Supplementary of "RayZer: A Self-supervised Large View Synthesis Model"

Hanwen Jiang[1] Hao Tan[2] Peng Wang[2] Haian Jin[3] Yue Zhao[1] Sai Bi[2]
Kai Zhang[2] Fujun Luan[2] Kalyan Sunkavalli[2] Qixing Huang[1] Georgios Pavlakos[1]
[1]The University of Texas at Austin [2]Adobe Research [3]Cornell University

A. Experimental Details

In this section, we introduce more details of RayZer.

Objaverse Data Details. We render Objaverse as continuous videos for training and evaluation. The frames are rendered with corresponding cameras on a unit sphere with a constant distance to the object center. Specifically, we render about 70 frames for azimuth 0° to 360°, where the elevation is randomly sampled between -20° to 60° for each shape instance. We sample frames with the distance between the first frame and the last frame being 50 to 65, covering the camera azimuth rotation for about one cycle.

Camera Interpolation Details. For the experiment of interpolating predicted cameras, we use Spherical Linear Interpolation (Slerp) for interpolating the camera pose rotation. This is based on the fact that the camera of Objaverse is moving at a constant speed. Thus, Slerp ensures the correct rotation interpolation. We then find the location on the unit sphere that corresponds to this interpolated rotation angle. Thus, we ensure the interpolated cameras are still on the unit sphere, which matches the camera sampling rule for rendering. In conclusion, this interpolation assumes that 1) the camera is moving in a constant speed, and 2) the rule of sampling camera location is known. Thus, this interpolation is only applicable to the synthetic Objaverse data, and does not apply to DL3DV and RealEstate.

More Training Details. For all transformer layers in RayZer, we apply QK-Norm [4] to stabilize the training. We use a latent dimension of 768 for RayZer and all baselines methods. RayZer and LVSM both use a latent set scene representation with 3072 tokens. We use mixed precision training [8] with BF16, further accelerated by FlashAttention-V2 [3] of xFormers [7] and gradient checkpointing [2].

We train RayZer and all baselines with the same training protocol. We use 32 A100 GPUs with a total batch size of 256. During training, we warm up with 3000 iterations, using a linearly increased learning rate from 0 to $4e-4$. We apply a cosine learning rate decay, while the final learning rate is $1.5e-4$. We train all baselines with $50,000$ steps. We clip the gradient with norm larger than 1.0. We follow all other hyper-parameters of LVSM.

More Model Details. Following LVSM, we do not use bias terms in linear and normalization layers. We also apply the depth-wise initialization for transformer layers.

Ablation details. In Table 7 (2), we use a two-layer MLP to encode the camera pose and intrinsics back to a latent pose representation in $\mathbb{R}^d$. In detail, for the predicted pose of each image (in 6D representation [11]), and the camera intrinsics (as the 4-dimensional focal length and principal points of x-axis and y-axis), we first concatenate them, getting a 10-dimensional pose representation. Then, we use the MLP to map it as a high-dimensional pose feature token. To predict the target views, we use a set of learnable patch-aligned spatial tokens shared across all target images as the initialization. Thus, the rendering decoder takes in the spatial tokens, the scene tokens, and the pose token. After using transformer for updating, we use the updated spatial tokens to regress the pixel values.

B. RayZer Training with Continuous Inputs

RayZer takes in multi-view image inputs, which can be sampled from either continuous video frames or an unordered image set. In this section, we present two design choices to improve self-supervised learning on video frames input.

Canonical View Selection. Prior works [5, 9] usually select the first image in an image sequence as the canonical view. In contrast, we select the frame at the middle time-step as canonical. In this context, the pose prediction MLP_{pose} initialized with a zero mean for its weights will have a small pose data variance. Otherwise, when using the first frame as canonical, the variance can be much larger. Note that this difference in pose variance can be easily handled with ground-truth camera supervision, thus, prior works choose the first image as the canonical view. However, this is more important for unsupervised methods, like RayZer.

Curriculum. We gradually increase the training difficulty by sampling video frames with an increasing distance range. With proper initialization of the model for camera pose estimation, it first learns from images with small camera base-

		Even Sample			Random Sample		
		PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
(0)	RayZer	**24.36**	**0.757**	**0.209**	**23.72**	**0.733**	**0.222**
(1)	first frame as canonical	23.86	0.736	0.224	23.78	0.737	0.225
(2)	no curriculum	23.87	0.734	0.226	23.87	0.735	0.226

Table 1. **Ablation study of RayZer techniques to train on continuous video frames**. (1) is a variant choosing the first image in the sequence as the canonical view, rather than choosing the middle frame. (2) does not use the frame sampling curriculum.

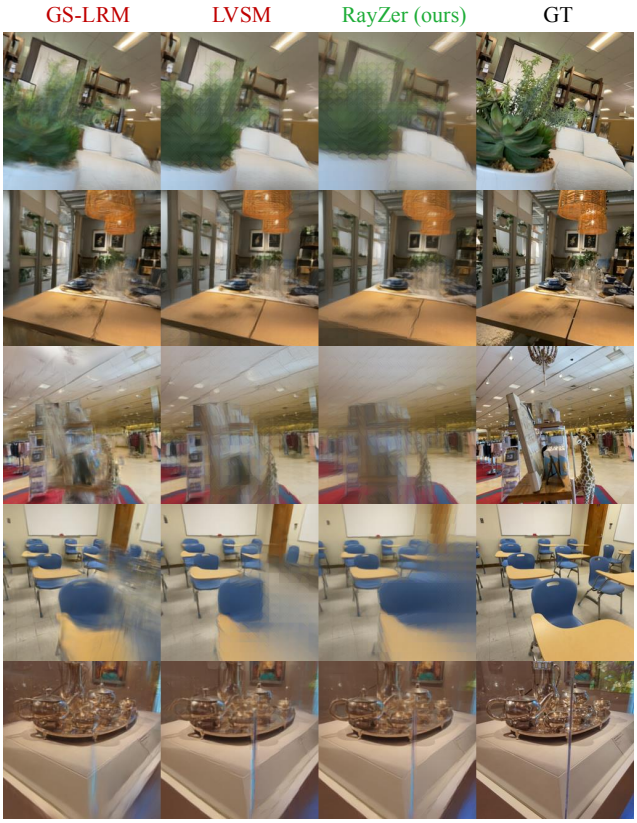

Figure 1. **Visualization of RayZer failure cases** on DL3DV.

lines, benefiting the training with larger camera baselines, that follows. In detail, we use a curriculum with a frame sampling range of 48-64, 96-128, and 24-32 at the beginning of training for DL3DV, RealEstate, and Objaverse, respectively. The frame sampling range is linearly increased to 64-96, 128-192, and 48-65 at the end of training for DL3DV, RealEstate, and Objaverse, respectively. The final frame sampling range is also used for the evaluation. The sampling ranges are set based on the difficulty (camera baseline) of each dataset, following prior works [1, 6, 9, 10, 12].

Experiments. We include ablations in Table 1, where removing any of the previously discussed techniques leads to a degraded performance. This demonstrates the effectiveness of our designs of selecting canonical view and using frame sampling curriculum during training.

C. More Results

In this section, we present more results for discussing RayZer's failure cases and show more visualizations.

Failure Case Pattern. We observe that RayZer can fail when dealing with fine-grained geometry, complicated materials, and occlusions. We present the visualization in Fig. 1. In detail, RayZer fails to handle complicated plant geometry (first row). This failure is not specific to RayZer – GS-LRM and LVSM also can not handle it. In the second and last row, RayZer fails to handle multiple stacked glasses and is not perfect on the specular reflection of the silver teapots. GS-LRM and LVSM also demonstrate imperfect results. In the third and fourth rows, all methods, including RayZer, fail to handle occlusions, where the side view of the exhibition stand is not observed in input views (third row), and the chairs in the fourth row have self-occlusion.

More Comparisons. We present more visualization results, comparing with GS-LRM and LVSM in Fig. 2. RayZer generally performs on par, while being a self-supervised method that does not require any camera pose annotations.

More Visualization. We present more visualization results comparing with ground-truth novel views in Fig. 3-5.

D. More Discussion

Why does RayZer demonstrates strong novel view synthesis quality while the fine-tuned pose estimation is not perfect (Table 7 in the main manuscript)? We conjecture RayZer's pose space jointly learns the actual pose information and 3D-aware video frame interpolation at the same time. On datasets with small camera baselines (RealEstate), which is easy to learn, RayZer mainly focuses on learning actual pose estimation. This is supported by the accurate pose estimation performance on RealEstate. On datasets that have large camera baselines (DL3DV and Objaverse), where pose estimation is harder to learn with only self-supervision, RayZer also leverages video interpolation cues together with pose estimation to perform novel view synthesis.

Thus, the method to further enhance disentanglement of interpolation and pose estimation would be an important future direction. In RayZer, using unordered image sets for training and using continuous video frames for training can be two extreme cases in the spectrum for learning this disentanglement. In detail, learning on continuous video frames with using image index positional embeddings strongly encourages the camera pose local smoothness to enhance training performance; while training on unordered image sets fully discards this prior. Finding a balance between the two and designing a better method to encourage the camera pose local smoothness is a promising avenue to solve the structure-and-motion problem with learning `SE(3)` camera poses in the real-world space.

References

[1] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, pages 19457–19467, 2024.

[2] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[3] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[4] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020.

[5] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. *arXiv preprint arXiv:2310.01410*, 2023.

[6] Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, et al. Megasynth: Scaling up 3d scene reconstruction with synthesized data. *arXiv preprint arXiv:2412.14166*, 2024.

[7] 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.

[8] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *International Conference on Learning Representations*, 2018.

[9] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023.

[10] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024.

[11] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[12] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint 2410.12781*, 2024.

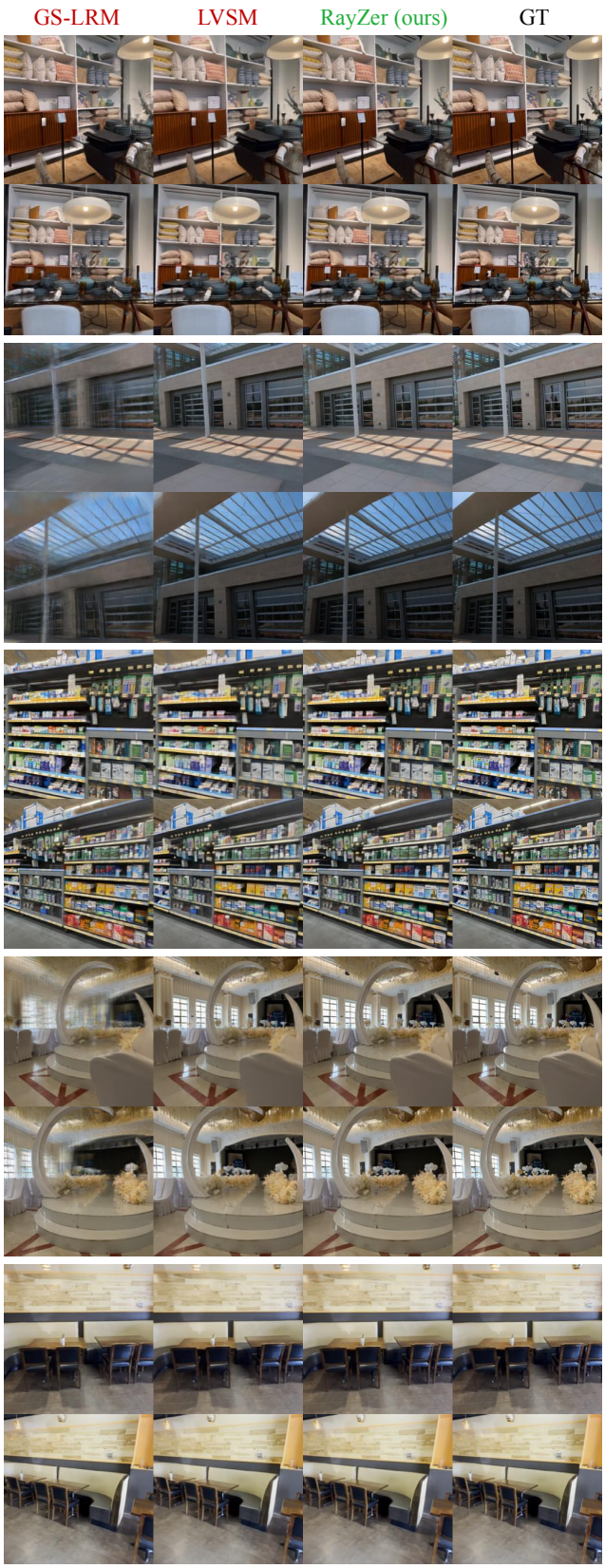

Figure 2. **Visual compression of RayZer and "oracle" methods** on DL3DV.

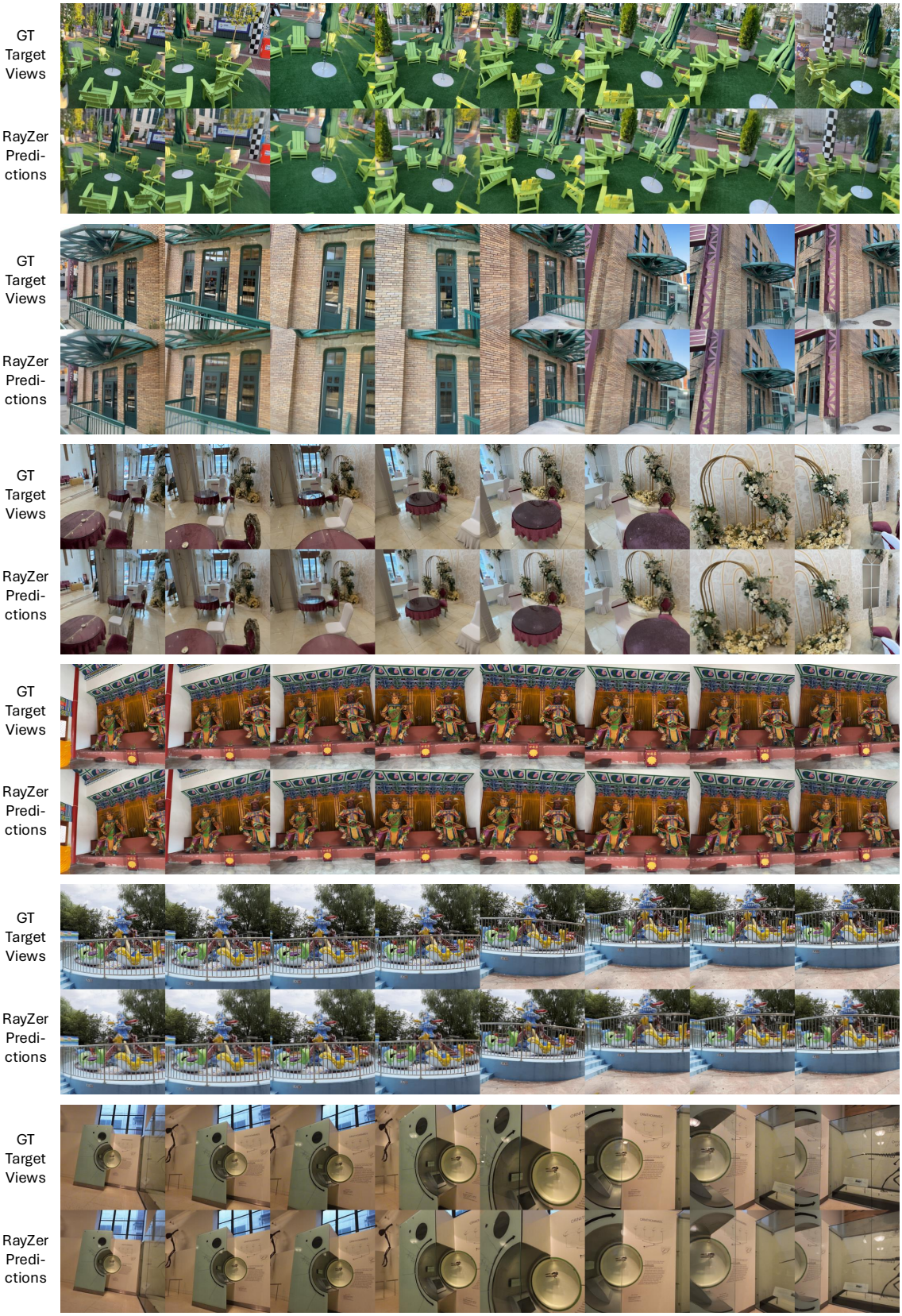

Figure 3. **Visual compression with ground-truth novel views** on DL3DV. The first row of each sample is the target novel views, and the second row are images rendered by RayZer.

Figure 4. **Visual compression with ground-truth novel views** on RealEstate. The first row of each sample is the target novel views, and the second row are images rendered by RayZer.

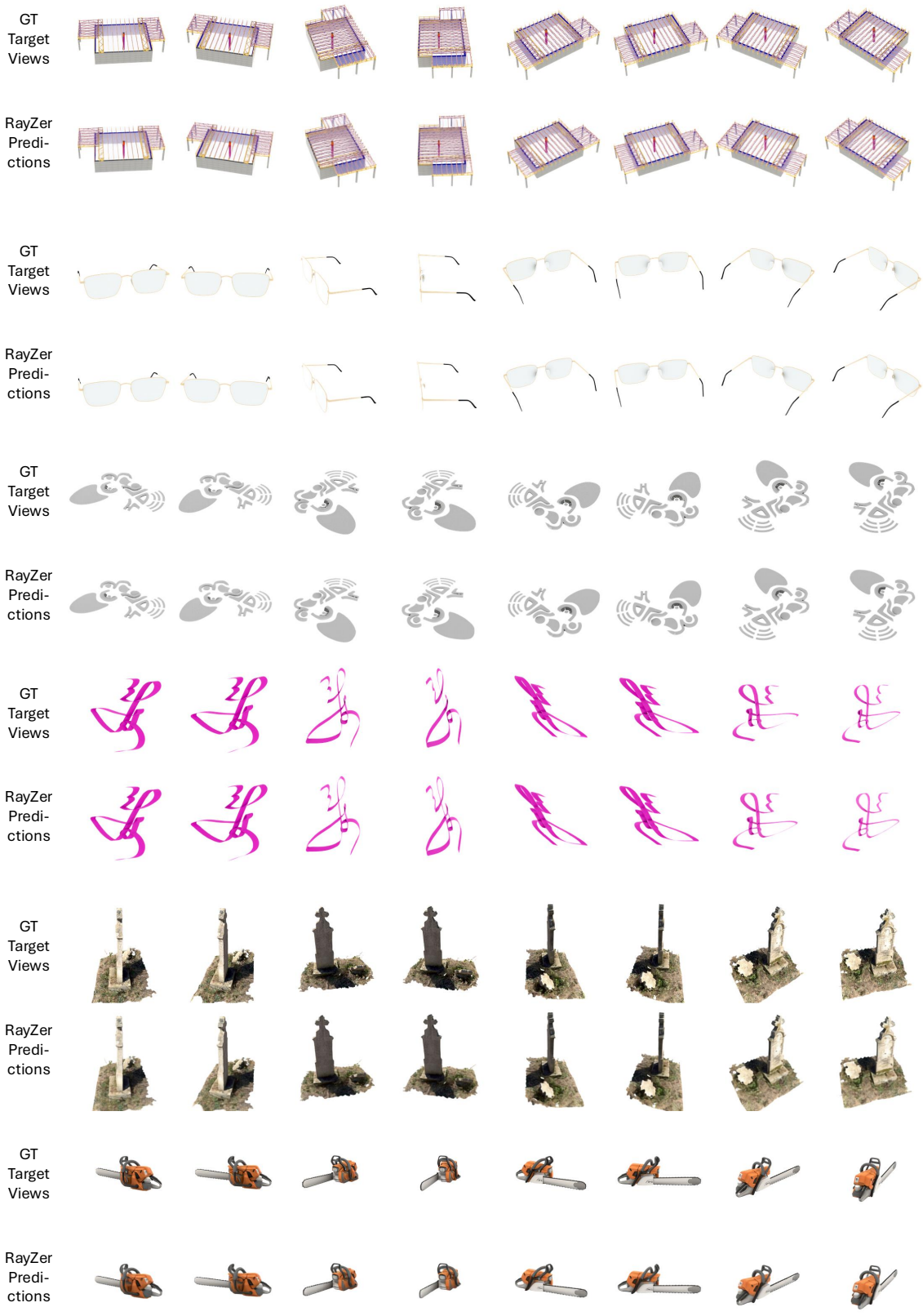

Figure 5. **Visual compression with ground-truth novel views** on Objaverse. The first row of each sample is the target novel views, and the second row are images rendered by RayZer.