VideoRFSplat: Direct Scene-Level Text-to-3D Gaussian Splatting Generation with Flexible Pose and Multi-View Modeling

Supplementary Material

A. Details of VideoRFSplat

Here, we present additional details on VideoRFSplat, expanding on the discussion in Section 4 to address space constraints. We first delve into the architecture and camera optimization of the dual-stream pose-video joint model in Section A.1, emphasizing key modifications from the Mochi video generation model. Following that, Section A.2 details the architecture of the Gaussian Splat decoder. Lastly, in Section A.3, we outline the previously omitted aspects of CFG for camera-conditioned generation.

A.1. Dual-Stream Pose-Video Joint Model

Architecture. We adopt Mochi [11] as the backbone for our video generation model without any architectural modifications. Mochi utilizes an Asymmetric Diffusion Transformer architecture with full 3D attention, enabling high-fidelity video synthesis. The model consists of approximately 10 billion parameters, and it employs the T5-XXL model [9] as its text encoder to extract text-conditional embeddings. For further architectural details, we refer the reader to [11].

For the pose generation module, we employ the same Asymmetric Diffusion Transformer architecture as Mochi but with a more efficient configuration. Specifically, we set the hidden size to 256, the patch size to 2, the number of attention heads to 4, and the input size of the ray embedding to 10×16. While the video generation model consists of 48 transformer blocks, the pose model is designed with 16 blocks for computational efficiency. Both models share the same text encoder. The pose model is trained from scratch.

The communication block is placed at every three blocks in the video model and at every single block in the pose model, allowing periodic information exchange between the two streams. Communication is omitted in the final layer to maintain independent final representations. To prevent significant changes in the initial output of the video model, weights and biases of the linear layer in the communication block are initialized to zero.

Ray to camera parameters. To recover camera parameters from generated rays, we follow a slightly optimized version of the RayDiffusion approach [16], which is utilized in SplatFlow [4]. In summary, the camera center is estimated by minimizing the mismatch between rays. Next, the projection matrix is computed using a least-squares approach and decomposed into the intrinsic matrix and rotation. Finally, optimization with the Adam optimizer [5] is applied to refine the intrinsic and rotation matrices, enforcing shared intrinsic parameters across all views. Then, we slightly refine camera poses to be consistent with source views.

A.2. Details of Gaussian Splat Decoder

The Gaussian Splat Decoder follows a 3D-CNN architecture, adopted from the decoder of Mochi [11]. To enhance global context modeling, we add two attention layers to each residual block in the lowest layer and do not use causal 3D convolutions [14]. Additionally, we incorporate Plücker ray embeddings as inputs to the decoder. Specifically, the embeddings follow the format of LGM [10], forming a 9-channel representation. These embeddings are transformed to match the resolution of the intermediate representations in each decoder block and are injected via additional 3D convolution layers. The Gaussian Splat Decoder outputs depth, opacity, RGB, rotation, and scale, forming an 11-channel output. To accommodate this, we introduce an additional 1×1 convolution layer in the output layer. For 3DGS rendering, we utilize Gsplat [13] library, which offers the efficient implementation.

A.3. Details of Camera Conditioned Generation

Classifier-Free Guidance details. To implement camera-conditioned generation, we follow the formulation presented in [1]. Since our generation task involves multi-view generation conditioned on both a text prompt and a camera trajectory, we decompose these two conditions within the Classifier-Free Guidance (CFG) framework like HarmonyView [12]:

$$\begin{aligned}\Big[&(1+s_c)u_\theta(\mathcal{I}_{t_\mathcal{I}}, \mathcal{R}_{t_\mathcal{R}}, c, t_\mathcal{I}, t_\mathcal{R}) \\ &- s_c u_\theta(\mathcal{I}_{t_\mathcal{I}}, \mathcal{R}_{t_\mathcal{R}}, c_{\text{null}}, t_\mathcal{I}, t_\mathcal{R}) \\ &+ (1+s_\mathcal{R})u_\theta(\mathcal{I}_{t_\mathcal{I}}, \mathcal{R}_{0.05}, c, t_\mathcal{I}, 0.05) \\ &- s_\mathcal{R} u_\theta(\mathcal{I}_{t_\mathcal{I}}, \mathcal{R}_1, c, t_\mathcal{I}, 1)\Big]/2,\end{aligned} \tag{1}$$

where $s_\mathcal{R}$ and s_c represent the guidance strengths for the camera pose and text conditions, respectively, and c_{null} denotes the null text condition used in CFG. Also, following DiffusionForcing [2], we slightly noise the camera pose condition to the better conditional mechanism as $t_\mathcal{R} = 0.05$.

B. Further Details of Experimental Setups

Training dataset. We use multiple datasets for training, each containing multi-view images with camera annotations. The MVImgNet [15] dataset originally consisted of 219,188 scenes with camera parameters. After filtering out erroneous scenes, we retained approximately 200K scenes.

For validation, 1.25K scenes were allocated for each specific task. The DL3DV [7] dataset, which initially contained 10K scenes, was similarly processed, with 300 sequences designated for the validation set. Additionally, we incorporate the RealEstate10K [17] and ACID [8] datasets in our training pipeline. However, during downloading and preprocessing, approximately 20% of the total data was lost due to filtering and quality control steps. These datasets collectively provide a diverse set of multi-view scenarios:

- **MVImgNet** consists of object-centric video, capturing various objects within controlled environments.
- **DL3DV** primarily contains outdoor scenes, featuring complex natural landscapes and diverse conditions.
- **ACID** focuses on aerial scenes, providing a wide range of viewpoints from aerial.
- **RealEstate10K** comprises indoor scenes, primarily focused around residential rooms and houses.

To generate text captions, we extract multiple captions per sequence for the DL3DV, RealEstate10K, and ACID datasets, enabling fine-grained descriptions for each sequence. Each sequence is divided into groups of 32 images, and captions are generated using the InternVL2.5-26B model [3]. In contrast, for the MVImgNet dataset, we generate a single caption per scene to provide a concise summary of the overall content. Since MVImgNet is an object-centric dataset, the main object remains consistent across all images within a sequence, making a single caption sufficient for describing the entire scene. To ensure that the captions accurately capture the main object while remaining concise, we randomly select one of the following three prompts:

- *"Briefly describe the main object in the image, including its color and key features, in a single concise sentence."*
- *"Describe the main object and its surroundings in 15 words or fewer, using keywords or a short phrase."*
- *"Summarize the main object's color, texture, and shape in no more than 15 words, using a concise phrase."*

Training Details. We train the joint pose-video model with a batch size of 16 for 120K iterations. We use an initial learning rate of 5×10^{-5} with a cosine decay schedule and a warm-up period of 1000 steps. Training is implemented using Fully Sharded Data Parallel (FSDP) for memory efficiency, and we optimize the model using the Adam optimizer. For timestep sampling, we uniformly sample both pose and multi-view timesteps. To further improve training efficiency, text embeddings are precomputed and stored prior to training. For training the Gaussian Splat decoder, we use a batch size of 8 for 400K iterations with a learning rate of 5×10^{-5}. During the first 300K iterations, we render 13 target views for training, and in the final 100K iterations, we increase the number of target views to 19.

Inference and Evaluation. For inference, we follow the default timestep schedule used in Mochi. Unless otherwise specified, we use 64 sampling steps. Additionally, as shown in the ablation study, we set $\delta = 0.2$ as the default. Metrics computation and evaluation splits are configured following the setup of SplatFlow [4]. Additionally, during sampling, one possible approach for modified Classifier-Free Guidance (CFG) is to continuously sample random poses from $\mathcal{N}(0, I)$. However, we found that pre-sampling the poses and keeping them fixed during generation leads to improved stability.

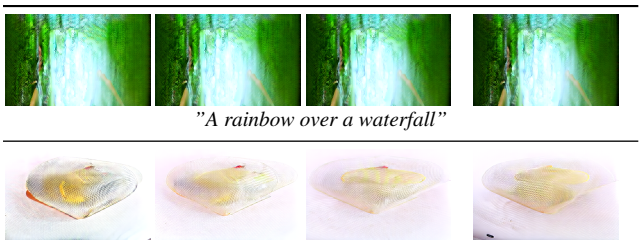

Figure A.1. **Generated multi-view images from image-first asynchronous sampling.** Accelerating the denoising of multi-view images, instead of the pose modality, leads to severe degeneration.

C. Additional Results

C.1. Image-First Asynchronous Sampling

We primarily accelerate the denoising of the pose modality to reduce mutual ambiguity during the sampling process. However, the effectiveness of asynchronous sampling for faster multi-view images remains an open question. To explore this, we conduct an experiment where images' denoising is performed at an accelerated rate with $\delta = 0.2$. We illustrate the generated multi-view images from such image-first asynchronous sampling in Fig. A.1. As shown in the figure, accelerating the denoising of multi-view images, rather than the pose modality, results in a significant failure in generation. This observation suggests that, unlike the pose modality, the image modality is not as robust to accelerated denoising, leading to severe artifacts.

C.2. Additional Qualitative Comparison

We present additional qualitative comparison results of text-to-3DGS in Fig. A.2. Consistent with Fig. 5, our VideoRFSplat can generate more realistic and detailed scenes than baselines without relying on SDS refinements.

C.3. Additional Qualitative Results

Figure A.3, A.4, and A.5 illustrate additional qualitative results of VideoRFSplat. As shown in the results, VideoRFSplat can generate high-quality 3DGS aligned with text inputs.

C.4. Results on Camera Conditioned Generation

Due to the limited space, we hereby supplement qualitative results on camera conditioned generation. Figure A.6 shows that VideoRFSplat accurately generates images following camera trajectories while aligned with text prompts.

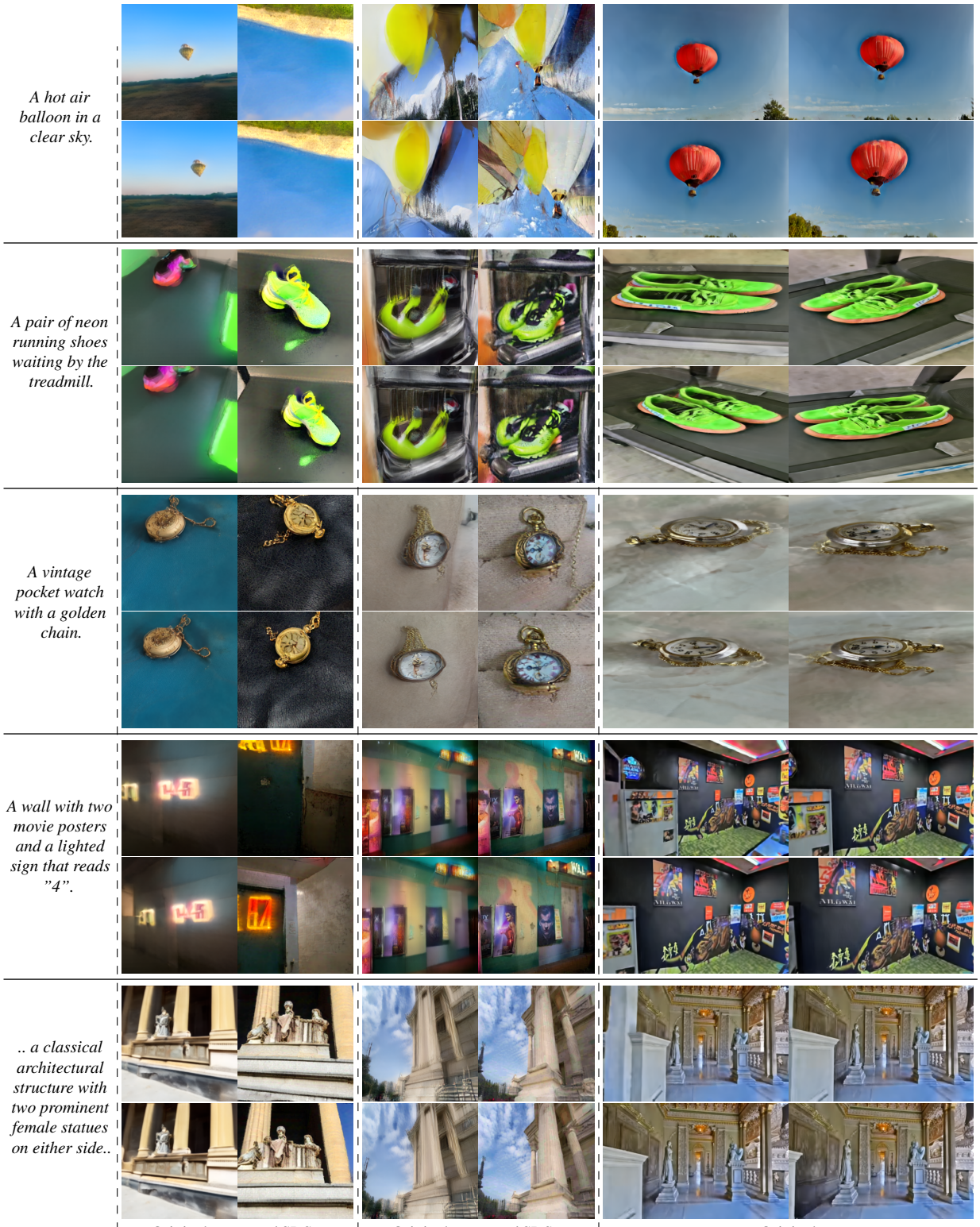

Figure A.2. **Additional qualitative comparison with Director3D [6] and SplatFlow [4].** Our VideoRFSplat generates more realistic scenes compared to baselines without relying on SDS++ [6].

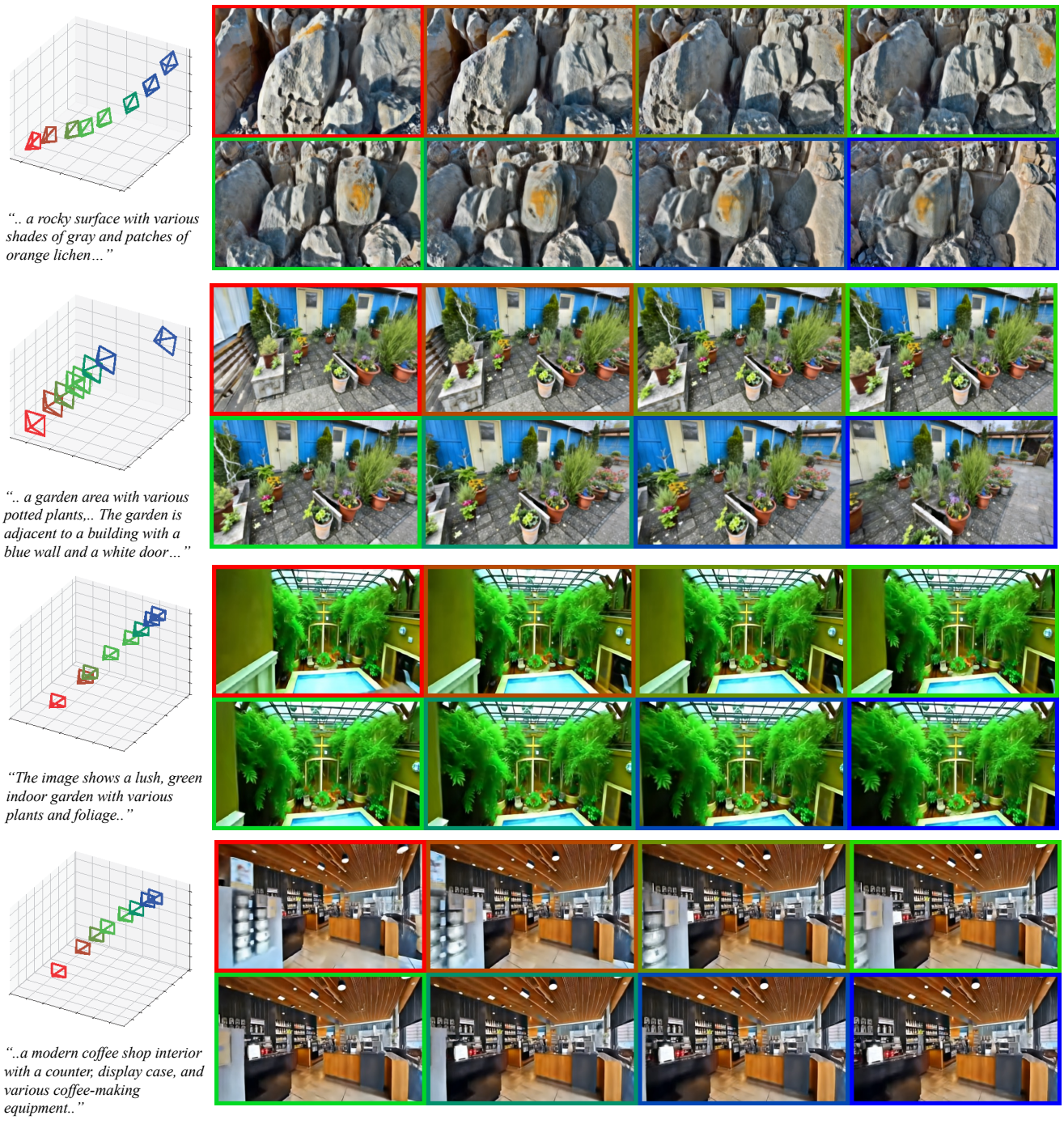

Figure A.3. **Additional qualitative results.**We present eight rendered scenes along with their corresponding camera poses from text prompts, with image border colors indicating the respective cameras.

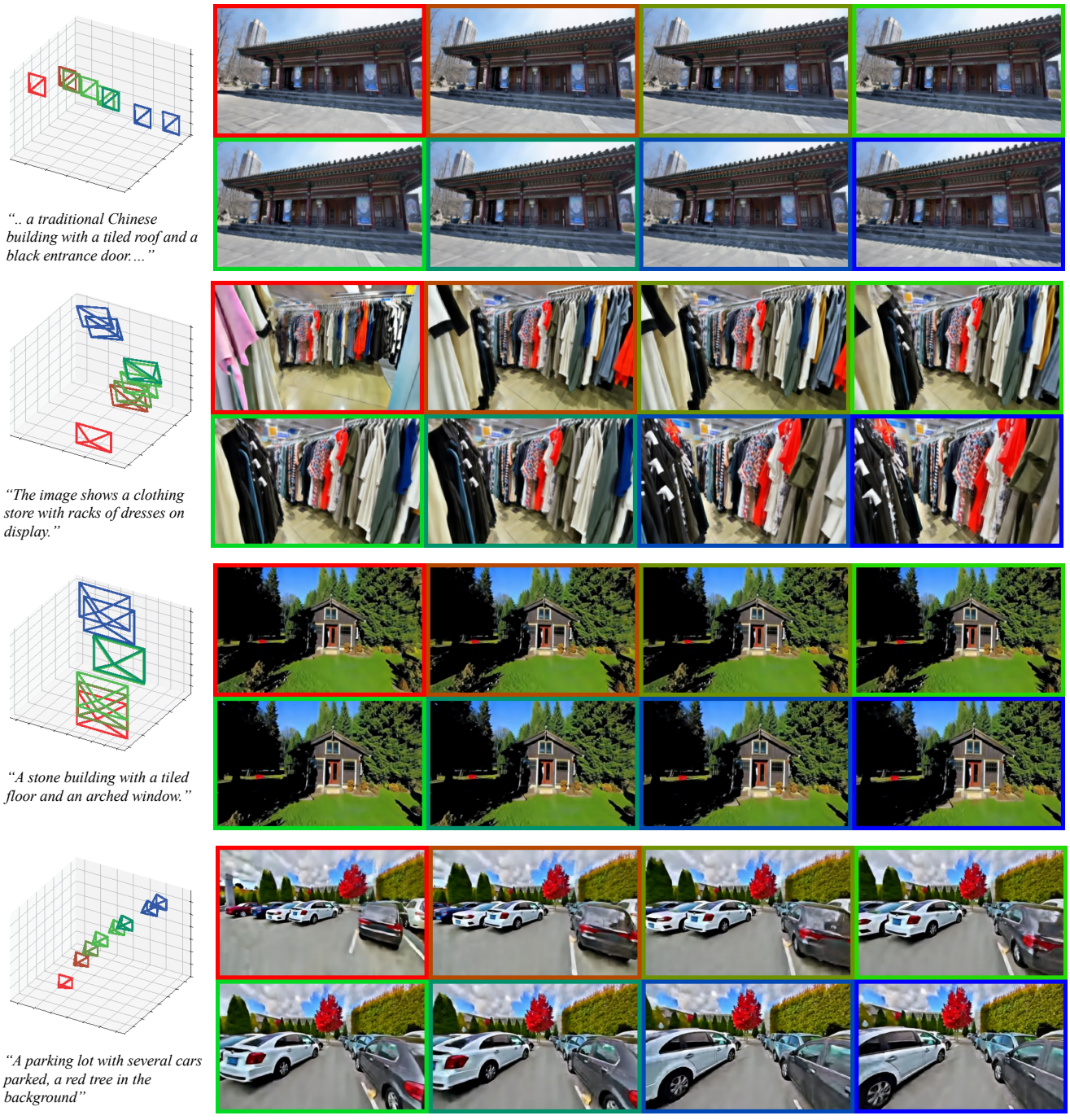

Figure A.4. **Additional qualitative results.** We present eight rendered scenes along with their corresponding camera poses from text prompts, with image border colors indicating the respective cameras.

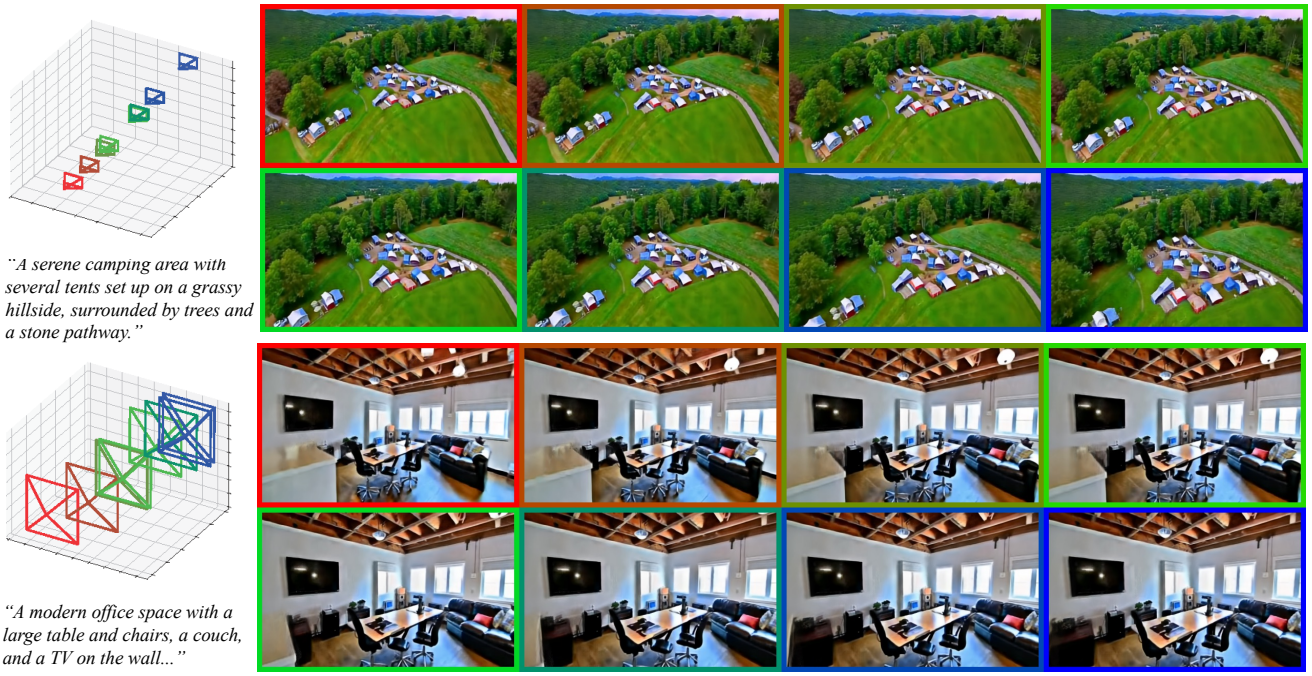

Figure A.5. **Additional qualitative results.**We present eight rendered scenes along with their corresponding camera poses from text prompts, with image border colors indicating the respective cameras.

Figure A.6. **Qualitative results on camera-conditioned generation.** We present generated multi-view images from given text and camera trajectory conditions. VideoRFSplat can perform camera-conditioned generation.

References

[1] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, pages 1692–1717. PMLR, 2023. 1

[2] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2025. 1

[3] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 2

[4] Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. Splatflow: Multi-view rectified flow model for 3d gaussian splatting synthesis. *arXiv preprint arXiv:2411.16443*, 2024. 1, 2, 3

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[6] Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3d: Real-world camera trajectory and 3d scene generation from text. *Advances in Neural Information Processing Systems*, 37:75125–75151, 2025. 3

[7] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 2

[8] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 2

[9] 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. *Journal of machine learning research*, 21(140):1–67, 2020. 1

[10] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 1

[11] Genmo Team. Mochi 1. https://github.com/genmoai/models, 2024. 1

[12] Sangmin Woo, Byeongjun Park, Hyojun Go, Jin-Young Kim, and Changick Kim. Harmonyview: Harmonizing consistency and diversity in one-image-to-3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10574–10584, 2024. 1

[13] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 1

[14] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 1

[15] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 1

[16] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. *arXiv preprint arXiv:2402.14817*, 2024. 1

[17] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2