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

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 Lab [5]UCLA

1. Datasets Details

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

CLEVR. CLEVR [4] is a synthetic dataset, containing cubes, spheres, and cylinders with different colors. We adopt the official script[1] for rendering. 80K images are collected in total. The camera positions are fixed for all the images. In Fig. 1, 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 [2, 3] is an indoor scene dataset, which contains different kinds of furniture with fine details. We use the public script[2] 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. 2 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 [1] 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. 3 shows sampled images and BEV maps.

[1]https://github.com/facebookresearch/clevr-dataset-gen
[2]https://github.com/DLR-RM/BlenderProc/blob/main/examples

2. 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 [5], 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.

3. 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. 4, 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

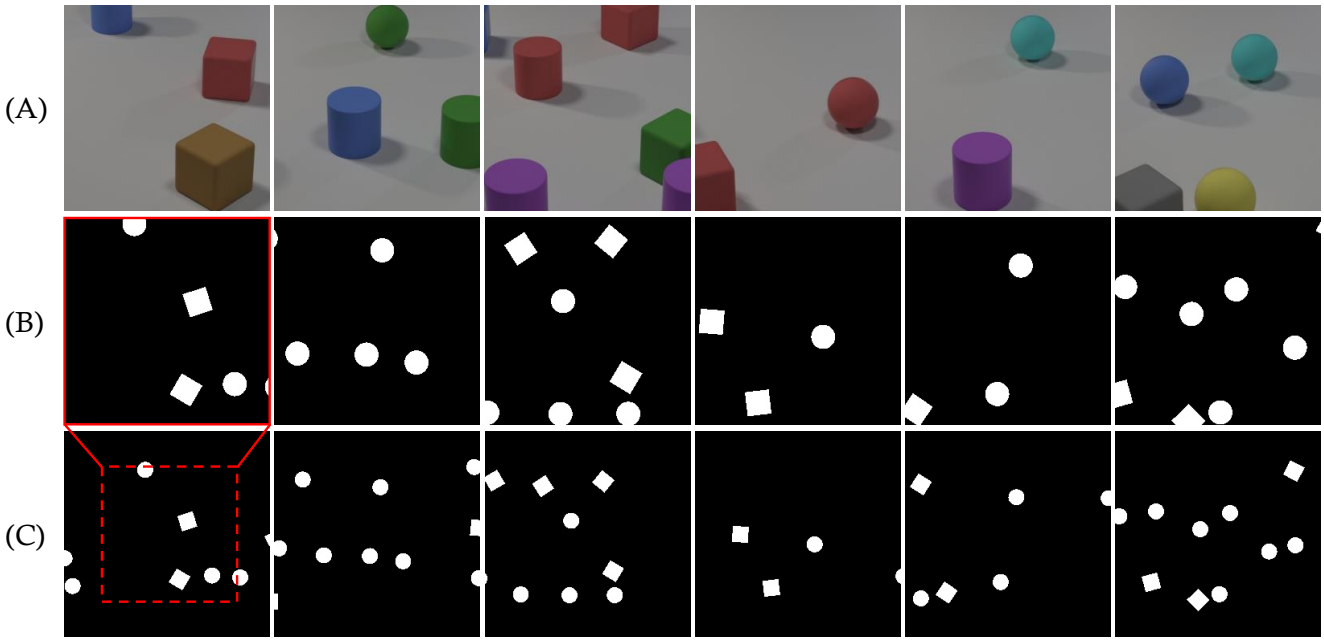

Figure 1. **Sampled front view images and BEV maps on CLEVR.** Row A shows rendered front view images. Row B and C show corresponding BEV maps without and with broader paddings.

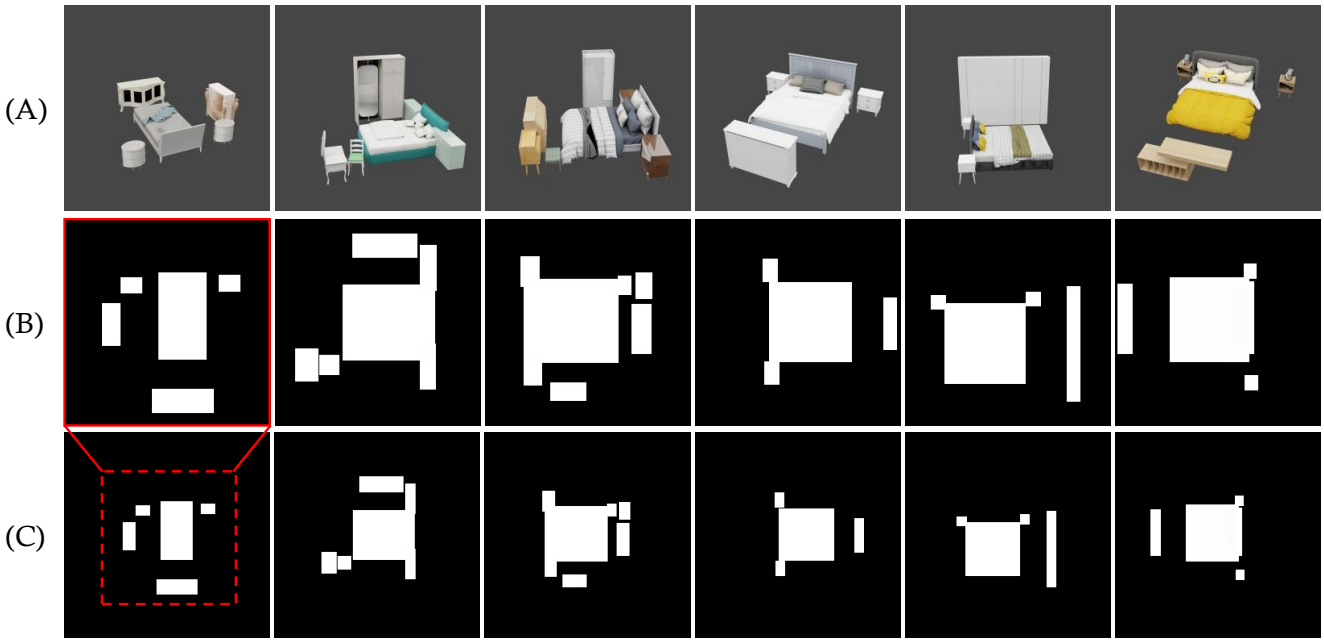

Figure 2. **Sampled front view images and BEV maps on 3D-Front.** Row A shows rendered front view images. Row B and C show corresponding BEV maps without and with broader paddings.

the 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

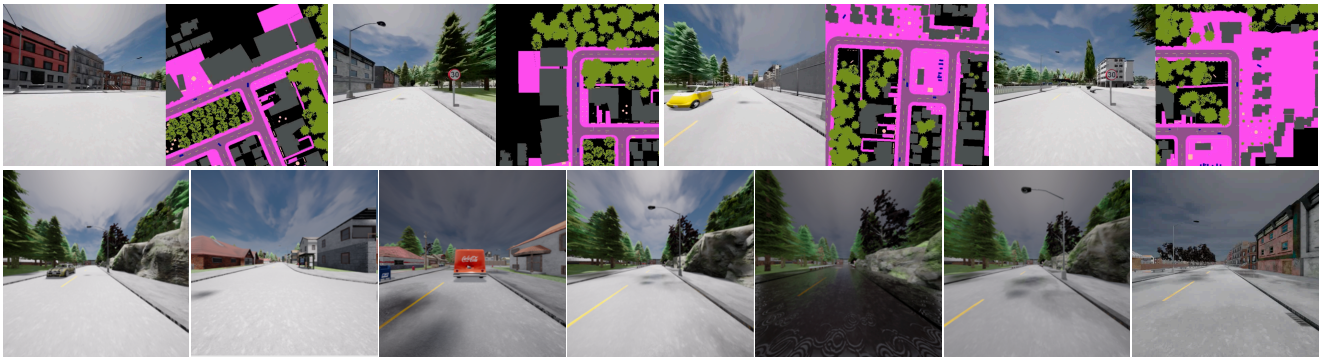

Figure 3. **Sampled front view images and BEV maps on Carla.** The up row shows paired front view images and BEV maps. The bottom row shows diverse weather conditions.

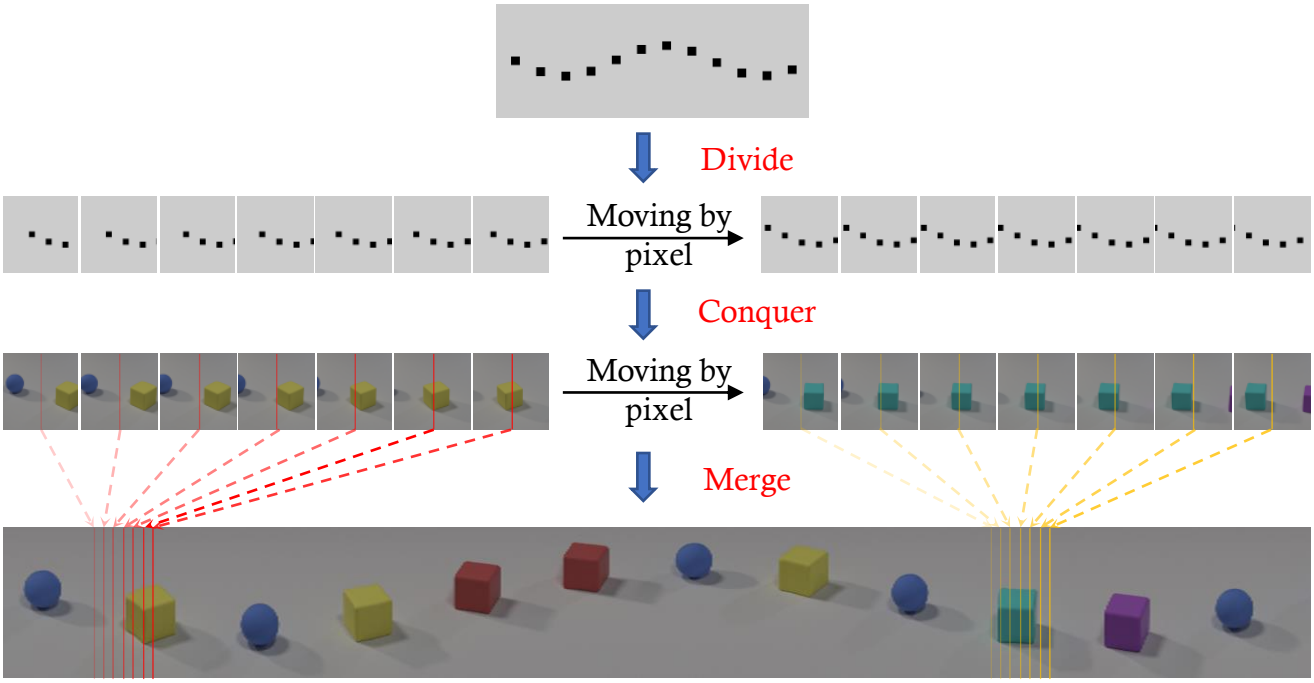

Figure 4. **Illustration of how to perform infinite-scale 3D scene generation.**

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 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. 5 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.

More samples. We show more synthesized large scene in Fig. 6. The corresponding traversing videos could be found at the *Supplementary Material*.

References

[1] 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. 1

[2] 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. 1

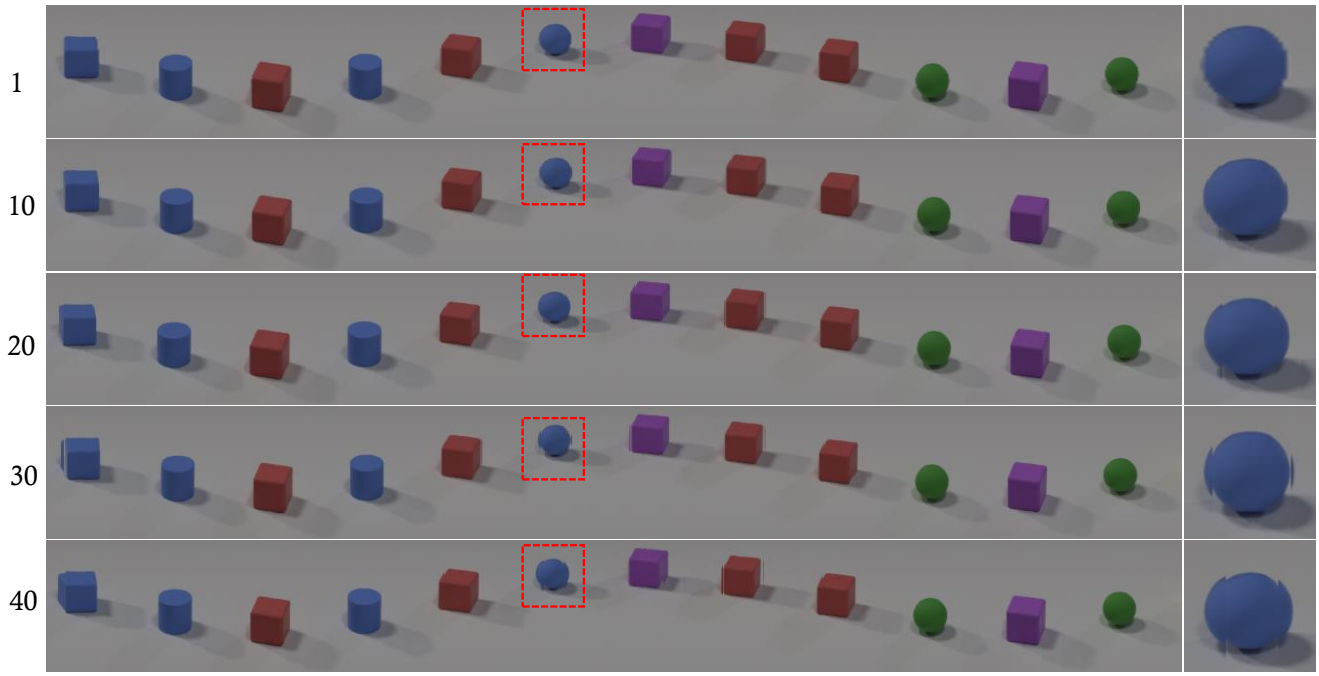

Figure 5. **Synthesized results over different N_{step} choices.**

[3] 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. 1

[4] 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. 1

[5] 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. 1

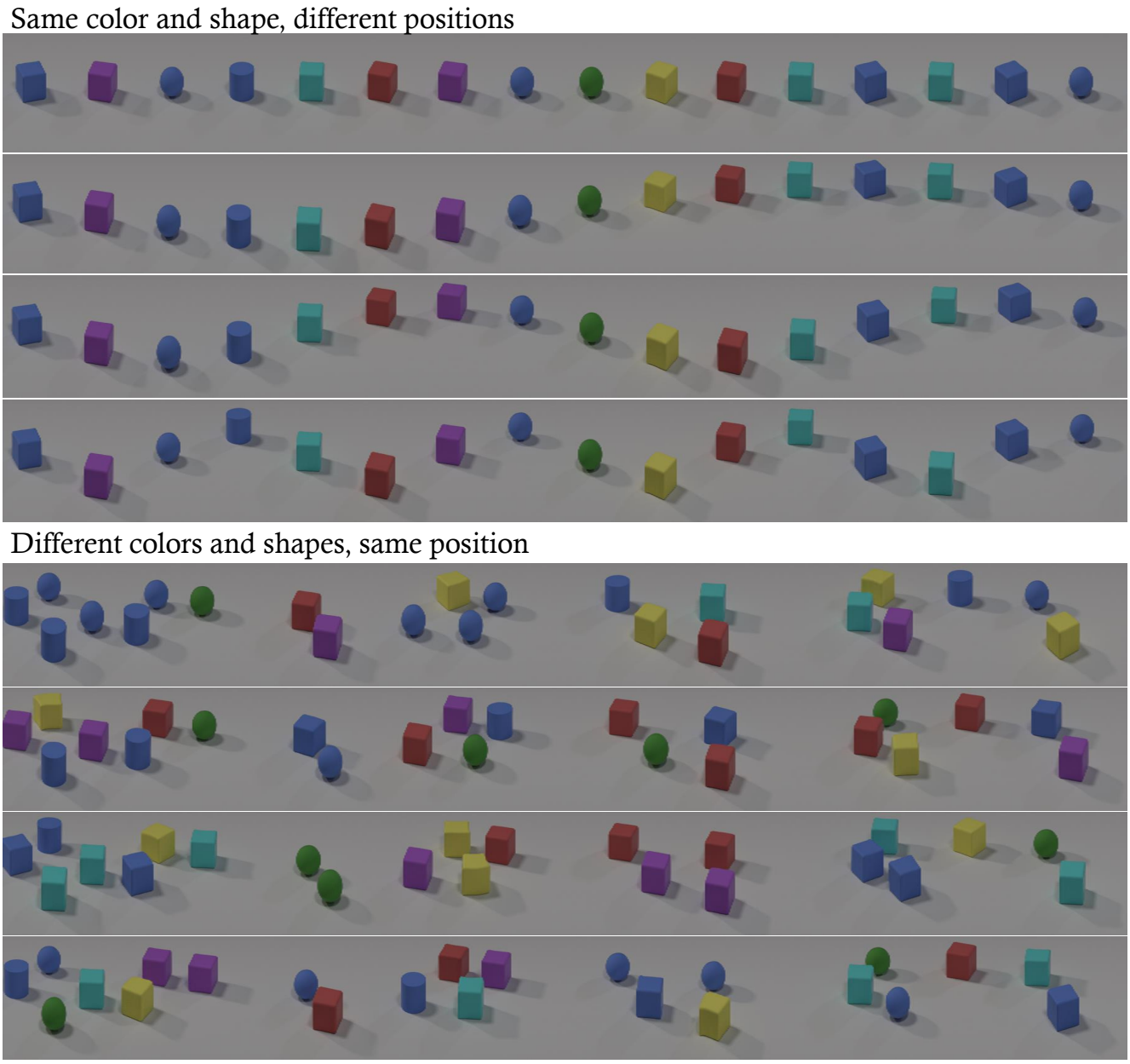

Figure 6. **Synthesized large-scale 3D scene.**