RnG: A Unified Transformer for Complete 3D Modeling from Partial Observations

Supplementary Material

1. Training Details

For a single object, we render RGBD images at 25 viewpoints with different radius. To form a training batch, we randomly sample 7 images from the multi-view renderings. The first 4 images will be used as source views, and the rest 3 are different target views. We then expand the batch size by 3 so that each training batch contains the same 4 source images and 1 target image.

The training batch size is 6 (objects), we accumulate every 2 forward steps for an increased batch size. The maximum learning rate is set to 6e-4. We use learning-rate warm-up for the first 3000 steps and use the cosine learning-rate decay for the rest of the training process. The single image feature extractor's parameters (the DINO vision transformer [2] in VGGT [3]) are kept frozen during training.

2. Recovering complete 3D

2.1. Recovering up-direction

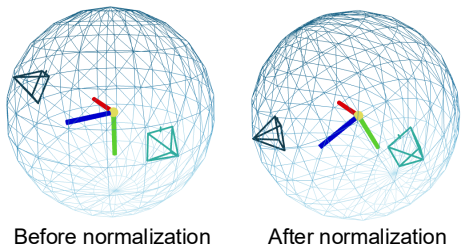

Figure 1. **The camera normalization process.** The first camera pose will be normalized on a unit sphere and transformed to a fixed position. This introduces additional roll angles to other cameras.

As illustrated in Figure 1, the training samples undergo a camera normalization process, where the first camera is scaled and transformed to $[I_{3\times3}\,|[0,0,-1]^\intercal]$. This will change the 'up direction' of the object in the world coordinate system. When querying from a novel view, if we directly use pre-defined camera locations, an extra 'rolling angle' will be introduced. In other words, the 'up-direction' of the world coordinate system does not align with the observed object's up-direction, but with an extra rolling angle. This is out of the training samples' distribution, so after model predicting the camera poses, we have to recover object's up-direction (if possible).

We assume all input images' up-directions are aligned with the object's up-direction, then the objective is to find a rotation along the x-axis (the red axis in Figure 1) so that all cameras do not have additional 'roll angles'. A simple solution is to search for the best angle of elevation that lead to the minimum of all rolling angles.

2.2. Accumulating views

When extracting complete 3D from the network, we can manually choose viewpoints to back-project the estimated 3D points using our user interface (shown in Figure 2). We can also uniformly place cameras on a sphere to get novel views. We utilize the estimated confidence map to filter the estimated points to remove flying points around sharp boundaries. The user interface is implemented using Viser [4].

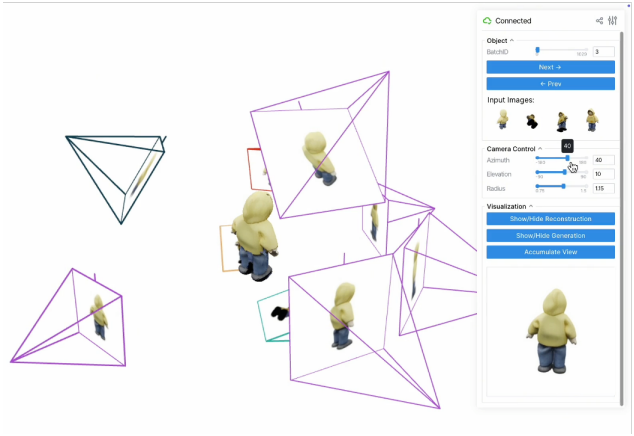

Figure 2. The user interface to visualize novel view images and accumulate complete 3D from multiple views.

3. More Experimental Results

3.1. Generalize to arbitrary input views

In **??** of the main paper, we demonstrated that the proposed method generalizes to arbitrary input views. In this section, we provide additional details of the experimental setup, as well as more comprehensive quantitative and qualitative results. Specifically, for each 3D object in the evaluation dataset GSO [1], we render RGBD frames at a 25 different places. We randomly pick 10 frames for evaluation, the source images are chosen from the rest 15 views. Our main comparison uses 4 random source views. To keep the metrics comparable when given different number of source views, we remove or append to these 4 views and keep the target views fixed.

Table 1. **Generalize to other number of input views.** Although our model is not trained to handle other number of input views, it still shows strong generalization ability to other number of source images.

# input views	*Reconstruction*					*Generation*				
	Pose			Input View Depth		Novel View Depth		Novel View Synthesis		
	RA@5↑	RT@5↑	AUC@30↑	Rel↓	a1↑	Rel↓	a1↑	PSNR↑	SSIM↑	LPIPS↓
8	85.922	86.893	87.942	0.535	99.927	0.593	99.875	27.608	0.903	0.0857
6	85.825	86.505	87.741	0.539	99.935	0.622	99.876	27.235	0.899	0.0891
4	85.146	86.019	86.942	0.584	99.929	0.717	99.850	26.276	0.891	0.0975
3	85.243	86.311	86.136	0.662	99.931	0.848	99.820	25.291	0.881	0.1063
2	83.010	87.961	85.405	0.931	99.906	1.243	99.670	23.403	0.860	0.1249
1	*not applicable*			3.328	99.160	3.443	98.121	19.452	0.811	0.1821

The detailed metric results are given in Table 1. As shown, both the target-view depth and image synthesis quality improves when denser source views are provided. Notably, RnG still achieves reasonable results even under two input-view settings. For instance, the PSNR of novel view appearance decreases by only 7.5% compared with our standard training setup using four input views. A similar trend is observed across other metrics, further validating the strong generalization ability of the proposed method.

We also provide visualizations in Figure 5. As shown in the first rows, the model fails to produce reasonable results with a single input view, but the performance improves notably with two input views. When the number of source images increases and more observations become available, the synthesized target images exhibit richer details and higher visual fidelity. We provide more visualizations in Figure 6. In these cases, the model performs reasonably well even when only a single source view is given. There are also hard cases where key structures are occluded or the structures are ambiguous. We show these samples in Figure 7, where RnG performs 'reasonably bad'. When sufficient amount of observations are available, the performance of RnG improves over these cases, shown in Figure 8.

3.2. Generalize to real-world inputs

Generalization results on CO3D are shown in Figure 3. RnG shows better zero-shot performance than LVSM and is comparable to Matrix3D trained with CO3D data. Due to the world-origin ambiguity discussed in the Limitations, predictions may be misaligned with the ground truth, which can be alleviated by improved training data preparation.

Figure 3. Qualitative results on CO3D.

3.3. Generalize to multi-objects scenerarios

RnG naturally generalizes to multi-object scenes without any modification. As shown in Figure 4, several examples from the GSO dataset support this claim. We find it valuable for future explorations to systematically evaluate on more complex scenarios.

Figure 4. RnG's novel view synthesis results on multi-objects.

References

[1] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, pages 2553–2560. IEEE, 2022. 1

[2] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[3] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025. 1

[4] Brent Yi, Chung Min Kim, Justin Kerr, Gina Wu, Rebecca Feng, Anthony Zhang, Jonas Kulhanek, Hongsuk Choi, Yi Ma, Matthew Tancik, et al. Viser: Imperative, web-based 3d visualization in python. *arXiv preprint arXiv:2507.22885*, 2025. 1

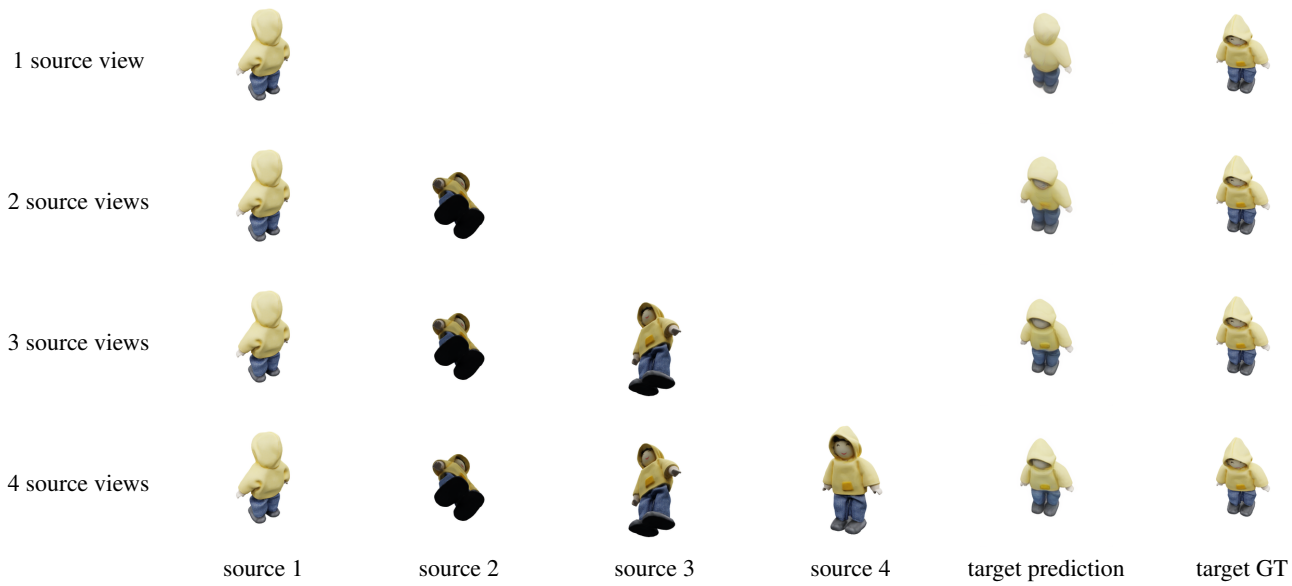

Figure 5. The evaluation protocol when models are given different number of input views. Each row represents an input configuration.

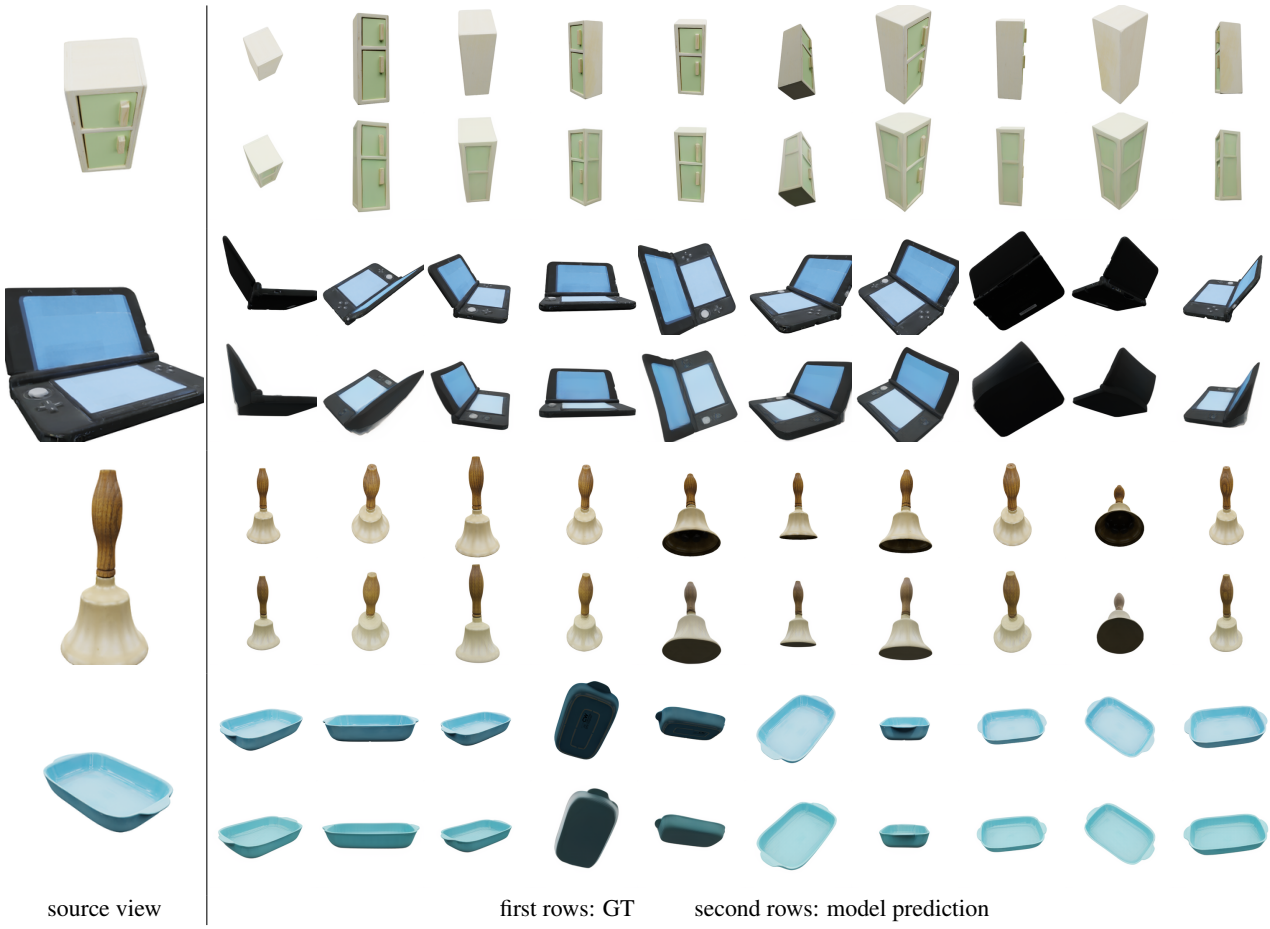

Figure 6. RnG performs well even when only a single image is given.

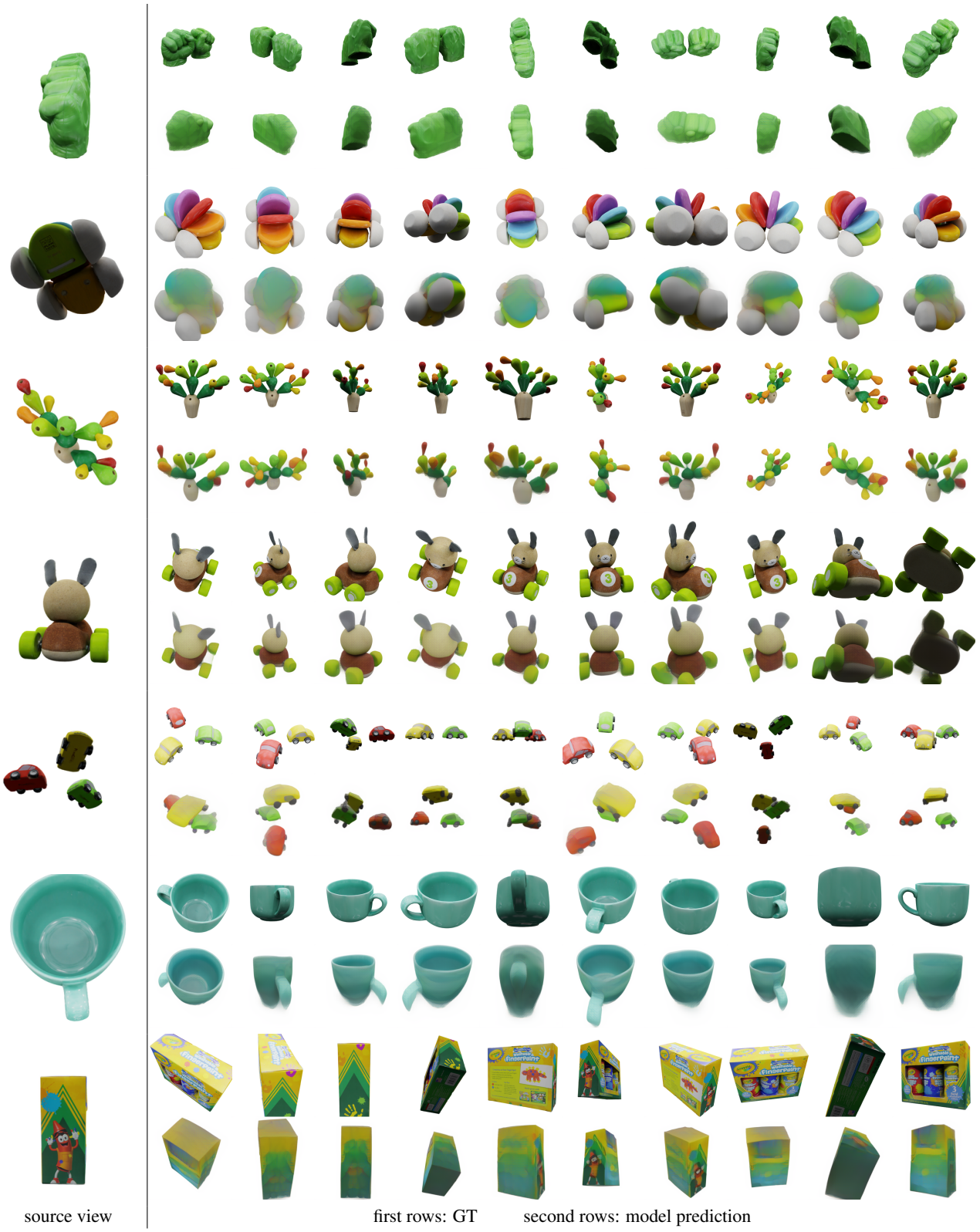

Figure 7. When generating novel views from only one source image is too hard, RnG performs reasonably bad.

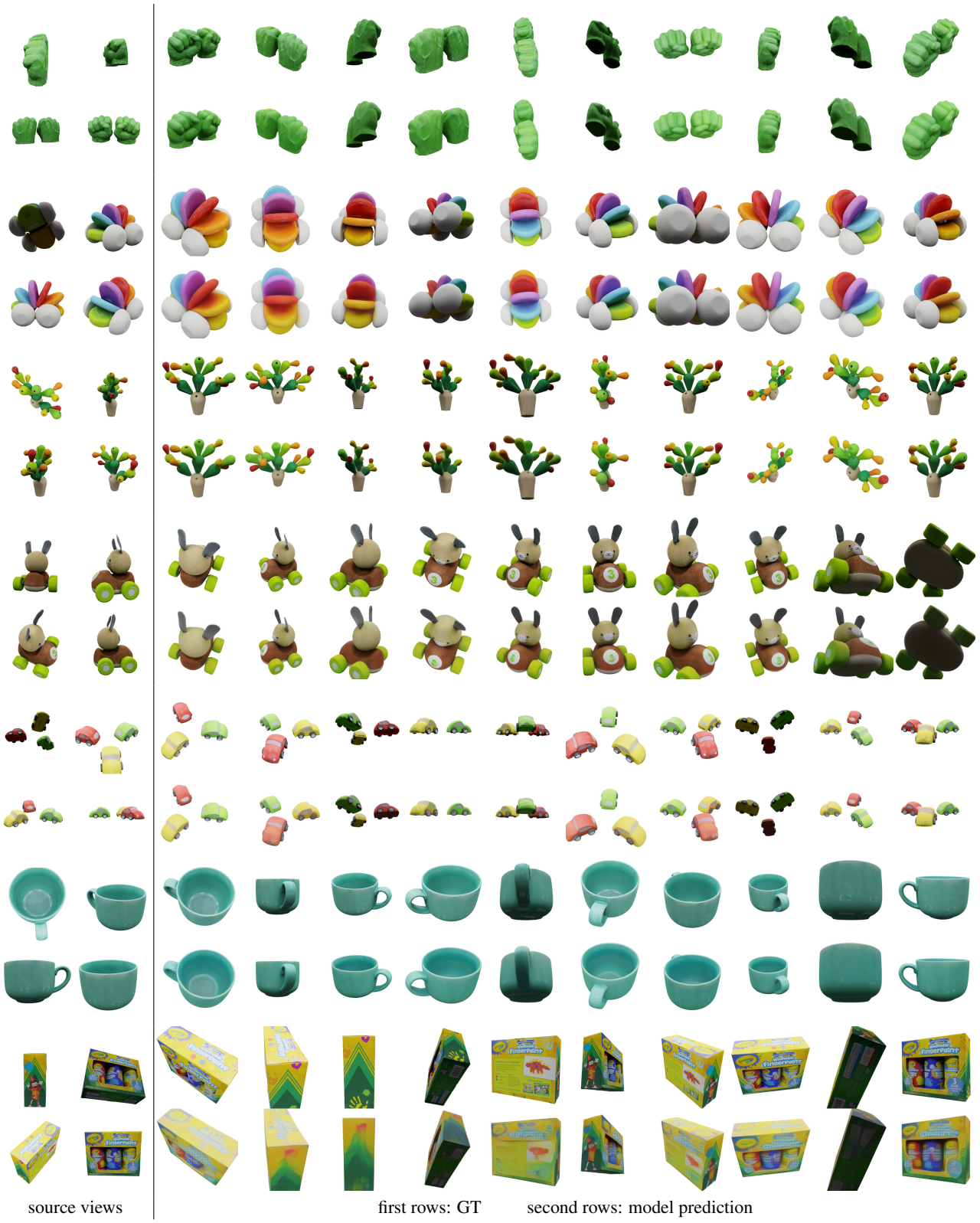

Figure 8. RnG performs better when ambiguities are resolved.