Learning 3D Reconstruction with Priors in Test Time

Supplementary Material

1. More Ablation Studies

1.1. Prediction Compatibility Objective

We ablate on the heuristic rules designed for the prediction compatibility objective, i.e., the rendering loss implemented with 2DGS rasterization. First, we try different scale factors α between the final 2DGS radius and the point map gradient magnitude, i.e., $\mathbf{r}_{i,x,y} = \alpha|\mathbf{n}_{i,x,y}[z]| \cdot [\|\nabla_x \mathbf{p}_{i,x,y}\|, \|\nabla_y \mathbf{p}_{i,x,y}\|]$. We also report the results of optimizing the 2DGS parameters directly, with MVT's predictions as the initialization. The results are shown in Tab. 1. First, our method is robust to a wide range of radius scale factors, from 0.05 to 5. More specifically, smaller radius scale factor performs slightly better. With direct 2DGS parameter optimization, the performance deteriorates significantly. This observation justifies that our heuristic rules are necessary for the success of our method by leaving the adaptation to the MVT's parameters.

1.2. Fine-tuning Strategy

We ablate on different LoRA ranks, including 1, 4 (default), and 16. The results are shown in Tab. 1. We find that our method achieves similar performance with different LoRA ranks. At the same time, even with 1 LoRA rank, our method already outperforms the base image-only model VGGT.

Table 1. **Ablation: Target view rendering loss and LoRA rank (ETH3D).** Acc., Comp., and N.C. are reported (lower is better for Acc./Comp., higher is better for N.C.).

Setting	Acc. ↓		Comp. ↓		N.C. ↑	
	Mean	Med.	Mean	Med.	Mean	Med.
VGGT (base)	0.280	0.185	0.305	0.182	0.853	0.950
Rendering loss						
Radius scale = 0.5 (def.)	**0.114**	**0.060**	**0.116**	**0.054**	**0.901**	**0.984**
Radius scale = 0.05	0.122	0.066	0.124	0.058	0.895	0.983
Radius scale = 5	0.143	0.072	0.207	0.123	0.871	0.972
Optimize 2DGS params	0.895	0.771	0.809	0.476	0.565	0.598
LoRA rank						
1	0.122	0.072	0.193	0.123	0.880	0.963
4 (default)	**0.114**	**0.060**	**0.116**	**0.054**	**0.901**	**0.984**
16	0.127	0.074	0.135	0.073	0.894	0.983

2. Implementation Details

For reconstruction tasks, we only use photometric loss to realize the prediction compatibility objective. We set rotation loss weight $\mu_1 = 1.0$, translation loss weight $\mu_2 = 2$, and focal length loss weight $\mu_3 = 0.01$. For ETH3D and 7-Scenes datasets, we use the a weaker photometric loss weight, i.e., $\lambda_1 = 0.2$. For DTU and NRGBD datasets, we use a stronger photometric loss weight, i.e., $\lambda_1 = 1.0$. All the datasets set test-time training steps as 40 except DTU adopts 50 steps due to more challenging scenes. The learning rates are set to 5×10^{-4}, 1×10^{-4}, 2×10^{-4}, and 1×10^{-3}, for ETH3D, NRGBD, DTU, and 7-Scenes datasets, respectively. For the camera pose estimation task, we only use the geometric loss to realize the prediction compatibility objective. The test-time training steps are set to 40 and learning rate is set to 2×10^{-4}.

3. Robustness to the Prior Noise

We test the robustness of our method to camera pose and intrinsic noise. We perturb the camera pose and intrinsic parameters by adding a small random perturbation to the ground truth values. We report the results in Tab. 2. Although the performance deteriorates as the perturbation increases, our method still outperforms the base image-only model VGGT in most cases.

Table 2. **Noise robustness (ETH3D).** We report mean Acc., mean Comp., and mean N.C. (lower is better for Acc./Comp., higher is better for N.C.).

Rot pert. (deg)	Trans pert. (%)	Focal pert. (%)	Acc. ↓	Comp. ↓	N.C. ↑
VGGT (base)			0.280	0.305	0.853
0	0	0	0.1765	0.1816	0.8822
1	0	0	0.1755	0.1755	0.8791
3	0	0	0.2111	0.2108	0.8669
5	0	0	0.2344	0.2143	0.8569
0	1	0	0.1702	0.1700	0.8790
0	5	0	0.2383	0.2013	0.8585
0	10	0	0.2816	0.2381	0.8489
0	0	1	0.1805	0.1657	0.8822
0	0	5	0.2328	0.2083	0.8740
0	0	10	0.2549	0.2654	0.8527
1	1	1	0.1643	0.1409	0.8850
3	5	5	0.2588	0.2112	0.8625
5	10	10	0.2901	0.2708	0.8413

4. Test-time Inference Time and Limitations

In this section, we report the test-time inference time of our method on the ETH3D dataset under different settings. As shown in Tab. 3, inference time is a limitation of our method. By trading off some efficiency, we improve the performance of our method on a series of benchmarks. We

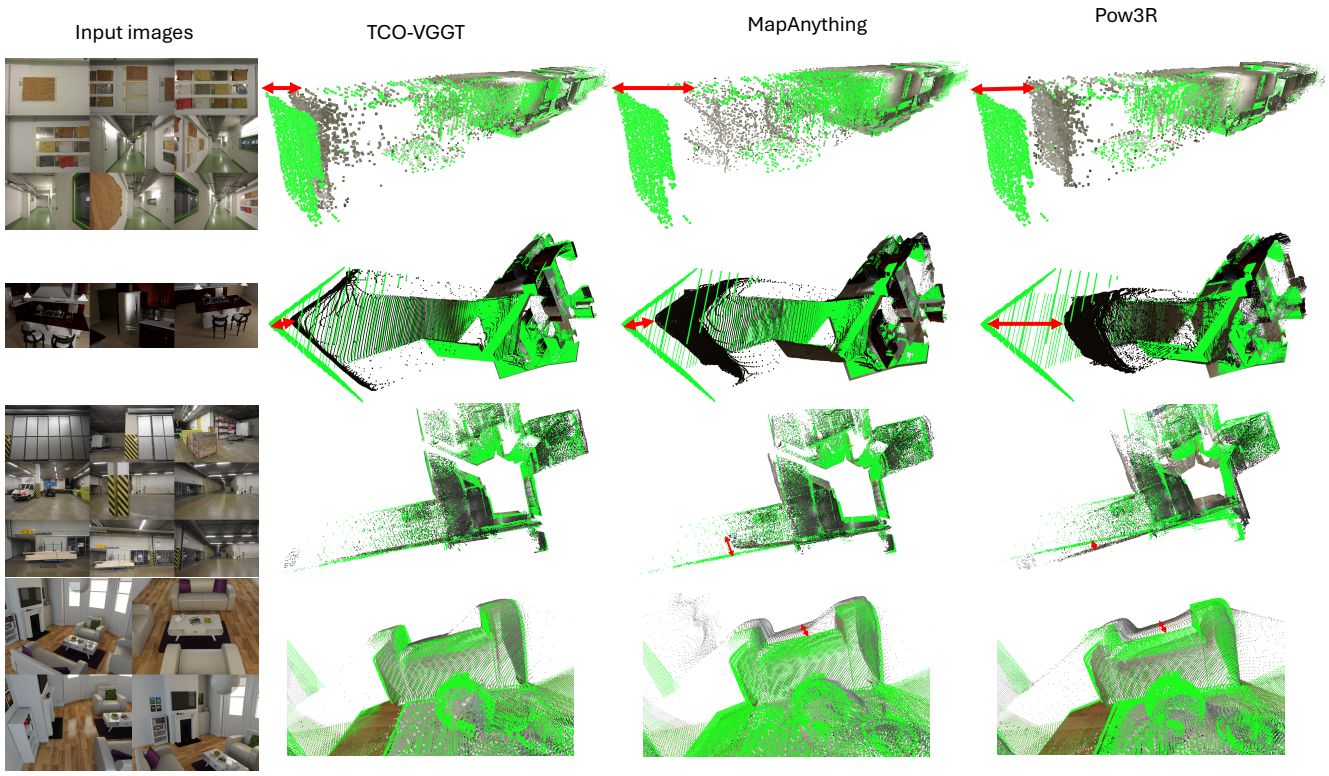

Figure 1. **Fine-grained Qualitative Results.** We compare TCO-VGGT with prior-aware feed-forward methods, including Pow3R and MapAnything. In each grid cell, the predicted geometry is overlaid with the ground truth geometry, whose points are shown in green. Discrepancies between the predicted and ground-truth geometries are highlighted by red double arrows, whose lengths indicate the magnitude of the errors. TCO-VGGT exhibits much smaller discrepancies than MapAnything and Pow3R in both scene structure and boundary regions.

will leave the exploration of more efficient test-time methods for future work.

Table 3. **Test-time training steps vs. time (10 views).** The reported times are wall-clock seconds per scene. The image resolution is 392(H)×518(W). The base model is VGGT. The GPU is NVIDIA TITAN RTX.

TTT Steps	Render Loss	Avg (s)	Std (s)	Min (s)	Max (s)
0	✗	1.301	0.198	1.104	1.499
10	✗	43.630	0.218	43.412	43.848
10	✓	45.598	0.440	45.158	46.037
40	✓	161.547	1.424	160.123	162.971

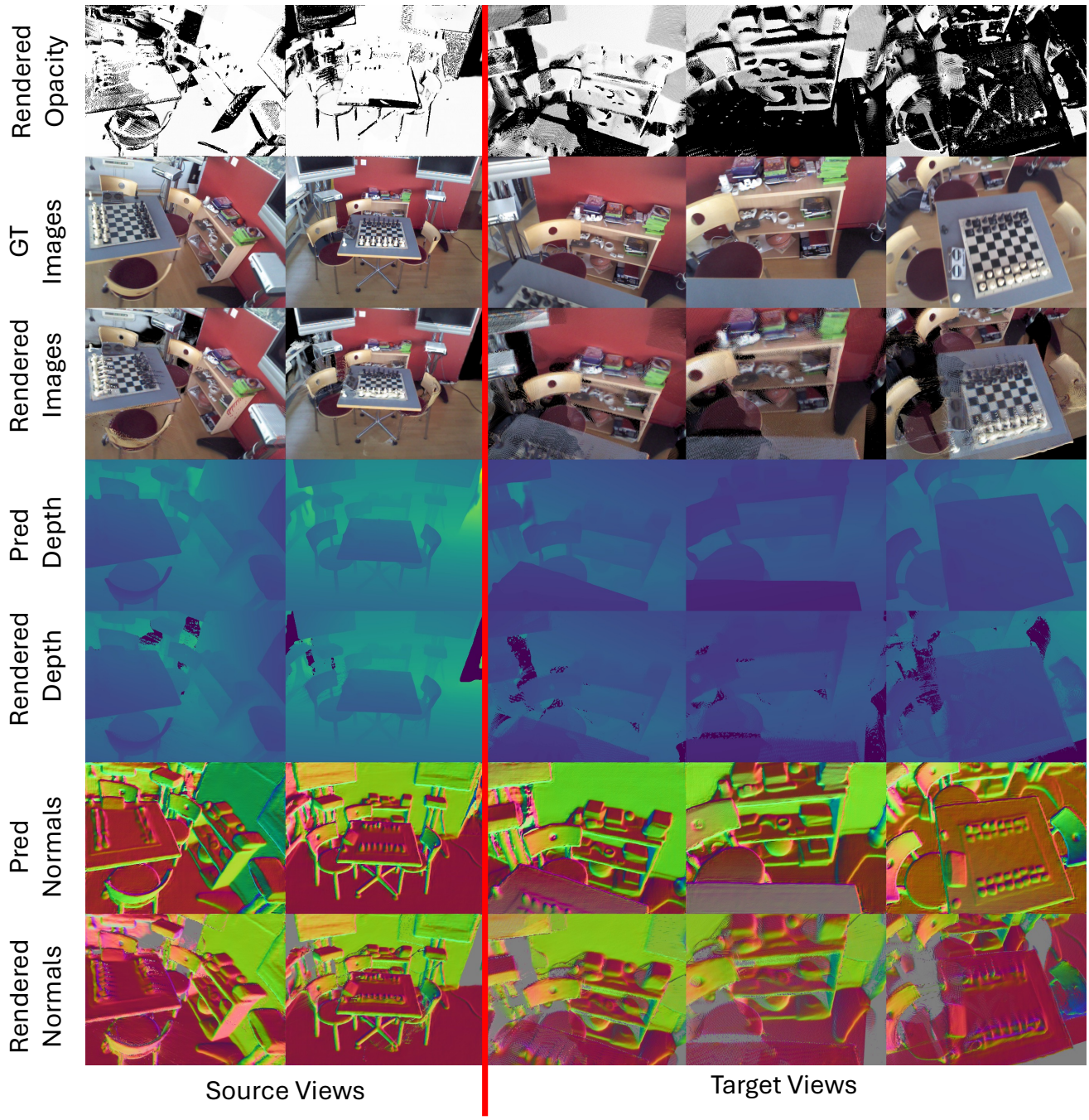

Figure 2. **2DGS Rendering Visualization.** We visualize the 2DGS rendering process for one scene from 7-Scenes. As shown in the Rendered Image row, our 2DGS heuristic parameterization produces rendered images that closely match the ground-truth images. We also compare the depth maps and normal maps rendered from 2DGS with the corresponding ground-truth depth and normal maps, i.e., those directly predicted from the MVT views.