

Appendices

The Supplementary material is organized as follows:

- appendix A: provides additional visualization results and quantitative results. Furthermore, it also shows the resource consumption which reveals the potential of 4DGS-1K for deployment on low-performance hardware.
- appendix B: provides additional ablation study. It discusses the role of each score in the Spatial-Temporal Variation Score, then it presents the ablation study to illustrate that our parameter selection is the result of a trade-off between rendering quality and storage size.
- appendix C: discusses the reason of improved performance for 4DGS-1K. Furthermore, we introduce the limitations and potential future directions of 4DGS-1K.

A Experimental Results

A.1 Per scene result

We provide per-scene quantitative comparisons on the N3V Dataset [2](Table 5) and D-NeRF Dataset [19](Table 6). Compared to vanilla 4DGS [1], our model significantly reduces the storage requirements and enhances rendering speed while maintaining high-quality reconstruction. Figure 11 and Figure 12 show more visual comparisons on the N3V Dataset. Figure 13, Figure 14 and Figure 15 show visual comparisons on the D-NeRF Dataset.

A.2 Resource consumption

We present the resource consumption metrics, including training time, GPU memory allocation and additional storage space. On the N3V dataset [2], 4DGS-1K only takes approximately 30 minutes to fine-tune, with GPU memory allocation of 10.54GB. During rendering, it only consumes 1.62GB of GPU memory. For storage requirement, 4DGS-1K requires additional storage for the mask of filter and codebook; however, these occupy only a minimal portion of the total storage, approximately 1 MB per scene. These parts are also included in the final experiment results.

The above results demonstrate the potential of 4DGS-1K for deployment on low-performance hardware. Consequently, we further test 4DGS-1K on TITAN X GPU, where 4DGS-1K maintains 200+ FPS on the N3V dataset, still far outperforming vanilla 4DGS (20 FPS).

A.3 Additional experiments for redundancy

In this section, we provide additional experiments for redundancy study as a supplement to Section 4.1. It is composed of two parts: first, the visualization of the Gaussian with short lifespan distribution, and secondly, the relationship between FPS and the number of inactive Gaussians.

Visualization of Gaussians with small lifespan. In Section 4.1, we argue that in vanilla 4DGS, nearly all Gaussians have a short lifespan, especially around the edge of fast-moving objects. Therefore, we visualize the spatial distribution of Σ_t to better support our redundancy study in Section 4.1.

Specifically, we visualize the distribution of Σ_t at several timestamps in *Sear Steak* Scene. The visualization results are shown in Figure 5. For visualization, we take the reciprocal of Σ_t during rendering and then normalize it. Therefore, brighter regions in the rendered image indicate smaller Σ_t .

As shown in Figure 5, Gaussians with short lifespan are primarily concentrated in regions of object motion, such as the moving person and dog. Moreover, we observe that Gaussians with small Σ_t also appear on the edges of some objects which exhibit significant color variation. This is because small Gaussians are preferred in these regions to capture the high-frequency details in the spatial dimension. As vanilla 4DGS [1] treats time and space dimensions equally, these Gaussians also have short lifespan in the temporal dimension.

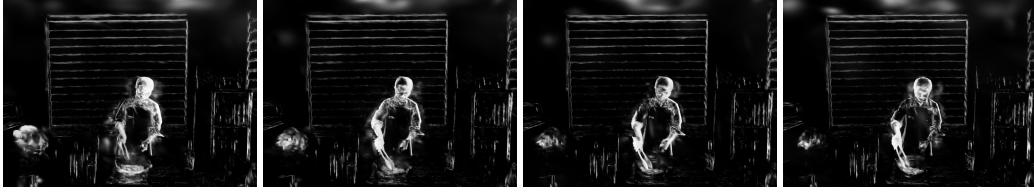


Figure 5: **Visualizations of Distribution of Σ_t .**

Relationship between FPS and the number of inactive Gaussians. In Section 4.1, our primary prior assumption is that the number of inactive Gaussians affects the FPS. Therefore, we visualize the relationship between FPS and the number of inactive Gaussians.

However, only limiting the total number of Gaussians is incorrect in this task. As the total number increases, the number of active Gaussians and inactive Gaussians also increases, which cannot clarify whether the FPS variation is caused by active or inactive Gaussians. Consequently, we first identify the active Gaussians by rendering and then add a mount of inactive Gaussians among these Gaussians.

We visualize the result in the *Sear Steak* Scene(See Figure 6). The FPS decreases as the number of inactive Gaussians increases. This phenomenon strongly supports our redundancy study in Section 4.1.

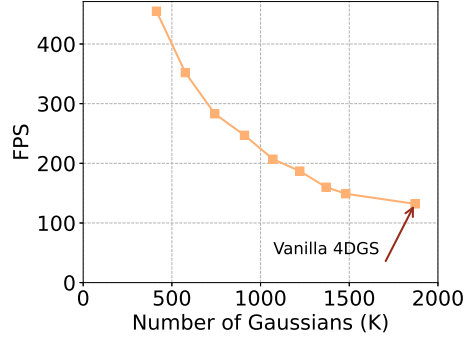


Figure 6: **Relationship between rendering speed and the number of inactive Gaussians.**

A.4 Visualizations of Pruned Gaussians

We provide the visualization of pruned Gaussians in the *Sear Steak* Scene, as shown in Figure 7. Our pruning strategy can accurately identify Gaussians with short lifespan(See Figure 7c) while maintaining the high quality reconstruction(See Figure 7d). The quantized results after pruning are presented in Table 3. Our pruning technique achieves $5\times$ compression ratio and $5\times$ faster rasterization speed while slightly improving rendering quality.

A.5 Video result

In this work, we propose a novel framework for dynamic 3D reconstruction. Therefore, we provide several videos that are rendered from testing viewpoints on the N3V datasets and D-NeRF datasets to show the reconstruction quality and temporal consistency of 4DGS-1K. These videos are composed by concatenating each frame of 4DGS and our method.

B Additional ablation study

B.1 Ablation study of Spatial-Temporal Variation Score.

We discuss the role of each score in the Spatial-Temporal Variation Score. Specifically, we introduced the following variants.

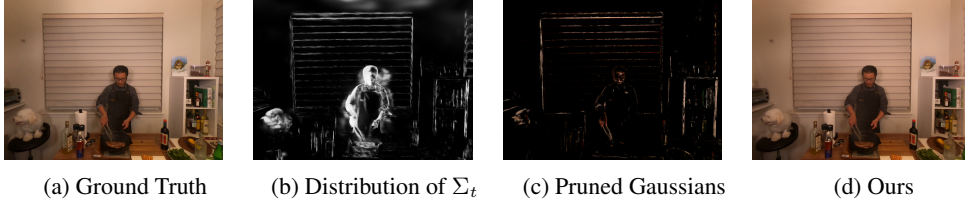


Figure 7: **Visualization of Pruned Gaussians.**

Table 4: **Ablation study of Spatial-Temporal Variation Score.** We compare our Spatial-Temporal Variation Score with other variants, and report the PSNR score of each scene.

ID	Model	Sear Steak	Flame Salmon
a	4DGS w/o Prune	33.60	29.10
b	\mathcal{S}_i^S Only	33.62	28.75
c	\mathcal{S}_i^T Only	33.59	28.79
d	\mathcal{S}_i (w. $p_i^{(1)}(t)$)	33.67	28.81
e	\mathcal{S}_i (w. Σ_t)	33.47	28.71
f	Ours	33.76	28.90

Variant Settings. As described in Section 4.2.1, our Spatial-Temporal Variation Score is composed of two parts, *spatial score* that measures the Gaussians contributions to the pixels in rendering, and *temporal score* considering the lifespan of Gaussians. By aggregating both spatial and temporal score, our score \mathcal{S}_i can be written as:

$$\mathcal{S}_i = \sum_{t=0}^T \mathcal{S}_i^T \mathcal{S}_i^S \quad (8)$$

Therefore, the variant scores in Table 4 can be written as follow.

- (b) \mathcal{S}_i^S Only: only considering the spatial part of our score. It can be written as:

$$\mathcal{S}_i = \sum_{t=0}^T \mathcal{S}_i^S \quad (9)$$

- (c) \mathcal{S}_i^T Only: only considering the temporal contribution part of our score. It can be written as:

$$\mathcal{S}_i = \sum_{t=0}^T \mathcal{S}_i^T \quad (10)$$

- (d) \mathcal{S}_i (w. $p_i^{(1)}(t)$): Replace the $p_i^{(2)}(t)$ with $p_i^{(1)}(t)$ in temporal score \mathcal{S}_i^T . It can be written as:

$$\begin{aligned} \mathcal{S}_i &= \sum_{t=0}^T \mathcal{S}_i^T \mathcal{S}_i^S \\ &= \sum_{t=0}^T \mathcal{S}_i^{TV} \gamma(S_i^{4D}) \mathcal{S}_i^S \\ &= \sum_{t=0}^T \frac{1}{0.5 \cdot \tanh(|p_i^{(1)}(t)|) + 0.5} \gamma(S_i^{4D}) \mathcal{S}_i^S. \end{aligned} \quad (11)$$

- (e) \mathcal{S}_i (w. Σ_t) Replace the \mathcal{S}_i^{TV} with Σ_t . It can be written as:

$$\begin{aligned} \mathcal{S}_i &= \sum_{t=0}^T \mathcal{S}_i^T \mathcal{S}_i^S \\ &= \sum_{t=0}^T \Sigma_t \gamma(S_i^{4D}) \mathcal{S}_i^S \end{aligned} \quad (12)$$

Performance of the variants. As shown in Table 4, using spatial and temporal scores separately reduce the PSNR. This occurs because separate scores can amplify extreme Gaussians. For instance, using only the spatial score (b) may retain Gaussians that cover just a single frame but occupy a large spatial volume. Our combined score balances these factors. For variant d, using the first derivative may cause some small Gaussians to have large \mathcal{S}_t^T compared to ours. Moreover, since most Gaussians have small Σ_t , it is difficult to distinguish them by using Σ_t along (See e).

Performance change with pruning ratio. As illustrated in Figure 9, we analyze the relationship between the pruning ratio and rendering quality. This reveals that our spatial-temporal variation score based pruning can even improve scene rendering quality when the pruning ratio is relatively low in the *Cook Spinach* and *Sear Steak* scenes. Moreover, at higher thresholds, it can maintain results comparable to the vanilla 4DGS [1]. Our default setting represents a balanced trade-off between rendering quality and storage size. This setting allows us to achieve a $5\times$ compression ratio while still maintaining high-quality reconstruction.

Performance change with key-frames intervals. As shown in Figure 10, although the temporal filter effectively improves rendering speed, its performance degrades significantly when the filter is with long-interval keyframes. However, by integrating the temporal filter into the fine-tuning process, this limitation can be mitigated. The fundamental reason is that some Gaussians which may carry critical scene information are being overlooked due to overly long intervals. However, the fine-tuning process effectively compensates for the loss of this portion of information. This allows us to utilize longer intervals to reduce the additional computational overhead caused by mask calculations.

C Discussion

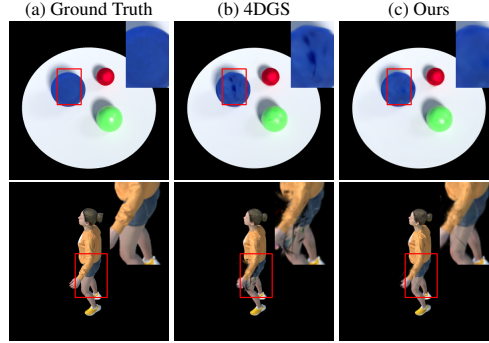


Figure 8: **Visualization of improved performance.**

Improved performance. As shown in Table 2, our model achieves a slight PSNR improvement on the D-NeRF Dataset [19]. This is because vanilla 4DGS often suffers from floaters and artifacts, due to the limited training viewpoints on the D-NeRF Datasets. However, in our study, 4DGS-1K not only can prune the Gaussians with short lifespan, but also reduce the occurrence of floaters and artifacts, as shown in Figure 8. We visualize two scenes, *Bouncingballs* and *Jumpingjacks*, on the D-NeRF Dataset. These two scenes exhibit floaters and artifacts issues due to limited training viewpoints, as shown in the red box. However, this issue does not appear in 4DGS-1K. Through pruning and filtering, 4DGS-1K successfully mitigates the occurrence of this phenomenon.

Limitations and Future work. As shown in Table 5 and Table 6, due to the acceleration provided by the temporal filter, the proportion of time spent on the rasterization process sharply decreases relative to the total rendering time. Therefore, the time consumed by preliminary preparation stages has not gradually become negligible. We hope that future work will focus on optimizing these additional operations within the rendering module to improve its computational performance. Moreover, during the pruning process, we specified a predefined pruning ratio. This pruning ratio is influenced by the inherent characteristics of the scene. As shown in Figure 9, an improper pruning ratio will cause a sharp drop in rendering quality. Therefore, identifying the minimal number of Gaussians required to maintain high-quality rendering across different scenes remains a challenge. Lastly, there is a significant amount of existing work on Gaussian-based novel view synthesis for dynamic

scenes, whereas our model is specifically tailored to a particular model, 4DGS [1]. Therefore, developing a universal compression method for these Gaussian-based models is a promising direction for subsequent research endeavors.

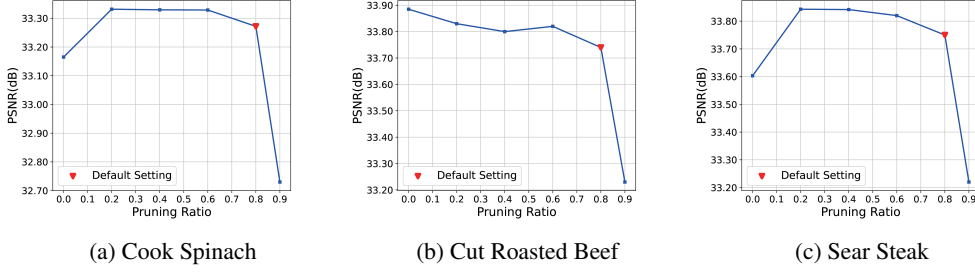


Figure 9: Rate-distortion curves evaluated on diverse scenes with different pruning ratios.

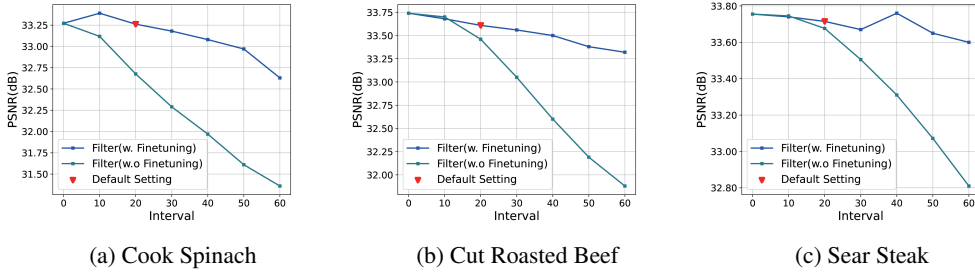


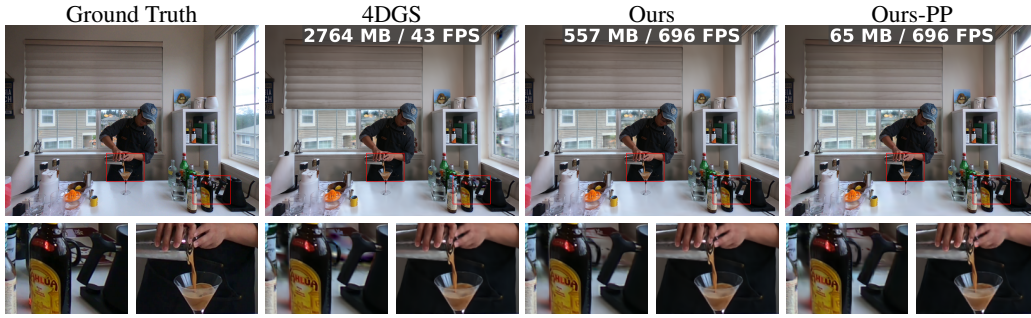
Figure 10: Rate-distortion curves evaluated on diverse scenes with different key-frames interval.

Table 5: Per-scene results of N3V datasets.

Scene	Coffee Martini	Cook Spinach	Cut Roasted Beef	Flame Salmon	Flame Steak	Sear Steak	Average
4DGS	PSNR	27.9286	33.1651	33.8849	29.1009	33.7970	31.9133
	SSIM	0.9160	0.9545	0.9589	0.9236	0.9615	0.9459
	LPIPS	0.0759	0.0449	0.0408	0.0691	0.0383	0.0518
	Storage(MB)	2764	2211	1863	2969	1536	2085
	FPS	43	89	103	31	122	90
	Raster FPS	75	103	122	70	148	118
	#NUM	4441271	3530165	2979832	4719443	2457356	3333160
Ours	PSNR	28.5780	33.2613	33.6092	28.8488	33.2804	31.8821
	SSIM	0.9185	0.9553	0.9570	0.9221	0.9598	0.9457
	LPIPS	0.0726	0.0459	0.0435	0.0707	0.0417	0.0524
	Storage(MB)	557.4	443.11	374.05	592.4	308.4	418.36
	FPS	696	803	853	680	864	805
	Raster FPS	901	1088	1163	879	1189	1092
	#NUM	888254	706033	595967	943889	491471	666632
Ours-PP	PSNR	28.5472	33.0641	33.7767	28.9878	33.2519	31.8722
	SSIM	0.9166	0.9540	0.9562	0.9209	0.9581	0.9444
	LPIPS	0.0744	0.0467	0.0445	0.0712	0.0421	0.0532
	Storage(MB)	64.94	52.04	44.54	69.24	36.94	49.50
	FPS	696	803	853	680	864	805
	Raster FPS	901	1088	1163	879	1189	1092
	#NUM	888254	706033	595967	943889	491471	666632

Table 6: Per-scene results of D-NeRF datasets.

Scene	Bouncingballs	Hellwarrior	Hook	Jumpingjacks	Lego	Mutant	Standup	Trex	Average
4DGS	PSNR	33.3472	34.7296	31.9369	30.8247	25.3320	38.9257	39.0411	32.9989
	SSIM	0.9821	0.9516	0.9635	0.9684	0.9178	0.9903	0.9896	0.9678
	LPIPS	0.0252	0.0652	0.0385	0.0340	0.0819	0.0090	0.0094	0.0353
	Storage(MB)	83.69	156.53	164.91	510.99	351.19	73.24	95.38	278.45
	FPS	462	426	414	267	317	463	457	376
	Raster FPS	1951	1433	1309	489	634	1861	1878	1232
Ours	#NUM	133762	250201	263593	816773	561357	117062	152454	1265408
	PSNR	33.4532	35.0316	32.5118	31.8045	26.8319	37.1916	39.3990	33.3370
	SSIM	0.9826	0.9530	0.9653	0.9716	0.9280	0.9886	0.9896	0.9699
	LPIPS	0.0248	0.0644	0.035	0.0322	0.0674	0.0124	0.0099	0.0330
	Storage(MB)	12.56	23.38	24.63	76.19	52.45	10.97	14.25	41.58
	FPS	1509	1517	1444	1491	1318	1518	1539	1462
Ours-PP	Raster FPS	2600	2665	2634	2476	2067	2598	2644	2482
	#NUM	20065	37368	39360	121776	83837	17527	22768	188986
	PSNR	33.4592	35.1570	32.5498	31.8467	27.2850	37.0218	39.0713	33.3746
	SSIM	0.9821	0.9537	0.9671	0.9728	0.9315	0.9883	0.9896	0.9709
	LPIPS	0.0259	0.0629	0.0345	0.0309	0.0646	0.0139	0.0109	0.0326
	Storage(MB)	4.12	5.29	5.39	11.04	8.48	3.56	3.88	7.23
Ours-PP	FPS	1509	1517	1444	1491	1318	1518	1539	1462
	Raster FPS	2600	2665	2634	2476	2067	2598	2644	2482
	#NUM	20065	37368	39360	121776	83837	17527	22768	188986



(a) Results on Coffee Martini Scene.

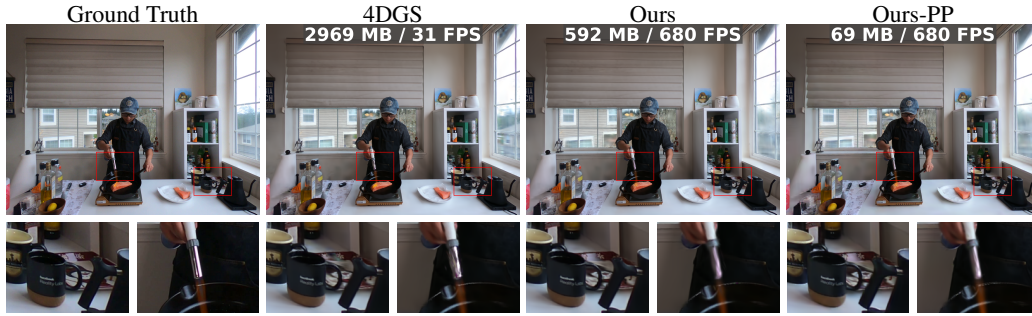


(b) Results on Cook Spinach Scene.

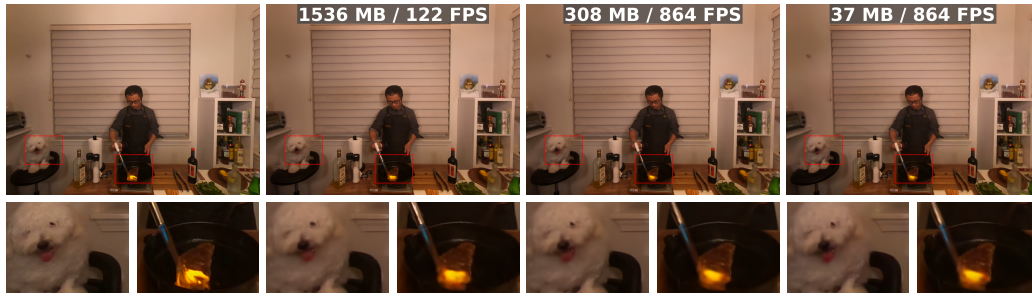


(c) Results on Cut Roasted Beef Scene.

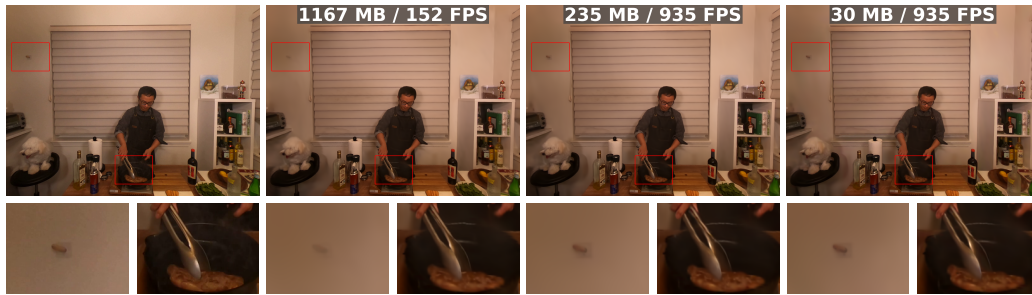
Figure 11: **Qualitative comparisons of 4DGS and our method on the N3V dataset.** To be continued in the next page.



(a) Results on Flame Salmon Scene.

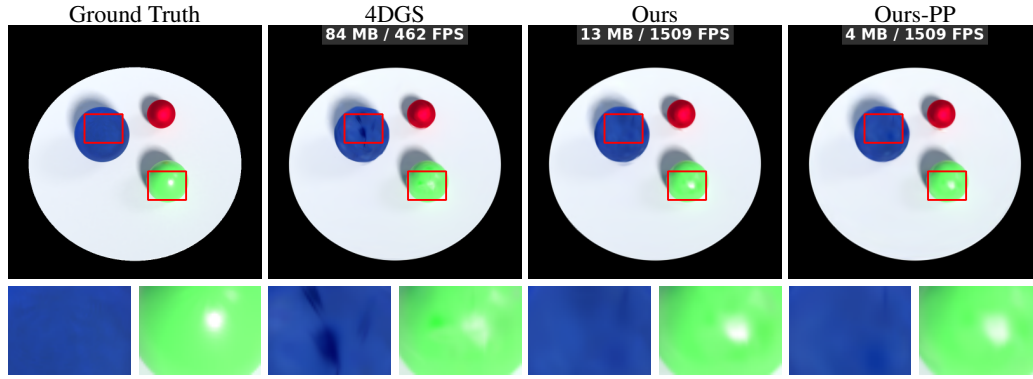


(b) Results on Flame Steak Scene.

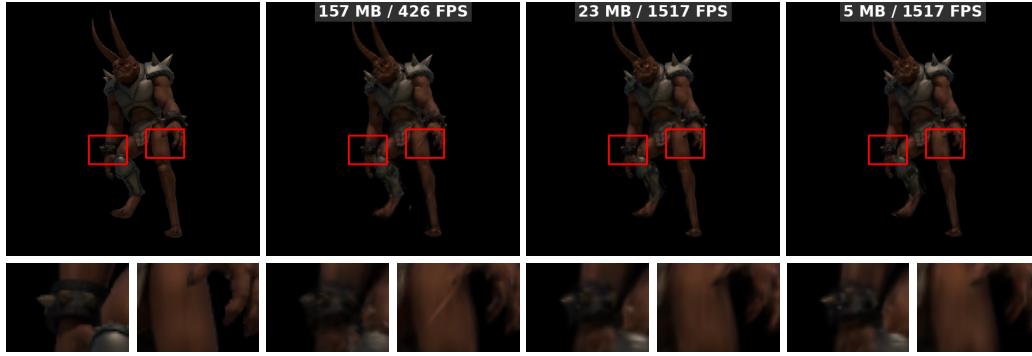


(c) Results on Sear Steak Scene.

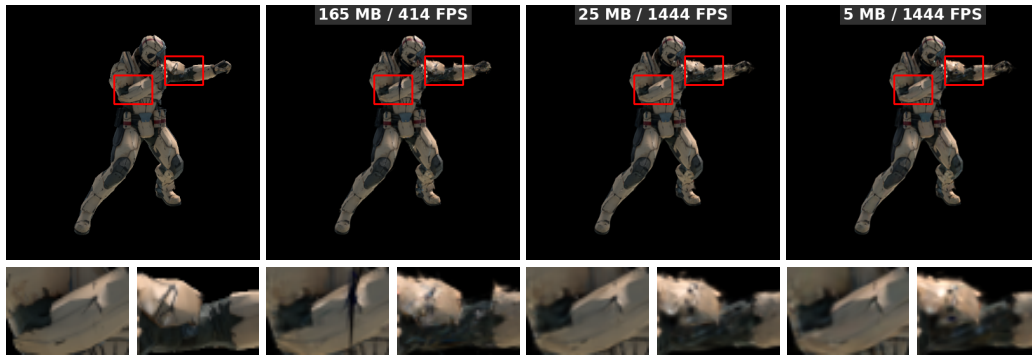
Figure 12: Qualitative comparisons of 4DGS and our method on the N3V dataset.



(a) Results on Bouncingballs Scene.

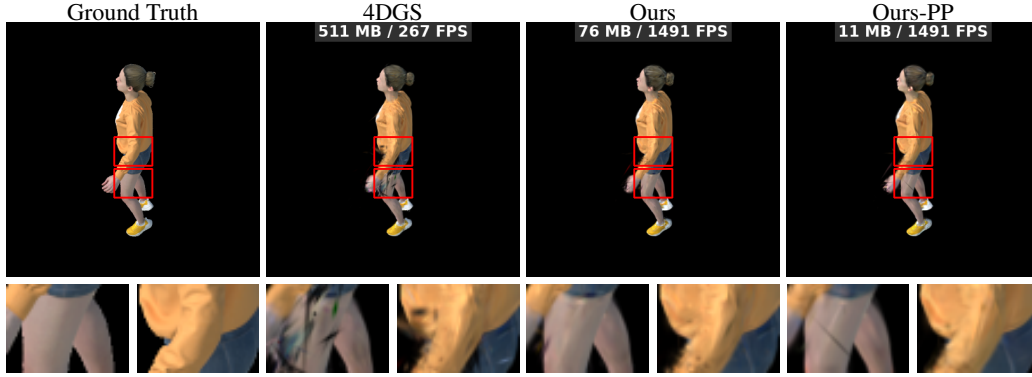


(b) Results on Hellwarrior Scene.



(c) Results on Hook Scene.

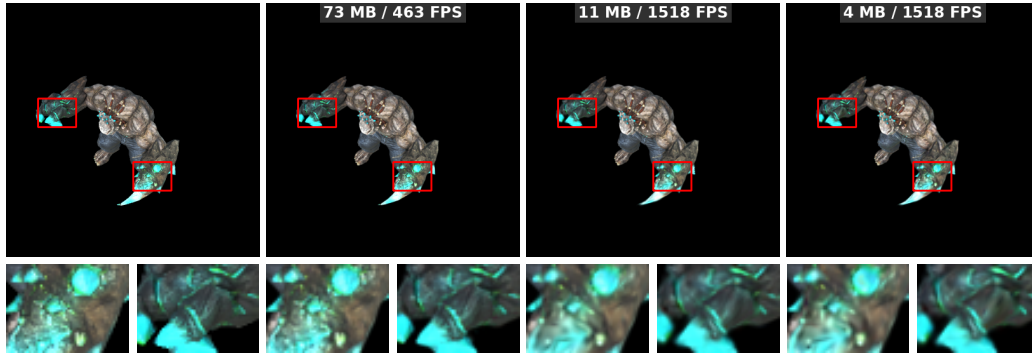
Figure 13: **Qualitative comparisons of 4DGS and our method on the D-nerf dataset.** To be continued in the next page.



(a) Results on Jumpingjacks Scene.

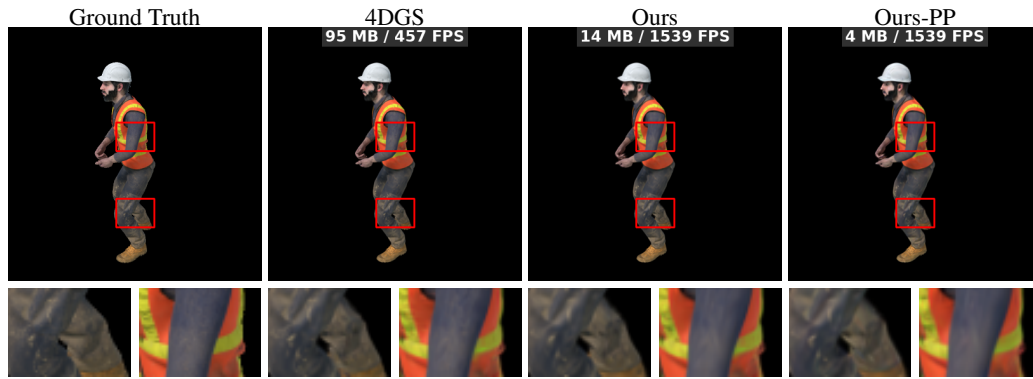


(b) Results on Lego Scene.

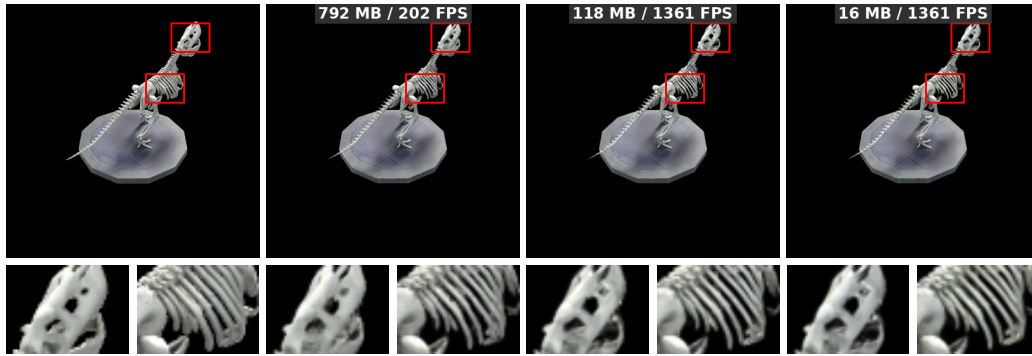


(c) Results on Mutant Scene.

Figure 14: **Qualitative comparisons of 4DGS and our method on the D-nerf dataset.** To be continued in the next page.



(a) Results on Standup Scene.



(b) Results on Trex Scene.

Figure 15: Qualitative comparisons of 4DGS and our method on the D-nerf dataset.