Text2NeRF: Text-Driven 3D Scene Generation with Neural Radiance Fields

Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao*

***Abstract*—Text-driven 3D scene generation is widely applicable to video gaming, film industry, and metaverse applications that have a large demand for 3D scenes. However, existing text-to-3D generation methods are limited to producing 3D objects with simple geometries and dreamlike styles that lack realism. In this work, we present Text2NeRF, which is able to generate a wide range of 3D scenes with complicated geometric structures and high-fidelity textures purely from a text prompt. To this end, we adopt NeRF as the 3D representation and leverage a pre-trained text-to-image diffusion model to constrain the 3D reconstruction of the NeRF to reflect the scene description. Specifically, we employ the diffusion model to infer the text-related image as the content prior and use a monocular depth estimation method to offer the geometric prior. Both content and geometric priors are utilized to update the NeRF model. To guarantee textured and geometric consistency between different views, we introduce a progressive scene inpainting and updating strategy for novel view synthesis of the scene. Our method requires no additional training data but only a natural language description of the scene as the input. Extensive experiments demonstrate that our Text2NeRF outperforms existing methods in producing photo-realistic, multi-view consistent, and diverse 3D scenes from a variety of natural language prompts. Our code and model will be available upon acceptance.**

***Index Terms*—Text-to-3D, NeRF, 3D scene generation, scene inpainting, depth alignment.**

I. Introduction

RECENT breakthroughs in text-to-image generation have also sparked great interest in zero-shot text-to-3D generation [1]–[4], as using natural language prompts to specify desired 3D models is intuitive and, therefore, could increase the productivity of the 3D modeling workflow and reduce the entry barrier for novices. However, contrary to the text-to-image case, in which paired data is abundant, it is impractical to acquire large quantities of paired text and 3D data, making the text-to-3D generation task still challenging [2], [5], [6].

To circumvent this data limitation, some pioneer works, including CLIP-Mesh [7], Dream Fields [1], DreamFusion [2], and Magic3D [6], use deep priors of pre-trained text-to-image models, such as CLIP [8] or image diffusion model [9], [10], to optimize a 3D representation, which thus empowers text-to-3D generation without the need for labeled 3D data. Despite the great success of these works, their generation results are still limited to 3D scenes with simple geometries and dreamlike styles. These limitations potentially stem from the fact that the deep priors derived from pre-trained image models, which are utilized to optimize the 3D representation, can only impose constraints on high-level semantics while neglecting low-level details. By contrast, recently concurrent arXived works, SceneScape [11] and Text2Room [12], directly employ the color image generated by text-image diffusion model to guide the reconstruction of 3D scenes. Although they support the generation of realistic 3D scenes, these methods mainly focus on indoor scenes and are hard to be extended into large-scale outdoor scenes due to the limitation of the explicit 3D mesh representation such that the stretched geometry caused by naive triangulation and noisy depth estimation. In contrast, our method utilizes NeRF as the 3D representation which has more advantage of modeling diverse scenes with complex geometry.

In this paper, we present Text2NeRF, a text-driven 3D scene generation framework by combining the best of Neural Radiance Field (NeRF) [13] and a pre-trained text-to-image diffusion model. We adopt NeRF as the 3D representation because of its superiority in modeling fine-grained and photorealistic details in various scenes [14]–[16], which could significantly suppress the artifacts caused by a triangular mesh. In addition, we use a pre-trained text-to-image diffusion model as the image-level prior to constrain the NeRF optimization from scratch without the demand of additional 3D supervision or multi-view training data.

Unlike the previous methods, e.g. DreamFusion [2], that supervise the 3D generation with the semantic priors, we leverage finer-grained image priors inferred from the diffusion model, which consequently allows our Text2NeRF to generate more delicate geometric structure and realistic texture in the 3D scenes. Specifically, we employ the diffusion model to generate a text-related image as the content prior and employ a monocular depth estimation method to offer the geometric prior of the generated scene. Both content and depth priors are leveraged to optimize the parameters of the NeRF representation. Moreover, to guarantee consistency between different views, we propose a progressive inpainting and updating strategy (PIU) for the novel view synthesis of the 3D scene. Through the PIU strategy, the generated scene can be expanded and updated in a view-by-view manner following a camera trajectory. In this way, the expanded area of the current view can be reflected in the next view by rendering the updated NeRF, which ensures that the same area will not be expanded repeatedly during the scene expansion process, thereby ensuring the continuity and view-consistency of the generated scene. Briefly, the 3D representation of NeRF together with our PIU strategy ensures the view-consistent images generated by the diffusion model for generating a

*: corresponding author.

J. Zhang, Z. Wan, C. Wang and J. Liao are with Department of Computer Science, City University of Hong Kong. E-mail: jbzhang6-c@my.cityu.edu.hk, ziyuwan2-c@my.cityu.edu.hk, cwang355-c@my.cityu.edu.hk, jingliao@cityu.edu.hk. X. Li is with Tencent AI Lab. E-mail: xliea@connect.ust.hk

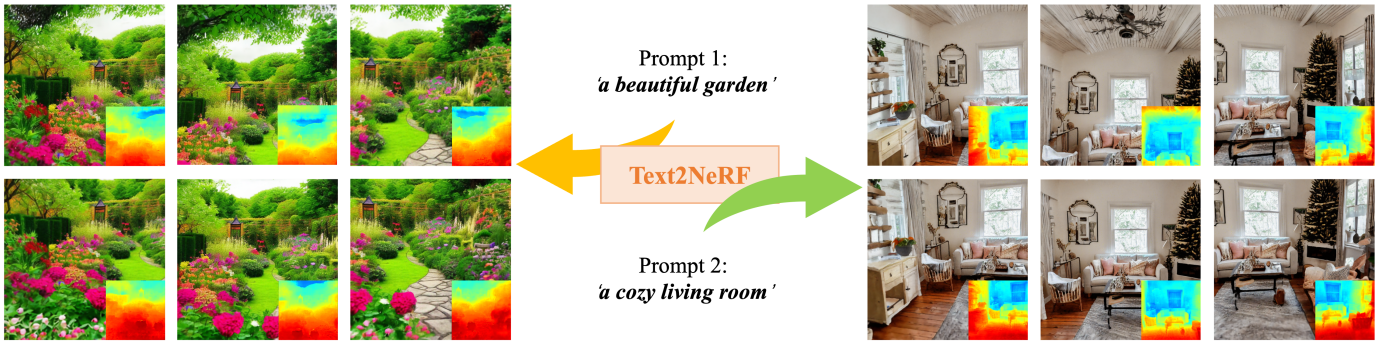


Fig. 1. We propose Text2NeRF, a text-driven 3D scene generation framework by combining the neural radiance field representation and a pre-trained text-to-image diffusion model. Our Text2NeRF is capable of generating diverse and view-consistent indoor/outdoor 3D scenes solely from natural language descriptions. Please refer to our supplementary video demo for more examples.

view-consistent 3D scene. In practice, we find that single-view training in NeRF leads to overfitting to this view and thus causes geometric ambiguity during view-by-view updating due to the lack of multi-view constraints. To overcome this issue, we build a support set for the generated view to offer multi-view constraints for the NeRF model.

Meanwhile, inspired by [17], in addition to image RGB loss, we also adopt a L_2 depth loss to achieve depth-aware NeRF optimization and improve the convergence rate and stability of the NeRF model. Considering that the depth maps at different views are estimated independently and could be inconsistent in the overlapped regions, we further introduce a two-stage depth alignment strategy to align the depth value of the same point from different views. Thanks to the above well-designed components, our Text2NeRF is capable of generating diverse, high-fidelity, and view-consistent 3D scenes solely from natural language descriptions, as shown in Fig. 1. Due to the generality of our method, Text2NeRF could generate a wide range of 3D scenes, including indoor, outdoor, and even artistic scenes (Fig. 7 and 8) and is not limited by the view range and can generate 360-degree scenes (Fig. 6). Extensive experiments demonstrate that our Text2NeRF outperforms the previous methods both qualitatively and quantitatively.

Our contributions are summarized as follows:

- We propose a text-driven realistic 3D scene generation framework combining diffusion model with NeRF representations, which supports zero-shot generation of various indoor/outdoor scenes from a variety of natural language prompts.
- We introduce the PIU strategy to progressively generate view-consistent novel contents for 3D scenes, and build the support set to provide multi-view constraints for the NeRF model during view-by-view updating.
- We employ the depth loss to achieve depth-aware NeRF optimization, and introduce a two-stage depth alignment strategy to eliminate estimated depth misalignment in different views.

II. Related Work

A. Text-Driven 3D Generation

The long-standing problem of 3D generation entails constructing diverse view-consistent 3D geometry and high-fidelity textures. Early works, like 3D-GAN [18], Pointflow [19], and ShapeRF [20] focus more on the category-specific texture-less geometric shape generation based on the representations of voxels or point clouds. Subsequently, PlatonicGAN [21], HoloGAN [22], and VolumeGAN [23] are proposed to generate textured 3D scenes by learning the structural and textual representations from a category-specific dataset such as cars, faces, indoor scenes, et al. Although such methods achieve yield promising 3D scenes on specific categories, they cannot handle text-driven generative tasks. To achieve text-driven 3D generation, Text2shape [24] uses two encoder networks to learn cross-modal connections between texts and 3D models in the embedding space from a specific paired scene-text dataset.

Thanks to the rapid development of text-to-image methods, recent works aim to employ the pre-trained text-to-image model to guide the 3D scene generation. For example, CLIP-Mesh [7] adopts a semantically supervised optimization strategy to deduce shapes and textures for 3D meshes under the guidance of a pre-trained CLIP [8] model. Similar to CLIP-Mesh, PureCLIPNeRF [25] and DreamFields [1] use the pre-trained CLIP model to guide the generation of 3D objects with implicit NeRF representations. Compared with the CLIP model, the state-of-the-art text image diffusion models [9], [10], [26], [27] undoubtedly have more powerful generation capabilities due to their abundant training data and excellent structure. Therefore, DreamFusion [2] and SJC [3] propose a score distillation sampling (SDS) loss to extract deep semantic priors from pre-trained text-to-image diffusion models [9], [10] and supervise the generative network of 3D models. Subsequently, some follow-up works, such as Magic3D [6], Latent-NeRF [28], and 3DFuse [4], are proposed to improve the quality of generated 3D models under the constraint of SDS loss. Although these methods enable producing diverse 3D models related to the input prompts, they fail to generate a photorealistic 3D scene with complex geometry and high-fidelity textures because only high-level semantic priors of the pre-trained model are used to constrain the 3D generation. In contrast, our method infers low-level content and depth priors from the pre-trained text-to-image diffusion model, with which geometry and texture details in a photorealistic 3D scene are well constrained.

More recently, SceneScape [11] and Text2Room [12], which are *independent and concurrent* to our work, propose text-to-3D schemes similar to our method. Differently, they employ explicit polygon meshes as the 3D representation during their generative procedure, which limits the representation of outdoor scenes and leads to stretched geometry and blurry artifacts in the fusion regions of mesh faces. In contrast, our implicit NeRF representation and reconstruction strategy could model fine-grained geometry and textures without specific scene requirements thus enabling our method to produce both indoor and outdoor scenes.

B. Novel View Synthesis from a Single Image

Some novel view synthesis methods constrained by 3D presentation are able to generate a 3D-consistent experience from a single image. For example, several existing 3D photography methods, like SVS [29], 3DP [30], and 3D-Ken-Burns [31], use multi-plane images (MPI) or layered depth images (LDI) as 3D representations, and then employ pre-trained inpainting models to complete occluded regions to synthesize plausible novel views. However, such methods can only produce views in a small range due to the limitation of their specific 3D representation. By contrast, some other methods achieve the 3D reconstruction and novel view synthesis by mapping single-view image information to conventional 3D models. For instance, SynSin [32] transforms the image features into a point cloud based on the predicted depth information and decodes the rendered feature map to synthesize a novel view of the 3D scene. PixelSynth [33] constructs a point cloud by directly mapping the pixel color to the 3D points and introduces outpainting and refinement modules to fill the missing information in novel views. Worldsheet [34] synthesizes novel views of the 3D scene by warping a planar mesh sheet according to the input image and predicted depth. Intuitively, directly applying one of these methods to extrapolate an image generated by a text-to-image model to novel views is a naive strategy for a text-driven 3D generation. However, this naive strategy is limited in several aspects. First, their scene extrapolation is based on the input image only, not conditioned on the text prompt. Consequently, their generated scene is within a limited view range around the input image to ensure semantic consistency. In contrast, our method allows for generating new content in novel views driven by the given text prompt. Therefore, ours is not limited by the view range and can even generate 360-degree scenes that are coherent with the text description. Besides, the explicit 3D representations, such as coarse mesh or point cloud, adopted in these methods restrict them from rendering fine results, while ours leveraging the implicit NeRF representation is superior in representing and rendering high-fidelity details.

III. Method

We propose a text-driven 3D scene generation framework to progressively generate 3D scenes according to given text prompts as shown in Fig. 2. We first generate an initial view by a text-to-image diffusion model. Based on the initial image, we build the support views and corresponding depth maps as the support set to offer multi-view constraints for the NeRF reconstruction using the depth image-based rendering (DIBR) method. After training this initialized NeRF model, we further introduce a progressive inpainting and updating (PIU) strategy to expand the generated scene view-by-view. Specifically, we render a novel view and complete its missing regions via the diffusion model with the text prompt. Then we take the inpainted view and constructed its support set as the additional supervision to update the NeRF model. By progressively adding new content consistent with the existing scene, our framework succeeds in generating realistic 3D scenes with fine-grained details.

A. Scene Initialization

Content Generation. To obtain the initial scene content with respect to the input prompt p, we first employ a pre-trained diffusion model f_d conditioned on p to generate a 2D scene image $I_0 = f_d(\epsilon \mid p)$, where ϵ is a random Gaussian noise. Due to the lack of geometric information in this single view I_0, a monocular depth estimation model f_e is adopted to offer the geometric inference $D_0 = f_e(I_0)$. The initial view I_0 and depth map D_0 will be used to produce a support set for the 3D scene initialization.

3D Scene Representation. Unlike explicit representations like polygon meshes or point clouds [33], [35], which are hard to represent complex geometry, NeRF shows its power in representing arbitrarily complex scenes. Therefore, we employ a NeRF network f_θ to represent the 3D scene. In NeRF, volume rendering [13] is used to accumulate the color in the radiance fields:

$$\mathbf{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})\,\mathrm{d}t, \tag{1}$$

where $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ represents the 3D coordinates of sampled points on the camera ray emitted from the camera center $\mathbf{o}$ with the direction $\mathbf{d}$. t_n and t_f indicate the near and far sampling bounds. $(\mathbf{c}, \sigma) = f_\theta(\mathbf{r}(t))$ are the predicted color and density of the sampled point along the ray. $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))\,\mathrm{d}s)$ is the accumulated transmittance. Different from NeRF that takes both the 3D coordinate $\mathbf{r}(t)$ and view direction $\mathbf{d}$ in Eq. 1 to predict the radiance $\mathbf{c}(\mathbf{r}(t), \mathbf{d})$, we omit $\mathbf{d}$ to avoid the effect of view-dependent specularity. Additionally, inspired by [17], we introduce the depth constraint into NeRF optimization to achieve depth-aware NeRF optimization and speed up model convergence. To this end, the predicted depth value $z(\mathbf{r})$ is required to be calculated:

$$z(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))t\,\mathrm{d}t. \tag{2}$$

To be convenient, we denote the volume rendering on view i as $(I_i^R, D_i^R) = VR(f_\theta \mid i)$, where I_i^R and D_i^R are the rendered image and depth map, respectively.

Support Set. Since the lack of multi-view supervision, directly adopting single-view I_0 and its depth D_0 to train the radiance fields easily leads to overfitting and geometric ambiguity. To overcome this issue, we adopt a depth image-based rendering (DIBR) method [36] to construct a support set $\mathbf{S}_0$ for the

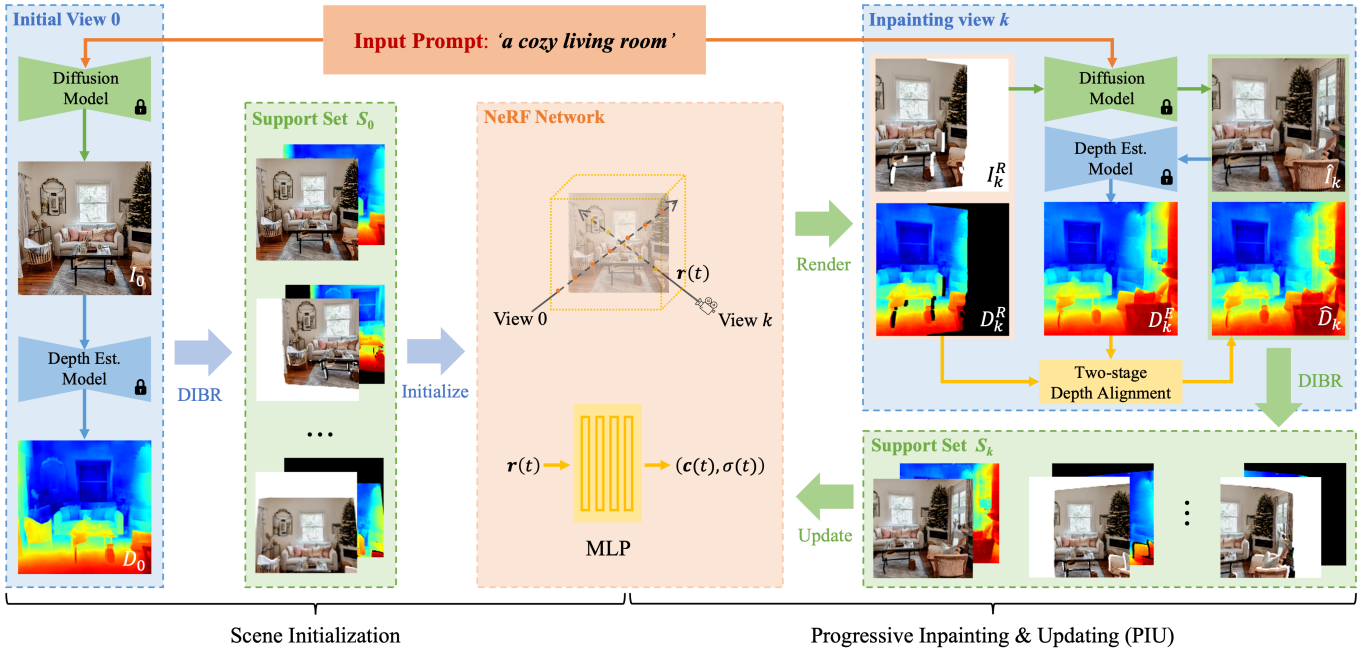


Fig. 2. Overview of our Text2NeRF. Given an input text prompt, we infer an initial view I_0 and estimate its depth D_0 via a pre-trained diffusion model and a depth estimation model. Then we use the depth image-based rendering (DIBR) to warp the initial view and its depth map to various views to build the support set $\mathbf{S}_0$ for initializing the neural radiance field (NeRF). Afterward, we design a progressive scene inpainting and updating strategy to complete missing regions consistently. During each update, we first render the initialized NeRF in a novel view k to produce the image I_k^R and depth D_k^R with missing regions. Then, the diffusion model is adopted to generate completed image $\hat{I}_k$ and the depth estimation model is used to predict its depth D_k^E. Furthermore, a two-stage depth alignment is implemented on D_k^R and D_k^E to obtain aligned depth $\hat{D}_k$. Finally, the support set $\mathbf{S}_k$ of view k is added into training data to update NeRF.

initialization. Specifically, for each pixel q in I_0 and its depth value z in D_0, we compute its corresponding pixel $q_{0\to i}$ and depth $z_{0\to i}$ on a surrounding view i:

$$[q_{0\to i}, z_{0\to i}]^T = \mathbf{K}\mathbf{P}_i\mathbf{P}_0^{-1}\mathbf{K}^{-1}\,[q, z]^T \tag{3}$$

where $\mathbf{K}$ and $\mathbf{P}_i$ indicate the intrinsic matrix and the camera pose in view i. For convenience, we denote the DIBR process from view 0 to view i as $DIBR_{0\to i}$.

We manually set the intrinsic matrix $\mathbf{K}$ and camera pose $\mathbf{P}_0$ and then use $\mathbf{P}_0$ to get surrounding camera poses $\mathbf{P}_i$. Specifically, we first define a surrounding circle of radius ζ centered at the current camera position and having the same z-coordinate as the current camera position. Then, we uniformly sample ξ points as the camera positions and employ the same camera direction as the current view to produce the warping views in the support set. Here, ζ is the shift distance and ξ is the number of warping views. In practice, we define $\xi = 8$ by shifting the camera position with $\zeta = 0.2$ in directions of up, down, left, right, upper left, lower left, upper right, and lower right, respectively. With these support views, along with the initial view I_0, we train a NeRF as the initialized 3D scene.

B. Text-Driven Inpainting

After the scene initialization, the radiance field can be rendered in arbitrary novel views. However, the rendered results other than the initial view 0 will inevitably have missing content since the information in the initial scene is derived from the single image I_0. To complement the missing regions, we employ a text-driven inpainting method based on the pre-trained diffusion model f_d. Specifically, we first render a novel view I_k^R to be inpainted. Then, we calculate the mask M_k of missing parts in I_k^R by warping all known views to the rendered view k according to Eq. 3. The rendered image I_k^R along with the mask M_k and input prompt p are fed into the diffusion model f_d to predict an inpainting result of I_k^R:

$$\hat{I}_k = f_d\left(I_k^R, M_k \mid p\right). \tag{4}$$

Considering that the inpainting process is stochastic, although the current diffusion model has a strong completion ability, it is difficult to guarantee that the quality of each result can meet the expected requirements. We thus perform the inpainting process many times for each view I_k^R to be completed, and automatically select the one from all candidate inpainting results $\hat{I}_k^j$ that is most similar as the initial view in the CLIP semantic space:

$$\hat{I}_k = \arg\max_j \cos\left(E_{CLIP}\left(I_0\right), E_{CLIP}\left(\hat{I}_k^j\right)\right), \tag{5}$$

where $E_{CLIP}(\cdot)$ is the image encoder of CLIP model [37]. In practice, we generate 30 inpainting results as candidates for each view to be completed.

Besides, we also use the depth estimation model f_e to estimate the depth map D_k^E for the inpainted image $\hat{I}_k$. Note that, unlike the depth map D_0 of the initial view, D_k^E cannot be directly taken as the supervision to update the radiance field since it is predicted independently and could conflict with known depth maps such as D_k^R in the overlapping regions. To

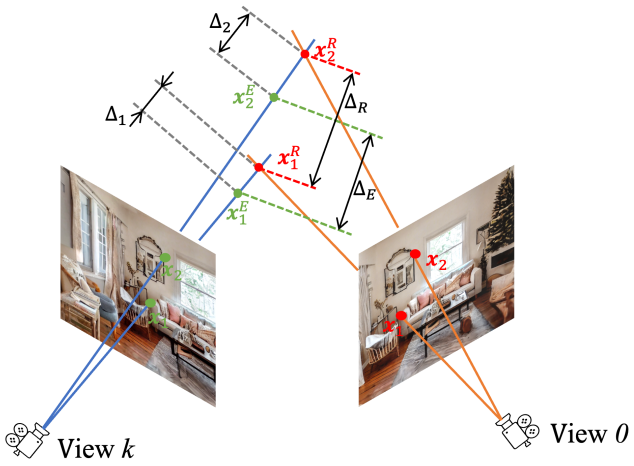
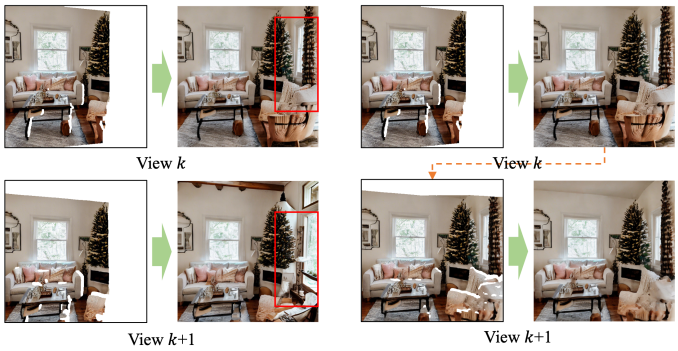


Fig. 3. Example of scale and distance disparities. x_1 and x_2 are two aligned pixels in different views. The spacial points x_1^E and x_2^E are projected based on the estimated depth D_k^E in view k. x_1^R and x_2^R are points projected according to the rendered depth D_k^R which is constrained by known views such as view 0. Here, $\Delta_E \neq \Delta_R$ indicates the scale disparity, and $\Delta_1 \neq 0$ or $\Delta_2 \neq 0$ indicate the distance disparity.



(a) Independent Inpainting (b) Progressive Inpainting

Fig. 4. Examples of two inpainting strategies. The intuitive independent inpainting strategy simultaneously performs rendering and inpainting for each view. Due to there is no 3D constraint during 2D inpainting, the overlapping regions inpainted in different views will be view-inconsistent, as shown in the red box. In contrast, our progressive inpainting strategy achieves view-consistent inpainting results by introducing NeRF as a 3D constraint and reflecting previously inpainted content into the next view.

solve this problem, we implement depth alignment to align the estimated depth map to the known depth values in the radiance field.

C. Depth Alignment

Due to the lack of geometric constraint during the depth estimation, the predicted depth values could be misaligned in the overlapping regions [38], for example, the estimated depth D_k^E of the inpainted view may be inconsistent with the depth D_k^R rendered from NeRF since D_k^R is constrained by previous known views. The inconsistency is manifested in two aspects: scale disparity and distance disparity. For example, the *depth difference* of the distance between two pixel-aligned spatial points and the *depth value* of a specific point could be both different in depth maps estimated from different views, as shown in Fig.3. The former difference is the scale disparity and the latter is the distance disparity. In the case of scale disparity, we cannot align both points by shift processing because even if we align the depth value of one of the points, the other point is still misaligned. To eliminate the scale and distance disparities between the overlapping regions of the rendered depth map D_k^R and the estimated depth map D_k^E of the novel view, we introduce a two-stage depth alignment strategy. Specifically, we first globally align these two depth maps by compensating for mean scale and distance disparities. Then we finetune a pre-trained depth alignment network to produce a locally aligned depth map.

To determine the mean scale and distance disparities, we first randomly select M pixel pairs from the overlapping regions and deduce their 3D positions under depth D_k^R and D_k^E, denoted as $\left\{(\mathbf{x}_j^R, \mathbf{x}_j^E)\right\}_{j=1}^M$. Next, we calculate the average scaling score s and depth offset δ to approximate the mean scale and distance disparities:

$$s = \frac{1}{M-1} \sum_{j=1}^{M-1} \frac{\|\mathbf{x}_j^R - \mathbf{x}_{j+1}^R\|_2}{\|\mathbf{x}_j^E - \mathbf{x}_{j+1}^E\|_2}, \tag{6}$$

$$\delta = \frac{1}{M} \sum_{j=1}^{M} \left(z\left(\mathbf{x}_j^R\right) - z\left(\hat{\mathbf{x}}_j^E\right) \right), \tag{7}$$

where $\hat{\mathbf{x}}_j^E = s \cdot \mathbf{x}_j^E$ indicates the scaled point and $z(\mathbf{x})$ represents the depth value of point $\mathbf{x}$. Then D_k^E can be globally aligned with D_k^R by $D_k^{global} = s \cdot D_k^E + \delta$.

Since depth maps used in our pipeline are predicted by a network, the disparities between D_k^R and D_k^E are not linear, that is why the global depth aligning process cannot solve the misalignment problem. To further mitigate the local difference between D_k^{global} and D_k^R, we train a pixel-to-pixel network f_ψ for nonlinear depth alignment. During optimization of each view, we optimize the parameter ψ of the pre-trained depth alignment network f_ψ by minimizing their least square error in the overlapping regions:

$$\min_{\psi} \left\| \left(D_k^{global} - D_k^R \right) \odot M_k \right\|_2. \tag{8}$$

Finally, we can derive the locally aligned depth using the optimized depth alignment network: $\hat{D}_k = f_\psi(D_k^{global})$. For convenience, we denote the two-stage depth alignment process as $align(D_k^E \mid D_k^R, M_k)$. In terms of the training of the depth alignment network, please refer to the implementation details in Sec. III-E.

D. Progressive Inpainting and Updating

After obtaining the inpainted image $\hat{I}_k$ and the aligned depth map $\hat{D}_k$ at iteration k, we could use the depth image-based rendering to construct the corresponding support set $\mathbf{S}_k$ which is used to update the radiance field. An intuitive solution is to render all the views from the initialized radiance field and inpaint them independently. However, there may be many

Algorithm 1 Progressive Inpainting & Updating Strategy

Input:
prompt p;
pre-trained diffusion model f_d;
pre-trained depth estimation model f_e;
initialized NeRF f_θ;
views to be updated $\mathbf{V} = \{1, 2, \cdots, N\}$;
views already updated $\widetilde{\mathbf{V}} = \{0\}$.
Updating Process:
for k in $\mathbf{V}$ **do**
rendering $(I_k^R, D_k^R) = VR(f_\theta \mid k)$
mask calculation $M_k \leftarrow \cap\{DIBR_{n\to k}\}$, where $n \in \widetilde{V}$
if $sum(M_k) > 0$ **then**
text-driven inpainting $\hat{I}_k = f_d\left(I_k^R, M_k \mid p\right)$
else
continue
end if
depth estimation $D_k^E = f_e(\hat{I}_k)$
depth alignment $\hat{D}_k = align(D_k^E \mid D_k^R, M_k)$
support set $\mathbf{S}_k \leftarrow \cup\{DIBR_{k\to support\_views}\}$
update views updated $\widetilde{\mathbf{V}} = \widetilde{\mathbf{V}} \cup \{k\}$
update NeRF model $f_\theta \leftarrow \mathbf{S}_k$
end for
Return: updated NeRF f_θ

overlapping regions to be inpainted among different views, so the 2D text-driven inpainting model cannot produce view-consistent content in all views without 3D constraints, as shown in Fig. 4(a). To guarantee the view consistency and avoid the ambiguity of geometry and appearance during the scene inpainting process, we propose a progressive inpainting and updating strategy to update the radiance fields view by view, as shown in Fig. 4(b) and Algorithm 1. In this strategy, we update the radiance field f_θ after every inpainting process. It means that the previous inpainted content will be reflected in the subsequent renderings, and these parts will be regarded as known regions and will not be inpainted again in other views.

E. Training and Implementation Details

Training Objective. We use a RGB loss, a depth loss, and a transmittance loss to optimize the radiance field of the 3D scene. Like previous NeRF-based works [13], [39], [40], the RGB loss L_{RGB} is defined as a L_2 loss between the rendered pixel color $\boldsymbol{C}^R$ and the color $\boldsymbol{C}$ generated by the diffusion model f_d. Different from previous works that employ regularized depth losses to handle uncertainty or scale-variant problem [17], [41], we adopt a stricter depth loss L_{Depth} to minimize the L_2 distance between the rendered depth D^R and the aligned estimated depth $\hat{D}$, since the aligned depth maps used in our framework are scale-invariant and can be regarded as ground truth. Besides, inspired by [1], we design a depth-aware transmittance loss L_T to encourage the NeRF network to produce empty density before the camera ray reaching the expected depth $\hat{z}$:

$$L_T = \|T(t) \cdot m(t)\|_2 \tag{9}$$

where $m(t)$ is a mask indicator that satisfies $m(t) = 1$ when $t < \hat{z}$, otherwise $m(t) = 0$. $\hat{z}$ is the pixel-wise depth value in the aligned depth map $\hat{D}$. $T(t)$ is the accumulated transmittance which is same as the $T(t)$ in Eq. 1. The total objective is then defined as:

$$L_{total} = L_{RGB} + \lambda_d L_{Depth} + \lambda_t L_T, \tag{10}$$

where λ_d and λ_t are constant hyperparameters balancing between terms.

Implementation Details. We implement the Text2NeRF with the Pytorch framework [42] and adopt TensoRF [39] as the radiance field. Note that, to make TensoRF satisfy the scene generation in a large view range, we let the camera position near the center of the NeRF bounding box and set outward-facing viewpoints. For scene generation, we use the stable diffusion [9] to generate the scene content related to the input prompt and use the boosting monocular depth estimation model [43] to estimate the depth for each view. In term of depth alignment, the super-parameter M in Eq. 6 is set as $\min(M_0, 10000)$ in practice, where M_0 indicates the number of all matched points in the overlapping regions. Besides, the depth alignment network in our framework uses the same pixel-to-pixel U-net architecture as the depth merging network in [43]. To train this network, we first predict 10000 depth maps using the depth estimation models and add continuous non-linear random noise into these depth maps, i.e., $\widetilde{D} = (D + \tau_1) \cdot D^{1/\tau_2}$ where D is the depth; τ_1 and τ_2 indicate the shift and scale factors, which are randomly sampled in the range $[0, 1]$ and $[30, 50]$, respectively. Then, we use the noisy depth maps as input and constrain the depth alignment network with the noise-free depth maps, so that the network acquires the ability to locally change the depth value. Finally, we finetune the network based on Eq. 8 to produce the local aligned depth for each inpainting view. During training, we use the same setting as [39] for the optimizer and learning rate and set the hyperparameters in our objective function as $\lambda_d = 0.005$ and $\lambda_t = 1000$.

IV. Experiments

In this section, we first briefly introduce several state-of-the-art text-to-3D baselines and metrics (Sec. IV-A), and then we apply our Text2NeRF to a variety of text prompts to evaluate its capability on photo-realistic indoor and outdoor 3D scenes generation and compare with the baseline methods (Sec. IV-B). Furthermore, we conduct ablation studies to investigate the effectiveness of major components in our method (Sec. IV-C).

A. Setup

Baseline Methods. To evaluate the performance of our method on text-driven 3D scene generation, we compare our method with seven baseline methods, as shown in Table. I, including four generation methods guided by the high-level semantic prior (i.e., CLIP-Mesh [7], SJC [3], DreamFusion [2], and DreamFusion-Scene) and three methods guided by the low-level image prior (i.e., 3DP [30], PixelSynth [33], and Text2Room [12]). Here, CLIP-Mesh, SJC, and DreamFusion are three existing state-of-the-art text-to-3D methods which

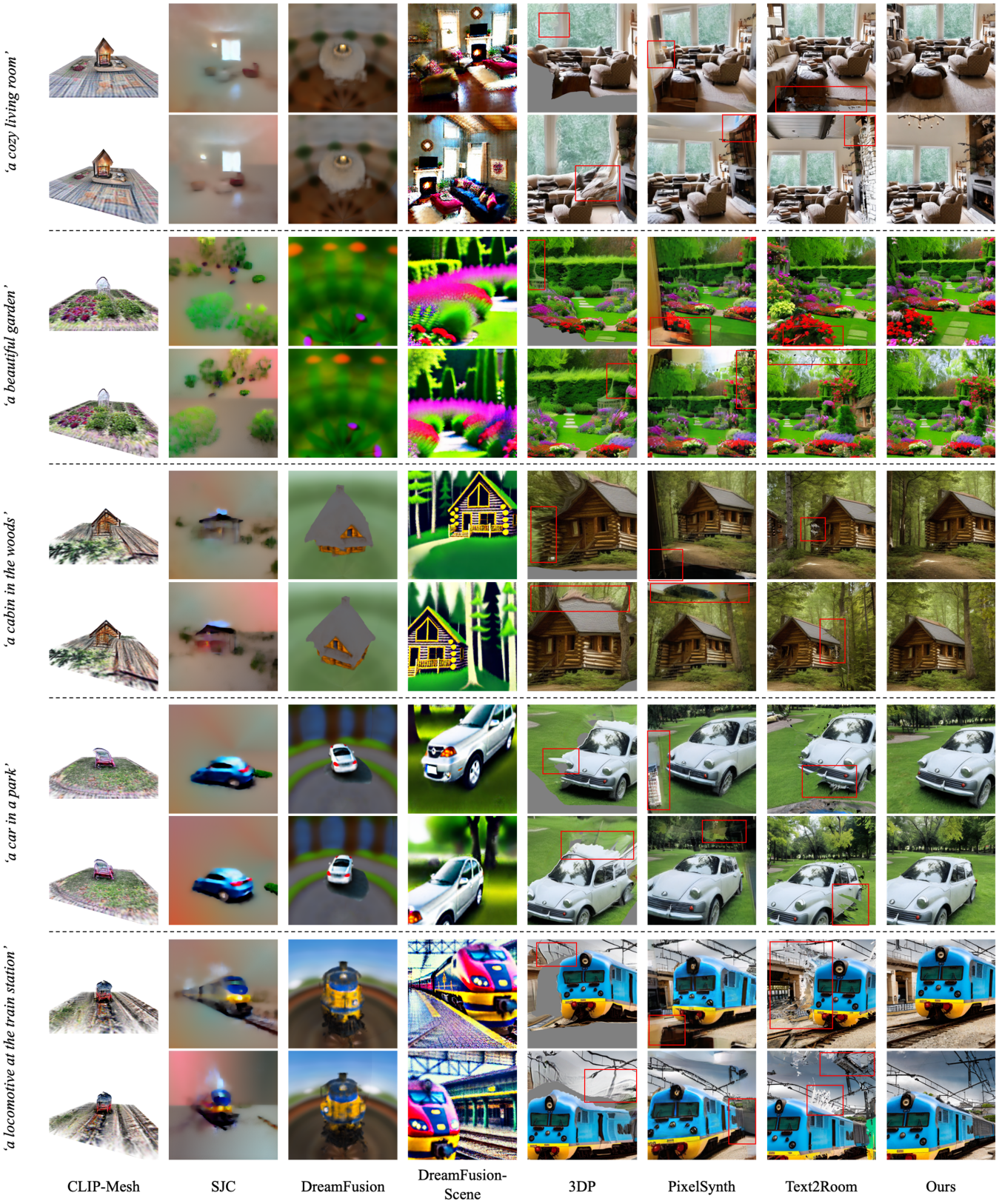


Fig. 5. Qualitative comparison of results generated by baselines and ours on different text prompts. Here, we only show two rendering results from different views for each generated scene of each method due to space limitations. Please refer to the supplementary material for video results.

TABLE I

DISCRIMINATION OF BASELINE METHODS AND OURS IN GUIDANCE TYPE AND 3D REPRESENTATION, AND QUANTITATIVE COMPARISON OF RESULTS GENERATED BY BASELINES AND OURS. HERE, *S* INDICATES HIGH-LEVEL SEMANTIC PRIOR AND *I* REPRESENTS LOW-LEVEL IMAGE PRIOR. COMPARED TO BASELINE METHODS, OUR TEXT2NERF YIELDS A LOWER METRIC SCORE ON BOTH BRISQUE AND NIQE AND A HIGHER SCORE ON CLIP SIMILARITY, WHICH MEANS THAT OUR METHOD ACHIEVES TO GENERATE MORE REALISTIC AND HIGHER-QUALITY TEXT-RELATED RESULTS.

Methods	CLIP-Mesh	SJC	DreamFusion	DreamFusion-Scene	3DP	PixelSynth	Text2Room	**Ours**
Guidance Type	S	S	S	S	I	I	I	I
3D Representation	Mesh	NeRF	NeRF	NeRF	LDI&Mesh	Point Cloud	Mesh	NeRF
BRISQUE ↓	46.266	39.543	67.012	37.799	32.469	25.924	28.395	**24.498**
NIQE ↓	6.652	11.971	12.022	6.402	6.124	6.604	5.415	**4.618**
CLIP Score ↑	27.480	24.152	22.576	28.032	26.638	27.267	28.056	**28.695**

(*a*) *'a beautiful garden'*

(*b*) *'a forest in CG style'*

Fig. 6. 360-degree scenes generated by our Text2NeRF. Please refer to the supplementary material for video results.

employ NeRF as their 3D representation. DreamFusion-Scene is a modified version of DreamFusion designed for generating 3D scenes, as the vanilla version focuses on 3D objects and is not suitable for outward-facing scene generation. 3DP and PixelSynth are two novel view synthesis methods using explicit polygon meshes or point clouds as 3D representation, which represent a naive strategy for the text-driven 3D generation, i.e., applying existing novel view synthesis methods to the single image generated by a text-to-image diffusion model. Text2Room is one recently arXived concurrent work which employ polygon meshes to represent the generated 3D scenes.

Notably, due to DreamFusion being performed based on the unavailable Imagen [10] diffusion model, we replace it with a Pytorth implementation[1] powered by the stable diffusion [9] model.

Metrics. Since there is no ground truth as a reference for generated 3D scenes related to the text prompts, previous reference-based metrics are not suitable for the generation tasks, like PSNR and LPIPS [44]. Instead, we use two metrics, blind/referenceless image spatial quality evaluator (BRISQUE) [45] and natural image quality evaluator (NIQE) [46], on no-reference image quality assessment to evaluate rendering quality of generated 3D scenes. Besides, we adopt the CLIP text-image similarity score [8] to measure how well the rendered images align with the input prompt.

B. Comparisons

We evaluate our Text2NeRF and compare it with baseline methods for text-driven 3D scene generation across various prompts, as shown in Fig. 5. Additionally, we provide the average evaluation scores of BRISQUE, NIQE, and CLIP for the rendered images produced by different methods, as shown in Table 1. Clearly, our method surpasses the baselines by generating higher-quality 3D scenes, as indicated by lower BRISQUE and NIQE values. Moreover, our method ensures the semantic relevance between the generated scene and the input text, resulting in a higher CLIP score. Overall, both qualitative and quantitative results unequivocally demonstrate the superiority of our approach over the baseline methods.

As shown in the first three columns of Fig. 5, CLIP-Mesh, SJC, and DreamFusion struggle to generate complex 3D scenes related to the given prompts since their primary design focus on simple 3D object generation. Consequently, their BRISQUE and NIQE values tend to be higher compared to other methods, indicating relatively poorer quality in the rendered images of their generated scenes. In particular, CLIP-Mesh generates 3D scenes by optimizing initial sphere and planar multi-mesh representations, guided by a pre-trained CLIP model. Due to the absorption of environmental semantics into the planar mesh during optimization, CLIP-Mesh is limited to producing object-centric scenes. Similarly, SJC and DreamFusion adopt a 'looking-inside' camera setting and sample the camera position in outer spherical coordinates of the radiation field. In this way, the unbounded background environment is difficult to optimize in the central radiance field, resulting in the tendency of both SJC and DreamFusion to also generate object-centric scenes. Unlike SJC, DreamFusion incorporates an additional background spherical surface outside the central radiance field. This design choice allows DreamFusion to include the scene environment in the background representation, fulfilling high-level semantic

[1]https://github.com/ashawkey/stable-dreamfusion

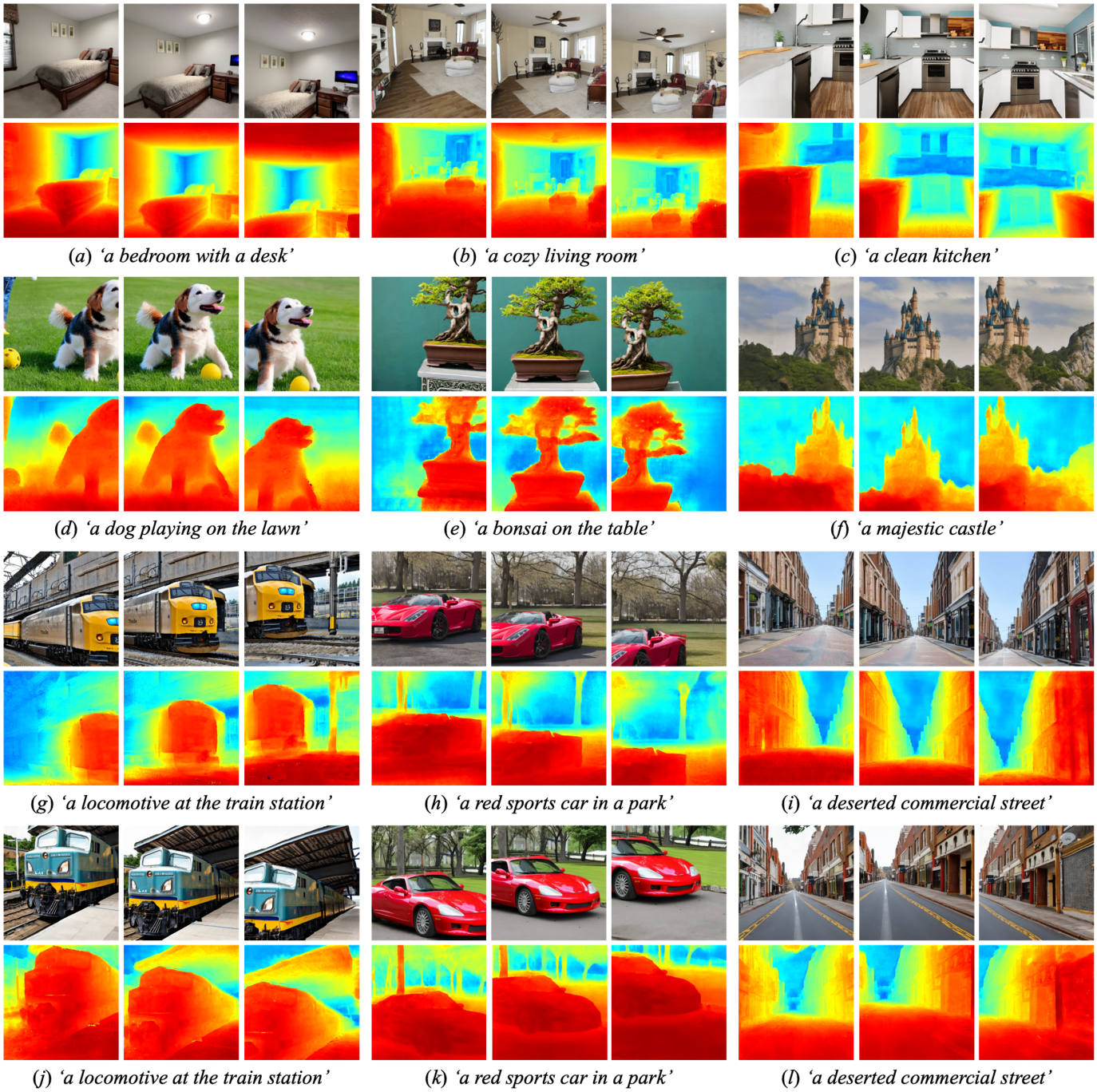


Fig. 7. More results of our 3D scene generation. It is worth noting that our method can generate diverse results from the same text prompt (g)&(j), (h)&(k), and (i)&(l). Please refer to the supplementary material for video results.

priors, as observed in the examples of the *garden* and *car*. Excluding completely failed cases, CLIP-Mesh, DreamFusion, and SJC exhibit the ability to generate object-centric scenes with a dreamlike style. However, they struggle to create 3D scenarios with complex spatial arrangements and geometry. In contrast, the modified DreamFusion-Scene successfully generates text-related 3D scenes with more complex geometry. Nevertheless, DreamFusion-Scene still falls short in deducing detailed structures and achieving photorealistic textures for the generated scenes. This limitation stems from the fact that the deep semantic priors provided by the text-image method are insufficient to fully constrain the low-level details.

Unlike existing text-to-3D methods guided by the deep semantic priors, the naive strategy that utilizes the novel view synthesis methods, 3DP and PixelSynth, to reconstruct the 3D scene from a single text-related image generated by the text-image model. The fifth and sixth columns of Fig. 5 demonstrate that such methods achieve to produce photo-realistic text-related 3D scenes with textual details, since they leverage the low-level content and depth priors to guide the

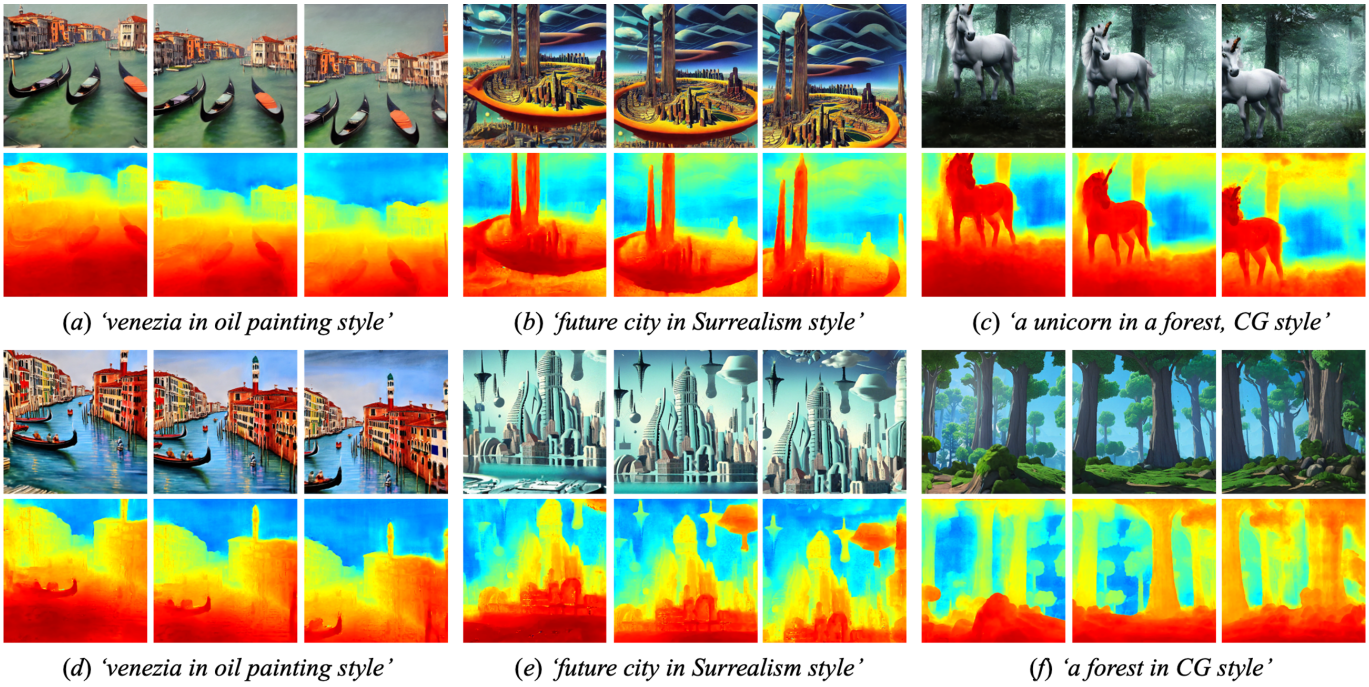


Fig. 8. 3D scenes in artistic styles generated by our Text2NeRF. Please refer to the supplementary material for video results.

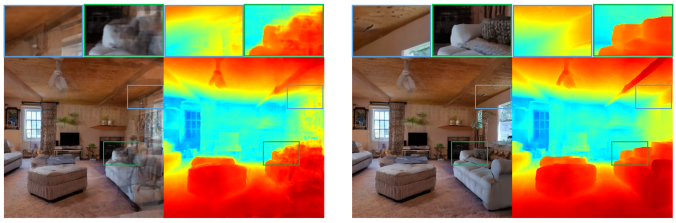
3D reconstruction process. As a result, their BRISQUE and NIQE values are substantially lower than those of previous semantic-guided generation methods, indicating superior scene quality and realism. However, their scene extrapolation is implemented within a limited view range and is independent of the input prompt, making it difficult for them to generate semantically consistent content in some novel views of the scene. Specifically, 3DP employs LDI and polygon meshes to represent the reconstructed 3D scene, which is susceptible to depth discontinuities. This can lead to missing content or stretched geometry in regions where depth is discontinuous, as illustrated in the gray area and red box in the fifth column of Fig. 5. By contrast, PixelSynth represents the 3D scene as point clouds, which mitigates the sensitivity to depth discontinuities to some extent. However, limited by its prompt-independent inpainting module, PixelSynth is prone to generating incoherent and blurry content, especially in the inpainted regions. Moreover, as shown in Fig. 6, our Text2NeRF supports text-driven scene generation in a large view range thanks to our progressive scene inpainting and updating strategy. On the other hand, other novel view synthesis methods produce blurred scene-filling results even at a small viewing angle since the text-related guidance is not considered in such methods.

In comparison to the novel view synthesis methods, both the concurrent work Text2Room and ours leverage the text-conditioned diffusion model as an inpainting module to complete missing regions in 3D scenes. To preserve the low-level textural details in the 2D images generated by the diffusion model, we both introduce a color objective as the low-level image guidance. This shared characteristic allows both approaches to generate 3D scenes that simultaneously exhibit high quality (as indicated by low BRISQUE and NIQE values) and high semantic relevance (as reflected in high CLIP scores). However, there are differences in how the generated scenes are represented. Unlike Text2Room that utilizes polygon meshes to represent the generated scenes, we adopt the NeRF (Neural Radiance Fields) framework, encoding the 3D scenes in an implicit network. This choice enables our method to effectively represent both bounded and unbounded scenes. As demonstrated in the seventh column of Fig. 5, Text2Room encounters challenges in generating certain outdoor scenes and often produces stretched geometry in regions with depth discontinuity. In contrast, our method successfully generates indoor and outdoor 3D scenes with complex structures and achieves a higher level of photorealistic detail.

Furthermore, we show more examples of 3D scenes generated by our Text2NeRF in Fig. 7. It is worth noting that our method can not only generate diverse results from the same text prompt (Fig. 7(g)&(j), (h)&(k), and (i)&(l)), but also support to generate 3D scenes in some artistic styles (Fig. 8). Please refer to the supplementary material for video results.

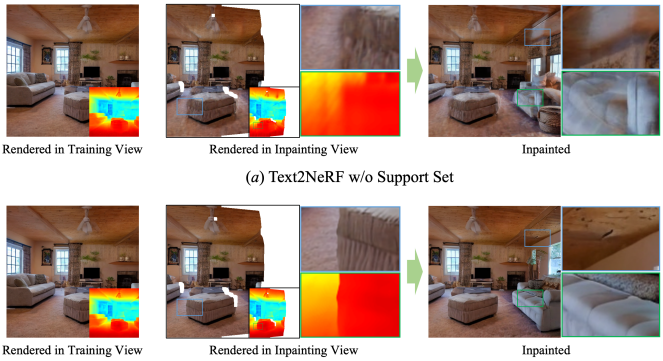
C. Ablation Studies

Ablation on PIU Strategy. To investigate the effectiveness of the PIU strategy in our pipeline, we conduct a comparative experiment by replacing it with the independent inpainting strategy. As shown in Fig. 9, in the absence of the PIU strategy, view-inconsistent inpainted views provide equal constraints on the content of the radiation field, which in turn produces significant artifacts in overlapping regions. By contrast, our PIU strategy enables the generation process to proceed view by view, effectively avoiding the view-inconsistent problem that may occur in the completion area.



(*a*) w/o PIU Strategy (*b*) w/ PIU Strategy

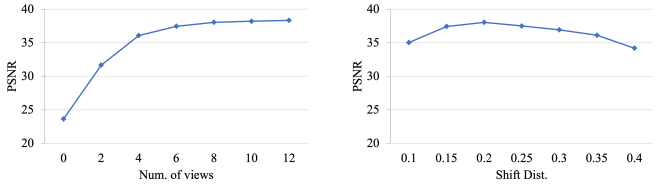
Fig. 9. Effectiveness validation of the PIU strategy. In the absence of the PIU (Progressive Inpainting and Updating) strategy, the missing regions in different views are independently inpainted, leading to noticeable artifacts in the final generated scene. However, by incorporating the PIU strategy, the generated scene is inpainted and updated in a view-by-view manner, ensuring view consistency and producing 3D scenes with distinct textures.



(*a*) Text2NeRF w/o Support Set

(*b*) Text2NeRF w/ Support Set

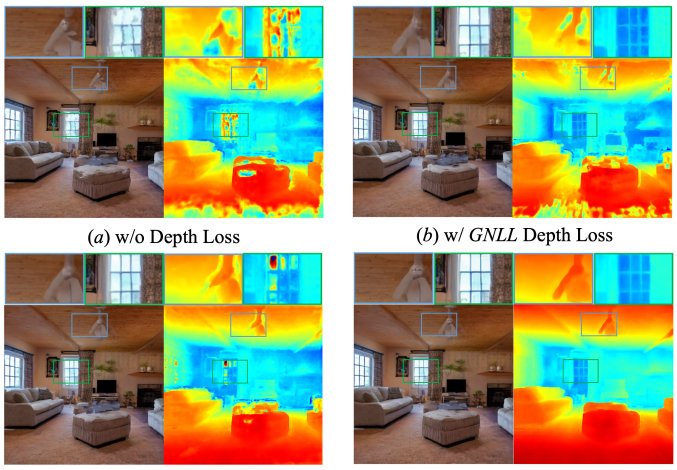
Fig. 10. Effectiveness validation of support set. Without the support set, although NeRF achieves good rendered image in the training view due to overfitting, it cannot produce a clear result in a novel inpainting view. By contrast, the case with support set enable to obtain images with desired quality in both training and inpainting views. Correspondingly, compared to the blurry rendering image, the clear one contributes to a better inpainted result.



(*a*) Quality curve for different number of views in the support set.

(*b*) Quality curve for different shift distance in the support set.

Fig. 11. Quality curves for different number of warping views and shift distance in the support set. Note that number 0 indicates the implementation without support set.



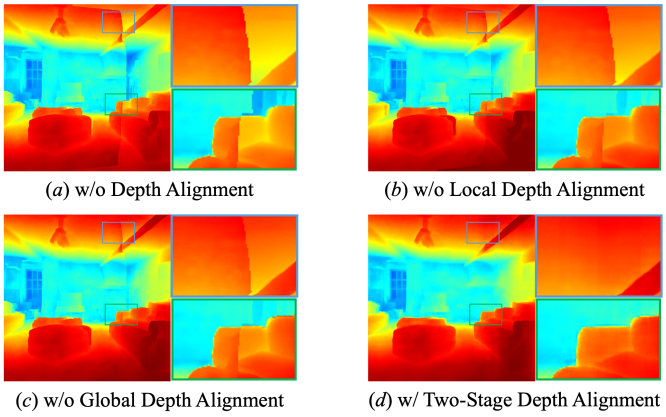
(*a*) w/o Depth Loss (*b*) w/ *GNLL* Depth Loss

(*c*) w/ *SSI* Depth Loss (*d*) w/ Our L_2 Depth Loss

Fig. 12. Effectiveness validation of our depth loss. Without the guidance of depth information, ambiguous depth values are produced in the near and far areas. In contrast, GNLL and SSI losses can constrain the depth values to a certain extent, but still cannot provide a strict constraint like our $L2$ depth loss.

Ablation on Support Set. To avoid overfitting and geometric ambiguity during single-view training of NeRF, we construct a support set for each view to provide multi-view constraints. Here, we further verify the effectiveness of the support set by removing this setting from our pipeline. As shown in Fig. 10, the radiance field in experiment (a) is trained under the constraint of a single initial view, i.e., without support set constraints. Obviously, the NeRF is overfitting in the training view and cannot produce clear results in the inpainting view, which further leads to poor inpainted results. By contrast, the case with a support set achieves high-quality rendering results in the inpainting view. Accordingly, a clear and concordant inpainted result can be estimated by the pre-trained diffusion model. Additionally, we design a series of experiments to determine the hyper-parameters of the support set, including the number of warping views ξ and shift distance ζ. Specifically, we use different number of warping views and shift distance to conduct the support set and initialize the NeRF model. Then, we calculate the PSNR values within valid pixels between the rendered images I_k^R and the DIBR-based warping results $I_k, M_k \leftarrow DIBR_{0\to i}$ to measure the quality of initialized NeRF: $psnr = \frac{1}{N_t}\sum_{k=1}^{N_t} 10\log_{10}\left(\left\|\left(I_k^R - I_k\right)\odot M_k\right\|_2\right)$, where N_t is the number of test poses. We generate 100 test poses using a generation method similar to the support set poses, i.e., randomly sample ζ in the range $[0.1, 0.4]$. As shown in Fig. 11(a), as the number of warped views increases, the benefit brought by the support set tends to saturate. To this end, we choose $\xi = 8$ warping views in the experiments to balance the computation cost and the training benefit of the support set. By changing the shift distance ζ of support sets, as shown in Fig. 11(b), we find that $\zeta = 0.2$ can make the support set achieve better performance than other parameters. Therefore, we set $\zeta = 0.2$ in all of our experiments.

Ablation on Depth Loss. Furthermore, to validate the effect of our depth loss, we compare our L_2 depth loss with the case without depth constraint and other two regularized depth losses, a Gaussian negative log likelihood (GNLL) [17] depth loss and a scale and shift invariant (SSI) [41] depth loss. Without the guidance of depth information, as shown in Fig. 12(a), the radiance field fails to synthesize novel views with plausible geometry and tends to produce ambiguous depth values in the near and far areas. In contrast, GNLL and SSI losses have better constraining effect on near or far depth, as shown in Fig. 12(b) and (c). Still, they fail to achieve satisfactory results because their constraints are weaker than our L_2 constraint (Fig. 12(d)). In fact, the depth information after alignment is view-consistent with the whole generated 3D



(*a*) w/o Depth Alignment (*b*) w/o Local Depth Alignment

(*c*) w/o Global Depth Alignment (*d*) w/ Two-Stage Depth Alignment

Fig. 13. Effectiveness validation of our two-stage depth alignment. In the absence of depth alignment, prominent demarcation lines arise due to depth disparities in the merged depth map. Global alignment helps bring the newly estimated depth values closer to the known depth map as a whole, but fails to eliminate the demarcation lines entirely. A comparison between (a) and (c) reveals that local alignment improves the alignment of unaligned depth maps, yet without global alignment, complete elimination of disparities remains challenging. In contrast, our two-stage strategy effectively achieves smoother transitions and harmonious results at the demarcation lines.

scene and can be directly seen as ground truth. In this case, a stricter objective function is more effective in constraining the generated scene than these flexible loss functions.

Ablation on Depth Alignment. Moreover, we conduct an ablation study on our two-stage depth alignment strategy. In Fig. 13(a), we present an example of scene generation without depth alignment, revealing noticeable demarcation lines caused by depth dislocations across different views. To address this issue, we introduce a two-stage depth alignment strategy. In the global alignment stage, we mitigate scale and distance disparities between known and newly predicted depth maps by computing the average scaling score and depth offset. Fig. 13(b) demonstrates the impact of global alignment, where the newly estimated depth values are pulled closer to the known depth map as a whole. However, due to the non-linear nature of depth estimation by a neural network, disparities among pixels do not vary linearly. Consequently, demarcation lines persist even with global alignment. In contrast, the local depth alignment fine-tunes a pretrained neural network to reduce local disparities among pixels. Comparing Fig. 13(a) and (c), we observe that local alignment partially brings unaligned depth maps closer in a non-linear manner. However, without global alignment, it is challenging to eliminate such disparities entirely. Therefore, we employ a two-stage depth alignment strategy to achieve smoother and more harmonious transitions at the demarcation lines, as depicted in Fig. 13(d).

V. Conclusion

In this paper, we propose the Text2NeRF for generating a wide range of 3D scenes with complicated structures and high-fidelity textures purely from a text prompt. We first leverage a pre-trained text-image diffusion model to generate an initial scene content and adopt a pre-trained monocular depth estimation model to provide geometric prior. Then, we initialize the radiance field of the scene according to the above information and update the 3D scene based on the PIU strategy. To avoid overfitting and geometric ambiguity during view-by-view updating, we introduce support sets to provide multi-view constraints for single-view training in NeRF. Moreover, we adopt depth and transmittance losses along with the RGB loss to achieve depth-aware NeRF optimization and propose a two-stage depth alignment strategy to eliminate depth disparity estimated in different views. Thanks to all well-designed modules and objectives, our Text2NeRF achieves to generate photo-realistic diverse 3D scenes with complex geometric structures and fine-fidelity textures.

Limitation. To generate 3D scenes in a large view range, we set the camera positions inside the radiance field and make the camera look outside. By this means, our method cannot generate an individual 3D object like other methods of setting the camera to look inside. To overcome this limitation, a flexible text-awarded camera setting strategy could be introduced in our framework in the future. Another limitation is that our method cannot handle dynamic 3D scene generation, which could be an interesting research direction.

References

[1] 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.

[2] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[3] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich, "Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation," *arXiv preprint arXiv:2212.00774*, 2022.

[4] J. Seo, W. Jang, M.-S. Kwak, J. Ko, H. Kim, J. Kim, J.-H. Kim, J. Lee, and S. Kim, "Let 2d diffusion model know 3d-consistency for robust text-to-3d generation," *arXiv preprint arXiv:2303.07937*, 2023.

[5] Z. Ye, M. Xia, Y. Sun, R. Yi, M. Yu, J. Zhang, Y.-K. Lai, and Y.-J. Liu, "3d-carigan: an end-to-end solution to 3d caricature generation from normal face photos," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[6] 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," *arXiv preprint arXiv:2211.10440*, 2022.

[7] N. Khalid, T. Xie, E. Belilovsky, and T. Popa, "Clip-mesh: Generating textured meshes from text using pretrained image-text models," *ACM Transactions on Graphics (TOG), Proc. SIGGRAPH Asia*, 2022.

[8] 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.

[9] 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.

[10] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022.

[11] R. Fridman, A. Abecasis, Y. Kasten, and T. Dekel, "Scenescape: Text-driven consistent scene generation," *arXiv preprint arXiv:2302.01133*, 2023.

[12] L. Höllein, A. Cao, A. Owens, J. Johnson, and M. Nießner, "Text2room: Extracting textured 3d meshes from 2d text-to-image models," *arXiv preprint arXiv:2303.11989*, 2023.

[13] 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.

[14] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.

[15] 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.

[16] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun, "Fov-nerf: Foveated neural radiance fields for virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3854–3864, 2022.

[17] B. Roessle, J. T. Barron, B. Mildenhall, P. P. Srinivasan, and M. Nießner, "Dense depth priors for neural radiance fields from sparse input views," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 892–12 901.

[18] 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.

[19] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4541–4550.

[20] R. Cai, G. Yang, H. Averbuch-Elor, Z. Hao, S. Belongie, N. Snavely, and B. Hariharan, "Learning gradient fields for shape generation," in *European Conference on Computer Vision*. Springer, 2020, pp. 364–381.

[21] P. Henzler, N. J. Mitra, and T. Ritschel, "Escaping plato's cave: 3d shape from adversarial rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9984–9993.

[22] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "Hologan: Unsupervised learning of 3d representations from natural images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7588–7597.

[23] Y. Xu, S. Peng, C. Yang, Y. Shen, and B. Zhou, "3d-aware image synthesis via learning structural and textural representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 430–18 439.

[24] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese, "Text2shape: Generating shapes from natural language by learning joint embeddings," in *Asian conference on computer vision*. Springer, 2018, pp. 100–116.

[25] H.-H. Lee and A. X. Chang, "Understanding pure clip guidance for voxel grid nerf models," *arXiv preprint arXiv:2209.15172*, 2022.

[26] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[27] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[28] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or, "Latent-nerf for shape-guided generation of 3d shapes and textures," *arXiv preprint arXiv:2211.07600*, 2022.

[29] R. Tucker and N. Snavely, "Single-view view synthesis with multiplane images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 551–560.

[30] M.-L. Shih, S.-Y. Su, J. Kopf, and J.-B. Huang, "3d photography using context-aware layered depth inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8028–8038.

[31] S. Niklaus, L. Mai, J. Yang, and F. Liu, "3d ken burns effect from a single image," *ACM Transactions on Graphics (ToG)*, vol. 38, no. 6, pp. 1–15, 2019.

[32] O. Wiles, G. Gkioxari, R. Szeliski, and J. Johnson, "Synsin: End-to-end view synthesis from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7467–7477.

[33] C. Rockwell, D. F. Fouhey, and J. Johnson, "Pixelsynth: Generating a 3d-consistent experience from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 104–14 113.

[34] R. Hu, N. Ravi, A. C. Berg, and D. Pathak, "Worldsheet: Wrapping the world in a 3d sheet for view synthesis from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 528–12 537.

[35] Y. Nehmé, F. Dupont, J.-P. Farrugia, P. Le Callet, and G. Lavoué, "Visual quality of 3d meshes with diffuse colors in virtual reality: Subjective and objective evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2202–2219, 2020.

[36] C. Fehn, "Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv," in *Stereoscopic displays and virtual reality systems XI*, vol. 5291. SPIE, 2004, pp. 93–104.

[37] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[38] X. Luo, J.-B. Huang, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," *ACM Transactions on Graphics (ToG)*, vol. 39, no. 4, pp. 71–1, 2020.

[39] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," *arXiv preprint arXiv:2203.09517*, 2022.

[40] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger, "Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2732–2742, 2023.

[41] K. Sargent, J. Y. Koh, H. Zhang, H. Chang, C. Herrmann, P. Srinivasan, J. Wu, and D. Sun, "Vq3d: Learning a 3d-aware generative model on imagenet," *arXiv preprint arXiv:2302.06833*, 2023.

[42] 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.

[43] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9685–9694.

[44] 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.

[45] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on image processing*, vol. 21, no. 12, pp. 4695–4708, 2012.

[46] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.