ACT-R: Adaptive Camera Trajectories for Single-View 3D Reconstruction

Supplementary Material

6. Additional Results

We provide additional qualitative results from the GSO dataset in Figures 14 and 15.

7. Estimation of Camera Poses and Object Bounding Boxes

We train a network to predict the camera elevation and distance from a single image using Objaverse data. Specifically, we render different views of an object and obtain the ground-truth (GT) camera parameters for each rendered view. The network backbone is a ResNet50 [15], with fully connected layers adapted to predict (1) the elevation e_0 and the camera distance d; and (2) the edge lengths in three dimensions (w, h, l) of the bounding box.

Instead of directly aligning the predicted elevation e_0' and distance d' with their GT counterparts e_0 and d, we adopt an alternative approach. From each shape, we sample a point cloud P with $1,024$ points. The camera elevation and distance are transformed into a matrix Θ, which performs the coordinate transformation from world coordinates to camera coordinates. For the predicted and ground-truth transformation matrices Θ' and Θ, the objective function minimizes the discrepancy between the transformed point clouds:

$$L_{cam} = ||P\Theta' - P\Theta||_2, \tag{2}$$

which measures the error by comparing the rotated point clouds.

For the bounding boxes prediction, we directly use MSE:

$$\mathcal{L}_{bbox} = ||w - w'||_2 + ||l - l'||_2 + ||h - h'||_2, \tag{3}$$

where (w', h', l') are the predicted edge lengths.

The overall training loss $\mathcal{L}$ combines the bounding box loss $\mathcal{L}_{bbox}$ and the camera alignment loss L_{cam}:

$$\mathcal{L} = \mathcal{L}_{bbox} + L_{cam}. \tag{4}$$

8. Robustness Against Camera Pose Errors

Our method demonstrates notable robustness to errors in camera pose estimation. For example, as shown in Fig. 10, the predicted camera distance and elevation deviate somewhat from the GT values. Despite this, our approach produces a trajectory closely resembling the one derived from GT camera poses, with only a minor reduction in the visibility score defined in Equation 2 of the main paper.

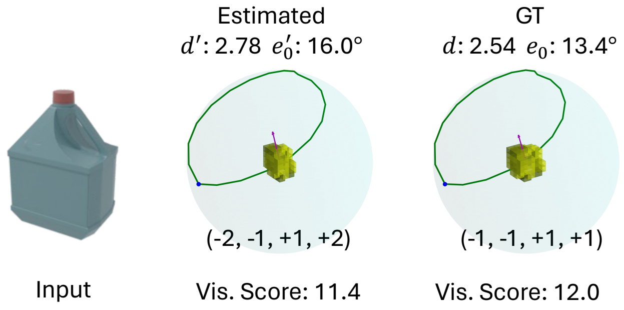

Figure 10. Robustness to camera pose prediction errors. The left and right show our predicted camera trajectories using predicted and GT camera poses, respectively. The numbers in brackets show the elevation changes in each segment of the orbit. "Vis. Score" denotes the visibility score defined in Equation (1) from the main paper.

9. Visibility Set Calculation

To calculate the visibility set $\psi(\pi(t))$ in Equation (1) from the main paper, we perform two key checks for each difference block: (1) a field of view check and (2) an occlusion check. The field of view check ensures blocks lie within the camera's viewing angle (33.6°), computed as the angle between the camera direction and the vector from camera to block center. The occlusion check determines if any other block intersects the line segment from the camera to the target block's center using a ray-box intersection algorithm. A block is considered visible only if it passes both checks. This visibility function allows us to evaluate each candidate trajectory by summing the weights of all visible blocks across all camera positions, leading to the selection of the optimal trajectory π^* that maximizes the visibility of important difference regions.

10. Calculation of Random Trajectory

Following the same setup as our adaptive trajectories, we apply a uniform azimuth angle change of $18°$ and divide the camera trajectory orbit into four segments. Instead of setting a constant elevation increment stepsize within each segment, we randomly sample an incrementation step with $[-5°, 5°]$ for each time of elevation change. To ensure the final trajectory forms a close loop, we apply the same mirroring and negating mechanism between segments.

11. Semantic Difference Map

We show additional examples on semantic difference map obatained by comparing feature differences between input

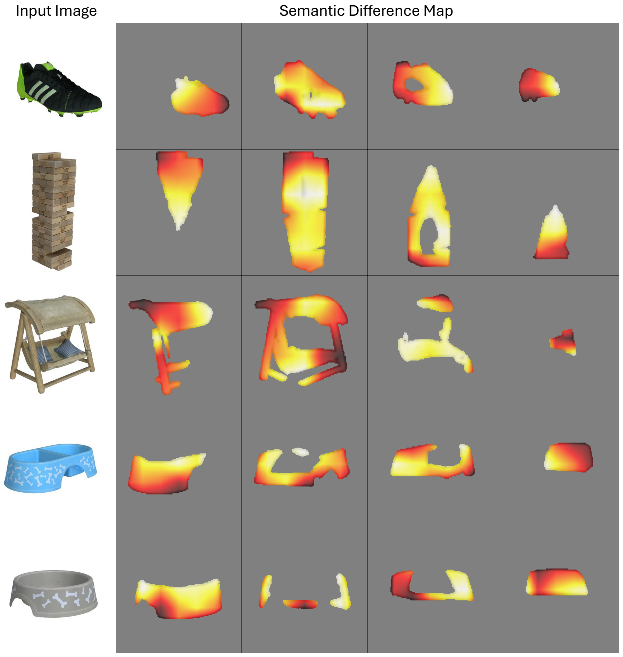

Figure 11. Visualization of semantic difference map computed form feature comparison between input image and the generated slice images by Slice3D [64]. The brighter color indicates higher semantic difference, and the darker colour indicates lower semantic differences.

image and the generated slices from Slice3D [64], as shown in Figure 11.

12. Alternative Feature Encoders

In our primary experiments, we utilized VGG16 [52] as the feature extractor to quantify semantic differences between slice images and the input image. To evaluate the impact of feature encoder architecture, we conducted ablation studies with DINOv2 [42] and CLIP [46] as alternative feature extractors. For the DINOv2 implementation, we extracted features from the final intermediate layer, resulting in a feature representation $\phi(\cdot) \in \mathbb{R}^{1536\times16\times16}$ while CLIP features remain in the same resolution as VGG. These features were then used to construct semantic difference maps following the same methodology applied to VGG features.

Qualitative and quantitative comparisons between static camera trajectory (SV3D), adaptive trajectory with CLIP, DINOv2, and VGG are presented in Figure 12 and Table 2, respectively. All methods were evaluated on the IM [73] reconstruction pipeline.

The results demonstrate that all feature extractors contribute to developing more effective camera paths, yielding improvements in both qualitative and quantitative metrics compared to static trajectories. Among the different feature encoder choices, VGG16 and CLIP exhibit similar performance with minimal differences in F1 scores, while DINOv2 falls slightly short.

Method	CD↓	F1↑	HD↓
SV3D_{IM} [60]	4.60	7.74	18.0
$\text{Ours}_{\text{CLIP}_{\text{IM}}}$	**3.93**	**9.99**	**16.1**
$\text{Ours}_{\text{DINO}_{\text{IM}}}$	3.97	9.91	16.3
Ours_{IM}	**3.93**	9.93	**16.1**

Table 2. Quantitative results of single-view 3D reconstruction on the full GSO dataset with static camera trajectory and adaptive trajectories with CLIP [46], DINOv2 [42] and VGG16 [52] as feature extractors, respectively. Numbers in bold are ranked first.

13. Coverage Metric

We further evaluate static, random, and adaptive trajectories using a coverage metric. Given the ground truth (GT) mesh as input, we render the GT mesh at 21 camera poses for each trajectory type. The visible area in each render is marked as visible on the corresponding mesh texture UV map, and the final coverage metric is measured as the accumulated visible region over the entire texturable UV map area. Averaged across all objects in the GSO dataset, the static trajectory achieved an 87.8% coverage rate, random trajectory 89.0%, adaptive trajectory with CLIP 90.3%, VGG16 90.5%, and DINOv2 93.4%.

While the coverage metric indicates the occlusion-revelation capability of different trajectories to some extent, we emphasize that it involves a trade-off with reconstruction quality. For example, using a greedy algorithm for extreme occlusion-revelation path-finding can achieve near 100% coverage, but it induces drastic elevation changes and degrades video generation quality due to sudden viewpoint transitions. Considering both metrics, feature encoders including VGG16 and CLIP strike an effective balance between coverage and quality.

14. Orbital Camera Trajectories

As described in the main paper, we implement a closed-loop orbital camera trajectory to optimize viewpoint sampling. The camera orbit is structured into four quarters, with camera elevation changing at a constant rate within each quarter. Our path planning algorithm determines the increment steps for the first two segments, while the final two segments mirror these changes with negated values to complete the loop.

This design approach is motivated by two key considerations:

- **Search Space Reduction:** Camera pose planning exists in a continuous space with virtually infinite candidate positions. By discretizing elevation changes, we significantly reduce the search space, enabling efficient execution of combinatorial optimization to determine an effective path.

- **Constraint Balance:** The closed-loop structure enforces zero total variation in elevation, creating a balance between exploring diverse elevation angles while remaining within the operational capabilities of the video diffusion model.

Our empirical observations indicate that unconstrained greedy algorithms—those without total variation constraints in elevation and closed-loop confinement—frequently produce steeply descending trajectories that result in distorted video generation outputs.

While we acknowledge the current limitations of our approach in exploring asymmetric objects, our method demonstrates significant improvements over static camera trajectories. We believe this work provides valuable insights for future research in reconstruction-centric camera path planning.

15. Complementary Role in Direct 3D supervision

Recent methods employing direct 3D supervision [71, 77] have demonstrated superior performance in 3D generation and reconstruction, achieving high-quality results with faster inference than multiview-based approaches. To investigate this further, we conduct an ablation study on Trellis [71] comparing single image input against multiview inputs sampled from orbital camera trajectories. For multiview evaluation, we sample 6 views on static camera trajectories and 6 views on adaptive camera trajectories, forming three comparison groups: Trellis, Multi-Trellis Static, and Multi-Trellis Adaptive.

The qualitative results in Figure 13 reveal that multiview input Trellis does not consistently outperform its single-view counterpart. This degradation can be attributed to blur and distortions introduced by hallucinated multiview images, which compromise geometric detail quality as evidenced in side-by-side comparisons. However, when input views exhibit significant self-occlusion—as observed in rows 1, 2, and 4 of the right column in Figure 13—additional multiviews enhance geometric awareness in occluded regions, supporting more regularized and occlusion-aware reconstruction. Notably, adaptive multiviews consistently outperform static multiviews in these scenarios due to their superior occlusion-revelation capabilities.

While these advances in direct 3D supervision might suggest that multiview reconstruction is becoming obsolete, we contend that both paradigms offer complementary strengths. Although multiview inputs do not universally improve results, particularly when hallucinated views suffer from distortion or inconsistency, our findings indicate that multiview generation and direct 3D reconstruction can synergistically complement each other when views are strategically selected. We hope this exploration of adaptive trajectory planning will inspire further research into harnessing the combined potential of these two paradigms.

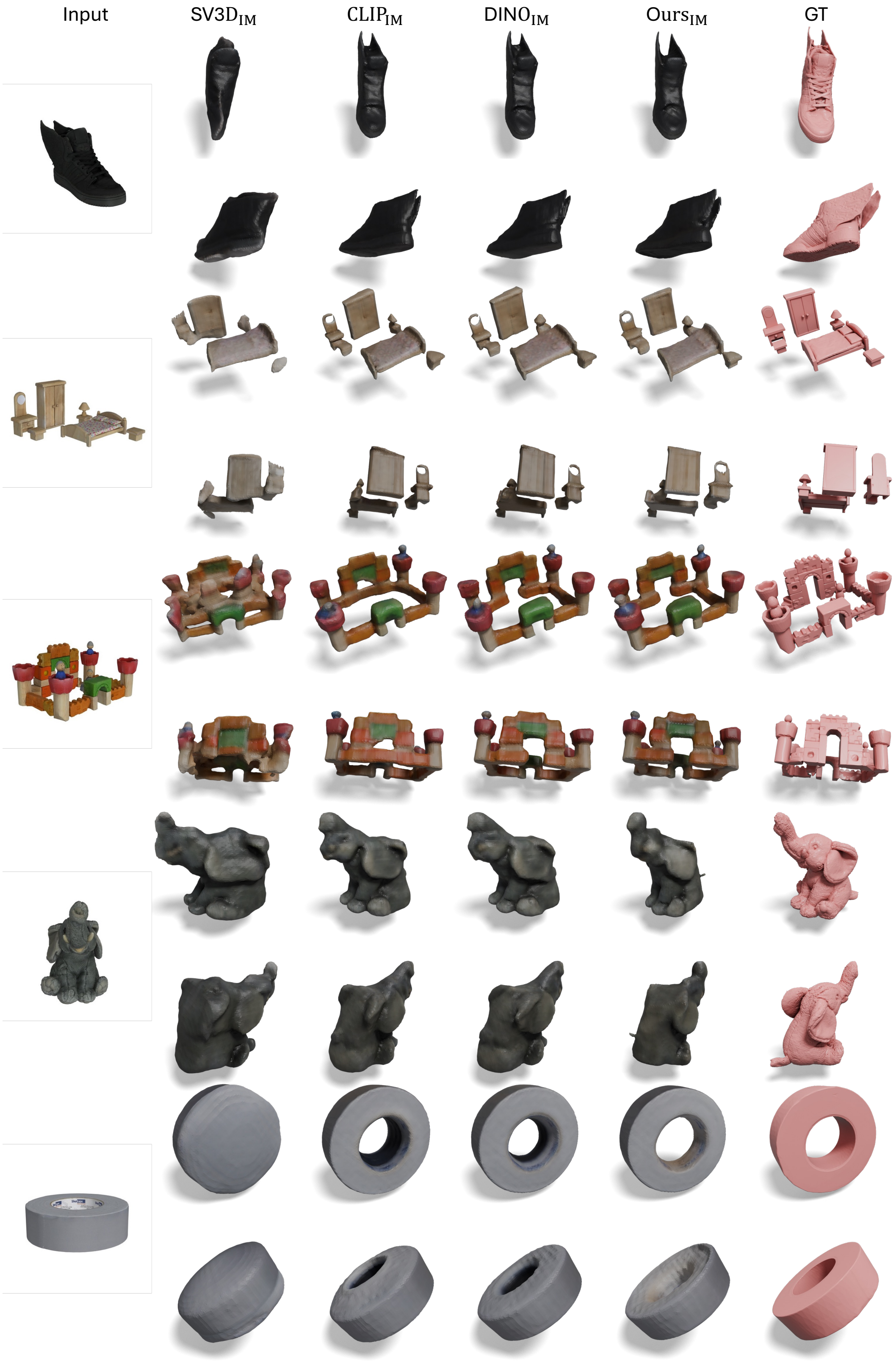

Figure 12. Qualitative visual comparisons between 3D reconstruction with static camera trajectory (SV3D), adaptive camera trajectory with CLIP [46], DINOv2 [42] (DINO) and VGG16 (Ours) as the feature extractor on the GSO dataset. Please zoom in for a closer inspection. All meshes are reconstructed with LRM. It can be seen that adaptive camera trajectory provides additional geometric awareness in comparison to static camera trajectory, regardless.

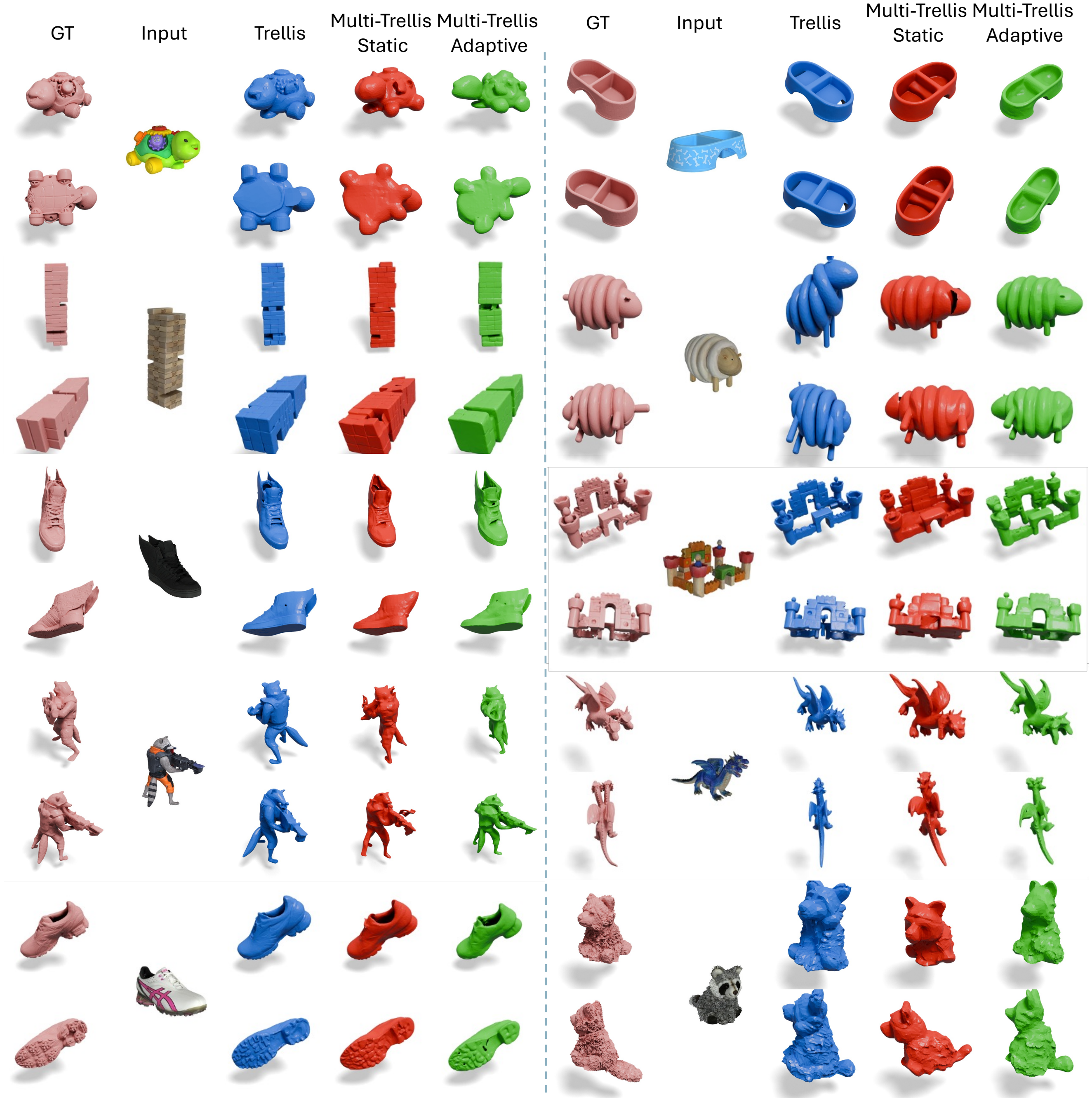

Figure 13. 3D reconstruction results obtained using Trellis [71] with single image input and multiview image inputs sourced from static and adaptive camera trajectories. Single-view inputs consistently produce cleaner results with richer geometric details, as multiview inputs introduce additional blur from hallucinated views, as universally observed across all samples. However, multiview inputs demonstrate advantages for handling self-occlusion scenarios. Notable examples include R1 (**R**ight column, row **1**), R2, and R4, where the dog bowl is reconstructed with a solid base, the distorted sheep model is regularized, and the dual-head feature of the dragon is properly emphasized. These cases highlight how strategic multiview selection can enhance reconstruction quality in geometrically challenging scenarios despite the general trade-off in detail fidelity.

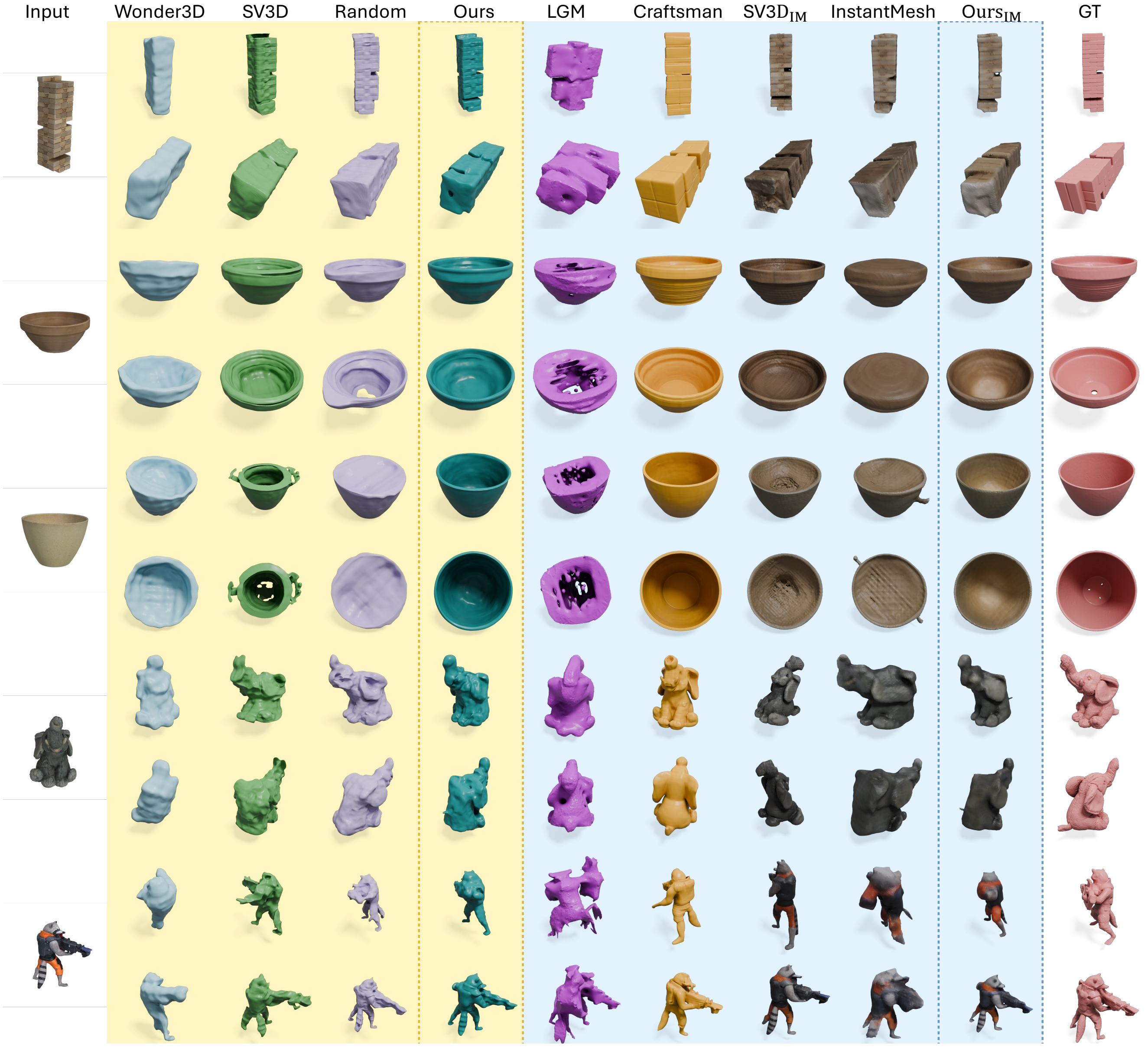

Figure 14. Additional qualitative visual comparisons between single-view 3D reconstruction methods on the GSO dataset. Please zoom in for a closer inspection. Meshes reconstructed with NeUS and LRM are in yellow and blue blocks, respectively. Our slice-guided approach demonstrates superior reconstruction of occluded regions, including bowl bottoms and extended features like elephant trunk. Please zoom in for a detailed comparison of the reconstruction quality.

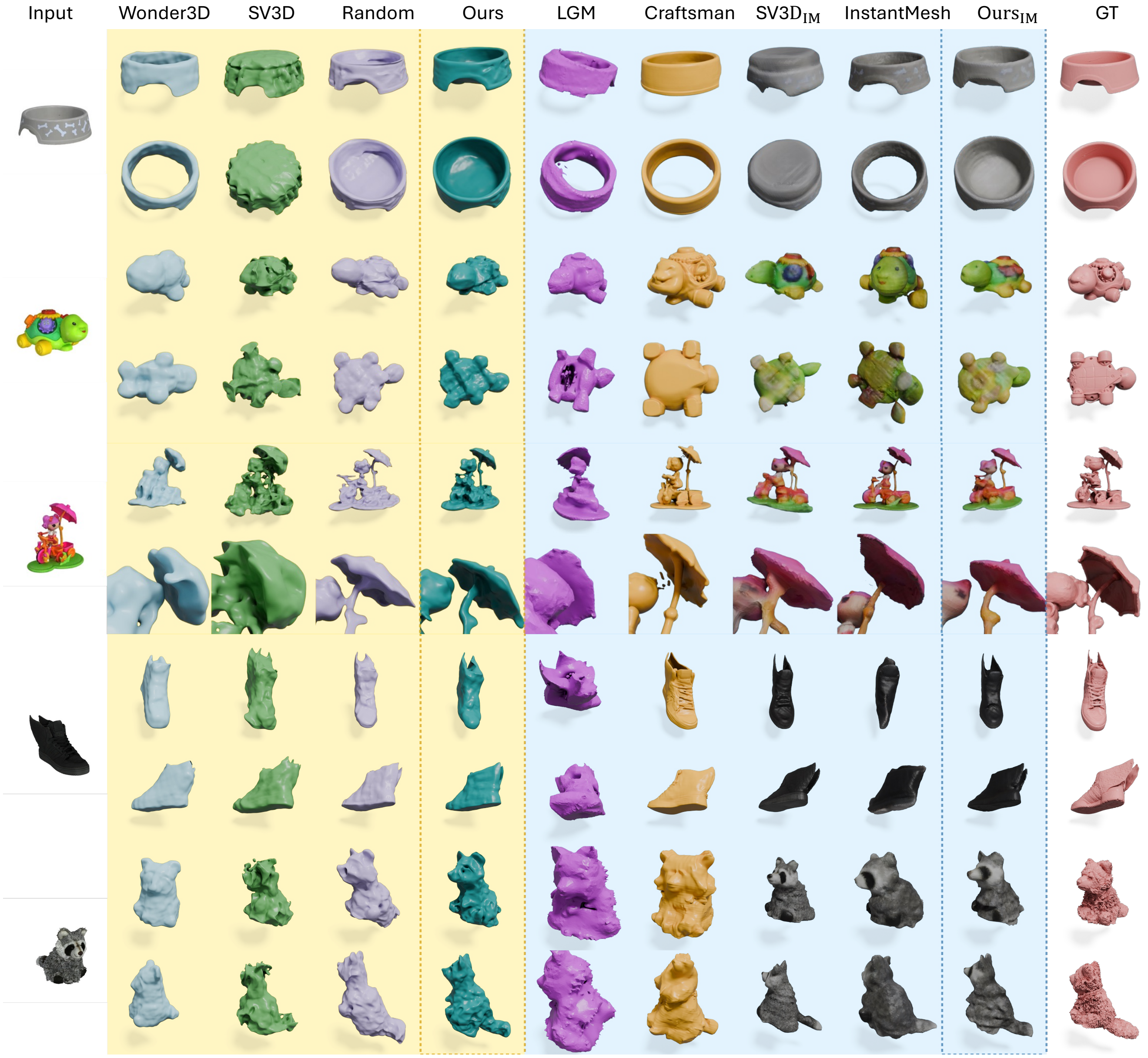

Figure 15. Additional qualitative visual comparisons between single-view 3D reconstruction methods on the GSO dataset. Please zoom in for a closer inspection. Meshes reconstructed with NeUS and LRM are in yellow and blue blocks, respectively. Our slice-guided approach demonstrates superior reconstruction of occluded regions, including bowl bottoms, hidden wings and raccoon tails. Please zoom in for a detailed comparison of the reconstruction quality.