

MM-TRELLIS: Point-Cloud Guided Multi-Modal 3D Vehicle Generation in Autonomous Driving

Hongli Xiao^{1,2,3*}, Youjian Zhang^{3*}, Yucai Bai³, Chaoyue Wang⁵, Yaohui Jin¹,
 Xiaoguang Ren², Wenjing Yang⁴, Long Lan^{4†}

Abstract—Recovering realistic 3D vehicle models from autonomous driving scenes is crucial for synthesizing training data and building simulation environment. However, most existing vehicle generation methods fail to fully exploit multimodal sensors (*i.e.* multi-view images and LiDAR point clouds) and rely on neural rendering based reconstruction, leading to low-quality mesh. Recently, native 3D generative models have made significant progress, yet they are not built for arbitrary multi-view inputs and often struggle with in-the-wild driving images. In this work, we present MM-TRELLIS, a multi-modal version of TRELLIS for in-the-wild 3D vehicle generation that integrates LiDAR and image sensors from autonomous driving datasets into native 3D generative models. Specifically, multi-view images are cycled as conditioning inputs, while LiDAR point clouds provide test-time guidance to ensure geometric accuracy and cross-view consistency. During denoising, we first align the guidance point cloud with the model priors, then enforce consistency between the generated geometry and the guidance point cloud. Finally, we introduce a voxel filtering strategy based on the opacity of 3D Gaussian Splatting to suppress floaters and produce clean meshes. Comprehensive experiments on Waymo dataset demonstrate our method outperforms existing methods in high-fidelity 3D vehicle generation. Code is available at <https://github.com/HongliXiao/MM-TRELLIS>.

I. INTRODUCTION

The development of autonomous driving and robotic systems increasingly relies on large-scale and diverse 3D assets. In particular, high-fidelity vehicle models are crucial for synthesizing training data for perception modules, constructing realistic simulation environments, and enabling safe interaction in complex urban scenes. Nevertheless, manual creation is costly and unscalable, calling for automatic approaches to recover 3D vehicle models directly from raw sensory data. In this work, we intend to generate high-fidelity 3D vehicle assets from existing multimodal self-driving datasets [1].

Recent efforts have explored creating 3D vehicle models in real-world road scenarios, yet fundamental limitations remain. Methods that reconstruct solely on single-view images [2]–[5] are inherently ill-posed and tend to produce ambiguous geometry. Multi-view methods [6], [7]

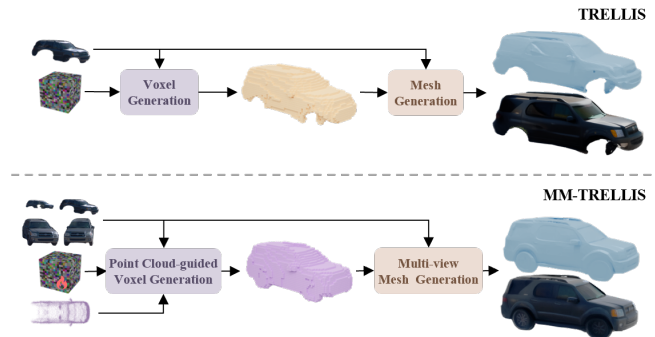


Fig. 1: TRELIS vs. MM-TRELIS. Top (TRELIS): single-image voxel generation followed by mesh decoding. Bottom (MM-TRELIS): We propose a point cloud-guided voxel generation and mesh generation with multi-view inputs, yielding robust and accurate geometry.

mitigate some of the ill-posedness, but the limited viewpoints and severe occlusions in driving datasets still lead to incomplete reconstructions and degraded fidelity. Moreover, most vehicle generation methods [4], [6]–[10] rely on neural rendering-based reconstruction, leading to slow inference and low-quality mesh extraction.

Recently, native 3D generative models that directly generate 3D representations [11]–[14] have achieved remarkable progress in synthesizing high-quality 3D assets. Trained on large-scale synthetic datasets [15]–[17], these models learn strong shape priors and show impressive generalization ability to unseen image textures. Among them, TRELLIS [14] represents a milestone by introducing a two-stage generation pipeline with structured voxel latents, where the first stage predicts a coarse geometric structure in a sparse 3D grid, and the second stage enriches it with local latent features. However, TRELLIS is not specifically designed for in-the-wild 3D vehicle generation, and several limitations arise when directly applying it to this task: i) the generative model is trained with single-image conditioning and tends to produce inaccurate geometry when handling multi-view inputs. In addition, it lacks the ability to incorporate LiDAR point clouds; ii) the framework assumes high-resolution, object-centered images as input, and its performance degrades significantly when conditioned on lower-quality inputs with diverse viewpoints and heavy occlusions (as shown in Fig. 1).

Our goal is to leverage the strong 3D geometric priors of state-of-the-art native 3D generation methods while fully exploiting the multimodal inputs specialized for autonomous driving datasets. Thus, we present MM-TRELLIS, a zero-shot 3D vehicle generation method with multi-modal inputs. First, given multiple viewpoints of a vehicle, TRELLIS can

*Equal contribution. Work done during internship at Bosch.

†Corresponding author: Long Lan (long.lan@nudt.edu.cn).

¹MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China (honglixiao@sjtu.edu.cn)

²Academy of Military Science, China

³Bosch innovation software development (Wuxi) Co., Ltd., China Technology, China (Youjian.Zhang@cn.bosch.com)

⁴College of Computer Science and Technology, National University of Defense Technology, China

⁵Shopee Pte. Ltd., China

cyclically inject different viewpoints as conditioning inputs to aggregate complementary visual information. However, naively adopting such cycle-conditioning is suboptimal, as different image conditions may provide contradictory optimization gradient, leading to inaccurate and low-quality geometry. To address this issue, we incorporate LiDAR point clouds as test-time guidance to encourage consistency between the generated voxel and the LiDAR structure. By leveraging the iterative nature of the diffusion process, our point cloud guidance scheme provides explicit geometric supervision without retraining the model, thereby avoiding the risk of degrading the learned 3D geometric prior.

Noted that the generated voxel does not have a fixed orientation (varies with the conditioning view), so we first optimize the orientation of the guidance point cloud to align with the generated voxel in order to make the guidance work properly. Finally, there are still floaters and extra geometry in the mesh generation, therefore, we propose a voxel filtering strategy based on the opacity of generated 3D Gaussian Splatting (3DGS) [18] to further refine and clean the geometry.

Our main contributions are summarized as follows:

- We introduce MM-TRELLIS, a zero-shot 3D vehicle generation framework that adapts native 3D diffusion priors to multimodal autonomous driving data.
- We propose a point cloud-guided test-time optimization scheme. By resolving the orientation disparity and regularizing the geometry generation with LiDAR structure during denoising, it imposes explicit geometric supervision and effectively enhances robustness under occlusions and challenging viewpoints.
- We propose an opacity-based voxel filtering strategy to remove floaters and further improve surface fidelity.
- Extensive experiments show that MM-TRELLIS surpasses existing 3D/vehicle generation methods, enabling high-quality in-the-wild 3D vehicle generation.

II. RELATED WORK

A. 3D Vehicle Modeling in the Wild

Recovering 3D vehicle models from in-the-wild driving data has been widely studied [2], [19]–[24]. Single-view methods [3]–[5], [8] take one image as input and often rely on category priors or large-scale training. While effective in constrained settings, they remain ill-posed in driving scenarios with occlusions and challenging viewpoints, leading to geometric ambiguity and inconsistent novel views. Multi-view approaches [6], [7] mitigate ambiguities by aggregating multiple observations, but the limited camera viewpoints and frequent occlusions in driving datasets still result in incomplete reconstructions and degraded fidelity. Approaches that incorporate LiDAR take different forms: GINA-3D [8] uses LiDAR depth as auxiliary supervision during training, GenAssets [10] learns latent priors with rendered LiDAR constraints, and ProtoCar [25] leverages completed LiDAR point clouds only as training labels. In all cases, LiDAR serves as supervision rather than directly guiding generation, which limits their ability to impose reliable geometric

constraints. In contrast, our method employs LiDAR as test-time guidance tightly coupled with the denoising trajectory, allowing it to directly guide geometry generation under occlusions and challenging viewpoints.

B. Native 3D Generative Models

Early work [26]–[30] in 3D generation often relied on optimization-based pipelines that lifted 2D priors into 3D representations, but these methods are slow and limited in fidelity. More recently, feed-forward paradigms have enabled direct 3D synthesis, comprising 2D-lifting methods [31]–[35] that exploit image priors and native 3D models [11]–[14], [36], [37] that learn directly in 3D space.

Among them, native 3D generative models have shown the strongest potential for high-fidelity and scalable synthesis. Trained on large datasets, these models learn powerful shape priors and support controllable generation across categories. Representative designs include latent set representations [11], [36], [38], [39], voxel-structured latents [14], [40], and scalable sparse encodings [41], [42]. In particular, TRELLIS [14] introduced a two-stage pipeline where coarse voxel grids are first generated and then enriched with local features, significantly improving scalability, stability, and interpretability. Despite these advances, existing models are primarily trained on clean synthetic assets and conditioned on single-view images or text prompts. This leaves a notable domain gap when applied to autonomous driving data, where vehicles appear under occlusions, extreme viewpoints, and sensor noise, limiting their applicability to real-world vehicle modeling.

III. METHOD

We aim to generate realistic 3D vehicle assets from multimodal data based on native 3D generation method. The overall framework is illustrated in Fig. 2. Given multi-view images of a vehicle, MM-TRELLIS performs cycle-conditioning to aggregate complementary appearance cues across different viewpoints. Meanwhile, LiDAR point clouds are incorporated as a test-time guidance signal, ensuring an accurate and synchronized geometry. Finally, an opacity-based voxel filtering step is applied to suppress floaters and produce clean, high-fidelity 3D vehicle models.

A. Multi-view Cycle Conditioning

In autonomous driving datasets, vehicles are typically captured from different onboard cameras and across multiple frames, offering complementary visual observations for 3D generation. While a single image often suffers from occlusion, truncation, or perspective distortion, aggregating information across views can provide richer and more complete appearance information. Although not explicitly stated in the original TRELLIS paper [14], a straightforward way to exploit such multi-view information is to employ a cycle-conditioning mechanism. Formally, given N selected images of a vehicle:

$$\mathcal{I} = \{I_1, I_2, \dots, I_N\}, \quad (1)$$

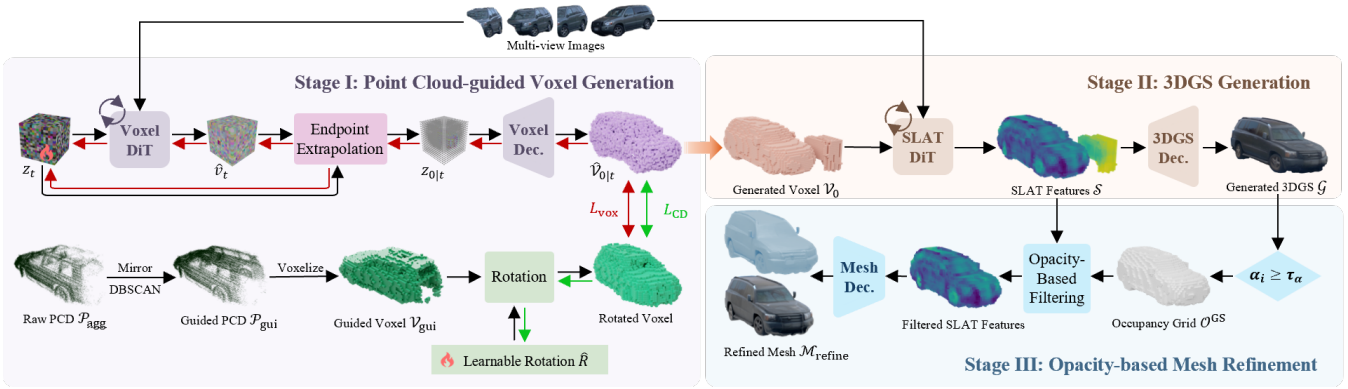


Fig. 2: **Overview of proposed MM-TRELLIS.** In stage I, we generate voxels with multi-view cycle conditioning and point cloud guidance. The LiDAR point cloud is first preprocessed and rotated by a learnable parameter \hat{R} to a aligned orientation. Then the voxel guidance is applied to optimize the sampled latent during the denoising process. Stage II perform a 3DGS generation with multi-view conditioning, ensuring the texture fidelity of the generation. Finally, opacity-based mesh refinement is performed in Stage III: a voxel mask is obtained by thresholding Gaussian opacity and filtering SLAT features, and decoding the filtered features produces the final clean mesh.

where each I_i may come from different cameras or different frames. During iterative denoising, the diffusion model predicts a cleaner latent sample \mathbf{x}_t from noisy input \mathbf{x}_{t+1} conditioned on an image $I_i \in \mathcal{I}$. We perform cycle-conditioning by switching the conditioning image at each step:

$$\mathbf{x}_t = f_\theta(\mathbf{x}_{t+1}, I_{(t \bmod N)+1}, t), \quad (2)$$

where f_θ is the denoising network, and the index cycles over the available views. This ensures that each denoising step is informed by a different viewpoint of the same vehicle.

B. LiDAR point cloud-guided Geometry Generation

Simply applying multi-view image conditioning can be suboptimal: i) images in autonomous driving scenes often suffer from occlusions and varying viewpoints, leading to incomplete or distorted geometry; ii) multi-view conditions may not cooperate effectively, as they can provide contradictory optimization gradients at each timestep.

To address these issues, we involve LiDAR point clouds as a test-time guidance signal for the denoising process of the voxel generation. Unlike RGB images, LiDAR observations can be easily aggregated across multiple frames, providing metric-accurate surface measurements that remain reliable under occlusions, varying illumination, and long-distance viewpoints. Consequently, LiDAR point clouds can serve as an effective modality for accurate and robust geometric constraints. Specifically, the point-cloud guidance can be decomposed into three stages: i) LiDAR preprocessing; ii) Voxel-guided geometry generation; iii) Voxel pose optimization, as detailed in the following.

LiDAR Preprocessing. Raw LiDAR point clouds collected in one timestep often contain noise, outliers, and missing regions due to sensor accuracy error and occlusions. To produce an accurate and complete reference for guidance, we first aggregate the raw vehicle point cloud $\mathcal{P}_{\text{agg}} = \sum_{i=0}^N \mathcal{P}_i$ through the sequence, while \mathcal{P}_i is the point of target vehicle in each frame.

However, in driving scenes, a vehicle is usually observed from only one side, leaving the opposite side of the point

cloud empty. Thus, we exploit the symmetry characteristic of vehicles by mirroring the raw point cloud across the vehicle’s center plane, producing a symmetrized point set:

$$\mathcal{P}_{\text{sym}} = \mathcal{P}_{\text{agg}} \cup \text{Mirror}(\mathcal{P}_{\text{agg}}). \quad (3)$$

The $\text{Mirror}(\cdot)$ operation flips the points by left-right symmetry plane, which can be estimated from bounding boxes.

We randomly downsample the point cloud if it contains more than 100,000 points, in order to preserves sufficient geometric fidelity while ensuring computational efficiency:

$$\mathcal{P}_{\text{down}} \subset \mathcal{P}_{\text{sym}}, \quad |\mathcal{P}_{\text{down}}| = \min(|\mathcal{P}_{\text{sym}}|, 10^5). \quad (4)$$

Finally, we apply DBSCAN [43] clustering to filter out scattered outliers, which can be formulated as:

$$\mathcal{P}_{\text{gui}} = \{\mathbf{p} \in \mathcal{P}_{\text{down}} \mid \text{DBSCAN}(\mathcal{P}_{\text{down}})_{\text{label}}(\mathbf{p}) \neq -1\}, \quad (5)$$

where points with label -1 are considered outliers in DBSCAN clustering results and are removed. This step effectively removes background noise and irrelevant points, resulting in a clean point cloud \mathcal{P}_{gui} as final guidance.

Voxel-guided Geometry Generation. One possible way to exploit point clouds as a conditioning signal is to encode the point cloud into a latent embedding and train cross-attention layers to impose the condition. However, this approach requires tedious post-training of the DiT. Furthermore, real-world 3D vehicle datasets are extremely scarce compared to the large-scale datasets used for training native 3D generative models. Post-training on such small datasets risks degrading the strong and generalizable 3D geometric priors.

Inspired by Marigold-DC [44], which formulates depth completion as monocular depth prediction guided by a sparse depth map, we aim to guide TRELLIS’s voxel generation towards the shape of a reference point cloud at test time without re-training model parameters.

First, we want to obtain both the guidance voxel and the predicted voxel in timestep t . Specifically, we voxelize \mathcal{P}_{gui} into a voxel grid $\mathcal{V}_{\text{gui}} \in \{0, 1\}^{H \times W \times D}$, where $\mathcal{V}_{\text{gui}}(u, v, w) = 1$ if the voxel in coordinate (u, v, w) contains at least one LiDAR point and 0 otherwise. Meanwhile, at

inference timestep t , the flow-matching DiT estimates a vector field \hat{v}_t , and we “preview” the denoised latent via an endpoint extrapolation [45], [46]:

$$z_{0|t} = z_t - t\hat{v}_t, \quad (6)$$

where z_t is the noisy latent at timestep t , $z_{0|t}$ is the corresponding estimated denoised result. Decoding $z_{0|t}$ with the voxel decoder yields the predicted occupancy $\hat{\mathcal{V}}_{0|t}$.

Then we calculate a masked Binary Cross Entropy (BCE) loss between the reference voxel grid \mathcal{V}_{gui} and the predicted voxel output $\hat{\mathcal{V}}_{0|t}$ only on the occupied voxels in \mathcal{V}_{gui} :

$$\mathcal{L}_{\text{vox}}(t) = -\frac{1}{|\mathcal{V}_{\text{gui}}|} \sum_{(u,v,w)} \mathcal{V}_{\text{gui}}(u,v,w) \log \hat{\mathcal{V}}_{0|t}(u,v,w), \quad (7)$$

where $|\mathcal{V}_{\text{gui}}|$ counts the nonzero voxels, and (u,v,w) is the 3D coordinates of the voxel grid.

By restricting supervision to non-empty voxels, we avoid penalizing regions without point cloud observations, enabling the use of incomplete point clouds as guidance. The gradients of the voxel guidance loss are back-propagated to the sampled latents, thereby steering the flow-matching process.

Notably, the final voxel generation is jointly conditioned on both the point cloud and input images. In other words, while the voxels are regularized to align with the reference point cloud, TRELLIS’s inherent 3D prior complements the missing regions. This leads to outputs with metric-accurate geometry while effectively addressing occlusions.

Voxel Pose Optimization. In early experiments, we observed that directly applying the voxel guidance loss sometimes led to poor convergence. This is because the orientation of voxels generated under the image condition is ambiguous and does not necessarily align with the canonical orientation of the reference voxel. When these two optimization objectives become too contradictory, the convergence becomes difficult.

To stabilize training, we optimize the pose of the guided voxel \mathcal{V}_{gui} such that it approximately aligns with the predicted coarse voxel. As the rectified-flow model holds a nice property that the sampling trajectory follows a stable probability transitions [45], empirically, the intermediate voxel output at around timestep $t = 0.9$ already exhibits geometry similar to the final result. This property allows us to first adjust the voxel orientation during the early denoising steps, after which voxel-guided geometry generation can proceed with well-aligned voxels.

Specifically, we convert both reference voxel and predicted voxel into point cloud:

$$\begin{aligned} P &= \{p \in \mathbb{R}^3 \mid \mathcal{V}_{\text{gui}}(p) = 1\}, \\ Q &= \{q \in \mathbb{R}^3 \mid \hat{\mathcal{V}}_{0|t}(q) = 1\}. \end{aligned} \quad (8)$$

Then we define a learnable rotation parameters as a 3 dimension axis angles, denoted as $R \in SO(3)$, and minimizing the one-sided Chamfer Distance (CD) between the predicted point cloud and the reference point cloud after rotation:

$$\mathcal{L}_{\text{CD}}(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \|Rot(p, R) - q\|_2^2, \quad (9)$$

where $Rot(\cdot, \cdot)$ is a rotation transformation operator. We only calculate one-sided Chamfer Distance since the reference voxel can be incomplete. The gradient of \mathcal{L}_{CD} is back-propagated to the rotation parameters R , thus optimize the orientation for voxel guidance.

In summary, our full test-time guidance uses a three-phase schedule within a 100-step sampler: i) Early sampling: run Stage I DiT and pause at step 30 ($t \approx 0.88$) to decode a coarse voxel; ii) Pose alignment: optimize a rotation R (applied to the guidance voxel grid \mathcal{V}_{gui}) for 100 iterations by minimizing \mathcal{L}_{CD} ; iii) Guided generation: resume denoising to $t=0$ by minimizing \mathcal{L}_{vox} at each step.

C. Opacity-based Voxel Filtering

Since the voxel guidance loss constrains only on voxels containing LiDAR points, it is not sufficient to suppress the generation of extra voxels, which may lead to floaters and noisy surface in the final generated mesh.

Interestingly, despite the presence of floaters in the mesh generation, TRELLIS’s Gaussian Splatting (3DGS) generation still produces clean renderings that faithfully match the multi-view image conditions. We infer that this robustness arises from the strong texture prior provided by multi-view images, as SLAT features are originally trained from the back-projection of image features. Leveraging this property, we perform voxel filtering based on the opacity of 3DGS.

Let \mathcal{S} be the generated SLAT features from TRELLIS stage II, decoding them as 3DGS yields a set of Gaussians

$$\mathcal{G} = \mathcal{D}_{\text{GS}}(\mathcal{S}) = \{(\mu_i, c_i, s_i, \alpha_i, r_i)\}_{i=1}^K, \quad (10)$$

where $\mu_i, c_i, s_i, \alpha_i, r_i, o_i$ denotes for positions, colors, scales, opacities, and rotations respectively. As aforementioned, the floaters are not visible in the rendering results, indicating the opacity of these regions are very low. Therefore, we first prune low-opacity splats through a threshold τ_α :

$$\mathcal{G}_{\text{prune}} = \{g_i \in \mathcal{G} \mid \alpha_i \geq \tau_\alpha\}. \quad (11)$$

The retained Gaussians $\mathcal{G}_{\text{prune}}$ are voxelized into an occupancy grid $\mathcal{O}^{\text{GS}} \in \{0, 1\}^{H \times W \times D}$, which is used as a mask for filtering the SLAT feature grid \mathcal{S} , and the refined mesh is generated using the TRELLIS mesh decoder:

$$\mathcal{M}_{\text{refine}} = \mathcal{D}_{\text{mesh}}(\mathcal{S} \odot \mathcal{O}^{\text{GS}}). \quad (12)$$

Similar to voxel-guided geometry generation, this procedure is a test-time refinement that does not require re-training the model parameters. We leverage the robustness of Gaussian Splatting generation to in turn refine the generated mesh, resulting in cleaner and high-fidelity mesh generations for downstream tasks.

IV. EXPERIMENTS

A. Experimental Setting

Datasets. We conduct experiments on the Waymo Open Dataset [1], which provides synchronized multi-view camera images and LiDAR scans captured in diverse real-world driving scenarios. Following the evaluation set in [7], we

TABLE I: Quantitative comparison on Waymo dataset. The quality of Novel-view Synthesis is measured by PSNR/SSIM/LPIPS. The accuracy of geometry is measured by Chamfer Distance (CD_{one} : LiDAR \rightarrow mesh, CD_{bi} : LiDAR \leftrightarrow mesh).

Image input	Method	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	CD_{one} \downarrow	CD_{bi} \downarrow	Time \downarrow
Single-view	InstantMesh [35]	22.34	0.9218	0.0843	0.1189	0.1892	9.6s
	TRELLIS [14]	19.81	0.9037	0.0912	0.0848	0.1050	25.8s
Multi-view	DreamCar [7]	20.19	0.9022	0.0972	0.1491	0.2128	118min
	MV-Hunyuan3D [12]	22.45	0.9231	0.0839	0.2443	0.1318	13.7min
	MV-TRELLIS [14]	20.39	0.9081	0.0834	0.0388	0.0479	24.3s
	Ours	22.61	0.9159	0.0720	0.0210	0.0361	37.3s



Fig. 3: Qualitative comparison with baseline methods in novel view synthesis. The four images in Input column represent the multi-view inputs for multi-view methods, while the image in the black box is used for the single-view method.

randomly select 100 vehicle instances that have been observed for more than 8 views from the first 30 scenes of the validation split.¹ For each vehicle, we prepare $N = 8$ images from different cameras or frames: 4 are used as conditioning inputs, and the remaining 4 are served as reversed views for image-based evaluation. The pseudo 3D ground-truth for geometry evaluation is derived by aggregating LiDAR scans in each frame and segmenting vehicle instances using the provided instance masks.

Implementation Details. Our framework builds upon the pretrained TRELLIS [14] without any additional finetuning. Stage I predicts a sparse 64^3 voxel occupancy grid. Differently, during inference, we extend the sampling steps from 25 to 100, and adjust the classifier-free guidance (CFG) to 3.0 Stage I. Moreover, voxel pose optimization is performed at the 30th step, followed by 70 steps of voxel-guided geometry generation using a BCE loss on LiDAR-occupied voxels. For voxel filtering, we decode SLAT features into 3DGS and prune Gaussians with opacity threshold $\tau_\alpha = 0.005$ before re-voxelization.

Baselines. We conduct comprehensive comparisons with state-of-the-art vehicle generation and 3D asset generation methods. Among them, DreamCar [7] is the SOTA vehicle generation method since some more recent methods [25] and [5] are not open source; InstantMesh [35], Trellis [14] and Hunyuan3D-2.0 [12] are several highly influential 3D generation methods. Moreover, we adopt MV-TRELLIS and MV-Hunyuan3D as the TRELLIS and Hunyuan3D model with multi-image conditioning. For fairness, InstantMesh and

TRELLIS, which take a single image as input, are evaluated by averaging the results over four runs for the same 4 conditioning images, as used in multi-view image methods. **Metrics.** We evaluate the generated 3D vehicles using both image- and geometry-level metrics. For image quality, we adopt PSNR, SSIM [47], and LPIPS [48] to measure the fidelity of rendered novel views against ground-truth images. For geometry, we compute Chamfer Distance (CD) between the generated mesh and aggregated LiDAR point clouds. Since LiDAR point clouds serve as pseudo ground truth and may be incomplete, we report both the one-sided Chamfer Distance (CD_{one}) and bidirectional (CD_{bi}). CD_{one} measures the distance from LiDAR points to the generated mesh and reflects how well the reconstruction matches the real-world shape. CD_{bi} further includes mesh-to-LiDAR distance, capturing the issue of extra-generation, where higher scores indicate floaters or extra structures.

B. Comparison with State-of-the-art Methods

Evaluation on Novel-view Synthesis. We evaluate the results of novel view synthesis (NVS) to mainly assess the geometric accuracy and generalization on real-world images of our method. The quantitative and qualitative results are presented in Table I and Fig. 3, respectively. As shown in Table 1, our method achieves best performance in both PSNR and LPIPS. Notably, despite reporting high PSNR and SSIM, InstantMesh [35] and MV-Hunyuan3D [37] often produce highly inaccurate geometry, as revealed by the Chamfer Distance metric and Fig. 4. Upon closer inspection, we find the failure cases of InstantMesh and MV-Hunyuan3D often

¹The selected scene IDs and vehicle IDs will be public.

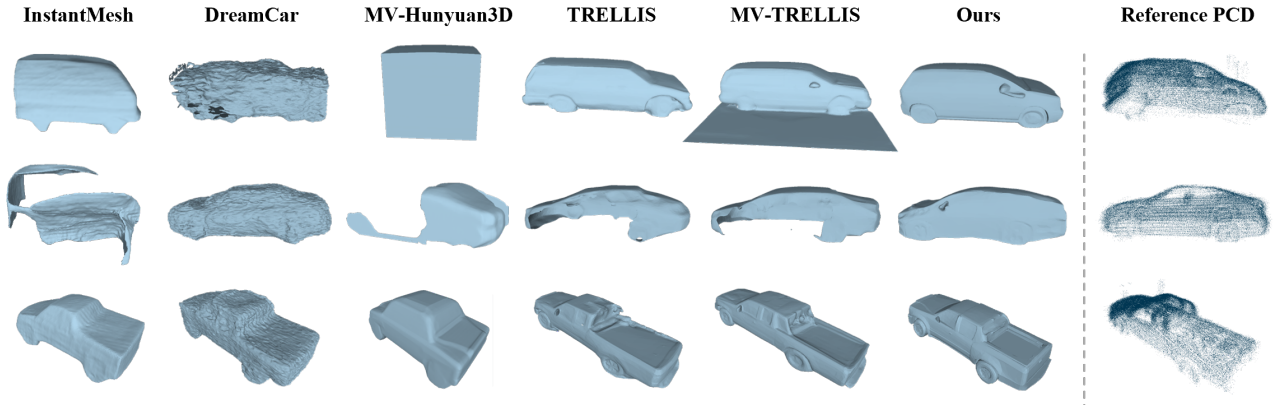


Fig. 4: Qualitative comparison with baseline methods in 3D geometry. Rightmost column shows the reference LiDAR point cloud.

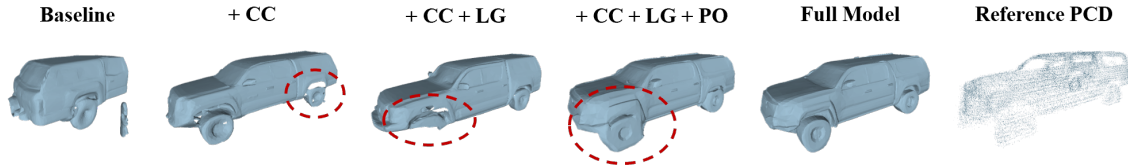


Fig. 5: An visual comparison for ablation study. CC, LG, PO stand for cycle conditioning, LiDAR guidance, and pose optimization respectively. Red dashed circles highlight improvements.

TABLE II: Ablation study of the proposed modules.

ID	Cycle Cond.	LiDAR Gui.	Pose Optim.	Mesh Refine	$CD_{one} \downarrow$	$CD_{bi} \downarrow$
1					0.0848	0.1050
2	✓				0.0388	0.0479
3	✓	✓			0.0292	0.1526
4	✓	✓	✓		0.0267	0.0496
5 (Ours)	✓	✓	✓	✓	0.0210	0.0361

produce severely inaccurate geometry and blurry textures. Since these meshes deviate significantly from true vehicle shape, they cannot be properly aligned in the canonical space. As a result, the rendering results only cover part of the vehicle with over-smoothed texture, which may be favored by PSNR and SSIM. In addition, since PSNR/SSIM are computed only on valid pixels, they fail to capture our method’s advantage in completing occluded regions.

Meanwhile, the comparison with TRELLIS [14], DreamCar [7], and MV-TRELLIS can more accurately reflect the geometry quality, as the orientation of their meshes align more faithfully with the true vehicle. Specifically, when comparing with the TRELLIS series, it can be observed that the cars generated by TRELLIS tend to be slim, with inaccurate length-to-height ratios (see Fig. 3 first-second rows). In contrast, our results preserve correct proportions, enabling consistent pixel-alignment with ground-truth and yielding over 2dB PSNR improvement.

Evaluation on Mesh Generation. Similar conclusions emerge when evaluating Chamfer Distance (CD) to directly measure the geometric accuracy of the reconstructed meshes. As shown in Tab. I, our method achieves the lowest error in both one-sided and bidirectional CD, significantly outperforming the TRELLIS series baseline. Combined with the visual comparisons in Fig. 4, we observe that single-

view methods (InstantMesh [35] and TRELLIS [14]) often produce incomplete or distorted results due to their strong reliance on the input image quality. DreamCar’s [7] reconstructions are coarse, limited by its NeRF-based mesh extraction. MV-Hunyuan3D [37] struggles with arbitrary multi-view inputs, as it is trained under fixed viewpoints. MV-TRELLIS [14], while better leveraging multi-view conditioning, still suffers from unstable extra-mesh generation (Fig. 4-first row), incompleteness (Fig. 4-second row), and inaccurate metric length (Fig. 4-third row). In contrast, our proposed point cloud guidance effectively resolves these issues. Although it may seem unfair that we use LiDAR as condition, yet our ability to fully exploit this information is the core contribution of our method. Both quantitative and qualitative results demonstrate that we generate meshes that are highly consistent with real-world vehicles shape.

In terms of efficiency, MM-TRELLIS reconstructs one vehicle in 37.3s on a single NVIDIA H20 GPU. The generation process (Stage I and Stage II) takes 23.6 s, followed by 13.7 s for mesh decoding.

C. Ablation Study

We conduct ablation experiments to validate the contribution of each component in our framework. As shown in Tab. II and Fig. 5, adding LiDAR guidance clearly improves the CD_{one} score, indicating better alignment between LiDAR point cloud and the generated mesh. However, the orientation ambiguity can lead to unstable optimization, causing the CD_{bi} becomes worse. Incorporating pose optimization effectively resolves this issue, leading to more stable training and more accurate geometry. Finally, the opacity-based mesh refinement further removes floaters and noisy artifacts, producing clean and realistic vehicle surfaces.

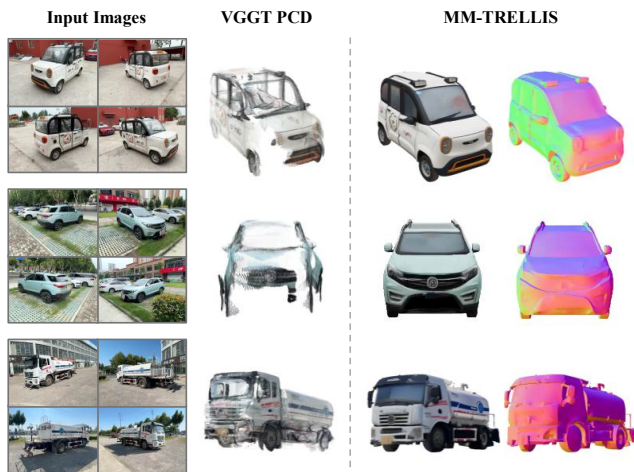


Fig. 6: Extend our method to VGGT [49] point cloud as 3D priors. Input images are sampled from the 3DRealCar [50] dataset, and point clouds reconstructed by VGGT are used as geometric guidance.

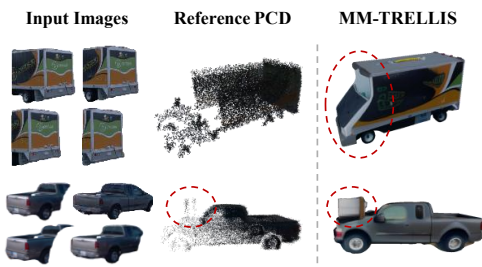


Fig. 7: Failure cases.

D. Extension to Feed-forward Point Cloud Priors

Our method not only can utilize LiDAR point as priors, but also works on point cloud acquired from sparse-view images by other approaches (i.e. point clouds obtained from fast feed-forward reconstruction VGGT [49]). As shown in Fig. 6, the reconstructed point clouds can be directly used as geometric guidance during voxel generation without any retraining. Although such point clouds are typically noisier and less complete than LiDAR data, MM-TRELLIS still produces geometrically consistent results. This demonstrates that our test-time voxel guidance mechanism is not restricted to LiDAR and can flexibly incorporate alternative 3D priors.

E. Limitations

As shown in Fig. 7, our method may degrade when the inputs are severely incomplete or noisy. Truncated image observations (Fig. 7-first column) can lead to missing geometry due to insufficient visual cues during cycle-conditioning. Noisy point clouds (Fig. 7-second column) may introduce incorrect geometric supervision, resulting in additional structures in the reconstructed mesh. Although opacity-based voxel filtering is applied, severe noise may still affect the final reconstruction. These cases indicate that reliable visual cues and clean geometric guidance are important for stable reconstruction.

V. CONCLUSIONS

In this work, we present MM-TRELLIS, a zero-shot framework for real-world 3D vehicle generation that adapts native 3D diffusion priors to multimodal autonomous driving data. Our method combines multi-view cycle-conditioning with LiDAR-guided optimization to aggregate image information and enforce geometric consistency. A pose optimization is proposed to resolve orientation ambiguity, and an opacity-based filtering step is further applied to get a clean mesh. Experiments on the Waymo dataset show that MM-TRELLIS outperforms existing methods, delivering geometric-accurate and realistic 3D vehicle assets and highlighting its potential to enrich large-scale autonomous driving datasets.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No. 62376282) and the Science and Technology Innovation Program of Hunan Province (No.2025RC3117).

REFERENCES

- [1] P. Sun, H. Kretschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.
- [2] N. Müller, A. Simonelli, L. Porzi, S. R. Bulo, M. Nießner, and P. Kotschieder, “Autorf: Learning 3d object radiance fields from single view observations,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3971–3980.
- [3] Y. Guo, A. Kumar, C. Zhao, R. Wang, X. Huang, and L. Ren, “Sup-nerf: A streamlined unification of pose estimation and nerf for monocular 3d object reconstruction,” in *European Conference on Computer Vision*. Springer, 2024, pp. 37–53.
- [4] T. Liu, H. Zhao, Y. Yu, G. Zhou, and M. Liu, “Car-studio: learning car radiance fields from single-view and unlimited in-the-wild images,” *IEEE Robotics and Automation Letters*, vol. 9, no. 3, 2024.
- [5] C. Lin, B. Zhuang, S. Sun, Z. Jiang, J. Cai, and M. Chandraker, “Drive-1-to-3: Enriching diffusion priors for novel view synthesis of real vehicles,” *arXiv preprint arXiv:2412.14494*, 2024.
- [6] Y. Xu, M. Chai, Z. Shi, S. Peng, I. Skorokhodov, A. Siarohin, C. Yang, Y. Shen, H.-Y. Lee, B. Zhou *et al.*, “DiscoScene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 4402–4412.
- [7] X. Du, H. Sun, M. Lu, T. Zhu, and X. Yu, “Dreamcar: Leveraging car-specific prior for in-the-wild 3d car reconstruction,” *IEEE Robotics and Automation Letters*, 2024.
- [8] B. Shen, X. Yan, C. R. Qi, M. Najibi, B. Deng, L. Guibas, Y. Zhou, and D. Anguelov, “Gina-3d: Learning to generate implicit neural assets in the wild,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 4913–4926.
- [9] Z. Liu, Z. Fu, G. Li, J. Hu, and Y. Yang, “Sterf: symmetric triplane neural radiance fields for novel view synthesis from single-view vehicle images,” *Applied Intelligence*, vol. 55, no. 5, p. 322, 2025.
- [10] Z. Yang, J. Wang, H. Zhang, S. Manivasagam, Y. Chen, and R. Urtasun, “Genassets: Generating in-the-wild 3d assets in latent space,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22392–22403.
- [11] L. Zhang, Z. Wang, Q. Zhang, Q. Qiu, A. Pang, H. Jiang, W. Yang, L. Xu, and J. Yu, “Clay: A controllable large-scale generative model for creating high-quality 3d assets,” *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–20, 2024.
- [12] Z. Zhao, Z. Lai, Q. Lin, Y. Zhao, H. Liu, S. Yang, Y. Feng, M. Yang, S. Zhang, X. Yang *et al.*, “Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation,” *arXiv preprint arXiv:2501.12202*, 2025.

- [13] Z. Lai, Y. Zhao, H. Liu, Z. Zhao, Q. Lin, H. Shi, X. Yang, M. Yang, S. Yang, Y. Feng *et al.*, “Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details,” *arXiv preprint arXiv:2506.16504*, 2025.
- [14] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang, “Structured 3d latents for scalable and versatile 3d generation,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 469–21 480.
- [15] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, “Shapenet: An information-rich 3d model repository,” *arXiv preprint arXiv:1512.03012*, 2015.
- [16] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. Vanderbilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, “Objaverse: A universe of annotated 3d objects,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 13 142–13 153.
- [17] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre *et al.*, “Objaverse-xl: A universe of 10m+ 3d objects,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 35 799–35 813, 2023.
- [18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [19] M. Niemeyer and A. Geiger, “Giraffe: Representing scenes as compositional generative neural feature fields,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 453–11 464.
- [20] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. J. Guibas, J. Tremblay, S. Khamis *et al.*, “Efficient geometry-aware 3d generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 123–16 133.
- [21] P. Karpikova, D. Selikhonovych, K. Struminsky, R. Musaev, M. Golitsyna, and D. Baranchuk, “Madrive: Memory-augmented driving scene modeling,” *arXiv preprint arXiv:2506.21520*, 2025.
- [22] Y. Li, J. Zhang, X. Teng, H. Zhang, X. Liu, and L. Lan, “Refsam: Efficiently adapting segmenting anything model for referring video object segmentation,” *Neural Networks*, p. 108000, 2025.
- [23] K. Xia, J. Jia, K. Jin, Y. Bai, L. Sun, D. Tao, and Y. Zhang, “D² gs: Dense depth regularization for lidar-free urban scene reconstruction,” *arXiv preprint arXiv:2510.25173*, 2025.
- [24] L. Lan, Z. Hu, H. Li, T. Liu, and X. Liu, “C-woe: Clustering for out-of-distribution detection learning with wild outlier exposure,” *IEEE Transactions on Image Processing*, 2026.
- [25] H. Liu, H. Yu, B. Zou, J. Lyu, Q. Mei, J. Chen, and H. Ma, “Protocar: Learning 3d vehicle prototypes from single-view and unconstrained driving scene images,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 5, 2025, pp. 5460–5468.
- [26] R. Chen, Y. Chen, N. Jiao, and K. Jia, “Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 22 246–22 256.
- [27] G. Qian, J. Mai, A. Hamdi, J. Ren, A. Siarohin, B. Li, H.-Y. Lee, I. Skorokhodov, P. Wonka, S. Tulyakov *et al.*, “Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors,” in *The Twelfth International Conference on Learning Representations*.
- [28] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, “Magic3d: High-resolution text-to-3d content creation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 300–309.
- [29] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “Dreamfusion: Text-to-3d using 2d diffusion,” in *The Eleventh International Conference on Learning Representations*, 2023.
- [30] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu, “Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [31] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, “Zero-1-to-3: Zero-shot one image to 3d object,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 9298–9309.
- [32] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt *et al.*, “Wonder3d: Single image to 3d using cross-domain diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9970–9980.
- [33] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su, “Zero123++: a single image to consistent multi-view diffusion base model,” *arXiv preprint arXiv:2310.15110*, 2023.
- [34] K. Wu, F. Liu, Z. Cai, R. Yan, H. Wang, Y. Hu, Y. Duan, and K. Ma, “Unique3d: High-quality and efficient 3d mesh generation from a single image,” *arXiv preprint arXiv:2405.20343*, 2024.
- [35] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, “Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models,” *arXiv preprint arXiv:2404.07191*, 2024.
- [36] B. Zhang, J. Tang, M. Niessner, and P. Wonka, “3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models,” *ACM Transactions On Graphics (TOG)*, vol. 42, no. 4, pp. 1–16, 2023.
- [37] T. Hunyuan3D, S. Yang, M. Yang, Y. Feng, X. Huang, S. Zhang, Z. He, D. Luo, H. Liu, Y. Zhao *et al.*, “Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material,” *arXiv preprint arXiv:2506.15442*, 2025.
- [38] R. Chen, J. Zhang, Y. Liang, G. Luo, W. Li, J. Liu, X. Li, X. Long, J. Feng, and P. Tan, “Dora: Sampling and benchmarking for 3d shape variational auto-encoders,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 16 251–16 261.
- [39] Z. Zhao, W. Liu, X. Chen, X. Zeng, R. Wang, P. Cheng, B. Fu, T. Chen, G. Yu, and S. Gao, “Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation,” *Advances in neural information processing systems*, vol. 36, pp. 73 969–73 982, 2023.
- [40] C. Ye, Y. Wu, Z. Lu, J. Chang, X. Guo, J. Zhou, H. Zhao, and X. Han, “Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging,” *arXiv preprint arXiv:2503.22236*, vol. 3, p. 2, 2025.
- [41] Z. Li, Y. Wang, H. Zheng, Y. Luo, and B. Wen, “Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling,” *arXiv preprint arXiv:2505.14521*, 2025.
- [42] S. Wu, Y. Lin, F. Zhang, Y. Zeng, Y. Yang, Y. Bao, J. Qian, S. Zhu, X. Cao, P. Torr *et al.*, “Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention,” *arXiv preprint arXiv:2505.17412*, 2025.
- [43] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [44] M. Viola, K. Qu, N. Metzger, B. Ke, A. Becker, K. Schindler, and A. Obukhov, “Marigold-dc: Zero-shot monocular depth completion with guided diffusion,” *arXiv preprint arXiv:2412.13389*, 2024.
- [45] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” *arXiv preprint arXiv:2210.02747*, 2022.
- [46] J. Choi, J. Kang, and B. Han, “Enhanced diffusion sampling via extrapolation with multiple ode solutions,” *arXiv preprint arXiv:2504.01855*, 2025.
- [47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [48] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
- [49] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, “Vggt: Visual geometry grounded transformer,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5294–5306.
- [50] X. Du, H. Sun, S. Wang, Z. Wu, H. Sheng, J. Ying, M. Lu, T. Zhu, K. Zhan, and X. Yu, “3drealcar: An in-the-wild rgb-d car dataset with 360-degree views,” *arXiv preprint arXiv:2406.04875*, 2024.