Time of the Flight of the Gaussians: Optimizing Depth Indirectly in Dynamic Radiance Fields

Supplementary Material

A. Supplemental Video + Full Figures

We have included a supplemental video showing all dataset sequences plus two ablation sequences. Further, we include full sequence qualitative figures for all three datasets at the back of this supplemental material; this reproduces some results from the main paper but groups and includes all results for completeness.

B. Ablations

We show quantitative (Tab. 1) ablations on F-TöRF synthetic scenes and also qualitative ablation results on F-TöRF real-world and synthetic scenes (Fig. 1). "No bias" refers to no consideration of either of our heuristics. "DD" means adding the depth distortion loss [2], and "H1" and "H2" means our control of the optimization using our occupancy bias (H1) and low initial reflectivity bias (H2). For result variants, 'd' refers to the mean depth from Eq. (5) in the main paper, and 'd_{ToF}' refers to the depth derived from the reconstructed quads using Eq. (1) in the main paper.

C. ToF Gradient Computation Details

We briefly explain the opacity gradient computation for our ToF image formation model, which is less trivial compared to other gradients computed via simple chain rules. Let L denote the loss function, and $\alpha_k = o_k \mathcal{G}_k^{2D}(\mathbf{x})$ the opacity term of the k-th Gaussian. Using the chain rule:

$$\frac{\partial L}{\partial \alpha_k} = \frac{\partial L}{\partial \mathbf{c}(\mathbf{x})} \frac{\partial \mathbf{c}(\mathbf{x})}{\partial \alpha_k}, \tag{S1}$$

the term $\frac{\partial \mathbf{c}(\mathbf{x})}{\partial \alpha_k}$ is computed recursively in the original 3D Gaussian Splatting (3DGS) method [3]:

$$\frac{\partial \mathbf{c}(\mathbf{x})}{\partial \alpha_k} = T_k(\mathbf{c}_k - \text{acc}_k), \tag{S2}$$

$$\text{acc}_k = \begin{cases} \alpha_{k+1}\mathbf{c}_{k+1} + (1-\alpha_{k+1})\text{acc}_{k+1}, & k < N, \\ 0, & k = N, \end{cases}$$

where T_k is the transmittance, $\mathbf{c}_k$ is the Gaussian's color, and acc_k aggregates contributions of later Gaussians.

In our model, this computation extends to C-ToF signals like phasor or quad pixels ($\mathbf{p}(\mathbf{x})$, $\mathbf{q}(\mathbf{x})$) as described in Eq. (4) in the main paper. We take $\mathbf{q}(\mathbf{x})$ as an example, the corresponding recursion becomes to:

$$\frac{\partial \mathbf{q}(\mathbf{x})}{\partial \alpha_k} = T_k^2(\mathbf{q}_k + 2(\alpha_k - 1)\text{acc}_k^q), \tag{S3}$$

$$\text{acc}_k^q = \begin{cases} \alpha_{k+1}\mathbf{q}_{k+1} + (1-\alpha_{k+1})^2\text{acc}_{k+1}^q, & k < N, \\ 0, & k = N. \end{cases}$$

Here, $\mathbf{q}_k$ represents the quad contribution of the k-th Gaussian, and acc_k^q accumulates later quad contributions.

References

[1] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. TöRF: Time-of-flight radiance fields for dynamic scene view synthesis. In *NeurIPS*, 2021. 4, 5, 6

[2] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, 2024. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. 1

[4] Mikhail Okunev, Marc Mapeke, Benjamin Attal, Christian Richardt, Matthew O'Toole, and James Tompkin. Flowed time of flight radiance fields. In *ECCV*, 2024. 4, 5

[5] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 4, 5, 6

Table 1. **Ablation depth error on the F-TöRF synthetic dataset**. Each number is a depth MSE$\times 100$. Bold marks the smallest result for each of d and d_{ToF}. On average, our approach is reliably more accurate without any catastrophic failures, though scene specific variations exist. *Sliding Cube* is too simple to induce large differences. *Ortho. ST* is the only sequence with large low reflectivity areas, and we see the advantage of our low initial reflectivity bias (H2) in the scene's reconstructed d. Using both heuristics H1, H2 plus the depth distortion (DD) loss theoretically should reduce the gap between d and d_{ToF}, but in practice, because of the spatial extent of the Gaussians, this usually led to oversmooth depth, worse thin structures, and is sometimes unstable because of the existence of large Gaussians representing large homogeneous textureless areas (Fig. 1).

		Mean	*Sliding Cube*	*Occ. Cube*	*Axial ST*	*3 Cubes ST*	*3 Chairs ST*	*Arcing Cube*	*Ortho. ST*
d_{ToF}	No bias	144.134	0.013	13.227	0.286	1.152	0.885	5.884	987.492
	+DD	146.189	0.119	11.422	5.413	7.289	1.224	13.415	984.440
	+H1	3.641	0.017	**0.024**	0.206	0.916	**0.605**	**1.113**	22.610
	+H1 +H2 (ours)	**3.360**	**0.008**	0.073	**0.154**	**0.294**	0.650	1.526	**20.818**
	+H1 +H2 +DD	255.717	0.012	148.578	0.422	1.010	0.895	1537.952	101.148
d	No bias	365.686	0.049	15.604	55.834	24.892	1.482	4.456	2457.487
	+DD	274.316	0.172	11.675	5.922	7.805	1.421	13.067	1880.148
	+H1	44.520	0.058	**0.310**	**0.726**	1.610	**1.011**	**1.025**	306.898
	+H1 +H2 (ours)	**5.824**	0.033	0.348	0.730	**1.059**	1.072	1.479	**36.048**
	+H1 +H2 +DD	310.564	**0.026**	147.856	0.787	1.362	1.106	1537.840	484.997

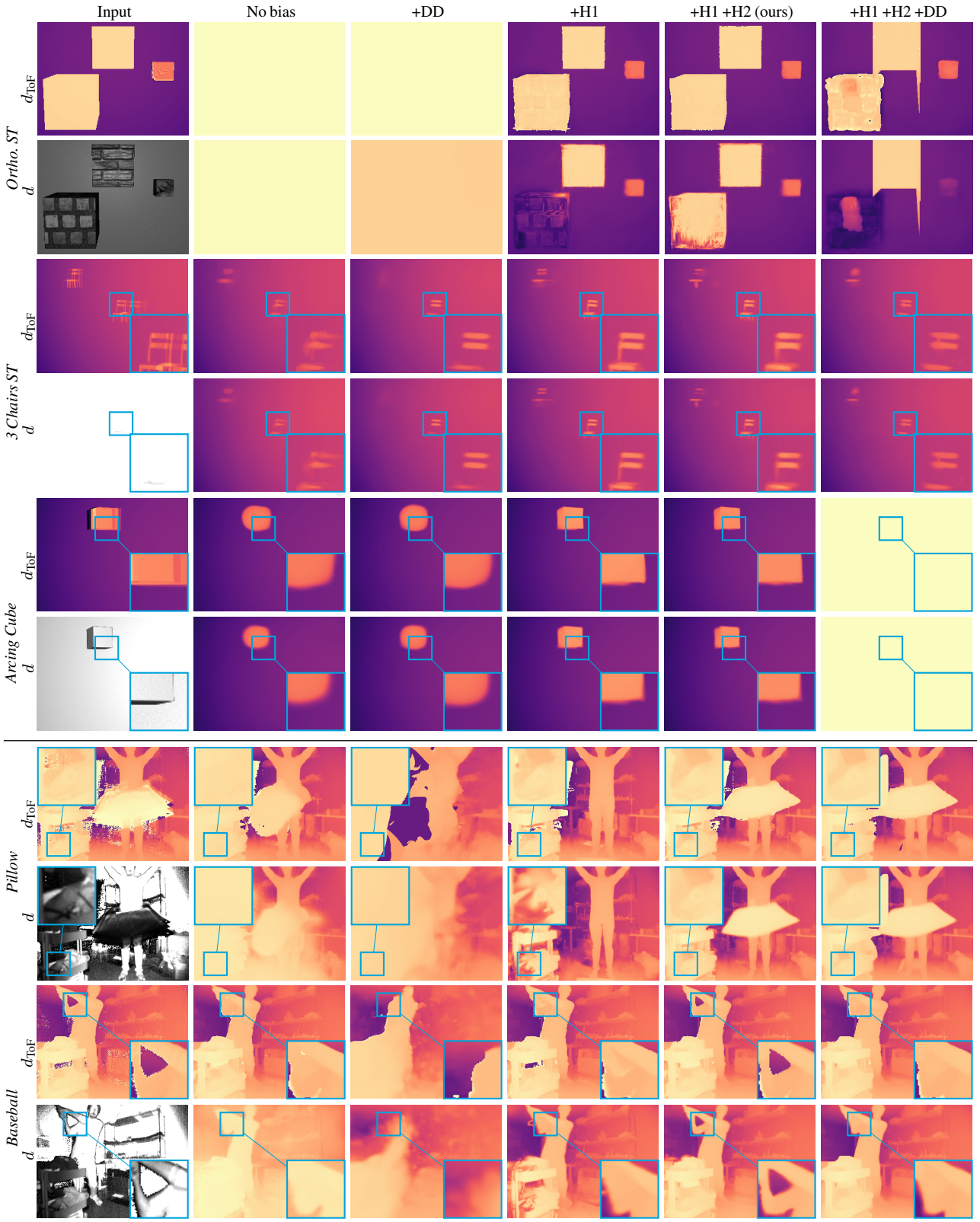


Figure 1. **Ablations**. In the first column, the even rows show the reflectivity map, which is computed as input amplitude multiplied with the square of input depth (light falloff), and can be understood as the expected of Gaussian reflectivity at the corresponding surface. For visualization, the map is clipped to the range [0, 1], where overexposed areas only indicate high target reflectivity. No bias led to arbitrary number of density peaks even for opaque surfaces, thus inaccurate depth from Eq. (5) in the main paper.Adding DD overly stack large Gaussians that led to overly smoothed mean depth. Occupancy bias improves the placement of Gaussians but still struggles in low-reflectivity areas (e.g., bottom left of the cart, and the pillow). Initializing Gaussians with low reflectivity mitigates this issue. Reintroducing the DD loss again after applying the two heuristics still led to oversmooth depth due to Gaussians' spatial extent.

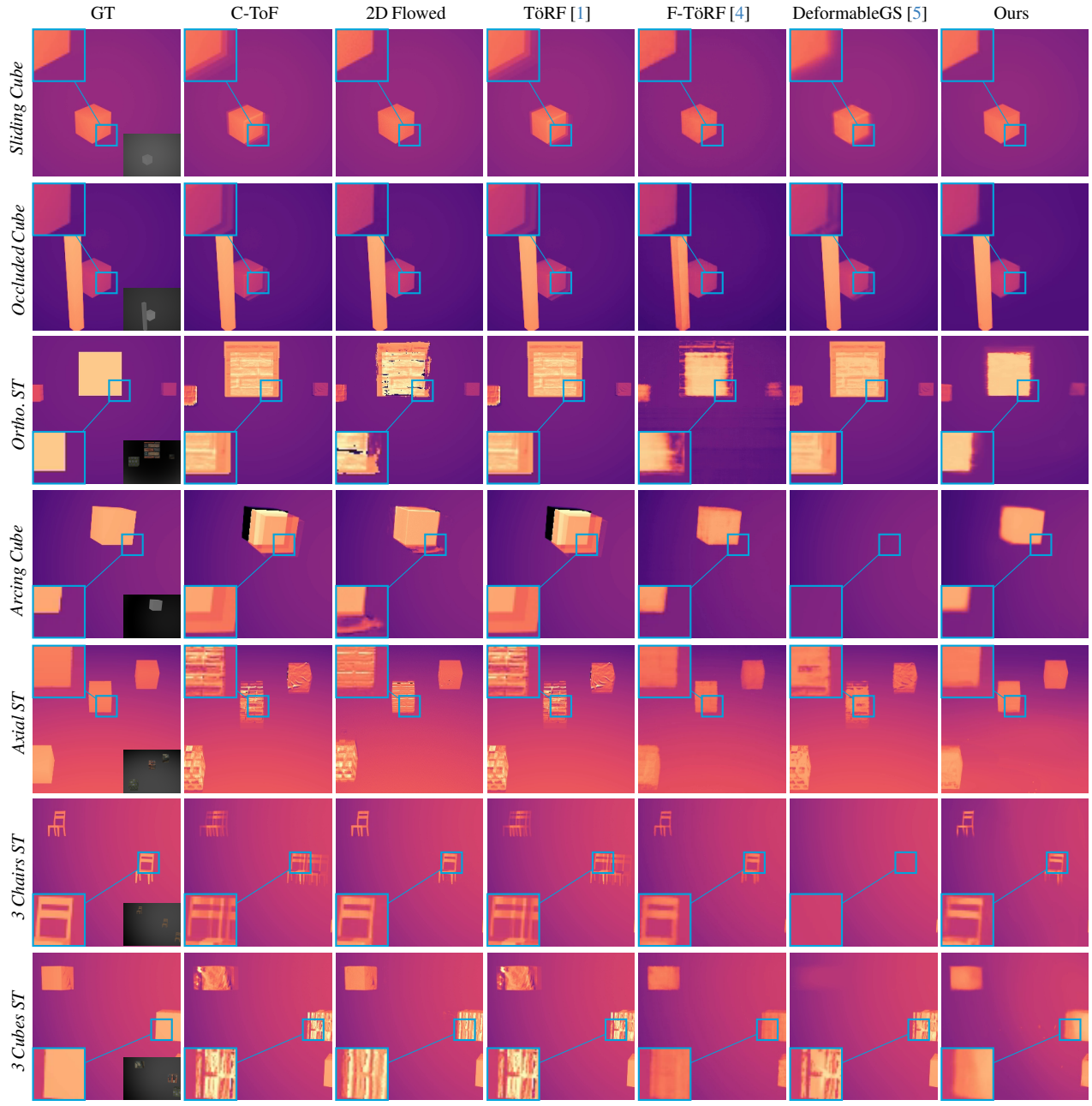


Figure 2. **Results on all F-TöRF synthetic scenes**. DeformableGS is given C-ToF-derived depth as an additional input. *Inset on ground truth: corresponding RGB image. Full RGB images can be found in Fig. 5.*

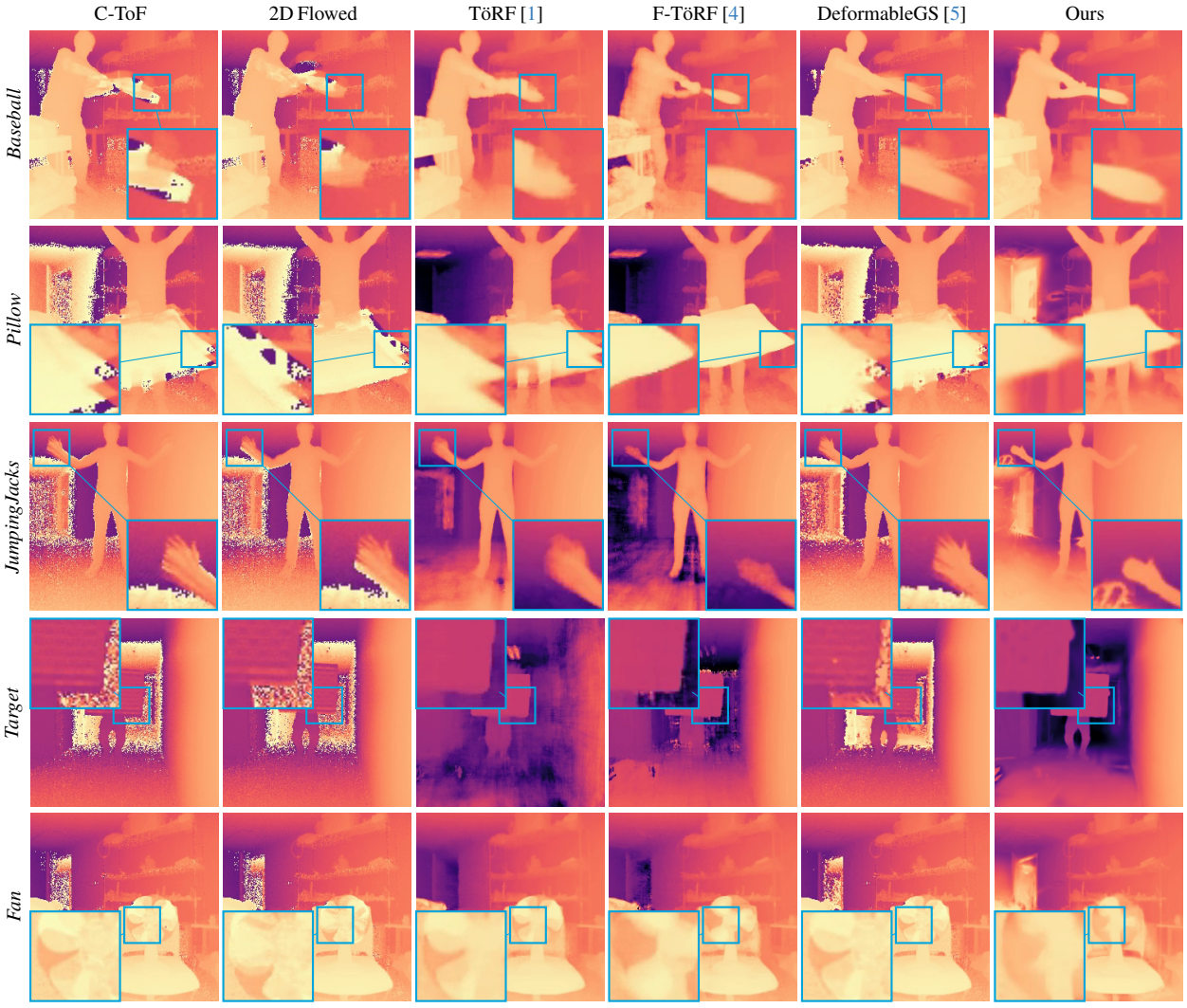


Figure 3. **Results on all F-TöRF real-world scenes**. DeformableGS is given C-ToF-derived depth as an additional input.

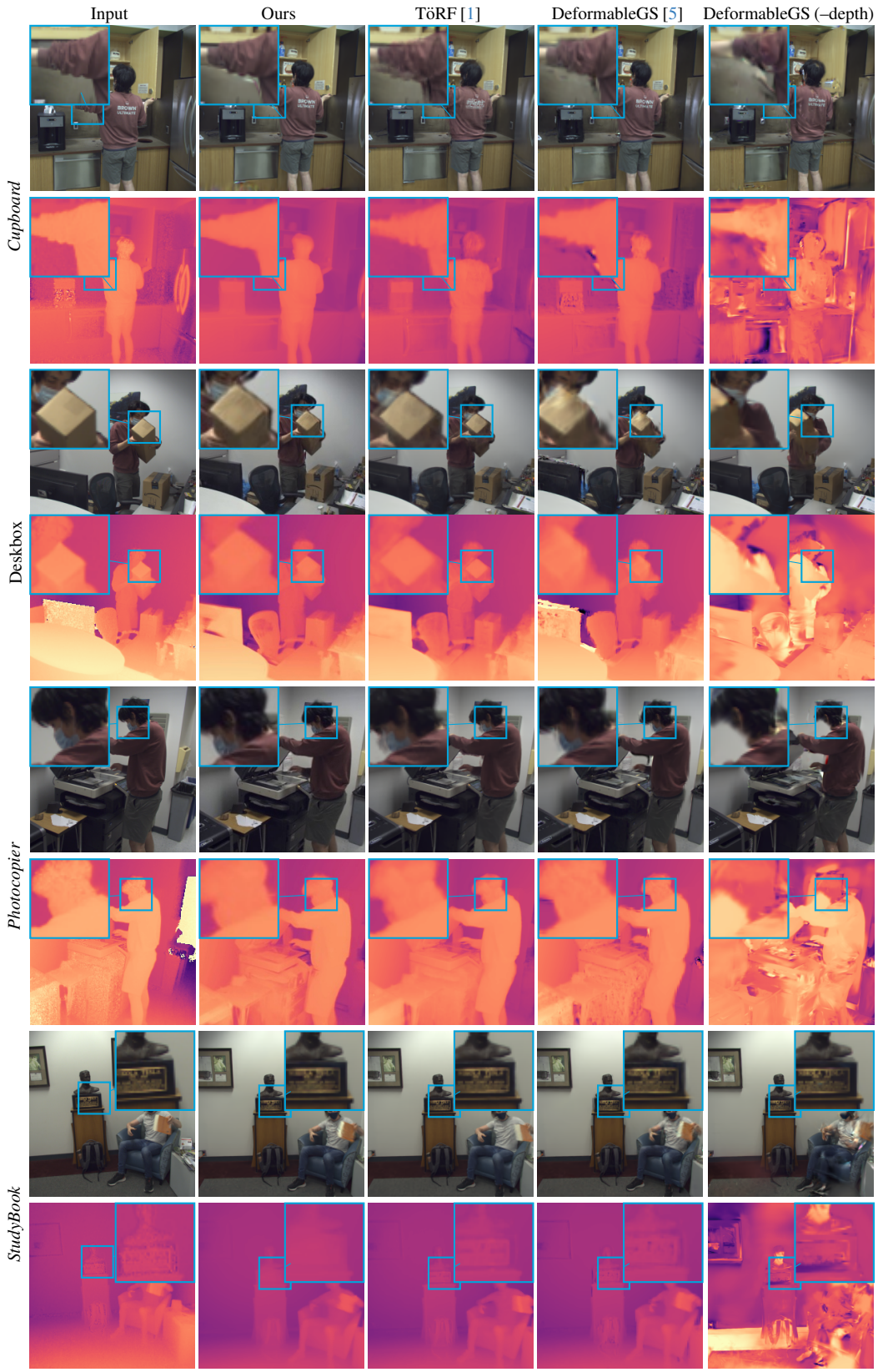
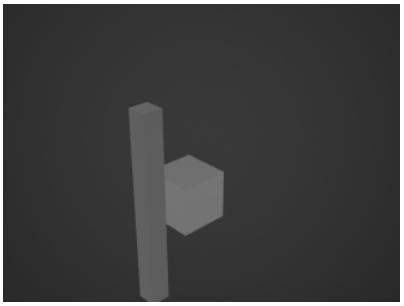
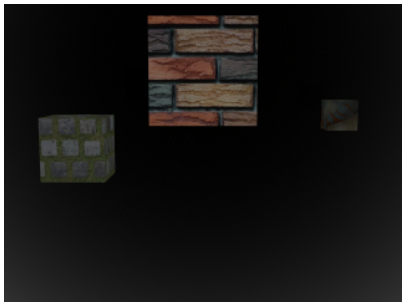


Figure 4. **More results on TöRF real-world scenes**. DeformableGS is given C-ToF-derived depth as an additional input.

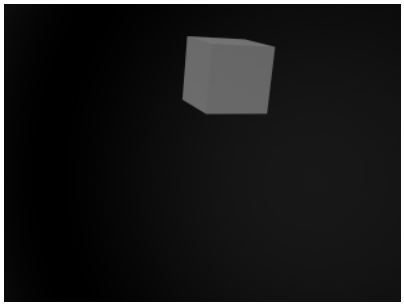
(a) *Sliding Cube*



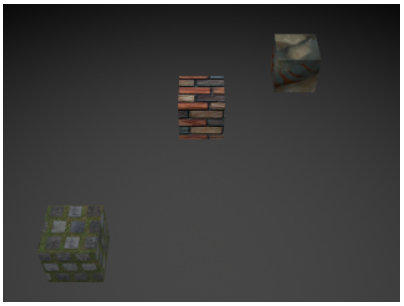
(b) *Occluded Cube*



(c) *Ortho. ST*



(d) *Arcing Cube*



(e) *Axial ST*

(f) *3 Chairs ST*

(g) *3 Cubes ST*

Figure 5. Corresponding RGB color images for the synthetic scenes for Fig. (4) in the main paper and Fig. 2. Note that *Sliding Cube*, *Occluded Cube*, and *Arcing Cube* have no texture.