Urban Architect: Steerable 3D Driving Scene Generation with Layout Prior

Fan Lu, Kwan-Yee Lin, Yan Xu, Hongsheng Li, Guang Chen, Changjun Jiang

Abstract—Text-to-3D generation has achieved remarkable success via large-scale text-to-image diffusion models. Nevertheless, scaling this methodology to large-scale driving scenes is underexplored. Driving scenes, characterized by numerous elements, intricate arrangement relationships, and vast scale, present a formidable barrier to the interpretability of ambiguous textual descriptions for effective model optimization. In this work, we surmount the limitations by introducing a compositional 3D layout representation into text-to-3D paradigm, serving as an additional prior. It comprises a set of semantic primitives with simple geometric structures and explicit arrangement relationships, complementing textual descriptions and enabling steerable generation. Specifically, we represent the scene via 2D Gaussian splatting and develop two innovative optimization strategies: (1) Layout-Guided Variational Score Distillation, which addresses model optimization inadequacies by conditioning the score distillation sampling process with geometric and semantic constraints of 3D layouts; (2) Semantic-Depth Score Distillation Sampling, which ensures alignment of semantics, geometry, and appearance by matching the rendered semantic and depth maps with 3D layouts. Extensive experiments substantiate the capability of our framework to scale text-to-3D generation to large-scale driving scenes that cover over 1000m driving distance for the first time. We also present various scene editing demonstrations, showing the powers of steerable urban scene generation. Website: https://urbanarchitect.github.io.

Index Terms—3D generation, diffusion model, driving scene, text-to-3D, editing

1 Introduction

A steerable paradigm to create large-scale 3D driving scenes with realism and flexibility hosts utmost significance to various applications (*e.g.*, autonomous driving simulation, virtual reality, games, and *etc*.). In previous research, tasks related to 3D driving scene creation, such as 3D-aware image synthesis [1], [2], [3], [4] and 3D scene reconstruction [5], [6], have made strides with different methodologies and specializations. Despite these advancements, they are all stuck in the dilemma of trading-off among scene scale, quality, flexibility, and geometric consistency. This prompts a pivotal inquiry: *What methodology could facilitate the realization of steerable 3D driving scene creation?*

Text-to-3D generation offers a promising direction. Rooted in the flexible and versatile textual condition, it enjoys the excellence of both remarkable quality from text-to-image diffusion models, and the geometric consistency from 3D representations. Notable examples include, among many others, DreamFusion [7], Magic3D [8] and ProlificDreamer [9], *etc*. At the heart of these advances is the idea of Score Distillation Sampling (SDS) [7]. It brings the best of two worlds together by optimizing a 3D model via aligning the distribution of rendered images with the target distribution derived from the diffusion model. Nevertheless, the ascendancy faces a formidable challenge when directly extending the paradigm to large-scale driving scenes.

Driving scenes, characterized by their high semantic and geometric complexity, pose challenges for existing text-to-3D paradigms to create high-quality 3D content. Firstly, existing paradigms rely on textual descriptions for 3D model optimization; however, text prompts are inherently ambiguous and struggle to deliver fine-grained conditional signals. This granularity gap between text and real-world driving scenes triggers the insufficient optimization predicament of SDS-based methodology. SDS (and its variant VSD [9]) would fail to capture the complex multimodal distribution of driving scenes, given only the ambiguous textual condition. Secondly, previous methods typically optimize 3D models by aligning the distribution of rendered perceptual images, tending to optimize scene appearance to satisfy constraints. However, driving scenes consist of numerous objects with intricate spatial relationships and significant occlusions, and the sampled camera views in street environments are often limited. Consequently, relying solely on perceptual-based optimization can easily lead to suboptimal solutions by focusing only on appearance tuning, while the underlying semantics and geometry are misaligned.

In this work, we present Urban Architect, a lightweight solution that utilizes the power of a compositional 3D layout representation to overcome the above dilemmas and extend the current text-to-3D paradigm to large-scale driving scenes. The 3D layout representation is an in-hand specific prior, which could complement the ambiguities of text prompts in a breeze. As shown in Fig. 1, the 3D layout comprises a set of semantic primitives, where instances of a driving scene are depicted using these simple geometric structures (*e.g.*, cuboids for buildings and cars, ellipsoids

- *Fan Lu and Guang Chen are with the Physical Intelligence Lab, Tongji University, Shanghai, China; the Shanghai Innovation Institute, Shanghai, China; and Robotscale AI, Shanghai China. Kwan-Yee Lin and Yan Xu are with the University of Michigan, Ann Arbor, United States. Hongsheng Li is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China. Changjun Jiang is with the School of Computer Science, Tongji University, Shanghai, China. (Fan Lu and Kwan-Yee Lin contributed equally to this work.)*
- *Corresponding author: Guang Chen (guangchen@tongji.edu.cn).*

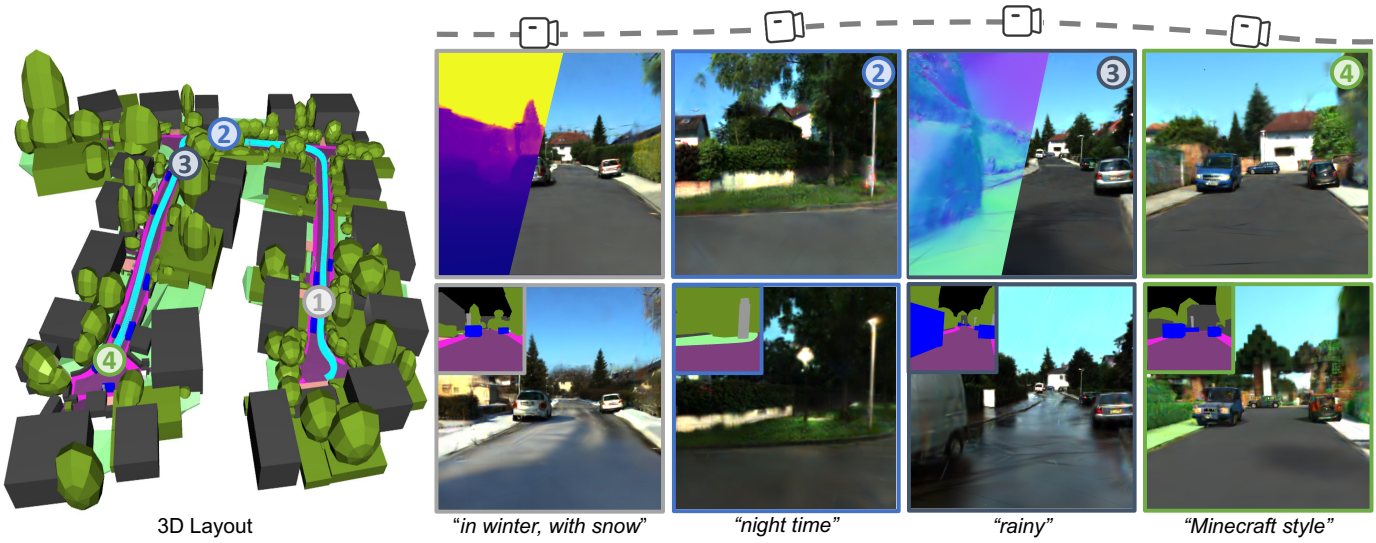


Fig. 1: **Urban Architect for steerable 3D driving scene generation.** We present Urban Architect, a method to generate steerable 3D driving scenes by introducing 3D layout as an additional prior to complement textual descriptions. The framework enjoys three key properties: (1) Large-scale driving scene creation. The scale goes beyond 1000m driving distance in our experiments. (2) High quality. The generated scene enables photo-realistic rendering quality (upper row) and obeys geometric consistency (We display the depth map of the first camera and the normal map of the third camera). (3) Steerable creation process. It supports various scene editing effects by fine-tuning the generated scene in a breeze (*e.g.*, style editing (lower row)).

for vegetation, and planes for roads), making it effective for guiding model optimization and easy for users to assemble. Centered on the representation, we provide two modifications to the current text-to-3D paradigm.

Firstly, to address model optimization inadequacies caused by ambiguous text prompts, we propose Layout-Guided Variational Score Distillation (LG-VSD). It is a conditional extension of SDS and VSD, which integrates additional conditions from the 3D layout into the optimization process. Concretely, we first map the 3D layout constraints to 2D space by rendering 2D semantic and depth maps at arbitrary camera views, which results in a sequence of 2D conditions with 3D consistency. To preserve the original power of the pre-trained 2D diffusion model, and meanwhile inject our explicit constraints, we leverage ControlNet [10] to integrate conditions from the rendered semantic and depth maps. Consequently, we could control the score distilling process with the 2D conditional signals derived from the 3D layout. As these signals provide explicit prior to semantic and geometric distribution of the scene, they could enforce the optimization process to match the 3D layout consistency and therefore lead to high-fidelity results. Moreover, as LG-VSD only changes the sampling process, the modification is seamlessly grafted to the original diffusion model. Thus, we could harness the inherent refinement ability of the diffusion model to further improve the rendering quality. To ensure the semantic and geometric consistency between coarse and refined stages, we introduce a layout-aware refinement strategy to fine-tune the generated scene in practice.

Secondly, we develop Semantic-Depth Score Distillation Sampling (SD-SDS) to ensure aligned semantics and geometry with appearance. Specifically, we present a semantic 2D Gaussian splatting representation (Sem2DGS) to model 3D driving scenes. Built upon 2DGS [11], Sem2DGS further incorporates fixed semantic labels as additional attributes of Gaussian points. We train a Semantic-Depth (SD) diffusion model, which generates fine-grained semantic and depth maps based on the projected maps from the 3D layout. Then, we derive a SDS loss using the SD diffusion model as the base model (*i.e.*, SD-SDS) to constrain the semantic and depth maps rendered from the Sem2DGS representation. Since the semantic labels in Sem2DGS are fixed, the SD-SDS loss focuses solely on optimizing the geometric attributes of the Gaussian points to match their distribution. By combining SD-SDS with LG-VSD, the rendered perceptual images, semantic maps, and depth maps are encouraged to match the distribution of 3D layouts, ensuring that the geometry and semantics of the Gaussian points align with appearance.

Extensive experiments substantiate the capacity of the proposed method to generate complex 3D driving scenes with high quality, diversity, and layout consistency based on the provided 3D layouts. The robust constraints from the 3D layout prior facilitate the generation of large-scale driving scenes covering over 1000m driving distance. By leveraging the flexible and fine-grained constraints of our layout representation, along with the text-based style control, our pipeline supports various scene editing effects, including instance-level editing (*e.g.*, object manipulation) and style editing (*e.g.*, transferring city style and weather).

2 Related Works

2.1 3D Representations for Driving Scene

Various 3D representations have been proposed to model 3D scenes. Explicit representations, such as point clouds and meshes, pose challenges for optimization through purely 2D supervisions given their discrete nature, especially in large-scale scenes due to the high complexity and ill-posed

optimization process. NeRF [12] adopts implicit neural representations to encode densities and colors of the scene and exploits volumetric rendering for view synthesis, which can be effectively optimized from 2D multi-view images. Given its superior performance in view synthesis, NeRF has been used to model diverse 3D scenes. Numerous works have enhanced NeRF in terms of rendering quality [13], [14], [15], [16], efficiency [17], [18], [19], [20], *etc*. Some works have also extended NeRF to large-scale scenes [5], [21], [22], [23], [24], [25], [26]. Recently, 3D Gaussian Splatting (3DGS) [27] has demonstrated significant success in scene modeling. 3DGS represents the scene using a set of 3D Gaussian points, and develops a differentiable rasterizer to enable optimization based on multi-view images. Due to the explicit representation, 3DGS achieves efficient training and real-time rendering. Consequently, various methods have attempted to improve vanilla 3DGS [11], [28], [29], [30]. For example, 2DGS [11] achieves higher geometric quality by using 2D Gaussian points. Several approaches have been proposed to extend 3DGS to large-scale scenes. For example, VastGaussian [28] proposes to divide a scene into multiple cells to enable large scene reconstruction. We use 2DGS as the scene representation due to its high efficiency and geometric quality.

2.2 Text-to-3D

Early text-to-3D methods adopted CLIP [31] to guide the optimization but encountered difficulties in creating high-quality 3D content [32], [33], [34], [35]. Recently, large-scale diffusion models [36], [37], [38] have demonstrated significant success in text-to-image generation, capable of generating 2D images with high fidelity and diversity. They further enable fine-grained control and editing by employing additional conditioning models [10], [39]. For example, ControlNet [10] incorporates 2D pixel-aligned conditional signals to control the generation. SDEdit [39] achieves conditional image synthesis and editing using reverse SDE. The achievements in 2D content creation have propelled text-to-3D generation [7], [8], [9], [40], [41], [42], [43]. As the core of most current text-to-3D methods, Score Distillation Sampling (SDS) [7] optimizes a 3D model by aligning 2D images rendered at arbitrary viewpoints with the distribution derived from a text-conditioned diffusion model. However, SDS suffers from issues such as over-smoothness and over-saturation. VSD [9] proposes a particle-based variational framework to enhance the generation quality of SDS. Some methods also attempt to incorporate 3D information into the generation process to enhance 3D consistency [40], [41], [44]. For example, AvatarCraft [44] utilizes SMPL [45] mesh for 3D human avatar generation and animation. Nonetheless, most prior works are tailored for single object-level generation. Some methods have attempted to achieve scene-level generation, while the scene scales are quite limited [46], [47]. When applied to large-scale driving scenes, current text-to-3D methods face challenges in modeling the intricate distribution from the purely text-conditioned diffusion model. Alternative methods have explored incrementally reconstructing and inpainting the scene to create room-scale [48] or zoom-out [49] scenes. However, this paradigm is susceptible to cumulative errors and poses challenges in scalability for larger scenes.

2.3 3D Generative Models

Many 3D generative models leverage GAN [50], VAE [51], and diffusion model [52] to model the distribution of 3D representations [53], [54], [55], [56]. However, this paradigm requires a large amount of 3D data and is often limited to single-object generation. NeuralField-LDM [57] trains a scene-level 3D diffusion model by preparing abundant voxel-based neural fields, while the scales are limited to the pre-defined 3D voxel grid. Another line of work employs 3D-aware GANs to optimize a 3D model from 2D image collections. Early 3D-aware GANs are mainly designed for object-level scene generation [58], [59], [60], [61]. Recently, some methods have extended the pipeline to larger scenes including indoor scenes [62], [63], landscapes [64], and driving scenes [1], [2], [3]. For example, GSN [63] proposes a local latent grid representation for indoor scene generation. SceneDreamer [64] aims at generating landscapes, which contain repeated textures and structures. CityDreamer [4] further enhances SceneDreamer's foreground generation quality for city scene generation. More recently, GaussianCity [65] adopts Gaussian Splatting for efficient unbounded 3D city generation, while CityDreamer4D [66] further extends compositional city generation to dynamic 4D environments by jointly modeling static urban structures and dynamic traffic objects. These methods demonstrate impressive large-scale city generation capabilities, but mainly focus on learning city distributions from training data, whereas our goal is to generate steerable and explicitly editable 3D driving scenes conditioned on user-specified 3D layouts and text prompts. CC3D [1] integrates 2D layout conditions in the 3D generative model for driving scene generation. UrbanGIRAFFE [2] and InfiniCity [3] incorporate 3D voxel grids to enhance quality and scalability. Nevertheless, owing to inherent constraints such as limited training data and model capacity, they still suffer from structural distortions and the lack of details, and often overfit to the training scenarios. In parallel, diffusion-based methods such as MagicDrive [67] and DriveDreamer [68] leverage structured traffic conditions to generate controllable driving images or videos. While these methods achieve strong visual synthesis and temporal modeling, their primary outputs are images or videos rather than persistent and explicitly editable 3D scene representations.

3 METHOD

We propose Urban Architect for large-scale 3D driving scene generation. The overall pipeline is illustrated in Fig. 2. Given the 3D layout of the desired scene, we represent the scene using Semantic 2D Gaussian Splatting (Sem2DGS) (Sec. 3.2) and propose two innovative distillation strategies for scene optimization: (1) Layout-Guided Variational Score Distillation (LG-VSD), which distills a pre-trained text-to-image diffusion model by complementing textual descriptions with additional 3D layout constraints (Sec. 3.3); (2) Semantic-Depth Score Distillation Sampling (SD-SDS), which optimizes the geometric attributes of Gaussian points by distilling a Semantic-Depth diffusion model (Sec. 3.4). Then, the generated scene is refined by a layout-aware refinement strategy to enhance quality (Sec. 3.5). Finally, we demonstrate the ability of our pipeline to achieve diverse

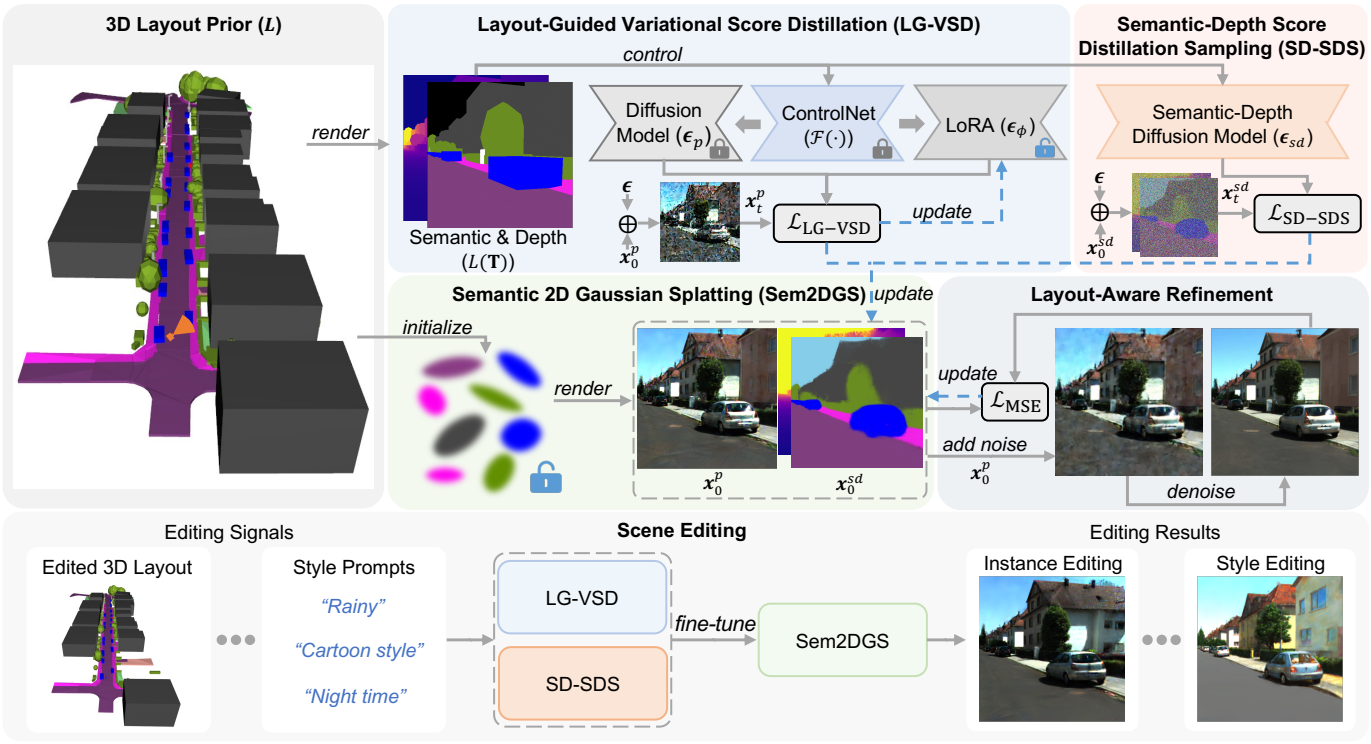


Fig. 2: **Overview of Urban Architect.** We introduce Urban Architect, a method that generates large-scale 3D driving scenes with 3D layout instruction and textual descriptions. We represent the scene using **Semantic 2D Gaussian Splatting (Sem2DGS)**, which augments 2DGS by introducing additional semantic labels. The scene is optimized via two distillation strategies: (1) **Layout-Guided Variational Score Distillation (LG-VSD)**, which controls the distilling process of VSD via the 3D layout of the desired scene, rather than relying solely on the text-based guidance; (2) **Semantic-Depth Score Distillation Sampling (SD-SDS)**, which optimizes the geometric attributes of Gaussian points, ensuring the generated 3D content exhibits alignment between semantics, geometry, and appearance. We further refine the local details via a **Layout-Aware Refinement** strategy. Moreover, we support various **Scene Editing** effects by fine-tuning the generated scene.

scene editing effects by fine-tuning the generated scene (Sec. 3.6).

3.1 Preliminaries

Score Distillation Sampling (SDS). SDS achieves text-to-3D generation by distilling a pre-trained text-to-image diffusion model $\boldsymbol{\epsilon}_p$ to optimize a differentiable 3D representation parameterized by θ. SDS encourages the image $\boldsymbol{x}_0^p = g(\theta, \mathbf{T})$ rendered at an arbitrary camera view $\mathbf{T}$ to match the distribution $p_0(\boldsymbol{x}_0^p|y)$ derived from the diffusion model conditioned on the text prompt y. In practice, SDS optimizes the 3D model by adding random noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to $\boldsymbol{x}_0^p$ and calculating the difference between the predicted noise from the diffusion model $\boldsymbol{\epsilon}_p$ and the added noise $\boldsymbol{\epsilon}$. The gradient of SDS loss can be formulated as:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{T}}\left[\omega(t)(\boldsymbol{\epsilon}_p(\boldsymbol{x}_t^p, t, y) - \boldsymbol{\epsilon})\frac{\partial \boldsymbol{g}(\theta, \mathbf{T})}{\partial \theta}\right], \quad (1)$$

where $t \sim \mathcal{U}(0.02, 0.98)$, $\omega(t)$ weights the loss given the time step t, and $\boldsymbol{x}_t^p$ is the randomly perturbed rendered image. Although SDS can produce reasonable 3D content aligned with the text prompt, it exhibits issues such as over-saturation, over-smoothing, and low diversity.

Variational Score Distillation (VSD). To address the limitations of SDS, instead of optimizing a single sample point, VSD proposes to optimize a distribution $q_0^\mu(\boldsymbol{x}_0^p|y)$ of possible 3D representations with $\mu(\theta|y)$ corresponding to the text prompt y. VSD adopts a set of 3D parameters $\{\theta\}_{i=1}^n$ as particles to represent the scene distribution and optimizes the 3D model by matching the score of noisy real images and that of noisy rendered images at each time step t. In practice, the score function of noisy rendered images is estimated by optimizing a low-rank adaptation (LoRA) model $\boldsymbol{\epsilon}_\phi$ [69] of the pre-trained diffusion model $\boldsymbol{\epsilon}_p$. The gradient of VSD loss can be formulated as:

$$\nabla_\theta \mathcal{L}_{\text{VSD}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{T}}\left[\omega(t)(\boldsymbol{\epsilon}_p(\boldsymbol{x}_t^p, t, y) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t^p, t, \mathbf{T}, y))\frac{\partial \boldsymbol{g}(\theta, \mathbf{T})}{\partial \theta}\right]. \quad (2)$$

VSD proves to be effective in alleviating issues of SDS such as over-saturation and over-smoothing, and successfully generates high-fidelity and diverse 3D content.

3.2 Semantic 2D Gaussian Splatting

Sem2DGS is an extension of vanilla 2DGS. Each Gaussian point in 2DGS is parameterized via its location $\boldsymbol{p}$, 2D covariance matrix $\boldsymbol{\Sigma}$, opacity α and view-dependent appearance $\boldsymbol{c}$. We introduce two additional attributes, *i.e.*, semantic label $\boldsymbol{s}$ and instance label i, which are fixed during the training process. The semantic labels facilitate the distillation process in SD-SDS (Sec. 3.4), while the instance labels enable scene

editing and subdivision, as discussed below. To initialize Sem2DGS, we uniformly sample points on the surface of the 3D layout, with semantic and instance labels trivially assigned. The sky is represented by sampling Gaussian points on a large sphere.

As previously discussed, our objective is to generate large-scale driving scenes, which necessitate a substantial number of Gaussian points for accurate representation. Consequently, this demands significant GPU memory to optimize these points simultaneously. Inspired by VastGaussian [28], we introduce a simple scene subdivision strategy to reduce memory consumption. Specifically, our layout-based representation inherently facilitates instance-level scene division. During optimization, we render an instance map from the 3D layout and retain only Gaussian points with matching instance labels for rendering and refinement. Additionally, to alleviate frequent GPU memory queries, we generate N_c nearby camera poses in a sub-training process and maintain a consistent pool of Gaussian points throughout this process.

3.3 Layout-Guided Variational Score Distillation

VSD relies on text prompts to guide model optimization. However, given the inherent ambiguity of text prompts y, it is challenging for VSD to estimate the distribution of a complex driving scene from a diverse target distribution $p_0(\boldsymbol{x}_0^p|y)$. Therefore, introducing additional constraints to obtain a more compact distribution is crucial for high-quality 3D driving scene generation. A straightforward approach is to simply fine-tune the diffusion model to constrain the distribution to the desired driving scene. However, this operation can only provide style constraints and VSD still struggles to capture the intricate geometric and semantic distribution of driving scenes, leading to unsatisfactory results.

Given the above observations, we introduce 3D layout L as a prior to provide additional constraints to the target distribution with a formula shift (*i.e.*, $p_0(\boldsymbol{x}_0^p|y) \rightarrow p_0(\boldsymbol{x}_0^p|y, L)$). Nonetheless, the direct integration of the 3D information into the 2D framework poses a challenge. We observe that the projected semantic and depth maps provide a comprehensive description of the 3D layout in 2D space. They offer both semantic and geometric constraints of the scene, and inherently exhibit multi-view consistency. Thus, we propose to condition the distilling process via the 2D semantic and depth maps rendered at the given camera pose $\mathbf{T}$. Specifically, we first train a ControlNet [10] $\mathcal{F}$ by utilizing the 2D semantic and depth maps rendered from 3D layouts as conditions, along with the corresponding 2D images as ground truths.

At each training step of VSD, we render the 2D signals from the conditional 3D layout L at a randomly sampled camera view $\mathbf{T}$. Subsequently, the features produced by $\mathcal{F}$ are integrated into both the diffusion model $\boldsymbol{\epsilon}_p$ and the LoRA model $\boldsymbol{\epsilon}_\phi$, leading to a compact, scene-specific target distribution $p_0(\boldsymbol{x}_0^p|y, \mathcal{F}(L(\mathbf{T})))$. Formally, the gradient of our layout-guided VSD (LG-VSD) loss can be written as:

$$\nabla_\theta \mathcal{L}_{\text{LG-VSD}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{T}}\Big[\omega(t)(\boldsymbol{\epsilon}_p(\boldsymbol{x}_t^p, t, y, \mathcal{F}(L(\mathbf{T}))) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t^p, t, \mathbf{T}, y, \mathcal{F}(L(\mathbf{T}))))\frac{\partial \boldsymbol{g}(\theta, \mathbf{T})}{\partial \theta}\Big]. \tag{3}$$

By maximizing the likelihood of the rendered 2D images in the layout-conditioned diffusion model at arbitrary camera views, LG-VSD gradually optimizes the 3D representation θ to align with the semantic and geometric distribution of the desired scene layout.

3.4 Semantic-Depth Score Distillation Sampling

Although LG-VSD provides fine-grained conditional signals for high-quality content generation, it relies on 2D rendered perceptual images for model optimization. However, in driving scenes, instances have complex spatial relationships with severe occlusions. With limited camera viewpoints, purely perceptual-based optimization can produce suboptimal results by tuning the appearance only, while the semantics and geometry are misaligned. For instance, perceptual-based optimization might generate a flat, planar car image that appears pasted onto the background.

We introduce SD-SDS to ensure consistent alignment between semantics, geometry, and appearance in the generated 3D content. To achieve this, we leverage our 3D layout representation as a natural bridge. Building on LG-VSD, the appearance of the 3D content is aligned with the 3D layout. Similarly, we enforce the semantics and geometry of the 3D content to match the 3D layout. Specifically, we train a conditional diffusion model $\boldsymbol{\epsilon}_{sd}$, where the projected coarse semantic and depth maps derived from 3D layouts serve as conditional inputs, and the fine-grained semantic and depth maps are the generation targets. Based on $\boldsymbol{\epsilon}_{sd}$, we derive SD-SDS:

$$\nabla_\theta \mathcal{L}_{\text{SD-SDS}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{T}}\Big[\omega(t)(\boldsymbol{\epsilon}_{sd}(\boldsymbol{x}_t^{sd}, t, \mathcal{F}(L(\mathbf{T}))) - \boldsymbol{\epsilon})\frac{\partial \boldsymbol{g}(\theta, \mathbf{T})}{\partial \theta}\Big], \tag{4}$$

where $\boldsymbol{x}_t^{sd}$ is perturbed from the rendered semantic and depth maps $\boldsymbol{x}_0^{sd}$ from 2DGS. Since the semantic labels of the Gaussian points are fixed, SD-SDS optimizes the geometric attributes of the Gaussian points to match the 3D layout, ensuring that the geometry and semantics of the 3D content align with the appearance generated through LG-VSD.

3.5 Layout-Aware Refinement

Inspired by SDEdit [39], we observe that the resampling process of diffusion models can effectively refine the rendered images. Specifically, given a random camera pose, the rendered image $\mathbf{I}_r$ is perturbed to $\mathbf{I}_p$ with random noise given the time step t. Then $\mathbf{I}_p$ will be denoised to the refined image $\mathbf{I}_f$ using the diffusion model. When $t \rightarrow 0$, the resampling process tends to refine local details of the image $\mathbf{I}_r$ while maintaining its overall structure. Still and all, the geometric and semantic consistencies in long-horizon trajectory cannot be guaranteed, as such a resampling process is agnostic to contextual 3D structures among frames. To this end, we further enable layout-aware refinement by conditioning the denoising steps using the rendered semantic and depth maps, leading to better consistency. Specifically, we first render a set of images from the generated scene. These images are then refined through a process of perturbation and denoising, as described earlier. Then, we follow the standard training process in 2DGS [11], using the refined images as

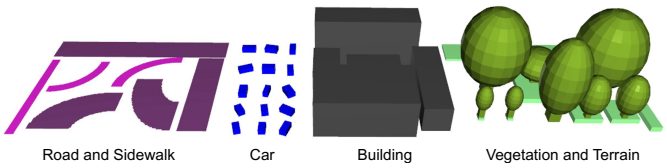


Fig. 3: **Basic primitives.** We provide several basic primitives of common objects in driving scenes (*e.g.*, road and sidewalk, car, building, *etc.*)

ground truth and updating the scene with the MSE loss between $\mathbf{I}_r$ and $\mathbf{I}_f$, *i.e.*, $\mathcal{L}_{\text{MSE}} = ||\mathbf{I}_r - \mathbf{I}_f||_2^2$. To further improve the scene geometry, we estimate monocular depth from each refined image using Depth Anything V2 [70], align it with the rendered depth to resolve scale and shift ambiguity, and additionally supervise the rendered depth and surface normals during refinement.

3.6 Scene Editing

Instance-Level Editing. The compositional 3D layout representation naturally supports instance-level scene editing. For background objects (*e.g.*, buildings, trees, *etc*), we can delete or insert instances in the scene layout as well as the Gaussian points and then fine-tune the generated scene using our LG-VSD and SD-SDS for possible missing region completion. For foreground objects (*i.e.*, cars), the manipulation (*e.g.*, rotation, translation, *etc*) can be applied simultaneously on the instance layout and the Gaussian points. For large transformations, we further apply layout-aware refinement to improve the consistency of the edited results.
Style Editing. Benefiting from the inherent capacity of the large-scale diffusion model, we can easily transfer the generated scene to various styles. Specifically, given the generated scene, we simply fine-tune it via our LG-VSD and SD-SDS, while adding a style text prompt in the distilling process of LG-VSD, *e.g.*, *"Cartoon style"*, *"Night time"*, *etc*. For some large style changes, we additionally reduce the ControlNet conditioning strength to increase the flexibility of text-based appearance control.

3.7 Layout Construction

Our layout representation is particularly suitable for outdoor driving scenes, where large-scale environments can be described by a relatively sparse set of recurring semantic elements, such as roads, vehicles, buildings, and vegetation. The coarse layout mainly captures their spatial organization and semantic composition, while fine-grained geometry and appearance are complemented by the pretrained diffusion prior. 3D layout construction is not the main focus of this work. In practice, we use in-hand layout data provided by the KITTI-360 dataset to train the ControlNet. Nonetheless, there are also several alternatives. For example, users can easily create the desired layout in 3D modeling software (*e.g.*, Blender) given the simple geometric structures of basic primitives. As shown in Fig. 3, we provide several basic primitives in driving scenes to ease the process.

We further provide an alternative method to generate 3D scene layout automatically based on SinDDM [71]. SinDDM learns the intrinsic distribution of the training image via a multi-scale diffusion process, enabling the generation of new images with arbitrary scales. Specifically, we first compress the example 3D layout to the 2D ground plane and obtain a 2D representation of the 3D scene. Then, we train the SinDDM model based on the single 2D example. The generated layout is then transformed into 3D space by clustering instances and assigning random heights empirically. As shown in Fig. 4, the trained model can produce diverse new samples with arbitrary scales and reasonable arrangements. We also generate scenes based on the generated 3D layouts and the rendered results are provided in the bottom row of Fig. 4.

4 Experiments

4.1 Experimental Setup

Dataset. We mainly perform experiments on the KITTI-360 dataset [72] for quantitative comparisons with baseline methods. KITTI-360 dataset is captured in urban scenarios covering a driving distance of around 73.7km and offers 3D bounding box annotations for various classes (*e.g.*, building, car, road, vegetation, *etc*), forming extensive 3D scene layouts. Each layout instance is modeled using a triangle mesh, enabling fast rendering of semantic and depth maps.
Implementation Details. We use Stable Diffusion 2.1 [36] as the text-to-image diffusion model. The overall framework is implemented using PyTorch [73]. We use diffusers [74] to implement the diffusion model. For the training of ControlNet [10], we concatenate the rendered semantic and depth maps from 3D layouts as the input conditional signal. We crop and resize the original image (with a resolution of 1408×376) in the KITTI-360 dataset [72] to a resolution of 512×512. The training of a single scene is conducted on an NVIDIA L40 GPU. We train and refine the scene at a resolution of 512×512.
Baselines and Settings. We compare the proposed method with several 3D generative methods applied to driving scenes, including EG3D [60], CC3D [1], UrbanGIRAFFE [2], Text2Room [48], SceneDreamer [64], GANCraft [75] and InfiniCity [3]. We use the officially released code and pretrained weights provided by the authors of CC3D for the evaluation of CC3D and EG3D. Due to the different settings of UrbanGIRAFFE, we re-train the model using its official released code. Text2Room is initially designed for room-scale textured mesh generation. We adapt it to our scenes by integrating our pre-trained ControlNet into the inpainting model of Text2Room, enabling the generation of driving scenes based on our 3D layout information. SceneDreamer aims to generate 3D landscapes and we apply it to driving scenes using the projected height and semantic field from our layout representation as the input. We then re-train both SceneDreamer and the required SPADE [76] model for driving scene generation. GANCraft also requires a pre-trained SPADE model, and we use the re-trained version for scene generation. For quantitative evaluation, we use Fréchet Inception Distance (FID) [77] and Kernel Inception Distance (KID) [78] as evaluation metrics, which are computed with 5000 randomly sampled images. Quantitative comparisons with InfiniCity are excluded due to the different settings and the incomplete released code.

TABLE 1: Quantitative comparison on the KITTI-360 dataset.

Method	Text2Room [48]	GANCraft [75]	UrbanGIRAFFE [2]	SceneDreamer [64]	EG3D [60]	CC3D [1]	Ours
FID↓	134.3	149.7	118.6	122.4	109.3	79.1	**52.1**
KID↓	0.116	0.161	0.143	0.113	0.121	0.082	**0.041**

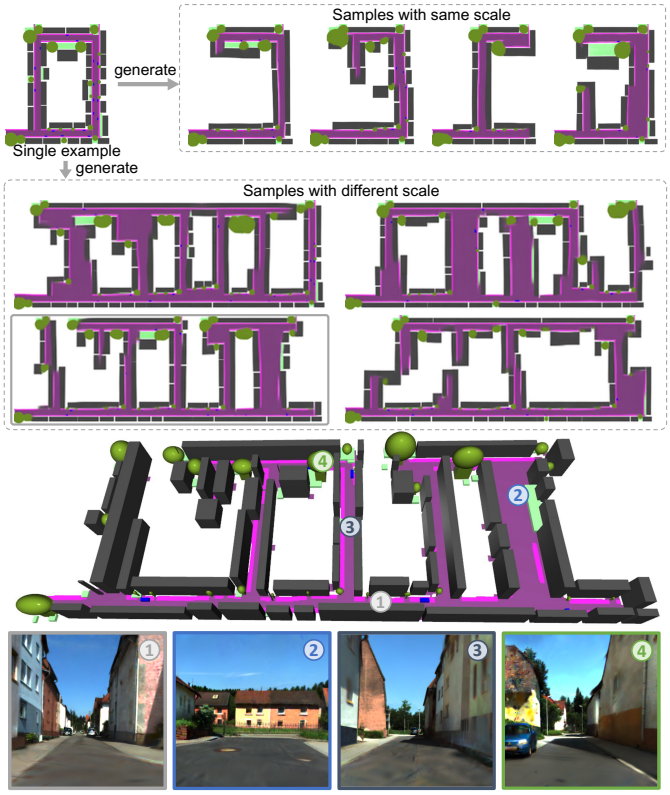


Fig. 4: **Automatic Layout Generation.** We present an alternative method for automatic 3D layout generation given a single example of the layout. In the top two rows, we display generated layouts with different scales and provide the corresponding 3D layout given the generated 2D sample in the third row. The rendering results of the generated scene are displayed in the bottom row.

TABLE 2: Quantitative comparison with CC3D+H.

Method	CC3D [1]	CC3D+H [1]	Ours
FID↓	79.1	77.8	**52.1**
KID↓	0.082	0.077	**0.041**

Results. According to the quantitative results in Table 1, the proposed method outperforms previous methods in terms of both FID and KID by a significant margin. For example, the FID of our method is about 34% lower than that of the most competitive baseline method, CC3D. We further provide visualization results of generated samples in Fig. 5 for qualitative comparison. According to the results, Text2Room tends to degrade rapidly under the influence of cumulative errors, resulting in a failure to produce valid results. Besides, 3D-aware GAN-based methods (*i.e.*, SceneDreamer, UrbanGIRAFFE, EG3D, CC3D, GANCraft and InfiniCity) suffer from structural distortions and lack fine-grained details, due to the limited model capacity and the complexity of driving scenes. In contrast, our method is capable of generating high-quality 3D driving scenes with rich details, demonstrating a notable advancement over previous methods. To further investigate the effect of our 3D layout prior, we add height fields of 3D layouts as an extra condition to CC3D, resulting in an enhanced version termed CC3D+H. As shown in Table 2 and Fig. 6, CC3D+H marginally outperforms CC3D, indicating that our 3D layout prior is beneficial for driving scene generation. However, it remains significantly inferior to our method, which demonstrates the effectiveness and necessity of our carefully designed pipeline to fully exploit the potential of 3D layout priors and large-scale text-to-image diffusion models. Please refer to our project page for video results.

TABLE 3: Camera-motion fidelity evaluation. Camera poses are estimated from generated image sequences using Pi3X.

Method	RPE		ATE
	Trans. (m)↓	Rot. (°)↓	RMSE (m)↓
CC3D	0.65	0.53	2.71
Ours w/o LG-VSD	0.22	0.31	1.94
Ours	**0.17**	**0.22**	**1.01**

User Study. We further conduct a user study with 20 volunteers, who rate the rendered videos of each method along three dimensions: *perceptual quality*, *realism*, and *3D consistency*. Perceptual quality evaluates local visual fidelity, including sharpness, texture quality, and visible artifacts. Realism measures the overall plausibility of the generated urban scene, considering scene composition, urban structure, and object appearance. 3D consistency evaluates the stability of scene structures and object geometry across viewpoint changes in the rendered videos. Each criterion is rated from 1 to 5, with 5 indicating the best quality. The results in Fig. 7 indicate users' preference for our results across all aspects.

Camera-Motion Fidelity. To evaluate multi-view geometric consistency, we introduce a camera-motion fidelity metric based on Pi3X [79]. We render image sequences along predefined trajectories, estimate camera poses with Pi3X, align the recovered trajectories with the inputs using Umeyama alignment, and report ATE and RPE [80]. As shown in Table 3, our method achieves the lowest errors, with an RPE translation error of 0.17 m, rotation error of 0.22°, and an ATE RMSE of 1.01 m, outperforming CC3D and the variant without LG-VSD and indicating better multi-view geometric consistency.

Large-Scale Generation Capability. Notably, the scales of scenes generated by most prior 3D-aware GAN-based methods are limited (typically less than 100m). SceneDreamer can generate scenes with larger scales, while the resolution of a local scene window is limited to 2048×2048 with 40GB GPU memory, which corresponds to a spatial



Fig. 5: **Qualitative comparison.** We display two scenes generated by different methods for comparison. Most of the results are directly borrowed from the original paper, except for Text2Room, GANCraft and SceneDreamer, which were adapted and re-trained within our settings. The proposed pipeline significantly outperforms previous baseline methods and achieves high-quality rendering. Best zoom in for more details.



Fig. 6: **Qualitative comparison with CC3D+H.** Our method significantly outperforms CC3D+H, which is an enhanced version of CC3D.

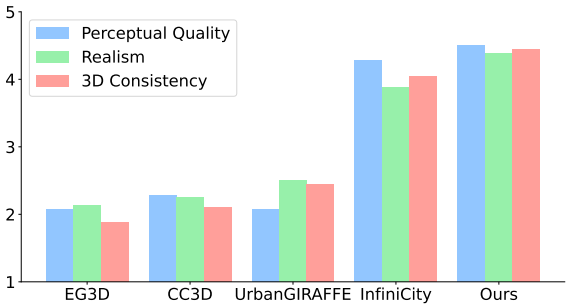


Fig. 7: **User study.** The score ranges from 1 to 5, with 5 denoting the best quality.

coverage of $\sim 400 \times 400\text{m}^2$ at a voxel resolution of 0.2m. Besides, the generated results also lack fine-grained details. In contrast, our method can achieve driving scene generation with longer driving distances and higher quality by a large margin. Fig. 8 shows our generation results in two regions with 40GB GPU memory, each covering an area of $\sim 600 \times 400\text{m}^2$ and spanning a driving distance of over 1000m. Even in such large-scale scenes, our method still achieves high-quality rendering results, showcasing the superior scalability of our pipeline in arbitrary-scale driving scene generation.

Runtime Analysis. We further evaluate the generation time under different driving distances. As shown in Fig. 11, the runtime increases approximately linearly with scene extent, benefiting from our scene subdivision strategy that optimizes local regions independently. Nevertheless, our method remains considerably slower than feed-forward baselines: generating a scene of approximately 100 m takes about 2.5 hours on an NVIDIA L40 GPU, while CC3D requires less than 10 seconds. We regard this efficiency gap as a limitation of the current optimization-based framework.

4.2 Scene Editing

Instance-Level Editing. As shown in Fig. 9 (a) (b), we showcase two kinds of instance-level editing: *(a) Foreground object manipulation.* We achieve object rotation, repeat, and scaling by simply manipulating the object in the layout. *(b) Background object editing.* We can further edit background elements in the scene (*e.g.*, building removing, tree adding) by editing the layouts while keeping the other components unchanged.

Style Editing. We present several style editing samples corresponding to the same generated scene in Fig. 9 (c). The results show that we can transfer the style of the generated scene via style text prompts (*e.g.*, *"Foggy"*, *"Vangogh paint"*, *etc*), while preserving the overall structure. Notably, we can also generate driving scenes of various city styles by simply adding text prompt-based conditions (*e.g.*, *"in Kyoto"*, *"in Hawaii"*, *etc*).

Transfer to Other Scenes. Although the conditioning model is initially trained on the KITTI-360 dataset, we can transfer our method to generate scenes of other driving datasets by fine-tuning the ControlNet. As shown in Fig. 12, we generate scenes on the NuScenes [81] dataset and further transfer them into a different style (*i.e.*, *"in winter, with snow"*).

TABLE 4: Ablation studies on different strategies.

Method	FID↓	KID↓
w/o LG-VSD	113.0	0.126
w/o SD-SDS	60.5	0.049
w/o refinement	57.1	0.048
Ours	**52.1**	**0.041**

4.3 Ablation Studies

Effectiveness of LG-VSD. To verify the effectiveness of LG-VSD, we conduct experiments by using the vanilla VSD, where the noise prediction network of Stable Diffusion is fine-tuned on the KITTI-360 dataset to constrain the style distribution. However, in the absence of semantic and geometric constraints from the 3D layout, VSD fails to capture the complex distribution of driving scenes. As shown in the top row of Fig. 13, the generated scene appears blurry and lacks detail in the absence of LG-VSD. Additionally, the quantitative comparison in Table 4 demonstrates that LG-VSD significantly outperforms VSD, achieving much lower FID and KID.

Effectiveness of SD-SDS. As previously discussed, we introduce SD-SDS to ensure alignment between semantics, geometry, and appearance. To evaluate its effectiveness, we remove SD-SDS (i.e., generating the scene using only LG-VSD and refinement). Visualization comparisons of the semantic map, depth map, and normal map are presented in Fig. 14, where the semantic map is rendered based on the semantic labels of the Gaussian points. The results show that, although the perceptual image without SD-SDS appears plausible, the geometry and semantics are misaligned and disordered. As highlighted in the zoomed-in patch, the car's geometry is inaccurate, while the perceptual results are rendered from the background road and trees rather than the car itself. In addition to the qualitative comparison, we provide quantitative results in Table 4. The results demonstrate that SD-SDS also enhances perceptual quality, reducing FID by 14%. This improvement is attributed to

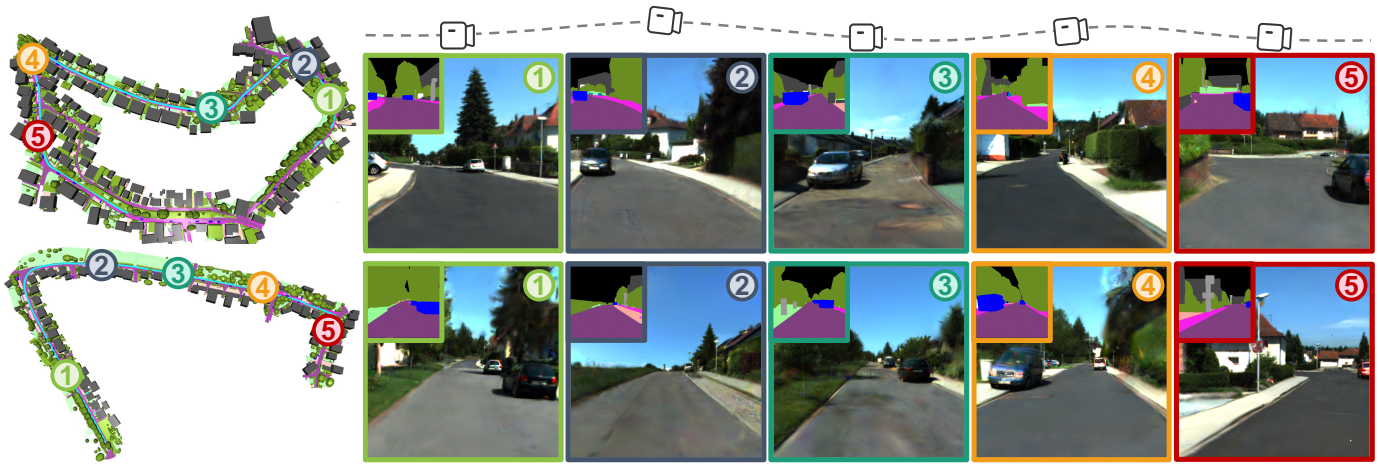


Fig. 8: **Large-scale generation capability.** We generate two large-scale 3D driving scenes with high quality, each covering an area of $\sim 600 \times 400\text{m}^2$ and spanning a driving distance of over 1000m.

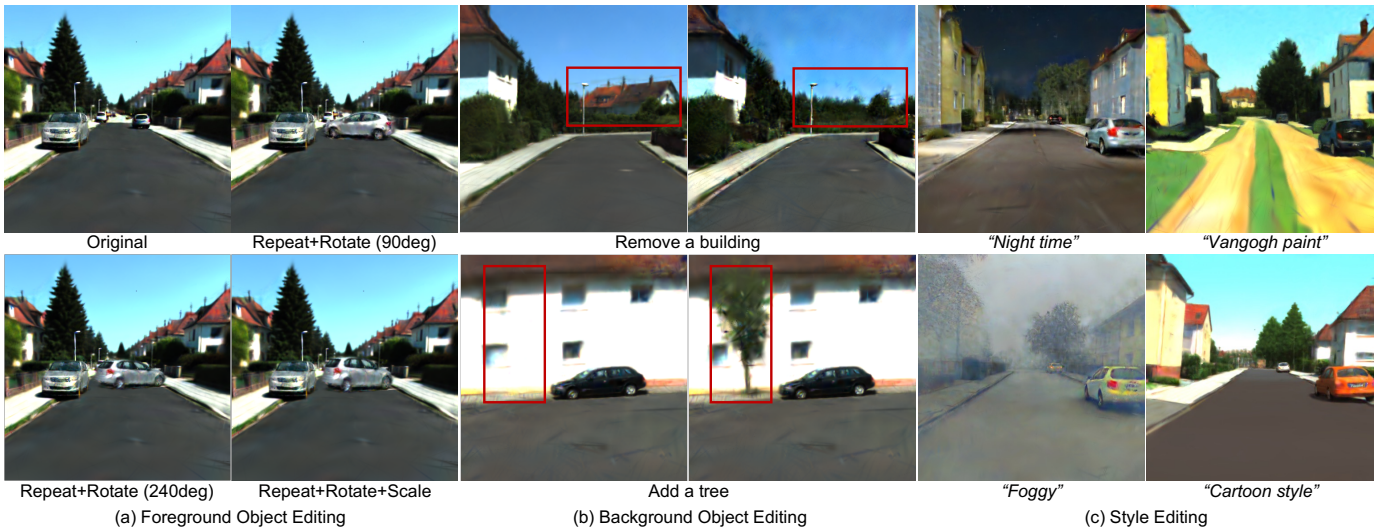


Fig. 9: **Samples of scene editing results.** The proposed pipeline achieves instance-level editing (including foreground and background object editing) and global style editing.

the fact that misaligned geometry and semantics introduce additional artifacts.

Effectiveness of Layout-Aware Refinement. We perform experiments to analyze the effectiveness of the layout-aware refinement strategy. As shown in Fig. 15, layout-aware refinement effectively reduces high-frequency artifacts and enhances the realism of the generated results. We also provide quantitative comparisons in Table 4. The results show that the introduction of layout-aware refinement improves both FID and KID, further demonstrating the effectiveness.

4.4 Further Analysis

Large Camera View Shifting. As shown in Fig. 16, we display the rendering results under three different camera view shifting settings: (a) rotating the camera right by $30\,\text{deg}$ and $60\,\text{deg}$; (b) translating the camera up by 1m and 2m; (c) close-range large-angle view shifting while keeping nearby scene elements visible. Owing to the high 3D consistency, the generated scene demonstrates strong robustness against large camera view shifting.

Diversity. To explore the diversity of generated results, we conduct experiments by employing different random seeds while maintaining the same layout. As shown in Fig. 17, given the same scene layout, the proposed pipeline can generate diverse scenes with elements (*e.g.*, cars, buildings) that have different appearances.

3D Visualization. To further explore the 3D consistency of the generated scene, we extract a point cloud from the generated scene and display the results in Fig. 18. The 3D point cloud reveals the consistent 3D structures of the generated scene. We also present the rendered normal maps along with the images to demonstrate 3D consistency.

Generalization to VoxCity. We further evaluate Urban Architect on VoxCity [82] to examine its generalization beyond KITTI-360. We convert the voxelized city representation into our semantic instance-level layout and construct street-level trajectories from the road network. As shown in Fig. 19, our framework can successfully generate 3D scenes from VoxCity-derived layouts, demonstrating its compatibility with alternative structured urban representations.



Fig. 10: **Transfer to other cities.** The generated driving scene can be transferred to different city styles given the text prompt by fine-tuning the generated scene.

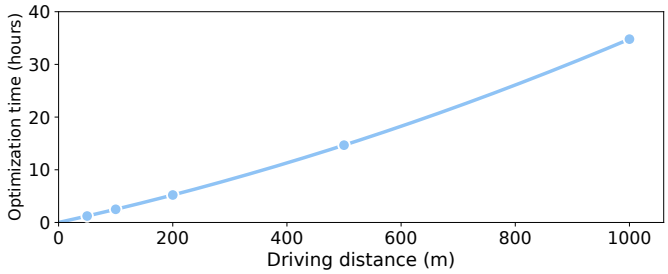


Fig. 11: **Runtime analysis under different driving distances.** The generation time scales approximately linearly with the scene extent, benefiting from our scene subdivision strategy.

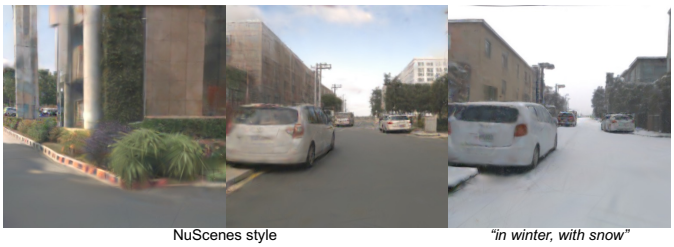


Fig. 12: **Transfer to other urban styles.** We generate scenes in NuScenes style and transfer them to a different weather.



Fig. 13: **Effectiveness of LG-VSD.** Without our 3D layout prior, VSD fails to generate high-quality driving scenes due to the absence of effective guidance.

5 Conclusion

We have presented Urban Architect, a method for steerable 3D driving scene generation. The core of our method lies in incorporating a 3D layout representation as a robust prior to complement textual descriptions. Based on the 3D layout, we introduce Layout-Guided Variational Score Distillation (LG-VSD) to integrate the 3D geometric and semantic constraints into the current text-to-3D paradigm, which is achieved by conditioning the score distillation sampling process with 3D layout information. Moreover, we further design Semantic-Depth Score Distillation Sampling (SD-SDS), which ensures the generated 3D content exhibits consistent alignment between semantics, geometry, and appearance. Our framework facilitates high-quality, steerable 3D driving scene generation, capable of scaling up to generate large-scale scenes covering a driving distance

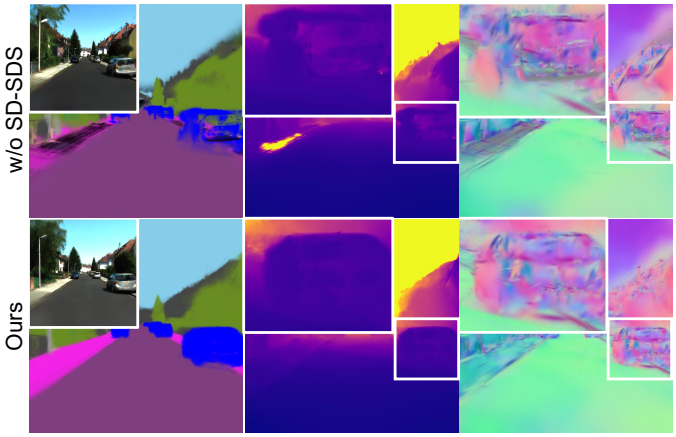


Fig. 14: **Effectiveness of SD-SDS.** Our method generates reasonable perceptual results w/o SD-SDS, while the geometry and semantics are misaligned and disordered.



Fig. 15: **Ablations on layout-aware refinement.** The layout-aware refinement strategy further enhances the generation quality, leading to more realistic results.

of over 1000m. Our flexible representation, coupled with the inherent capability of text-to-image diffusion models, empowers our pipeline to support diverse scene editing effects, including instance-level editing and style editing. Despite these capabilities, the current framework still relies on computationally expensive scene-specific optimization and does not provide pixel-level control or high-precision geometric reconstruction. In addition, while individual objects can be explicitly manipulated through their layouts, realistic traffic dynamics would require integration with dedicated traffic or trajectory-generation models. Extending the framework toward more efficient optimization, dynamic driving scenes, and broader scene domains such as indoor environments constitutes interesting directions for future work.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 62506264), National Key Research and Development Program of China (No. 2024YFE0211000), in part by the National Natural Science Foundation of China (No. 62372329, 62506263), in part by the Shanghai Scientific Innovation Foundation (No. 23DZ1203400), in part by the China Postdoctoral Science Foundation (No. BX20250383, GZB20250385, 2025M771530, 2025M771539), in part by the Fundamental Research Funds for the Central Universities, in part by Tongji-Qomolo Autonomous Driving Commercial Vehicle Joint Lab Project, and in part by Xiaomi Young Talents Program.

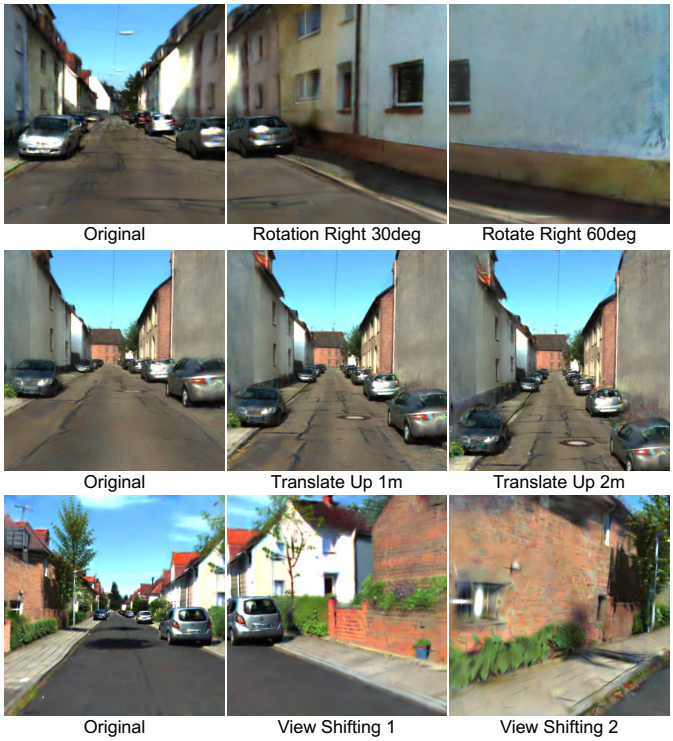


Fig. 16: **Large camera view shifting.** We evaluate three camera view shifting settings: (a) The camera rotates to the right by 30 deg and 60 deg. (b) The camera moves upward by 1m and 2m. (c) A close-range large-angle view shift while keeping nearby scene elements visible.

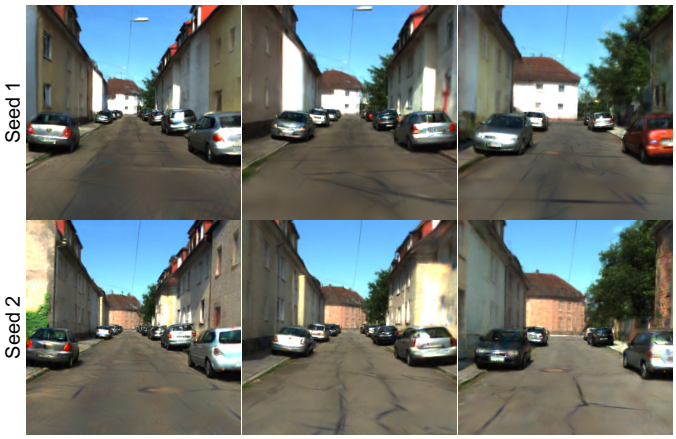


Fig. 17: **Generated results with different random seeds.** We display the generated scenes with two different random seeds given the same scene layout.

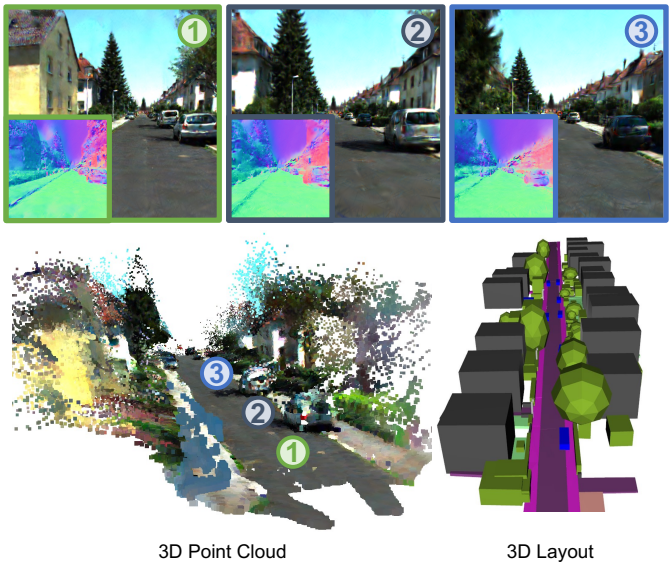


Fig. 18: **3D point cloud visualization.** We extract a 3D point cloud from the generated scene and provide rendered 2D images and normal maps from corresponding camera poses.



Fig. 19: **Generalization to VoxCity.** A representative scene generated by Urban Architect on VoxCity.

REFERENCES

[1] S. Bahmani, J. J. Park, D. Paschalidou, X. Yan, G. Wetzstein, L. Guibas, and A. Tagliasacchi, "Cc3d: Layout-conditioned generation of compositional 3d scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 7171–7181.

[2] Y. Yang, Y. Yang, H. Guo, R. Xiong, Y. Wang, and Y. Liao, "Urbangiraffe: Representing urban scenes as compositional generative neural feature fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 9199–9210.

[3] C. H. Lin, H.-Y. Lee, W. Menapace, M. Chai, A. Siarohin, M.-H. Yang, and S. Tulyakov, "Infinicity: Infinite-scale city synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 22 808–22 818.

[4] H. Xie, Z. Chen, F. Hong, and Z. Liu, "Citydreamer: Compositional generative model of unbounded 3d cities," *arXiv preprint arXiv:2309.00610*, 2023.

[5] F. Lu, Y. Xu, G. Chen, H. Li, K.-Y. Lin, and C. Jiang, "Urban radiance field representation with deformable neural mesh primitives," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 465–476.

[6] J. Y. Liu, Y. Chen, Z. Yang, J. Wang, S. Manivasagam, and R. Urtasun, "Real-time neural rasterization for large scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8416–8427.

[7] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *International Conference on Learning Representations*, 2023.

[8] 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.

[9] 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," *arXiv preprint arXiv:2305.16213*, 2023.

[10] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 3836–3847.

[11] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH 2024 conference papers*, 2024, pp. 1–11.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[13] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[14] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[15] ——, "Zip-nerf: Anti-aliased grid-based neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 19 697–19 705.

[16] W. Hu, Y. Wang, L. Ma, B. Yang, L. Gao, X. Liu, and Y. Ma, "Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 774–19 783.

[17] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[18] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.

[19] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[20] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[21] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[22] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.

[23] L. Xu, Y. Xiangli, S. Peng, X. Pan, N. Zhao, C. Theobalt, B. Dai, and D. Lin, "Grid-guided neural radiance fields for large urban scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8296–8306.

[24] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *European conference on computer vision*. Springer, 2022, pp. 106–122.

[25] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 922–12 931.

[26] C. Reiser, S. Peng, Y. Liao, and A. Geiger, "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 335–14 345.

[27] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (ToG)*, vol. 42, no. 4, pp. 1–14, 2023.

[28] J. Lin, Z. Li, X. Tang, J. Liu, S. Liu, J. Liu, Y. Lu, X. Wu, S. Xu, Y. Yan *et al.*, "Vastgaussian: Vast 3d gaussians for large scene reconstruc-

tion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5166–5175.
[29] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang, "Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 634–21 643.
[30] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-splatting: Alias-free 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 447–19 456.
[31] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
[32] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 867–876.
[33] A. Sanghi, H. Chu, J. G. Lambourne, Y. Wang, C.-Y. Cheng, M. Fumero, and K. R. Malekshan, "Clip-forge: Towards zero-shot text-to-shape generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 603–18 613.
[34] N. Mohammad Khalid, T. Xie, E. Belilovsky, and T. Popa, "Clip-mesh: Generating textured meshes from text using pretrained image-text models," in *SIGGRAPH Asia 2022 conference papers*, 2022, pp. 1–8.
[35] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, "Avatarclip: zero-shot text-driven generation and animation of 3d avatars," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–19, 2022.
[36] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
[37] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro *et al.*, "ediffi: Text-to-image diffusion models with an ensemble of expert denoisers," *arXiv preprint arXiv:2211.01324*, 2022.
[38] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.
[39] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," in *International Conference on Learning Representations*, 2021.
[40] 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 (ICCV)*, October 2023, pp. 22 246–22 256.
[41] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or, "Latent-nerf for shape-guided generation of 3d shapes and textures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 663–12 673.
[42] Y. Huang, J. Wang, Y. Shi, X. Qi, Z.-J. Zha, and L. Zhang, "Dreamtime: An improved optimization strategy for text-to-3d content creation," *arXiv preprint arXiv:2306.12422*, 2023.
[43] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," *arXiv preprint arXiv:2309.16653*, 2023.
[44] R. Jiang, C. Wang, J. Zhang, M. Chai, M. He, D. Chen, and J. Liao, "Avatarcraft: Transforming text into neural human avatars with parameterized shape and pose control," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 14 371–14 382.
[45] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.
[46] Y. Lin, H. Bai, S. Li, H. Lu, X. Lin, H. Xiong, and L. Wang, "Componerf: Text-guided multi-object compositional nerf with editable 3d scene layout," *arXiv preprint arXiv:2303.13843*, 2023.
[47] D. Cohen-Bar, E. Richardson, G. Metzer, R. Giryes, and D. Cohen-Or, "Set-the-scene: Global-local training for generating controllable nerf scenes," *arXiv preprint arXiv:2303.13450*, 2023.
[48] L. Höllein, A. Cao, A. Owens, J. Johnson, and M. Nießner, "Text2room: Extracting textured 3d meshes from 2d text-to-image models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 7909–7920.
[49] R. Fridman, A. Abecasis, Y. Kasten, and T. Dekel, "Scenescape: Text-driven consistent scene generation," *arXiv preprint arXiv:2302.01133*, 2023.
[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
[51] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[52] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
[53] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," *Advances in neural information processing systems*, vol. 29, 2016.
[54] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *International conference on machine learning*. PMLR, 2018, pp. 40–49.
[55] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, "Lion: Latent point diffusion models for 3d shape generation," *arXiv preprint arXiv:2210.06978*, 2022.
[56] T. Wang, B. Zhang, T. Zhang, S. Gu, J. Bao, T. Baltrusaitis, J. Shen, D. Chen, F. Wen, Q. Chen *et al.*, "Rodin: A generative model for sculpting 3d digital avatars using diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4563–4573.
[57] S. W. Kim, B. Brown, K. Yin, K. Kreis, K. Schwarz, D. Li, R. Rombach, A. Torralba, and S. Fidler, "Neuralfield-ldm: Scene generation with hierarchical latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8496–8506.
[58] Y. Deng, J. Yang, J. Xiang, and X. Tong, "Gram: Generative radiance manifolds for 3d-aware image generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 673–10 683.
[59] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger, "Graf: Generative radiance fields for 3d-aware image synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 154–20 166, 2020.
[60] 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.
[61] J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic, and S. Fidler, "Get3d: A generative model of high quality 3d textured shapes learned from images," *Advances In Neural Information Processing Systems*, vol. 35, pp. 31 841–31 854, 2022.
[62] M. A. Bautista, P. Guo, S. Abnar, W. Talbott, A. Toshev, Z. Chen, L. Dinh, S. Zhai, H. Goh, D. Ulbricht *et al.*, "Gaudi: A neural architect for immersive 3d scene generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 102–25 116, 2022.
[63] T. DeVries, M. A. Bautista, N. Srivastava, G. W. Taylor, and J. M. Susskind, "Unconstrained scene generation with locally conditioned radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 304–14 313.
[64] Z. Chen, G. Wang, and Z. Liu, "Scenedreamer: Unbounded 3d scene generation from 2d image collections," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 12, pp. 15 562–15 576, dec 2023.
[65] H. Xie, Z. Chen, F. Hong, and Z. Liu, "Generative gaussian splatting for unbounded 3d city generation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 6111–6120.
[66] ——, "Compositional generative model of unbounded 4d cities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
[67] R. Gao, K. Chen, E. Xie, L. Hong, Z. Li, D.-Y. Yeung, and Q. Xu, "Magicdrive: Street view generation with diverse 3d geometry control," in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 22 841–22 860.

[68] X. Wang, Z. Zhu, G. Huang, X. Chen, J. Zhu, and J. Lu, "Drivedreamer: Towards real-world-drive world models for autonomous driving," in *European conference on computer vision*. Springer, 2024, pp. 55–72.
[69] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.
[70] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *Advances in Neural Information Processing Systems*, vol. 37, pp. 21 875–21 911, 2024.
[71] V. Kulikov, S. Yadin, M. Kleiner, and T. Michaeli, "Sinddm: A single image denoising diffusion model," in *International Conference on Machine Learning*. PMLR, 2023, pp. 17 920–17 930.
[72] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.
[73] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
[74] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, and T. Wolf, "Diffusers: State-of-the-art diffusion models," https://github.com/huggingface/diffusers, 2022.
[75] Z. Hao, A. Mallya, S. Belongie, and M.-Y. Liu, "Gancraft: Unsupervised 3d neural rendering of minecraft worlds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 072–14 082.
[76] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.
[77] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.
[78] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," *arXiv preprint arXiv:1801.01401*, 2018.
[79] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He, "π^3: Permutation-equivariant visual geometry learning," *arXiv preprint arXiv:2507.13347*, 2025.
[80] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.
[81] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
[82] K. Fujiwara, R. Tsurumi, T. Kiyono, Z. Fan, X. Liang, B. Lei, W. Yap, K. Ito, and F. Biljecki, "Voxcity: A seamless framework for open geospatial data integration, grid-based semantic 3d city model generation, and urban environment simulation," *Computers, Environment and Urban Systems*, vol. 123, p. 102366, 2026.



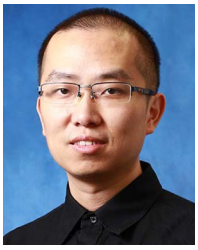
Fan Lu received the B.E. degree and Ph.D degree in vehicle engineering from Tongji University, Shanghai, China in 2020 and 2024. He is currently a post-doc researcher at the Tongji University, Shanghai, China. His research interests include 3D computer vision, neural rendering and 3D generative models.



Kwan-Yee Lin received the PhD degree from the Department of Information Science, School of Mathematical Sciences, Peking University, China. She is currently a Research Fellow at the University of Michigan, Ann Arbor. Her research interests lie at the intersection of computer vision, computer graphics, and machine learning.



Yan Xu received the BS degree in information science and engineering from Southeast University, in 2015, and the MS degree in computer science and technology from Zhejiang University, in 2018, and the PhD degree from the Multimedia Laboratory, at the Chinese University of Hong Kong, in 2023. He is currently a postdoctoral research fellow in Computer Science and Engineering (CSE) at the University of Michigan, Ann Arbor, His research interests include 3D computer vision, robotics, and deep learning.



Hongsheng Li received the BS degree in automation from the East China University of Science and Technology, Shanghai, China, in 2006, and the MS and PhD degrees in computer science from Lehigh University, Bethlehem, Pennsylvania, in 2010 and 2012, respectively. He is currently an associate professor with the Department of Electronic Engineering, Chinese University of Hong Kong. His research interests include computer vision, medical image analysis, and machine learning.



Guang Chen is a research professor at Tongji University and a senior research associate (guest) at the Technical University of Munich. His research interests include bio-inspired vision, machine learning, and computer vision with applications in autonomous systems. He received his Ph.D. degree from the Technical University of Munich. He received the B.S and M.Eng degree from Hunan University. He was awarded the program of Shanghai Rising Star 2021, and Shanghai S&T 35U35 2021. He serves as an Associate Editor for several international journals. He is the program chair of IEEE MFI 2022.



Changjun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995. He is currently a professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His current research interests include network computing, concurrency theory, formal verification of software, intelligent systems, and service-oriented computing. He has published more than 200 publications.