi-view RGB images as input (row 1), we lift the features into 3D and render them from novel viewpoints (row 4) by combining the PCA-projected feature (row 3) with the predicted depth (row 2) from the pre-trained GaussianDiffuser. The resulting novel-view visualizations preserve coherent scene geometry and view-consistent feature structures, indicating that pre-training already organizes visual features in a geometry-aware 3D space. This helps explain the strong transfer of GaussianDiffuser to downstream occupancy prediction and 3D object detection.

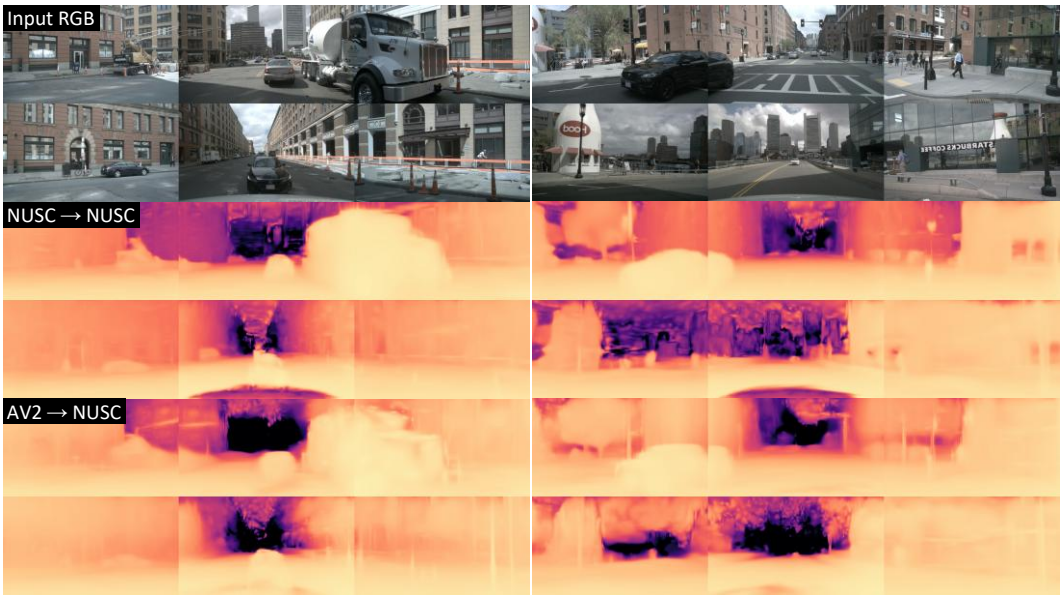

Figure 6: **Robust depth prediction under dataset shift.** We analyze the depth prediction from a model pre-trained on nuScenes (Caesar et al., 2020) and evaluated at the matched resolution of 448×800 as a reference setting (row 2), given the input multi-view RGB images (row 1). Against this reference, we examine zero-shot inference on nuScenes using a model pre-trained on Argoverse 2 (Wilson et al., 2021) without nuScenes fine-tuning (row 3). The shifted setting still preserves plausible road geometry, foreground object structure, and distant scene layout, indicating that GaussianDiffuser learns a geometric prior that remains effective under dataset shift.

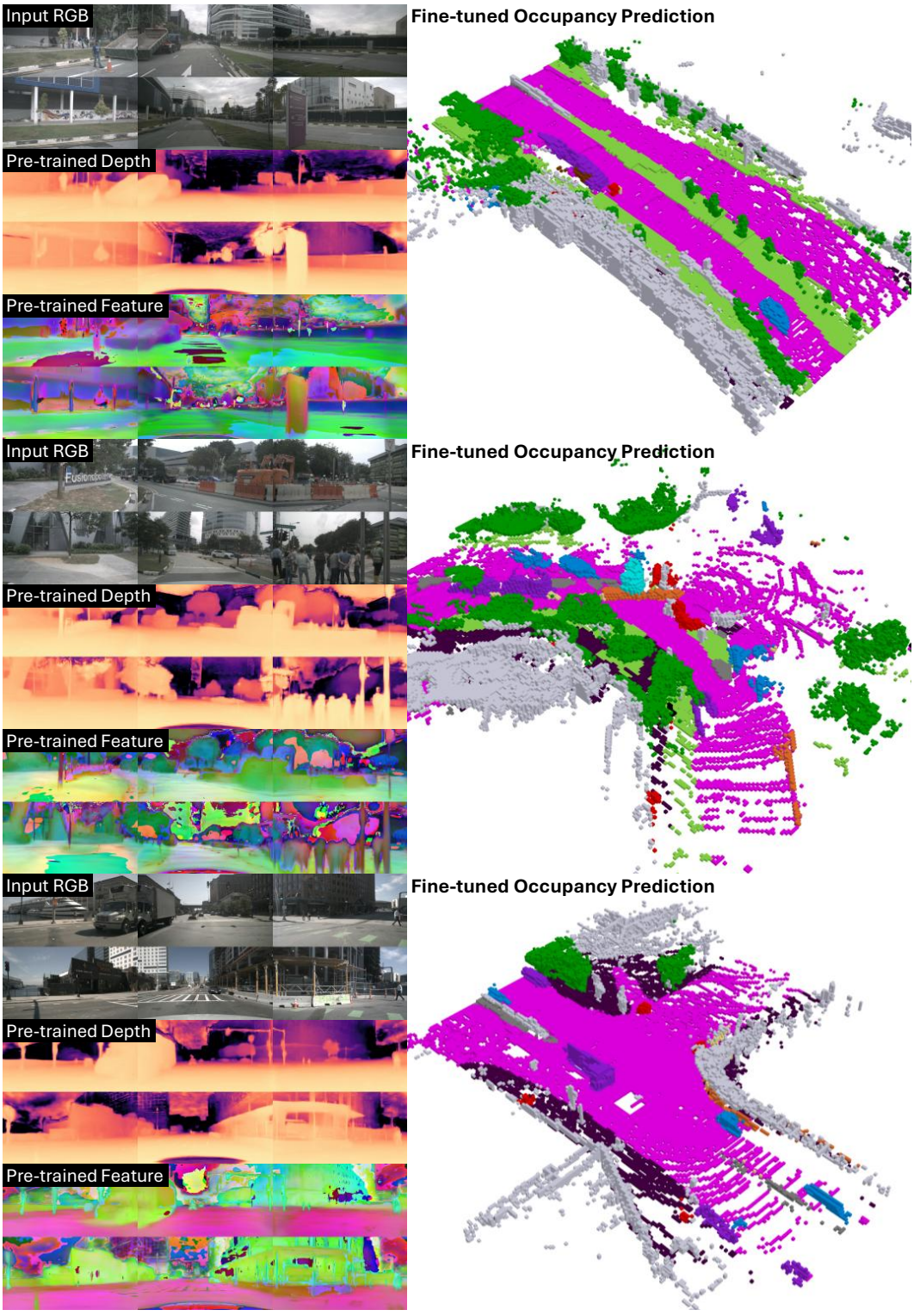

Figure 7: **Qualitative results of pre-trained depth/features and fine-tuned occupancy prediction.** For each scene, left panels show pre-trained depth predictions and PCA-projected features from multi-view images, while right panels show fine-tuned occupancy predictions on Occ3D (Tian et al., 2023). Pre-trained results capture coherent geometry and structured feature regions, which transfer to dense occupancy prediction with consistent scene layout and object-level details.