BerfScene: Bev-conditioned Equivariant Radiance Fields for Infinite 3D Scene Generation

Qihang Zhang[1] Yinghao Xu[2] Yujun Shen[3] Bo Dai[4] Bolei Zhou[5†] Ceyuan Yang[4†]
[1]CUHK [2]Stanford [3]Ant Group [4]Shanghai AI Laboratory [5]UCLA

Abstract

Generating large-scale 3D scenes cannot simply apply existing 3D object synthesis technique since 3D scenes usually hold complex spatial configurations and consist of a number of objects at varying scales. We thus propose a practical and efficient 3D representation that incorporates an equivariant radiance field with the guidance of a bird's-eye view (BEV) map. Concretely, objects of synthesized 3D scenes could be easily manipulated through steering the corresponding BEV maps. Moreover, by adequately incorporating positional encoding and low-pass filters into the generator, the representation becomes equivariant to the given BEV map. Such equivariance allows us to produce large-scale, even infinite-scale, 3D scenes via synthesizing local scenes and then stitching them with smooth consistency. Extensive experiments on 3D scene datasets demonstrate the effectiveness of our approach. Our project website is at: https://zqh0253.github.io/BerfScene/.

1. Introduction

The advancement in implicit and explicit 3D representations has driven the rapid progress in high-quality 3D object generation [3, 12, 14, 33, 38, 42, 50, 58]. However, directly applying object synthesis methods to 3D scene generation poses challenges due to inherent variations in spatial scales and composited objects within 3D scenes. Considering that urban architects construct city scenes, they won't place building randomly but always starts from a detailed map, serving as a foundational guide outlining the spatial configurations of blocks and buildings. This highlights the need for a suitable representation tailored for 3D scenes, capable of streamlining the scene generation process.

A well-structured scene representation must capture spatial relationships between objects and have the flexibility to scale up, facilitating the generation of scenes on a large or infinite scale. Previous approaches often relied on scene graphs [6, 19, 39, 52] for representation, containing rich object relations but facing limitations in processing due to unstructured topology. Recent work DiscoScene [59] proposes representing scenes with a set of 3D bounding boxes. However, despite offering a volumetric depiction of objects, it introduces complexity in interpreting the entire scene and faces scalability challenges.

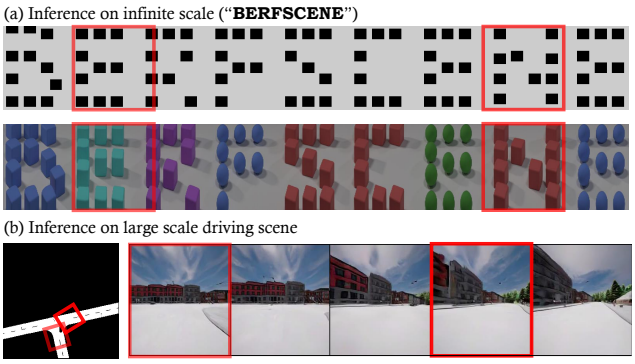

Figure 1. **BerfScene focuses on unbounded 3D scene synthesis.** Above: a CLEVR scene labeled "BERFSCENE". Below: a driving scenario before and after executing a right turn.

To overcome this, we choose a 2D bird's-eye-view (BEV) map to describe the scene structure, providing a practical and efficient way to represent and analyze spatial information, thereby simplifying the scene generation process. Concretely, BEV map could specify the composition and scales of objects clearly. Generating a large-scale scenes could be thus formulated as synthesizing local scenes first and then composing them together. However, composing the local blocks into a coherent global scene in 3D space always leads to the severe artifacts such as jittering and inconsistency, as BEV maps can be ambiguous to fine-grained semantics *i.e.*, primarily deliver a global layout and locations of objects but lack insights into the detailed visual appearance of the objects.

To avoid the ambiguity of BEV maps, recent attempts like InfiniCity [28] and SceneDreamer [5] incorporate explicit 3D structures (e.g., voxels) as a hard constraint to ensure the continuity of the composition process. How-

† Corresponding authors

ever, collecting and loading large-scale 3D structures always pose the computational overhead and inefficiency. Alternatively, we tackle this issue by integrating the equivariance property with a carefully-designed architecture into the BEV-conditioned representation. The consistency across various local scene generation is accordingly enhanced.

By introducing BEV-conditioned Equivariant Radiance Fields based on such representation, we present `BerfScene`, a framework allowing for large-scale 3D scene synthesis and flexible editing of camera pose and composite objects, as shown in Fig. 1. Our generator is conditioned on a local BEV patch to learn the entire scene's distribution, utilizing a specific network architecture to maintain equivariance across the same semantic regions in different BEV maps. This design incorporates extra padding and low-pass filters [24, 62] in the generator to reduce aliasing, ensuring consistent synthesis reflecting specified spatial configurations under any local coordinates. Thanks to the equivariance of the BEV-conditioned representations, our method can learn from 2D images showing scenes with limited spatial extent, while also being capable of generating infinite-scale 3D scenes. We evaluate our method on 3D scene datasets including CLEVR [18], 3D-Front [10, 11], and Carla [9]. Through qualitative and quantitative experiments, we demonstrate that our method achieves state-of-the-art performance in generating large-scale 3D scenes.

2. Related work

3D-aware image synthesis. We have witnessed amazing progress in image generation with 2D GANs [13, 21–24]. Recent works lift 2D GANs with 3D inductive bias for 3D-aware generation from unstructured single-view image collections. Early works leverage the voxel [33, 43, 63], mesh [12, 55], depth [47] or 2D feature plane [50, 51] to explicitly model object structure, but suffer from poor visual fidelity and geometric consistency. Another line of research integrates the neural radiance fields [2, 7, 40, 42, 48, 53, 57] into the GAN generator to alleviate these limitations. Recently, diffusion models have been used to synthesize 3D-aware images by distilling knowledge from large pretrained text-to-image models [41] or by training from scratch with direct 3D supervision [16, 20, 35, 37, 49] or adopt image-to-image translation framework using view conditioning [4, 15, 27, 29–31, 60]. However, these methods primarily focus on object modeling and have limited capacities in generating large-scale scenes, which our method focuses on.

3D scene generation. Although 3D aware-image and object generation has been significantly advanced in recent years, 3D scene generation remains a challenging task since generating a 3D scene usually considers the composition of objects and their corresponding scales. To tackle these issues, recent attempts explore to leverage existing prior like layout [34, 36, 59, 61], grid plane [8, 26], depth maps [45, 46, 51], or voxels [5, 28] to generate 3D scenes. We follow this philosophy yet incorporate a BEV-map as the conditions as it enables the flexible division of a large-scale scenes *i.e.*, specifies the scene configuration clearly. A very related work CC3D [1] shares similarities with our approach in utilizing a bird's-eye-view (BEV) map as a conditioned layout for generating scene radiance fields. However, CC3D is limited in its ability to generate infinite 3D scenes due to its lack of composition modeling. In contrast, our model overcomes this limitation by employing an equivariant representation conditioned on BEV maps, enabling seamless composition and facilitating the generation of infinite-scale scenes.

3. Method

`BerfScene` employs a BEV map as an input to specify a scene and generates a radiance field conditioned on the BEV representation, which is then used for image synthesis through volume rendering. To support large-scale scene generation, the BEV-conditioned radiance field is further extended into an equivariant representation through a carefully designed feature extractor. We first introduce preliminary knowledge about volume rendering in Sec. 3.1. In Sec. 3.2, we discuss the design of the equivariant representation. Sec. 3.3 describes the scene generation framework, including implementation, training, and inference details.

3.1. Preliminaries

The neural radiance field [32] has gained tremendous popularity among recent works in view synthesis and image generation. Specifically, to render an image given a camera viewpoint, multiple rays are cast out, with N points $\{p_i | i = 1, \cdots, N\}$ sampled along each ray r. For each point $p_i = (x_i, y_i, z_i)$, we query its color c_i and density σ_i:

$$c_i, \sigma_i = \Theta(f(p_i), d), \tag{1}$$

where $f(p_i)$ is the encoding feature of p_i, d is the ray direction, and Θ is parameterized as a Multi-Layer Perceptron (MLP). The color of the ray $C(r)$ is further calculated as the weighted average of each point's color:

$$C(r) = \sum_{i=1}^{N} \left[\prod_{j=1}^{i} e^{(-\sigma_j \delta_j)} \cdot (1 - e^{(-\sigma_i \delta_i)}) \right] c_i, \tag{2}$$

where δ_i is the length of the i-th interval on the ray.

As for $f(\cdot)$, there are different design choices to encode each single point, like positional embedding: $f(p_i) = (\mathrm{pe}(x_i), \mathrm{pe}(y_i), \mathrm{pe}(z_i))$, and sampled feature from 2D feature map: $f(p_i) = (\Phi(U_{xy}, x_i, y_i), \Phi(U_{xz}, x_i, z_i), \Phi(U_{yz}, y_i, z_i))$, where U_{xy}, U_{xz}, U_{yz} denote learnable 2D feature map and Φ denotes feature sampling operation.

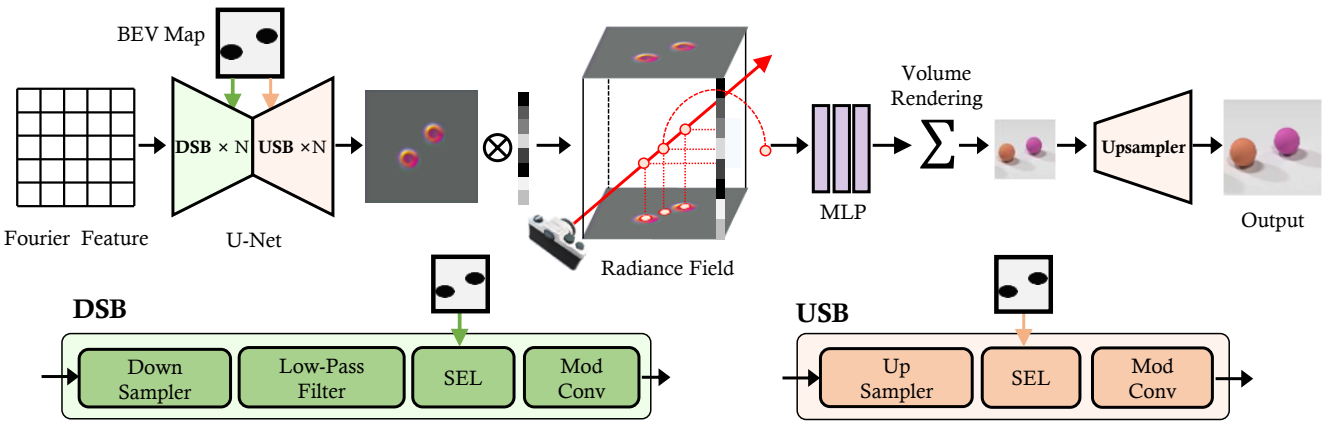

Figure 2. **Illustration of `BerfScene`:** A U-Net takes the fourier feature as input and gradually down-samples (DSB) and up-samples (USB) the features. The internal features would be spatially modulated by the BEV maps via SEL block, resulting in a BEV-conditioned radiance field. With the anti-aliasing design (*e.g.*, low-pass filters), the entire synthesis pipeline becomes equivariant to the BEV maps.

Recent works additionally sample latent code s and incorporate it into the encoding feature $f(\cdot)$ for 3D-aware image generation [2, 7, 42]. For example, EG3D [3] encodes each point feature as: $f(p_i) = (\Phi(U_{xy}(s), x_i, y_i), \Phi(U_{xz}(s), x_i, z_i), \Phi(U_{yz}(s), y_i, z_i))$, where $U_{xy}(s), U_{xz}(s), U_{yz}(s)$ are generated 2D feature map conditioned on latent code s. Our work follows this line of works.

3.2. Equivariant BEV-conditioned representation for radiance field

Given that various scenes (*e.g.*, traffic scenes) could be represented by a ground plan, we propose to leverage Bird-Eye-View (BEV) map to steer the generation of the radiance field. We also improve the equivariance of the representation for large-scale scene synthesis. In this section, we will provide a detailed explanation of our design.

BEV-conditioned radiance field. In order to incorporate the prior information provided by the BEV map, we introduce a generator U that generates a conditioned 2D feature map. The network architecture of U is a U-Net architecture with StyleGAN blocks. As illustrated in Fig. 2, the generator takes a 2D Fourier feature map γ as input and progressively modifies the feature map using sequential encoders and decoders, which are modulated by a randomly sampled latent code s and the BEV map $\mathcal{B}$.

We incorporate the 2D Fourier feature map γ as the input to provide positional information for local radiance field. It is defined on a positional grid $\mathbf{v}$ that spans the global coordinates. Each position in the grid is associated with a specific value.:

$$\gamma(\mathbf{v}) = \big[a_1 \cos\left(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{v}\right), a_1 \sin\left(2\pi \mathbf{b}_1^{\mathrm{T}} \mathbf{v}\right), \ldots, a_m \cos\left(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{v}\right), a_m \sin\left(2\pi \mathbf{b}_m^{\mathrm{T}} \mathbf{v}\right)\big]^{\mathrm{T}}, \tag{3}$$

where $a_i, \mathbf{b}_i$ denotes predefined amplitudes and Fourier basis frequencies. Each subsequent encoder or decoder uses Spatial Encoding Layer (SEL) [54] to incorporate the BEV map $\mathcal{B}$. Concretely, given an intermediate feature map a, one block in U-Net operates as:

$$a' = \texttt{SEL}(\mathcal{T}(a), \mathcal{T}(E(\mathcal{B}))), \tag{4}$$

$$a'' = \texttt{ModConv}(a', s), \tag{5}$$

where E is an encoder with two convolutional layers that extracts BEV's feature map, $\mathcal{T}(\cdot)$ denotes the interpolation operation that resizes the two feature maps, and `ModConv` performs the modulated convolution [23] to further modify the features based on the latent code s.

The output feature map of the U-Net U is then lifted to 3D by computing cross product with the positional embedding of coordinate z: $U(\mathcal{B}, \gamma, s) \times \{\text{pe}(Z)\}$. Consequently, color and density can be obtained via:

$$c, \sigma = \Theta(\Phi(U(\mathcal{B}, \gamma, s) \times \{\text{pe}(Z)\}, x, y, z)), \tag{6}$$

where Φ is the feature sampling operation, and Θ is the MLP that takes sampled features as the input.

Equivariant property. Building upon previous designs, we have the capability to create scenes according to the BEV map. As a result, this allows us to synthesize a scene of infinite scale in a divide-and-conquer fashion, *i.e.*, dividing a global map into local patches, generating local scenes, and composing them together. However, the BEV conditioned radiance field can result in uncertainty in terms of fine-grained details. This uncertainty may lead to the synthesis inconsistency since the same objects may appear in multiple local scenes, substantially deteriorating the quality when composing several local scenes for large-scale scene synthesis. We thus seek for the guarantee of the equivariant property.

In particular, regular convolutions with padding and down-sampling tend to cause aliasing [24, 62], *i.e.*, the synthesized concepts are strongly related to their coordinates.

Considering this, we carefully design the operations in U-Net to maintain the equivariance to a maximum extent. 1) BEV with a wide margin: as border padding would leak the absolute positional information to the internal representations [17, 25, 56], we follow [24] to leave a large margin around BEV map to ensure the representation unimpeded by padding. 2) Low-pass filters: it is inevitable to downsample the internal feature maps, for the sake of memory efficiency. According to Nyquist Law [44], the representation capacity of a regularly sampled signal is bound by half of the sampling rate. Otherwise, excessive signals can cause aliasing. Therefore, we introduce the low-pass filter before down-sampling to restrict the representation within a reliable region. The transform for the downsample becomes

$$\mathcal{T}(\cdot) = \texttt{Low-Pass}(\cdot) \circ \texttt{Interp}(\cdot), \tag{7}$$

where the low pass filter is designed as a finite impulse response (FIR) filter. With this equivariant property, generating large scale scenes becomes simply composing multiple local scenes, with the consistent concepts.

3.3. Scene generation framework

With the equivariant BEV-conditioned representation designed above, we now introduce `BerfScene`, the proposed method for infinite 3D scene generation.

Generator. The generator consists of a U-Net encoder that produces the spatial feature map for the volume rendering in Eq. (6). Concretely, this U-Net encoder takes Fourier feature as input, where the internal features would be modulated via the latent code. Besides, BEV map $\mathcal{B}$ would be incorporated into this encoder through the SEL, which could further guide the spatial configurations of the final synthesis. As internal feature maps are gradually down-sampled, we apply the low-pass filters to remove the excessive frequencies, improving the equivariance of theis encoder. To this end, the output feature map of this unet manages to correctly and equivariantly reflect the spatial structure determined by BEV maps. Given this feature maps, images would be obtained through the neural rendering.

Discriminator. We follow the dual-discriminator design of EG3D [3]. A bi-linearly upsampled version of the rendered image is concatenated with the super-resolved version. The discriminator takes as input the resulted six-channel image.

Training objectives. During training, style code s is randomly sampled from Gaussian distribution. BEV map $\mathcal{B}$ and camera pose γ are randomly sampled from the dataset. We optimize traditional adversarial loss $\mathcal{L}_{adv}$, R_1 regularization loss $\mathcal{L}_{R_1}$, and density regularization loss $\mathcal{L}_{density}$ as proposed in [3]. The overall training target is a weighted sum of the above loss terms:

$$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{R_1}\mathcal{L}_{R_1} + \lambda_{density}\mathcal{L}_{density}, \tag{8}$$

where $\lambda_{adv}, \lambda_{R_1}, \lambda_{density}$ are weighted coefficients.

Inference of infinite-scale synthesis. Rather than generating high-quality local scene images, `BerfScene` supports scene generation at an infinite scale. After defining a global BEV map, we divide it into several local BEVs and render images conditioned on them. One can get a progressively moving video by continuously cropping local BEVs. In addition, broad-view image can be generated by stitching rendered results. We also adopt supersampling anti-aliasing (SSAA) to perform ray marching at a temporary higher resolution and downsample the feature map to the original resolution. SSAA suppresses aliasing effect and provides better visual quality.

4. Experiments

We evaluate `BerfScene` on diverse datasets, and compare it with baseline methods of both image generation and 3D-aware image generation.

4.1. Settings

Datasets. We conduct experiments on three datasets: CLEVR [18], 3D-Front [10, 11], and Carla [9]. CLEVR is a multi-object dataset with a 3D rendering engine. We use the official script to render images for training and evaluation. The camera position is fixed in the global coordinate. For each scene, we randomly place 3 to 8 objects with various colors and shapes. We collect 80,000 images in 256×256 resolution. 3D-Front is a 3D indoor scene dataset with diverse furniture including bed, wardrobe, *etc*. We randomly place the camera and collect 50,000 images in 256×256 resolution on this dataset, covering 2535 different scenes in total. Carla is a driving simulator with realistic visual appearance. It covers different weather conditions, and diverse road environments (from rural to urban). We collect 28,000 frames in 256×256 resolution.

Metrics. Following the prior, we use the Frechet Inception Distance (FID) as a quantitative metric to evaluate the quality of our image synthesis results. We sample 50K real images and 50K generated samples to compute the FID score. Additionally, we measure the consistency of the same scene under different local coordinates to test composition feasibility. Since it is challenging to directly compare generated 3D scenes, we approximate the scene using the rendered image $G(\mathcal{B}, z)$. Following [24], we report the peak signal-to-noise ratio (PSNR) in decibels (dB) between two sets of images obtained by translating the input and output by a random amount:

$$\text{EQT} = 10 \cdot \log\left(\frac{I_{max}^2}{\mathbb{E}_{s,x}(\|G(t_x[\mathcal{B}], s) - t_x[G(\mathcal{B}, s)]\|)}\right), \tag{9}$$

where $t_x[\cdot]$ stands for translation operation by x margin, and the intended dynamic range of generated images from -1 to $+1$ gives $I_{max} = 2$.

Table 1. **Evaluation with baselines.** FID and EQT are reported as evaluation metrics. Note that we highlight the best results among 3D-aware models.

Method	CLEVR		Front-3D		Carla
	FID (↓)	EQT (↑)	FID (↓)	EQT (↑)	FID (↓)
StyleGAN2	6.95	-	31.01	-	16.89
GSN	-	-	130.70	-	-
EG3D	4.67	-	80.70	-	46.8
CC3D	3.61	21.94	42.88	14.74	45.2
Ours	**0.96**	**22.02**	**36.78**	**15.76**	**40.7**

Baseline. We compare our method to both 2D and 3D GANs. Specifically, we evaluate our approach against StyleGAN2, EG3D, and GSN to explore the impacts of introducing inductive biases, such as equivariance, on image quality. Additionally, we assess our capacity of model for generating compositional 3D scenes using CC3D, which is a scene generation framework conditioned on BEV.

Implementation details. We follow the architecture design of EG3D except our equivariant BEV-conditioned generator. To determine the best R_1 regularization weight, we performed a grid search across various datasets and methods. The values of R_1 regularization weight used in our experiments are available in the supplementary material. All other hyper-parameters were kept the same as EG3D. We conducted all experiments on 8×A100 GPUs with a batch size of 64. More details can be found in the supplementary material.

4.2. Generation Results

Qualitative results. In Fig. 3, we present results of local and global scene synthesis from our method and the baselines. For local scene synthesis, StyleGAN2, as a 2D image generator, cannot support the explicit camera control. On the contrary, we show two different views of one single scene for EG3D, CC3D, and ours.

When tested on the CLEVR dataset, StyleGAN2 fails to generate consistent object appearances. In the first example of StyleGAN2, the generated cylinder has a mixed color that is not present in the dataset. Both EG3D and CC3D suffer from blurry results, with slight blurs found in the generated output and twisted edges can be seen in CC3D's results. In contrast, our method consistently produces high-fidelity images and also supports excellent camera control, as evidenced by the consistent results across different camera angles. On the 3D-Front dataset, StyleGAN2 generates indoor scenes with high fidelity. EG3D fails to generate consistent results as the texture and shape vary across different camera poses. CC3D generates inaccurate shapes for small objects like nightstands and is leaning to generate blurry textures. In contrast, our method can generate indoor furniture with decent and consistent appearance across different camera views, demonstrating the effectiveness of our proposed scene representation.

Since both CC3D and our method are conditioned on BEV maps, we can continuously roll out BEV patches and generate and compose local scenes for global scene synthesis. We test the capacity of large scene synthesis on CLEVR and Carla. For CLEVR, CC3D generates transient color of a single object and blurry edges, indicating fractional shaking across local patches. Our method can generate a high-fidelity global scene without any inconsistencies or blurs. For Carla, our method can generate high quality driving videos with consistent visual appearance and 3D geometry of buildings and trees. Yet CC3D produces flickering frames with severe inconsistency.

Quantitative evaluations. Tab. 1 reports the quantitative results (FID and EQT) over different methods*. On CLEVR, `BerfScene` achieves a FID score of 0.96, a far better result compared to other methods. Regarding 3D-Front, our method also gains a significant lead among all 3D GANs. Additionally, our method consistently outperforms other 3D GANs in terms of EQT, demonstrating that our approach not only generates realistic 3D scene images but also enjoys good equivariance. This property is essential for composing local scenes into a large-scale scene, making our method a promising solution for generating 3D scenes of arbitrary scales.

4.3. Ablation Study

To better understand the individual contributions, we ablate main components by comparing quantitative metrics and qualitative large-scale scene synthesis results.

Radiance field representation design. To guide the generation process using BEV maps, we incorporated the Spatial Encoding Layer (SEL) into our generator to fuse the BEV. The output BEV feature map is further extended by positional embedding over the coordinate z to create the radiance field representation. We compare this design to the triplane representation and 2D-to-3D extrusion proposed by [1]. To ensure a fair comparison, all designs share the same backbone, with the last convolutional layer having different output channels. Our design output 32 channels, while the triplane representation triples the channel number, and the 2D-to-3D extrusion produces $32 \times N$ channels, where N is the number of height dimension channels. In Tab. 2, worse performance on FID and EQT is observed for both triplane and extruded plane designs. In Fig. 4, the generated global scenes with these two designs also suffer from severe artifacts.

Padding BEV. To analyze how additional padding suppresses aliasing, we compare models trained on BEVs with and without padding. As can be seen in Tab. 3, EQT drops by a large margin. This result indicates that positional in-

*We failed to train GSN on CLEVR with the official implementation, hence we do not report the quantitative results.

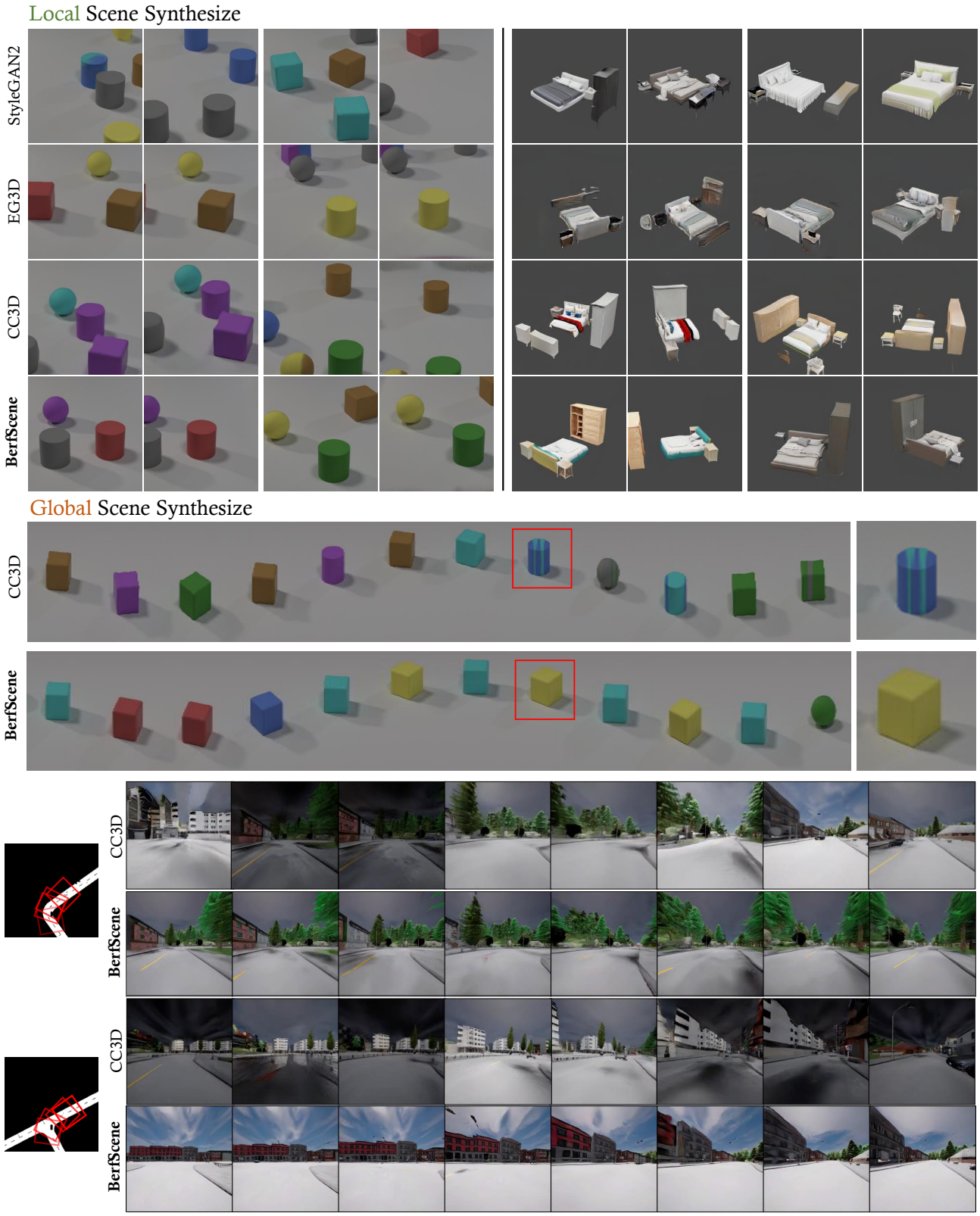

Figure 3. **Qualitative results of Local Scene Synthesize** in 256 × 256 resolution on various datasets, and **Global Scene Synthesis** on CLEVR.

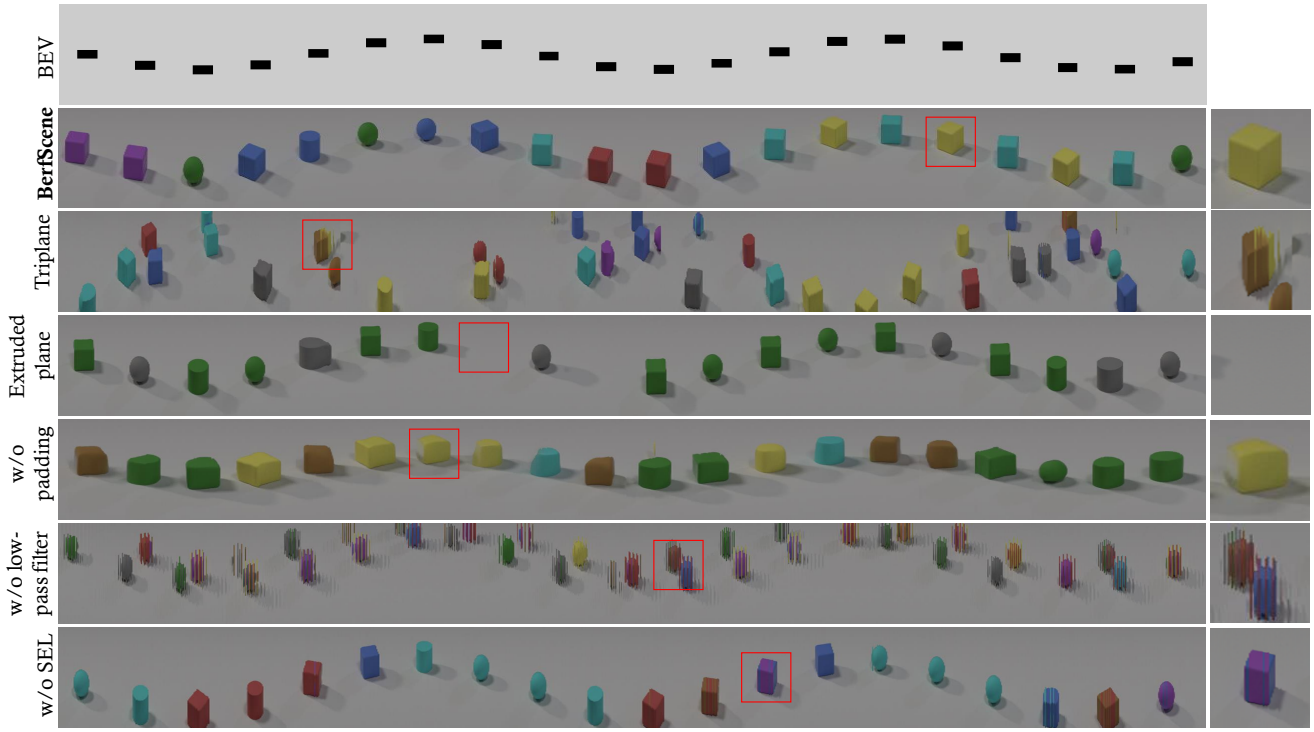

Figure 4. **Qualitative comparison for ablations** on large-scale scene synthesis.

Table 2. **Ablation study over different backbone design choices.**

Configuration	CLEVR		Front-3D	
	FID (↓)	EQT (↑)	FID (↓)	EQT (↑)
Triplane	18.11	19.58	39.17	14.10
Extruded plane	5.60	20.13	50.40	15.29
Ours	**0.96**	**22.02**	**36.78**	**15.76**

Table 3. **Ablation study over design components.**

Configuration	CLEVR		Front-3D	
	FID (↓)	EQT (↑)	FID (↓)	EQT (↑)
w/o padding BEV	2.50	19.01	51.30	13.32
w/o low-pass filter	5.53	18.19	36.87	14.45
w/o SEL	6.27	22.00	45.90	15.41
Ours	**0.96**	**22.02**	**36.78**	**15.76**

formation leaks into the generator and disrupts the equivariance property, limiting models for large-scale scene generation (also see wierd shapes in Fig. 4).

Low-pass filters. Beside padding in CNNs, aliasing could also be caused by excessive high frequency noise after down sampling layers. We study whether low-pass filter helps alleviate it in our 3D generation scenario. After removing low-pass filters in the network, EQT goes down, with intense discontinuity observed in the generated global scene, demonstrating that low-pass filters are essential to guarantee the equivariance property.

SEL layer. In our U-Net backbone, BEV map is repeatedly fused into the feature map through SEL layer to achieve precise layout control. An alternative choice is to directly feed BEV into the backbone. As shown in Tab. 3, FID increases by a large margin compared to our method with SEL. We hypothesize that repeated SELs could make the best of the geometry guidance from BEV, and thus generates scenes with more realistic and relevant spatial configurations.

5. Applications of BerfScene

5.1. Infinite scene generation

Our method can generate large-scale, even infinite, scenes, by dividing a global scene into local patches, generating and then seamlessly composing them. Concretely, we use sliding window to get continuous local BEV maps. These maps serve as the conditioning input for generating a *navigating video*. Then, we extract the middle vertical line from each frame in the video and stack them to form a holistic view of the entire scene. We demonstrate generated large-scale scenes with various layouts in Fig. 4.

5.2. Scene editing

Our generator is conditioned on the BEV map, thus it is easy to edit the scene by varying the input BEV map. In Fig. 5, we demonstrate different scene editing results including **1) translation**, a user can rearrange objects' layout; **2) restyling**, a user can directly modify single object's semantic to achieve restyling; **3) removal and insertion**, a user can delete or copy objects from the scene.

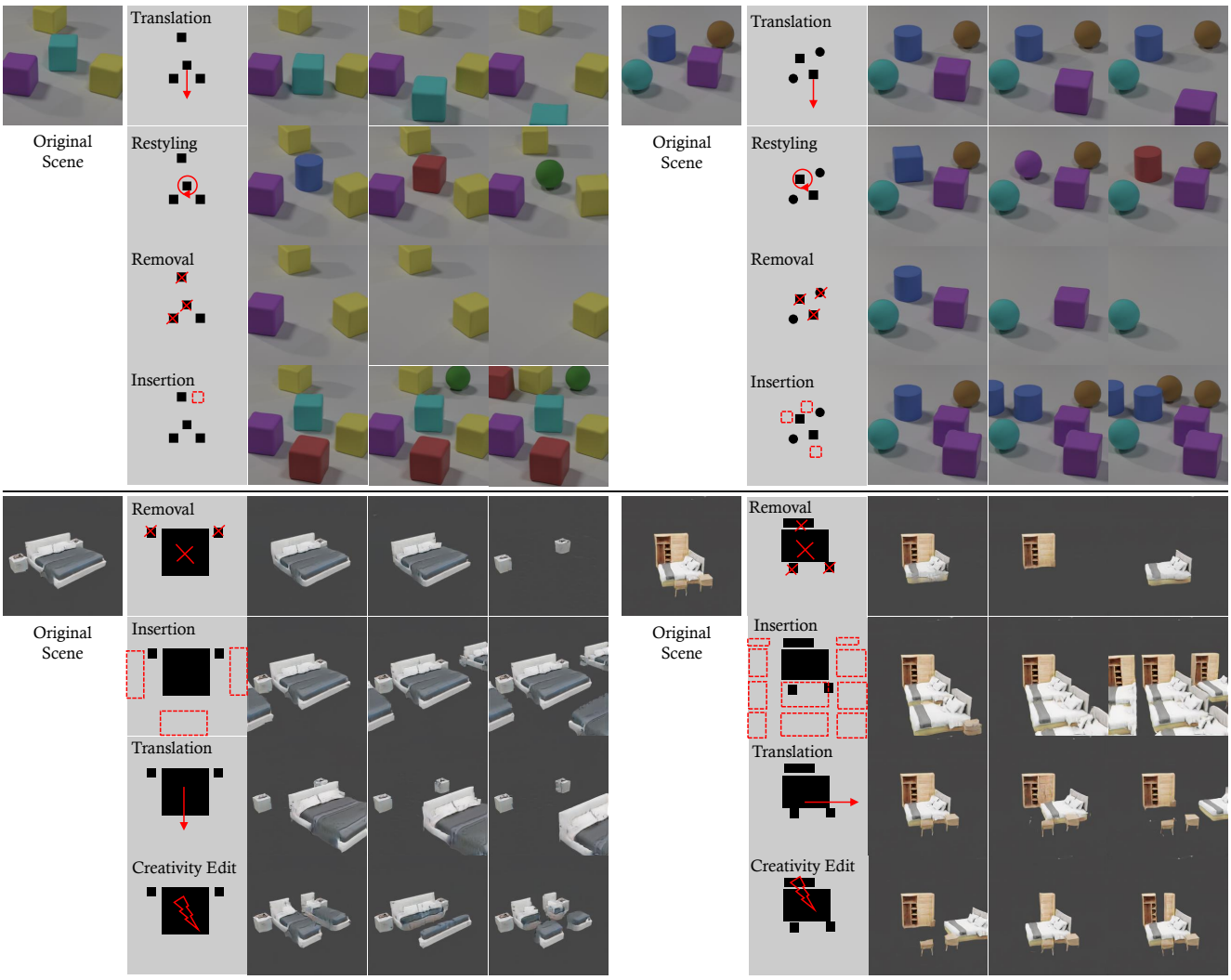

Figure 5. **Controllable 3D scene synthesis in** 256×256 **resolution.** We perform versatile user control on the scene objects by varying BEV map, such as translation, restyling, removal, insertion.

6. Discussion

Limitations. Although infinite-scale scene generation has been enabled, there remain several limitations we would like to discuss. First, as we follow the generative radiance field that mainly learns from the training set, the view of camera for inference is quite limited for large-scale scene synthesis. Collecting data with more diverse observations may help alleviate this issue. Second, current design only supports the static scene generation. How to enable the large-scale dynamic scene synthesis remains open for future work. Furthermore, it is important to note that our method may encounter challenges in achieving precise attribute control due to the absence of explicit supervision. For instance, when specifying a particular color in the BEV map, the synthesized output may exhibit a different color. This could potentially be enhanced by incorporating CLIP supervision.

Conclusion. This work introduces `BerfScene` that can generate 3D scenes of arbitrary scales. We propose a BEV-conditioned radiance field to represent a 3D scene. This approach enables users to directly steer the generated spatial configurations via BEV maps. To ensure smooth and consistent composition of multiple scenes, we further ensure the equivariance of the BEV-conditioned representations. We introduce several architectural designs, including a wider margin and low-pass filters, to achieve this goal. As a result, we can synthesize infinite-scale scenes by simply composing multiple syntheses controlled by local BEV maps. Experimental results on various 3D scene datasets demonstrate the effectiveness of our proposed method.

References

[1] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. Cc3d: Layout-conditioned generation of compositional 3d scenes. *arXiv preprint arXiv:2303.12074*, 2023. 2, 5

[2] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 2, 3

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 1, 3, 4

[4] Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. *ICCV*, 2023. 2

[5] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Scenedreamer: Unbounded 3d scene generation from 2d image collections. In *arXiv*, 2023. 1, 2

[6] Steve Cunningham and Michael J Bailey. Lessons from scene graphs: using scene graphs to teach hierarchical modeling. *Computers & Graphics*, 2001. 1

[7] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *CVPR*, 2022. 2, 3

[8] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *ICCV*, 2021. 2

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 2, 4, 12

[10] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *CVPR*, 2021. 2, 4, 12

[11] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *Int. J. Comput. Vis.*, 2021. 2, 4, 12

[12] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *NeurIPS*, 2022. 1, 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[14] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In *Int. Conf. Learn. Represent.*, 2022. 1

[15] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *ICML*, 2023. 2

[16] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 2

[17] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*, 2020. 4

[18] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 2, 4, 12

[19] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 1

[20] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Int. Conf. Learn. Represent.*, 2018. 2

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[23] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 3

[24] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021. 2, 3, 4

[25] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *CVPR*, pages 14274–14285, 2020. 4

[26] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8496–8506, 2023. 2

[27] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *https://arxiv.org/abs/2311.06214*, 2023. 2

[28] Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. InfiniCity: Infinite-scale city synthesis. *arXiv preprint arXiv:2301.09637*, 2023. 1, 2

[29] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928*, 2023. 2

[30] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023.

[31] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[33] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019. 1, 2

[34] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. BlockGAN: Learning 3D object-aware scene representations from unlabelled images. In *NeurIPS*, 2020. 2

[35] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[36] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[37] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc Van Gool, and Sergey Tulyakov. Autodecoding latent 3d diffusion models. *arXiv preprint arXiv:2307.05445*, 2023. 2

[38] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *CVPR*, 2022. 1

[39] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, pages 2856–2865, 2021. 1

[40] Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. In *NeurIPS*, 2021. 2

[41] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[42] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 1, 2, 3

[43] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. *arXiv preprint arXiv:2206.07695*, 2022. 2

[44] Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. 4

[45] Yuan Shen, Wei-Chiu Ma, and Shenlong Wang. Sgam: Building a virtual 3d world through simultaneous generation and mapping. *NeurIPS*, 35:22090–22102, 2022. 2

[46] Zifan Shi, Sida Peng, Yinghao Xu, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey. *arXiv preprint arXiv:2210.15663*, 2022. 2

[47] Zifan Shi, Yujun Shen, Jiapeng Zhu, Dit-Yan Yeung, and Qifeng Chen. 3d-aware indoor scene synthesis with depth priors. In *ECCV*, pages 406–422. Springer, 2022. 2

[48] Zifan Shi, Yinghao Xu, Yujun Shen, Deli Zhao, Qifeng Chen, and Dit-Yan Yeung. Improving 3d-aware image synthesis with a geometry-aware discriminator. *NeurIPS*, 2022. 2

[49] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, 2023. 2

[50] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022. 1, 2

[51] Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 3d generation on imagenet. *arXiv preprint arXiv:2303.01416*, 2023. 2

[52] Henry Sowizral. Scene graphs in the new millennium. *IEEE Computer Graphics and Applications*, 2000. 1

[53] Ayush Tewari, Xingang Pan, Ohad Fried, Maneesh Agrawala, Christian Theobalt, et al. Disentangled3d: Learning a 3d generative model with disentangled geometry and appearance from monocular images. In *CVPR*, 2022. 2

[54] Jianyuan Wang, Ceyuan Yang, Yinghao Xu, Yujun Shen, Hongdong Li, and Bolei Zhou. Improving gan equilibrium by raising spatial awareness. In *CVPR*, pages 11285–11293, 2022. 3, 12

[55] Zhangyang Xiong, Di Kang, Derong Jin, Weikai Chen, Linchao Bao, and Xiaoguang Han. Get3dhuman: Lifting stylegan-human into a 3d generative model using pixel-aligned reconstruction priors. *arXiv preprint arXiv:2302.01162*, 2023. 2

[56] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *CVPR*, pages 13569–13578, 2021. 4

[57] Xudong Xu, Xingang Pan, Dahua Lin, and Bo Dai. Generative occupancy fields for 3d surface-aware image synthesis. In *NeurIPS*, 2021. 2

[58] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, 2022. 1

[59] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. *CVPR*, 2023. 1, 2

[60] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Wang Peng, Jihao Li, Zifan Shi, Kaylan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Zhang Kai. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arxiv: 2311.09217*, 2023. 2

[61] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. Giraffe hd: A high-resolution 3d-aware generative model. In *CVPR*, 2022. 2

[62] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, pages 7324–7334. PMLR, 2019. 2, 3

[63] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, and William T. Freeman. Visual object networks: image generation with disentangled 3D representations. In *NeurIPS*, 2018. 2

BerfScene: Bev-conditioned Equivariant Radiance Fields for Infinite 3D Scene Generation

Supplementary Material

A. Datasets Details

In this section, we introduce the datasets we use and show sampled BEV maps and front view images.

CLEVR. CLEVR [18] is a synthetic dataset, containing cubes, spheres, and cylinders with different colors. We adopt the official script[†] for rendering. 80K images are collected in total. The camera positions are fixed for all the images. In Fig. A1, we show rendered images with their corresponding BEV maps. We demonstrate tight BEV maps used in the ablation study which represent just the right amount of objects as in the front views. In addition, we also show BEV maps with broader paddings for improving the equivariance of the BEV-conditioned representation. We concatenate together a one-hot vector which indicates color and a one-hot vector which indicates shape at each pixel of the BEV map.

3D-Front. 3D-Front [10, 11] is an indoor scene dataset, which contains different kinds of furniture with fine details. We use the public script[‡] for rendering. We filter out objects with abnormal sizes and collect 2535 different scenes in total. For each scene, we render 20 images from different camera poses. Fig. A2 shows sampled pairs of rendered images and BEV maps. Similar to CLEVR, for each scene, we prepare a tight BEV map for the ablation study, and also a broader version for sake of equivariance. The channel number of the BEV map is one. For each pixel, 0 indicates not occupied by any furniture, while 1 indicates occupied. We do not include any categorical information in the BEV map. Instead, the generator shall infer such knowledge from size, shape, and relative positions between different objects.

Carla. Carla [9] is a self-driving research simulator that offers a variety of realistic visual patterns, including diverse weather conditions and different types of scenes ranging from rural to urban. In our research, we employ a car equipped with a PID controller to autonomously navigate through the town, capturing images with a front-facing camera. A total of 80K images are collected during the process. The relative camera positions to the car remain fixed for all the images. Additionally, we generate the semantic bird's-eye view (BEV) map following the official primitive guidelines. Fig. A3 shows sampled images and BEV maps.

[†]https://github.com/facebookresearch/clevr-dataset-gen

[‡]https://github.com/DLR-RM/BlenderProc/blob/main/examples

B. Implementation Details

We implemented a U-Net architecture for our generator, which consists of four encoders followed by four decoders. Our input is a Fourier feature of shape $256 \times 256 \times 256$, which is computed by StyleGAN3's `SynthesisInput` module. Each encoder downsamples the feature map by a factor of 2 until it reaches a resolution of 16×16.

Each encoder in our U-Net architecture includes a downsample layer, a low-pass filter, an SEL module, and two layers of modulated convolutions. The low-pass filter is designed as a finite impulse response (FIR) filter. The kernel size in the modulated convolutions is 3, while it is 1 in the SEL module. The SEL module takes the similar design as in [54], while we add a low-pass filter after the downsampling operation. The decoders share a similar architecture design with the encoders, except that there is no low-pass filter in the decoders. This is because the upsampling operation in the decoders does not limit the bandwidth of the signal.

C. Infinite Generation

In this section, we make a detailed discussion about how to perform infinite generation over CLEVR and provide more visual examples.

How to synthesize infinite 3D scene? As illustrated by Fig. A4, we generate arbitrary-scale 3D scenes in a *divide-and-conquer* manner. To generate global scenes, we begin by dividing the global BEV maps into smaller local ones by a sliding window. Using these local BEV maps as input, we generate 3D scenes and obtain multiple first view images. To form the final global scene, we extract the middle line of pixels from each image and concatenate them together. This process allows us to combine the information from all the local BEV maps and generate a complete representation of the global scene. It is worth mentioning that, during the *divide* stage, the moving window is shifted pixel by pixel.

Such a design for infinite-scale scene generation places a significant demand on the equivariance property of the generator, as it requires the generator to maintain consistency at a pixel granularity level. An additional benefit of this approach is that by generating local frames and combining them, we can obtain a traversing video: by simply stacking the generated frames, we can create a video that allows for seamless exploration of the entire scene. Videos are zipped in the *Supplementary Material*.

If we do not need traversing video, but only want to get a composite image of the global scene, we can optimize

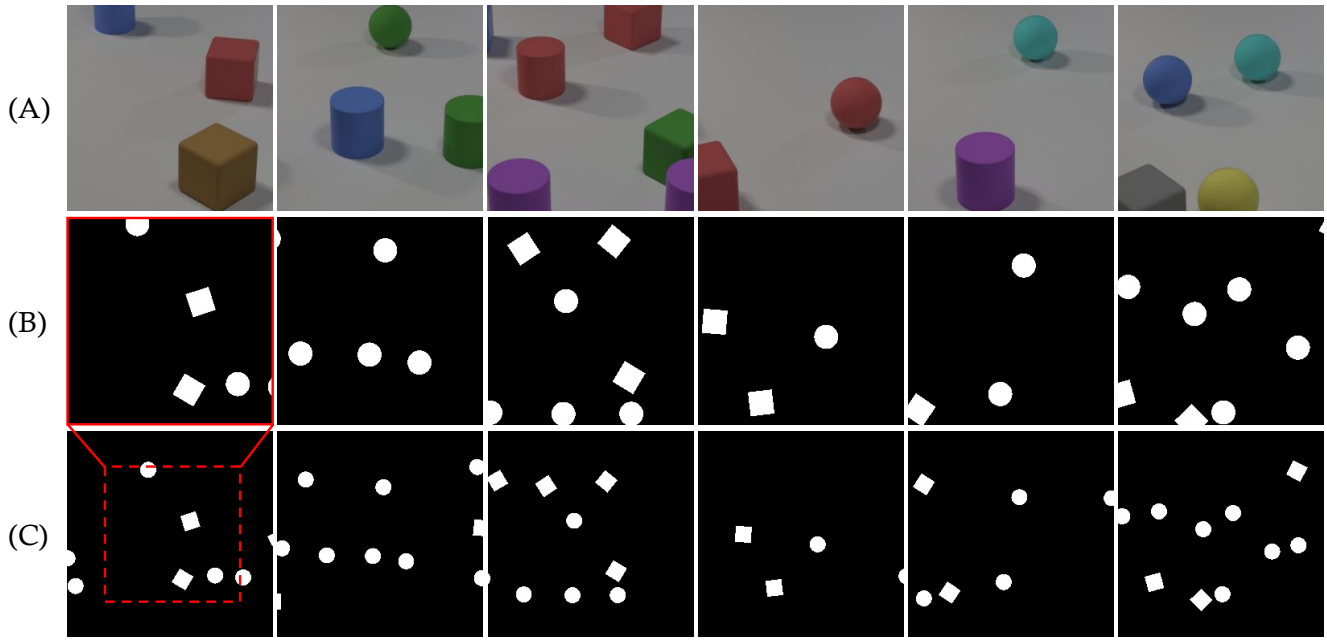

Figure A1. **Sampled front view images and BEV maps on CLEVR.** Row A shows rendered front view images. Row B and C show corresponding BEV maps without and with broader paddings.

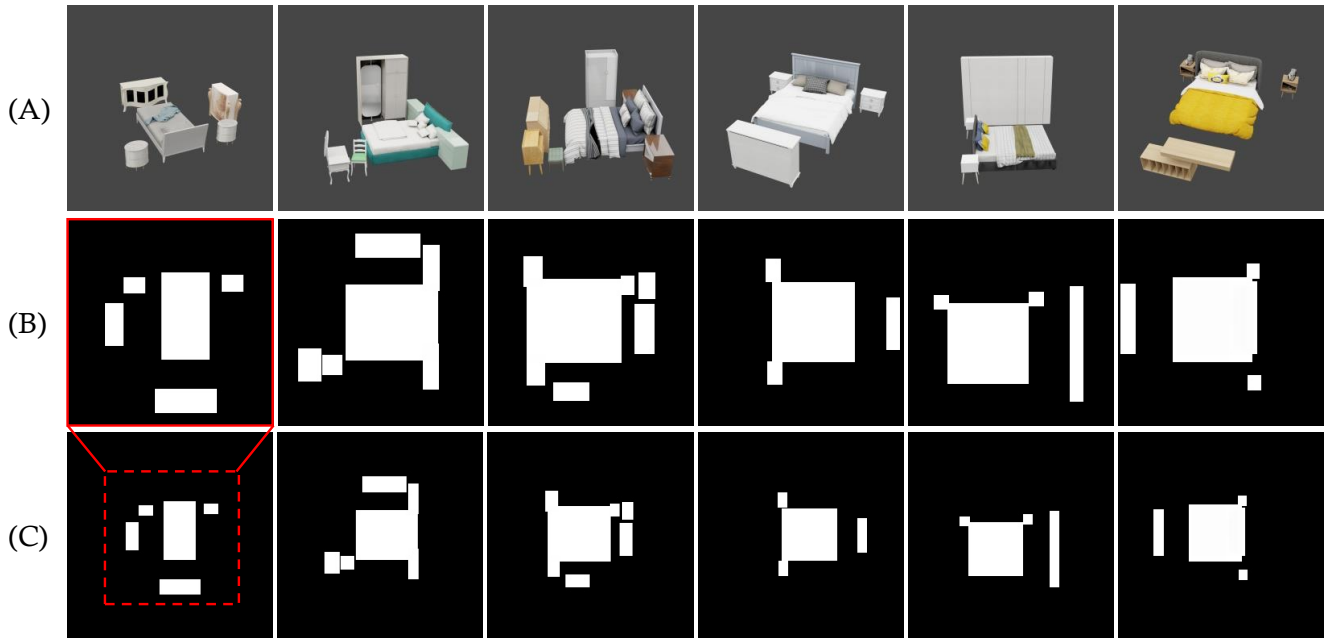

Figure A2. **Sampled front view images and BEV maps on 3D-Front.** Row A shows rendered front view images. Row B and C show corresponding BEV maps without and with broader paddings.

the pipeline by increasing the sliding window step size to N_{step}. This approach involves collecting N_{loc} consecutive lines of pixels from each synthesized image and concatenating them to form the global view. Leveraging the perspective relationship, we can determine that N_{loc} is equal to $\frac{1}{f_{norm}} \cdot N_{step}$, where f_{norm} represents the normalized focal length. Fig. A5 shows the results when N_{step} equals 1, 10, 20, 30, 40. Serrated artifacts can be observed as N_{step} increases, while $N_{step} = 10$ achieves a good balance between efficiency and quality of large-scale 3D scene synthesis.

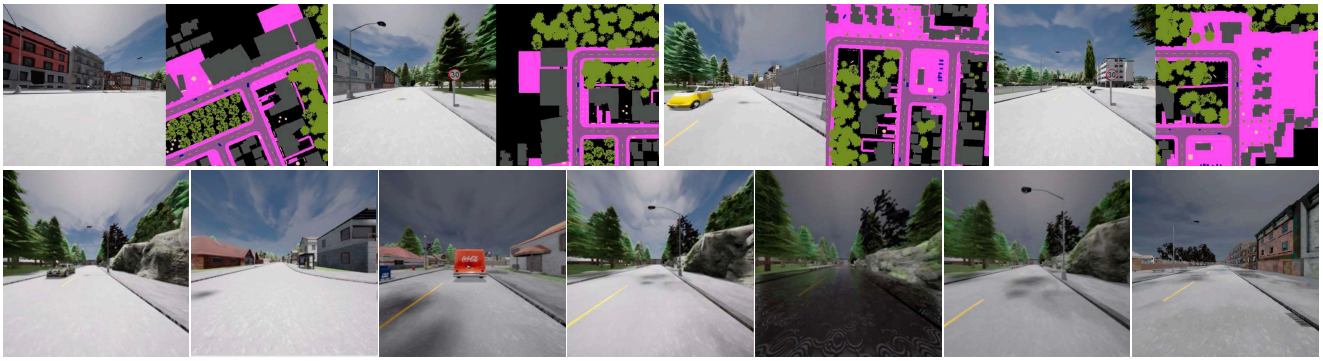

Figure A3. **Sampled front view images and BEV maps on Carla.** The up row shows paired front view images and BEV maps. The bottom row shows diverse weather conditions.

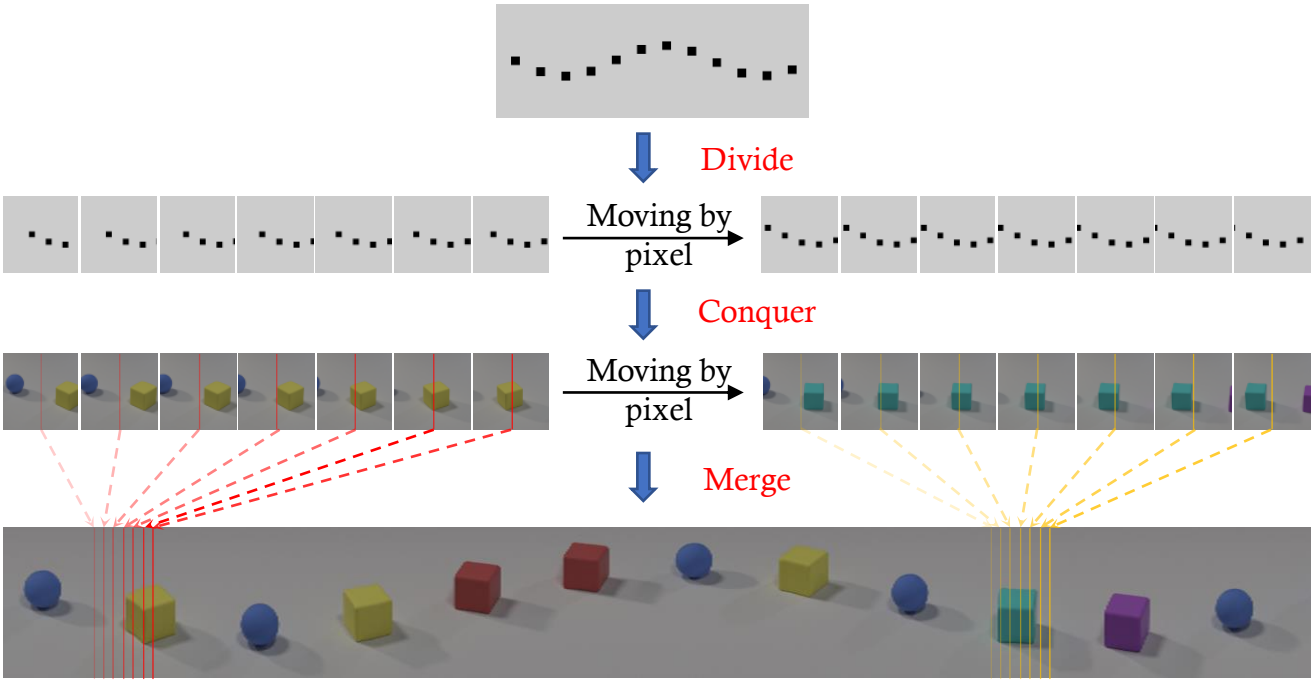

Figure A4. **Illustration of how to perform infinite-scale 3D scene generation.**

More samples. We show more synthesized large scene in Fig. A6. The corresponding traversing videos could be found at the *Supplementary Material*.

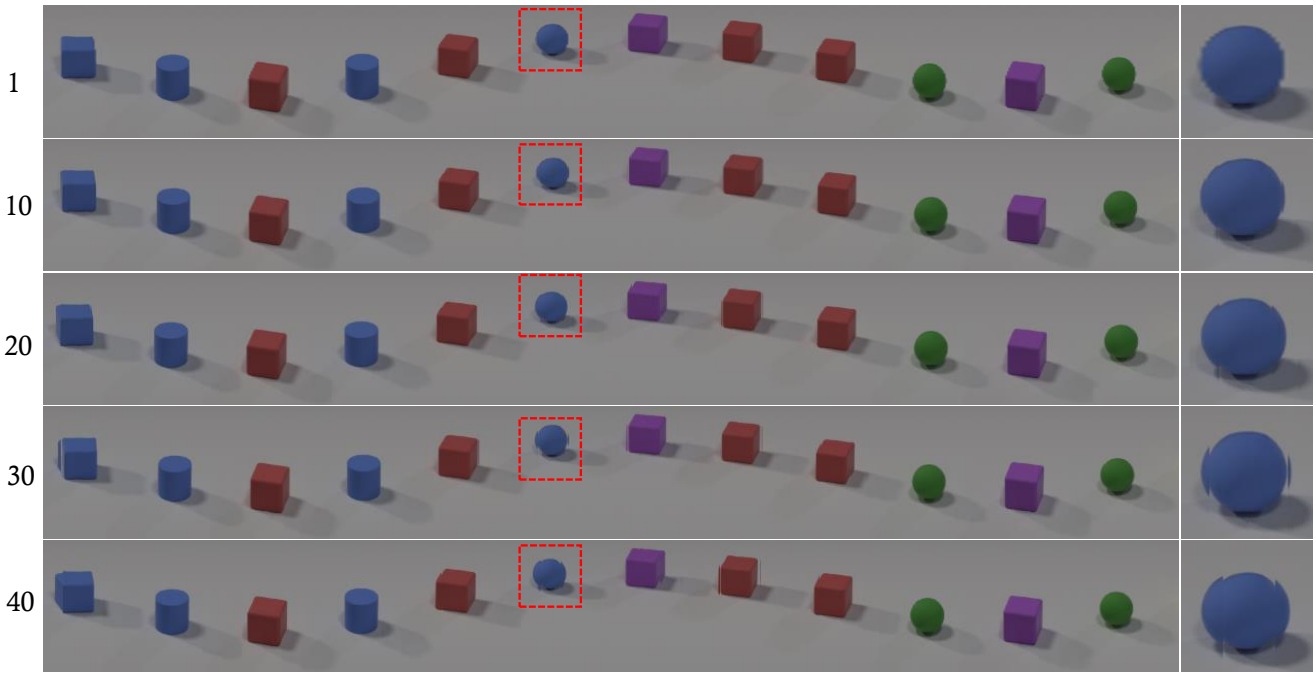

Figure A5. **Synthesized results over different N_{step} choices.**

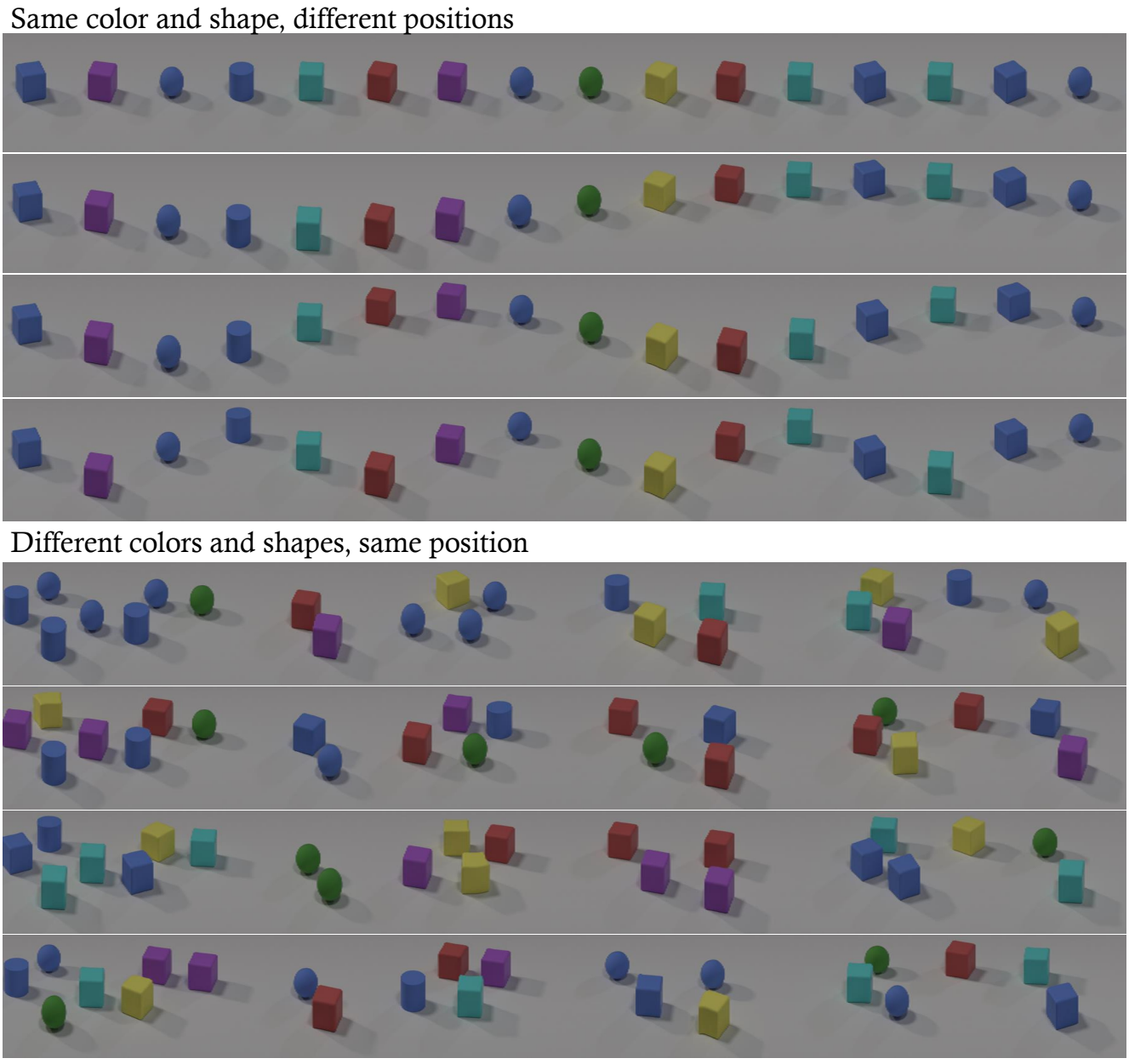

Figure A6. **Synthesized large-scale 3D scene.**