

StarGen: A Spatiotemporal Autoregression Framework with Video Diffusion Model for Scalable and Controllable Scene Generation

Supplementary Material

1. Network Architecture

We provide the pseudo code in Algorithm 1 to give a detailed explanation of the network architecture for the proposed Spatiotemporal-Conditioned Video Generation (SCVG).

2. Additional Training Details

We train our model using the AdamW [16] optimizer with a learning rate of 0.0004. To accelerate the training process, we employ xFormers [9] and mixed-precision [17] techniques. When projecting reconstructed latent features from the spatial conditioning images onto novel views, certain areas in the novel views may not be visible in the spatial conditioning images, resulting in regions without projections. During the loss calculation, we mask these regions to prevent them from influencing the loss. Our backbone is based on CogVideoX [27], with the T5 model [18] serving as the text encoder. Since the training datasets—RealEstate-10K [32], ACID [14] and DL3DV [13]—do not provide captions, we use PLLAVA [26] to generate captions for each video clip.

3. Downstream Tasks Details

3.1. Sparse View Interpolation

Method Details. We provide a detailed explanation of generating a long-range video in scenarios where the first and last input images have minimal or no overlap. Given two input image $\mathbf{I}_{\text{first}}$ and \mathbf{I}_{last} , we invoke the proposed SCVG to generate the image sequence $\mathbf{x} = \{\mathbf{x}_0, \dots, \mathbf{x}_{L-1}\}$. To interpolate those sparse frames to a long dense video, we uniformly sample $m + 1$ frames $\{\mathbf{x}_{i_0}, \mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_m}\}$ from the first-pass result and process the second video generation. In the second-pass process, each pair of adjacent frames $(\mathbf{x}_{i_j}, \mathbf{x}_{i_{j+1}})$ serves as the input of SCVG and generate L novel views, resulting a long final video. By adjusting the value of m , we can control the length of the final generated video.

Experiment Details. To ensure a fair comparison with ReconX [15] (32 frames) and ViewCrafter [29] (25 frames), we set the generated video length to 33 frames. This choice is also influenced by the limitation of the CogVideoX [27] 3D VAE, which only supports videos with $4n + 1$ frames. We filter the training datasets from RealEstate-10K, ACID, and DL3DV-10K to include only videos with at least 33 frames. Furthermore, we remove videos that could not be

Algorithm 1 Pseudo code of the proposed SCVG.

```
# Input list:
# spatial_images: [b, 2, h, w, 3]; b is the batch
# size; the spatial_images are two spatial
# conditioning images; h and w are the height
# and width
# temporal_images: [b, 1, h, w, 3]; the temporal
# conditioning image is the last temporal image
# of the previous prediction output;
# text_prompt
# spatial_extrinsics: [b, 2, 4, 4]; the extrinsic
# parameters of the three input images.
# spatial_intrinsics: [b, 2, 4]; the intrinsic
# parameters of the three input images.
# out_extrinsics: [b, n, 4, 4]; the extrinsic
# parameters. n is the frame count.
# out_intrinsics: [b, n, 4]; the intrinsic
# parameters.

# Output list:
# video: [b, n, h, w, 3]

# LRM
depths = DepthAnythingV2(spatial_images) # [b, 2,
# h, w, 1]
rays_os, rays_ds = get_rays(h, w,
# spatial_intrinsics, spatial_extrinsics) #
# rays_origin, rays_direction: [b, 2, h, w, 3]
x = concat([spatial_images, depths, rays_ds,
# cross(rays_os, rays_ds)], dim=-1) # [b, 2, h,
# w, 10]
x = conv(x, out=d, kernel=8, stride=8) # patchify
# to [b, 2, h/8, w/8, d]
x = x.reshape(b, -1, d) # transformer input [b, 2
# h/8 * w/8, d]
x = transformer(LN(x))
x = LN(x)
x = x.reshape(b*2, h//8, w//8, d)
x = deconv(x, out=17, kernel=8, stride=8) # [b,
# 2, h, w, 17]
x = x.reshape(b, -1, 17) # [b, 2 * h * w, 17]
distance, feature = split(x, [1, 16], dim=-1)
w = sigmoid(distance)
point_cloud = rays_o + rays_d * (near * (1 - w) +
# far * w)

# Render spatial condition
feature_map = render(point_cloud, feature,
# out_extrinsic, out_intrinsic) # render
# feature maps from point cloud with features
# based on the camera's intrinsic and extrinsic
# parameters [b, n, h/8, w/8, 16]

# Video Diffusion Model
spatial_latents = causalConv3d(feature_map) # [b,
# n/4, h/8, w/8, 16]
temporal_latents = vae_encode(temporal_images) # [
# b, 1, h/8, w/8, 16]
control_latents = concat([temporal_latents,
# spatial_latents[:, 1:]], dim=1) # [b, n/4, h
# /8, w/8, 16]
latents = CogVideoX(noise, Controlnet(
# control_latents), text_prompt) # [b, n/4, h
# /8, w/8, 16]
video = vae_decode(latents)

return video
```

downloaded and those where the pose counts did not match the frame counts. After these steps, a total of 66,859 videos remain in the final training dataset. For evaluation, we select 100 videos from the test sets of RealEstate-10K and

ACID. Following previous methods [3, 5, 15], we calculate the evaluation metrics using three frames per clip, specifically the 5th, 15th, and 25th frames.

3.2. Perpetual View Generation

Method Details. Given the first image $\mathbf{I}_{\text{first}}$ and a pose trajectory, we invoke the proposed SCVG Algorithm 1 in an autoregressive manner to generate a sequence of novel views. In generating the first clip, the input image $\mathbf{I}_{\text{first}}$ is duplicated to create paired spatial conditioning images for SCVG, aligning with the zero-shot novel view synthesis task in ViewCrafter [29]. Note that depth predictions from identical images can suffer from scale ambiguity. To align different methods to the same scale for fair comparison, we use DUST3R [20] to calculate the mean depth of the first image as a reference. We then align the mean depth of the first image from each method to this reference scale. In the subsequent clip generation, we simply use the first and last frames of the previously generated clip as the spatial conditioning pair. More sophisticated frame selection strategies can also be employed as alternatives.

Experiment Details. Apart from using the datasets detailed in Sec. 3.1, we also evaluate perpetual view generation on the Tanks-and-Temples dataset [8] to further validate its generalization capabilities. We use 6 scenes from its test set without utilizing any training data from Tanks-and-Temples.

3.3. Layout-Conditioned City Generation

Method Details. As described in the main paper, we adopt a two-stage approach, primarily integrating different ControlNets. The semantic ControlNet output is denoted as \mathbf{C}^s , the depth ControlNet output is \mathbf{C}^d , and the output of our SCVG ControlNet, as described in Algorithm 1, is \mathbf{C}^{scvg} . We combine different ControlNets by linearly weighting their output features. The weighted and combined features are then added to each block of CogVideoX to produce the output video. The distinction between the two stages lies in the combination of these ControlNets. In the first stage, the features are combined as $\alpha_1 \mathbf{C}^s + \beta_1 \mathbf{C}^d$, while in the second stage, they are combined as $\alpha_2 \mathbf{C}^s + \beta_2 \mathbf{C}^d + \gamma_2 \mathbf{C}^{\text{scvg}}$. In our experiments, α_1 and β_1 are both set to 0.5, while α_2 and β_2 are set to 0.3, and γ_2 is set to 0.4.

Experiment Details. In this task, we additionally utilize the CityGen dataset from CityDreamer [23], which comprises city layout maps derived from OpenStreetMap [2] and renderings generated by Google Earth Studio [1]. Each trajectory in this dataset contains only 60 frames, while spanning a large scene with a radius of approximately 400 meters, leading to very sparse frame intervals. To address this issue, we use Google Earth Studio to perform frame interpolation on the original trajectories, increasing the number of frames per trajectory to 600. We observe that Google Earth

Studio’s mesh-based representation introduces inconsistencies due to misalignment or lighting changes among the images used for mesh reconstruction, leading to artifacts in the training data. For instance, distortions in zebra crossings are frequently observed in the training data. We observe that the artifacts in the training data impact the learning process, as our model also learns and replicates these distortions during training. Despite this, compared to CityDreamer trained on the same data, our model produces significantly better results.

4. Evaluation Details

4.1. Baselines

The quantitative results for pixelNeRF [28], GPNR [19], AttnRend [4], and MuRF [24] are sourced from the MV-Splat paper [5]. The results for pixelSplat [3], MVSplat, GS-LRM [30], DepthSplat [25], and ReconX [15] are taken from their respective original papers. DepthSplat is not evaluated on ACID due to the lack of a released model trained on this dataset. The quantitative results for ViewCrafter [29], InfNat0 [12], LucidDreamer [6], MotionCtrl [22], and CityDreamer [23], as well as all qualitative results for each method, are reproduced using their publicly available code.

InfNat0 first resizes input images to 384×384 and then crops them to 256×256 . MotionCtrl resizes images proportionally to a 1024-pixel short side, followed by cropping to 1024×576 . We adhere to their default configurations. For LucidDreamer, we crop the input images to square dimensions and resize them to 512×512 . LucidDreamer supports only predefined camera intrinsics and pose trajectories. We modify the code to accommodate input camera control. Additionally, as LucidDreamer does not support long text prompts, we use text prompts generated with LAVIS [11] as specified in their original paper. For ViewCrafter, the images are first cropped to square dimensions and then resized to 512×288 to avoid further cropping in ViewCrafter. Since DUST3R [20] re-estimates camera intrinsics and extrinsics, the resizing process has minimal impact on the final results. In perpetual view generation, the necessary code for point cloud stitching is not released. We implement this functionality based on the paper’s description. For the task of layout-conditioned city generation, the results for CityDreamer are obtained by bypassing their unbounded layout generator and instead using our selected layout maps and the same trajectories as input.

4.2. Metrics

Short-Range Video Quality. For sparse view interpolation and short-range perpetual view generation, we use the PSNR, SSIM [21] and LPIPS [31] to measure the similarity between generated and groundtruth images. Since differ-

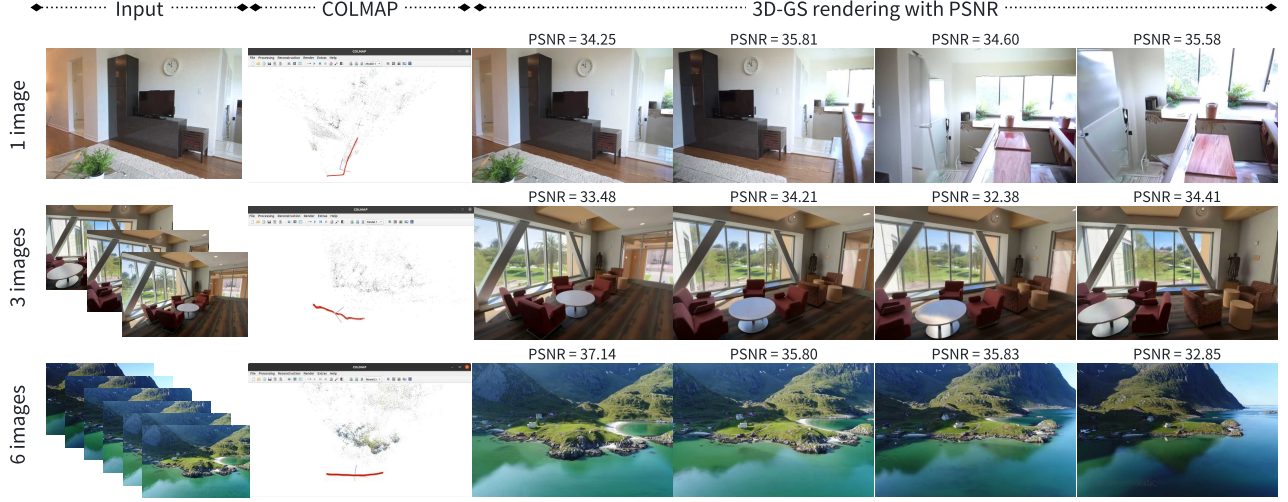


Figure 1. The reconstruction scenes of the generated videos: the reconstruction scene of single image generated video (the first row); the scene result interpolated from three images(the second row); the scene result interpolated from six images(the third row).

ent methods for comparison may operate at different resolutions, we consistently resize both the groundtruth and generated images to 256×256 for a fair comparison.

Long-Range Video Quality. For long-range perpetual view generation, since the output video contains a large amount of generated content, we calculate Fréchet Inception Distance (FID) [7] to assess the performance. For consistent evaluation of generated videos with different lengths, we use a fixed number of images in each FID calculation. Specifically, when calculating FID for generated videos with different lengths, we sample different numbers of clips from the test set to ensure that the total number of frames for each calculation is 5000. Again, we resize both the groundtruth and generated images to 256×256 for a fair comparison.

Pose Accuracy. We evaluate the pose-control ability of different methods by comparing the groundtruth poses with estimated poses of the generated images. In our experiments, we use MAST3R [10] to estimate poses as it is more robust than traditional methods. After the pose estimation, we transform the estimated pose to the GT coordinate system and align the scale for evaluation. Specifically, we first transform the estimated trajectory by aligning the first camera pose to groundtruth, then calculate the scale factor by comparing the lengths of groundtruth and estimated trajectories, and finally apply the scale factor to the estimated trajectories. We calculate the average rotation error R_{dist} and

translation error T_{dist} by:

$$R_{dist} = \frac{1}{n} \sum_{i=1}^n \arccos \left(\frac{\text{tr}(\mathbf{R}_{gen}^i \mathbf{R}_{gt}^{iT}) - 1}{2} \right), \quad (1)$$

$$T_{dist} = \frac{1}{n} \sum_{i=1}^n \|\mathbf{T}_{gt}^i - \mathbf{T}_{gen}^i\|_2,$$

where \mathbf{R}_{gen}^i and \mathbf{T}_{gen}^i are the rotation and translation estimated from the generated video corresponding to the i -th frame, while \mathbf{R}_{gt}^i and \mathbf{T}_{gt}^i are the groundtruth.

For the evaluation of long-range video, considering the computational efficiency of MAST3R, the cost of recovering all frames would be unbearable. Therefore, we only estimate the poses of keyframes for evaluation. We select one keyframe from every tenth frame in the first 200 frames of the generated videos, resulting in 21 keyframes (0, 10, ..., 200).

5. Reconstruction Results

To further demonstrate the temporal consistency of the long videos generated by the proposed method, we employed COLMAP to perform sparse reconstruction on the generated video sequences and subsequently trained a 3D Gaussian Splatting (3DGS) model. As illustrated in Fig. 1, both the sparse reconstruction results and the rendered outputs indicate that the generated videos exhibit excellent multi-view consistency, which significantly facilitates the reconstruction of large-scale scenes. This consistency not only validates the robustness of our method but also highlights its potential for applications requiring high-fidelity scene reconstruction and rendering.

Method	part (a)	part (b)	backbone	FID↓	$R_{\text{dist}}\downarrow$	$T_{\text{dist}}\downarrow$
Ours	LRM	CN	CogVideo	41.72	2.088	0.453
DUST3R+CN	DUST3R	CN	CogVideo	55.14	2.580	0.694
DUST3R+Concat	DUST3R	Concat	CogVideo	<u>51.40</u>	2.739	0.716
ViewCrafter	DUST3R	Concat	DynamicCrafter	62.91	11.32	0.86

Table 1. Ablation results for different module combinations.

6. Effect of VDM backbone

To ensure a fair comparison with ViewCrafter, we reimplemented its modules using the same CogVideo backbone, eliminating backbone-related discrepancies. The key differences between StarGen and ViewCrafter are: (a) For novel view conditioning, StarGen uses an LRM for latent feature reconstruction, while ViewCrafter relies on DUST3R for RGB point cloud reconstruction; (b) StarGen employs ControlNet for condition injection, whereas ViewCrafter uses concatenation. We evaluated different module combinations under identical experimental settings (Section 4.3, long-range video), benchmarking performance on the RealEstate-10K dataset. As shown in Tab. 1, our proposed method demonstrates significant advantages even under the same backbone.

References

- [1] <https://earth.google.com/studio/>. 2
- [2] <https://www.openstreetmap.org/>. 2
- [3] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. PixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 16-22, 2024*, pages 19457–19467. IEEE, 2024. 2
- [4] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations, Vienna, Austria, May 7-11, 2024*. 2
- [5] Yuedong Chen, Haoifei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2025. 2
- [6] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. LucidDreamer: Domain-free generation of 3D gaussian splatting scenes. *arXiv preprint*, arXiv:2311.13384, 2023. 2
- [7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. 3
- [8] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph.*, 36(4):78:1–78:13, 2017. 2
- [9] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers>, 2022. 1
- [10] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3D with MAST3R. In *European Conference on Computer Vision, Milan, Italy, September 29-October 4, 2024*, pages 71–91. Springer, 2024. 3
- [11] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C. H. Hoi. LAVIS: A one-stop library for language-vision intelligence. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2023, Toronto, Canada, July 10-12, 2023*, pages 31–41. Association for Computational Linguistics, 2023. 2
- [12] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. InfiniteNature-Zero: Learning Perpetual View Generation of Natural Scenes from Single Images. In *European Conference on Computer Vision, Tel Aviv, Israel, October 23-27, 2022*, pages 515–534. Springer, 2022. 2
- [13] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 16-22, 2024*, pages 22160–22169. IEEE, 2024. 1
- [14] Andrew Liu, Ameesh Makadia, Richard Tucker, Noah Snavely, Varun Jampani, and Angjoo Kanazawa. Infinite Nature: perpetual view generation of natural scenes from a single image. In *IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, October 10-17, 2021*, pages 14438–14447. IEEE, 2021. 1
- [15] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. ReconX: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint*, arXiv:2408.16767, 2024. 1, 2
- [16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations, New Orleans, LA, USA, May 6-9, 2019*. 1
- [17] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Damos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. *arXiv preprint*, arXiv:1710.03740, 2017. 1
- [18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67, 2020. 1
- [19] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural render-

- ing. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022. 2
- [20] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. DUS3R: Geometric 3D vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 16-22, 2024*, pages 20697–20709. IEEE, 2024. 2
- [21] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612, 2004. 2
- [22] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers, Denver, CO, USA, 27 July 2024- 1 August 2024*, page 114. ACM, 2024. 2
- [23] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. CityDreamer: compositional generative model of unbounded 3D cities. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 16-22, 2024*, pages 9666–9675. IEEE, 2024. 2
- [24] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. MuRF: Multi-baseline radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, June 16-22, 2024*, pages 20041–20050. IEEE, 2024. 2
- [25] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 2
- [26] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See-Kiong Ng, and Jiashi Feng. Pllava : Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint, arXiv.2404.16994*, 2024. 1
- [27] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihang Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint, arXiv.2408.06072*, 2024. 1
- [28] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, virtual, June 19-25, 2021*, pages 4578–4587. IEEE, 2021. 2
- [29] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint, arXiv.2409.02048*, 2024. 1, 2
- [30] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: large reconstruction model for 3D gaussian splatting. In *European Conference on Computer Vision, Milan, Italy, September 29-October 4, 2024*, pages 1–19. Springer, 2024. 2
- [31] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. IEEE, 2018. 2
- [32] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4): 65, 2018. 1