Turbo-GS: Accelerating 3D Gaussian Fitting for High-Resolution Radiance Fields

Supplementary Material

Contents

A. Overview

This supplementary material is organized as follows:

- **Implementation Details**: Additional information on the budget schedule (Section 4.1) and hyper-parameter settings.
- **Additional Experimental Results**: Per-scene results and an ablation study on dilated rendering.

B. Implementation Details

B.1. Convergence-aware Budget Control

To enhance the adaptiveness of the training process based on convergence patterns, we propose a dynamic scheduling mechanism that modulates the budget for each stage according to the deviation between recent and historical trends. As illustrated in Algorithm 1, this approach implements two adaptations: it dynamically adjusts both the power law exponent and the final budget target. The power law exponent is tuned based on the convergence behavior in log space, while the final budget is automatically scaled up or down depending on the loss decline rate. This dual-adaptation strategy enables the scheduler to respond effectively to varying convergence dynamics while maintaining training stability. Fig. 7 shows that with the adaptive budget control, the final number of primitives are significantly reduced while the rendering quality keeps comparable, implying that the budget control helps to add the proper number of Gaussians. And, fewer number of Gaussians denote a faster training process.

Algorithm 1 Adaptive Power Law Scheduling

Input: N, M: Numbers for initialization and final budget
$steps$: Total steps
$window\_size$: Window for trend analysis
t: Current step
Output: $B(t)$: Current scheduled value
Initialize EMA smoother with $\alpha_{ema} = 0.1$
if $len(loss\_history) \leq warmup\_steps$ **then**
  Use default power law with $\alpha = 1.0$
else
  `// Smoothing and Base Trend`
  Compute smoothed losses: $ema_t = \alpha_{ema} \cdot loss_t + (1 - \alpha_{ema}) \cdot ema_{t-1}$
  Fit in log space: $\log(ema\_losses) \sim \alpha_{base} \cdot \log(steps)$
  $\alpha_{base} \leftarrow$ Moving average of recent α_{base} values
  `// Dynamic Range Adjustment`
  $rate \leftarrow -\frac{d\log(ema\_losses)}{d\log(steps)}$ in recent window
  if $rate > 0.05$ **then**
    $M_{adaptive} \leftarrow \min(M_{adaptive} \cdot 1.1, M \cdot 1.5)$
  else if $rate < -0.05$ **then**
    $M_{adaptive} \leftarrow \max(M_{adaptive} \cdot 0.9, M \cdot 0.5)$
  end
  `// Adaptive Power Law`
  $deviation \leftarrow -rate - \alpha_{base}$
  $\alpha \leftarrow \alpha_{base} + 0.5 \tanh(deviation)$
  Clip α to $[0.1, 2.0]$
end
return $N + ((t^\alpha - 1)/(100^\alpha - 1)) \cdot (M_{adaptive} - N)$

C. More Experiments and Results

C.1. Per-scene Results

Here we list the error metrics used in our evaluation in Sec.4 across all considered methods and scenes, as shown in Tab 4- 7. **drjohnson-playroom** [5] belongs to the deep blending dataset; **train-truck** come from the Tanks and Temple [7] dataset; **bicycle-bonsai** are from MipNeRF360 [1].

C.2. Dilated Rendering

Since the tile in low resolution corresponds to multiple tiles in the high-resolution image space, there are inevitable misalignment if mimic high-resolution rendering in low-resolution space. To analyze the distribution of this part of the error, we render a series of high-resolution and their corresponding dilated low-resolution images. The difference map is shown in Fig. 8.

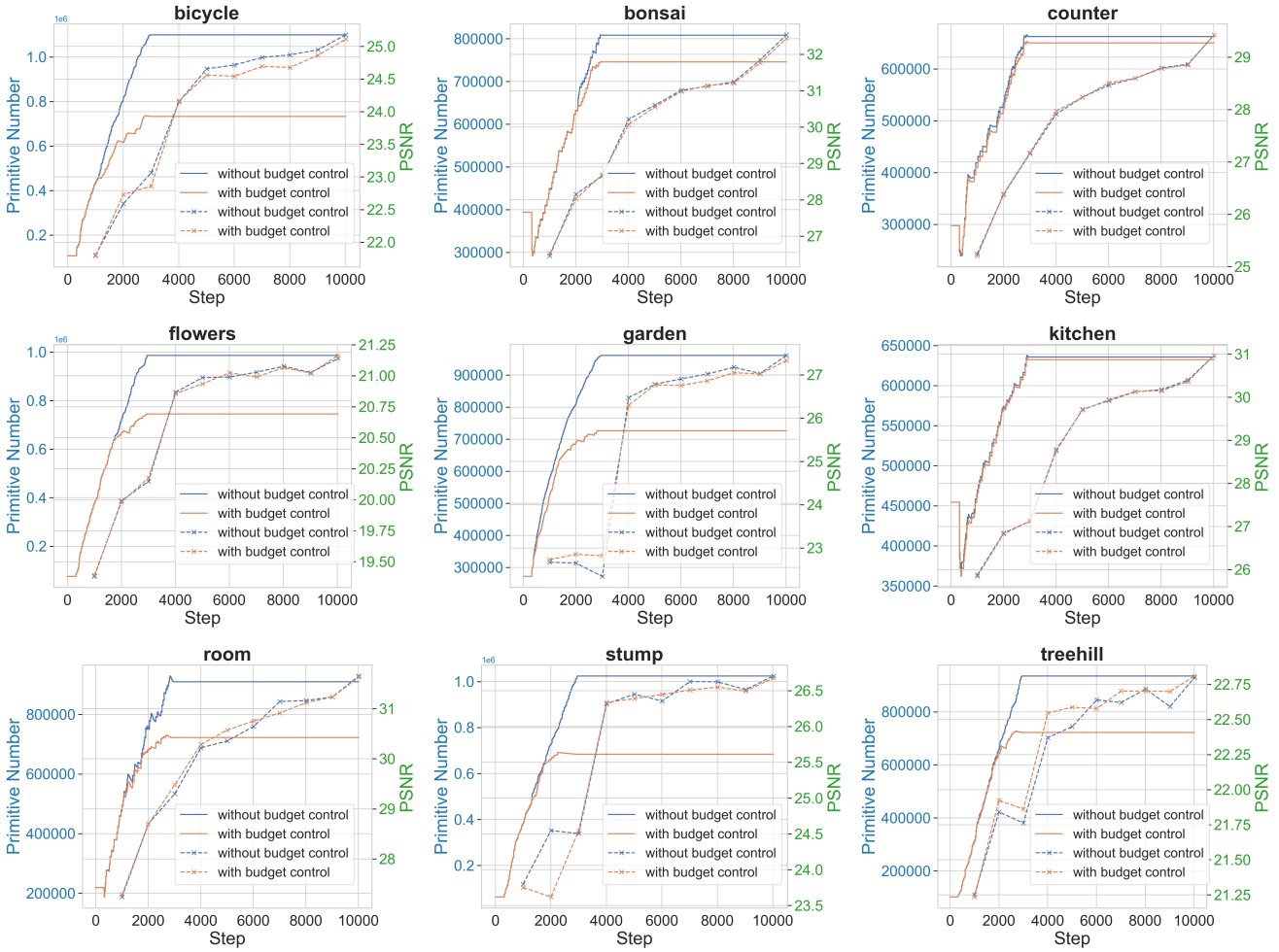

Figure 7. **Convergence.** We show "Number of primitives vs Step" and "PSNR vs Step" plots for scenes in MipNeRF-360 [1] dataset for with and without budget control in the optimization process. The proposed budgeting strategy prevents the number of primitives from increasing uncontrollably, while maintaining the overall quality. This is evident by the comparable PSNR plots, which demonstrate that the strategy maintains the balance between computational efficiency and visual fidelity.

Table 4. PSNR comparison across DeepBlending [5] ("drjohnson", "playroom"), Tanks & Temples [7] ("train", "truck") and MipNeRF-360 [1] scenes.

Method	drjohnson	playroom	train	truck	bicycle	garden	stump	flowers	treehill	counter	kitchen	room	bonsai
3DGS [6]	29.49	30.02	22.11	25.45	25.23	27.38	26.59	21.44	22.49	29.08	31.09	31.48	32.31
Taming-3DGS [9]	29.39	30.04	22.09	25.46	25.22	27.35	26.62	21.50	22.59	29.07	30.98	31.62	32.22
Taming-3DGS* [9]	29.36	29.94	21.77	25.29	25.13	27.24	26.46	21.43	22.53	28.95	31.09	31.18	32.03
Mini-Splatting [2]	29.35	30.46	21.46	25.06	25.22	26.85	27.22	21.60	22.68	28.60	31.31	31.36	31.21
EAGLES [3]	29.35	30.19	21.36	25.00	24.97	26.87	26.61	21.31	22.62	28.30	30.52	31.38	31.23
ScaffoldGS [8]	29.67	30.90	22.53	25.86	25.21	27.51	26.56	21.43	23.12	29.45	31.61	31.98	32.53
Mip-Splatting [11]	28.75	29.98	22.06	25.74	25.56	27.69	26.91	21.70	22.35	29.13	31.59	31.54	32.27
Turbo-Scaffold-GS	29.85	31.12	21.75	25.11	25.01	27.41	26.64	21.18	22.91	29.29	30.86	31.50	32.06
Turbo-3DGS	29.85	31.12	21.75	25.11	25.01	27.41	26.64	21.18	22.91	29.29	30.86	31.50	32.06

C.3. Senistivity to λ

We evaluate λ described in Equation 6 $\lambda \in \{0.2, 0.5, 0.7, 1.0\}$ in Tab. 8. We observe optimal PSNR at $\lambda = 0.5$, which is used for all our experiments.

C.4. Different Budget Schedules

We evaluate alternative budget schedules for Equation 7 in Tab. 9. We observe that our adaptive convergence-aware schedule consistently outperforms other schedules.

Table 5. SSIM comparison across DeepBlending [5] (“drjohnson”, “playroom”), Tanks & Temples [7] (“train”, “truck”) and MipNeRF-360 [1] scenes.

Method	drjohnson	playroom	train	truck	bicycle	garden	stump	flowers	treehill	counter	kitchen	room	bonsai
3DGS [6]	0.903	0.902	0.818	0.881	0.765	0.864	0.770	0.602	0.633	0.907	0.925	0.918	0.940
Taming-3DGS [9]	0.902	0.904	0.818	0.881	0.765	0.863	0.770	0.601	0.634	0.906	0.925	0.919	0.939
Taming-3DGS* [9]	0.903	0.901	0.812	0.878	0.751	0.859	0.764	0.596	0.630	0.905	0.923	0.915	0.939
Mini-Splatting [2]	0.903	0.912	0.798	0.874	0.773	0.848	0.806	0.626	0.654	0.905	0.926	0.922	0.939
EAGLES [3]	0.906	0.908	0.796	0.872	0.757	0.844	0.770	0.589	0.637	0.897	0.920	0.917	0.934
ScaffoldGS [8]	0.907	0.912	0.822	0.886	0.760	0.863	0.766	0.594	0.643	0.911	0.927	0.924	0.944
Mip-Splatting [11]	0.898	0.908	0.827	0.893	0.793	0.878	0.791	0.640	0.639	0.913	0.930	0.925	0.944
Ours	0.907	0.915	0.816	0.886	0.757	0.861	0.769	0.593	0.636	0.913	0.926	0.925	0.944

Table 6. LPIPS comparison across DeepBlending [5] (“drjohnson”, “playroom”), Tanks & Temples [7] (“train”, “truck”) and MipNeRF-360 [1] scenes.

Method	drjohnson	playroom	train	truck	bicycle	garden	stump	flowers	treehill	counter	kitchen	room	bonsai
3DGS [6]	0.238	0.243	0.198	0.143	0.211	0.108	0.217	0.339	0.329	0.201	0.127	0.220	0.205
Taming-3DGS [9]	0.239	0.243	0.200	0.144	0.210	0.109	0.217	0.341	0.328	0.202	0.127	0.219	0.206
Taming-3DGS* [9]	0.244	0.253	0.206	0.147	0.236	0.116	0.229	0.346	0.341	0.204	0.130	0.229	0.208
Mini-Splatting	0.256	0.249	0.245	0.160	0.225	0.150	0.199	0.327	0.313	0.198	0.129	0.211	0.200
EAGLES [3]	0.244	0.253	0.240	0.166	0.232	0.146	0.229	0.361	0.338	0.217	0.138	0.226	0.218
ScaffoldGS [8]	0.252	0.253	0.206	0.142	0.227	0.118	0.236	0.347	0.319	0.200	0.127	0.210	0.203
Mip-Splatting [11]	0.243	0.235	0.189	0.123	0.167	0.094	0.188	0.274	0.274	0.187	0.119	0.202	0.188
Ours	0.241	0.224	0.200	0.132	0.224	0.116	0.221	0.320	0.293	0.185	0.126	0.196	0.191

Figure 8. The error map ($\times 100$ for better view) between ”render a high-resolution image” and ”render 4 low resolution with different offsets then merge into a high-resolution image”. The artifact locates near the tile edge.

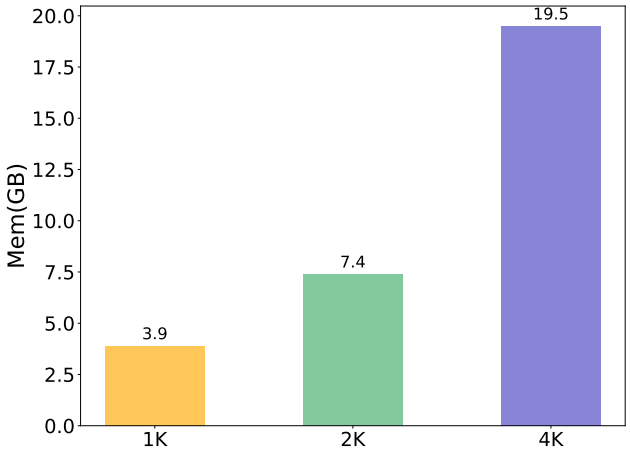

Figure 9. The memory consumption of rendering the same Gaussian checkpoint in different resolutions. We could observe a big increasement happens from 2K to 4K.

C.5. Impact of position-appearance based densification

We visualize the magnitude of color and position gradients via alpha-blending in Figure 4 and observe that color gradients are significantly more active in fine-textured (background) regions than position gradients. Impact of using both gradients during densification improve results qualitatively (Figure 6) and is quantitatively validated in Tab. 10, where the “Only Position” variant yields degraded performance.

C.6. Analysis of power function fitting on 4K scenes

Our convergence-aware training schedule is motivated by the observation that the log(loss) vs. log(iteration) curve exhibits stable linear behavior after initial stage, indicating a power-law function (Figure 5). We further validate this

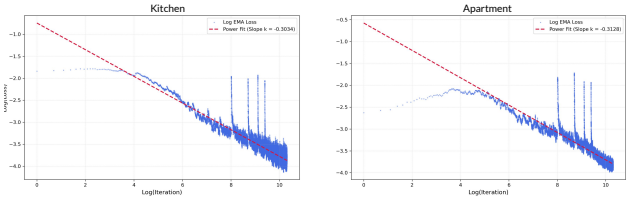

Figure 10. Loss analysis with power function fitting. **(Zoom-in)**

hypothesis in [**?**] using two 4K scenes from the Eyeful-Tower [10] dataset, which confirms that the log-linear relationship holds after the initial optimization phase.

Table 7. Training time (in minutes) comparison across DeepBlending [5] ("drjohnson", "playroom"), Tanks & Temples [7] ("train", "truck") and MipNeRF-360 [1] scenes.

Method	drjohnson	playroom	train	truck	bicycle	garden	stump	flowers	treehill	counter	kitchen	room	bonsai
3DGS	35.04	26.13	16.25	19.27	35.60	34.64	29.31	24.94	27.21	29.27	34.97	30.32	24.52
Taming-3DGS	15.71	11.02	8.11	11.74	22.58	22.31	18.88	14.65	15.37	11.44	15.51	11.35	9.92
Taming-3DGS*	9.40	6.81	6.22	7.51	11.95	12.72	9.17	8.35	8.53	9.37	14.52	8.28	7.60
Mini-Splatting	19.23	16.89	13.94	13.87	16.72	18.65	16.55	18.07	18.14	29.22	28.83	24.67	24.47
EAGLES	28.48	19.45	12.34	13.18	23.70	21.88	22.58	17.36	20.85	22.18	27.81	24.08	19.57
ScaffoldGS	17.64	16.87	15.02	14.68	22.41	23.99	18.08	21.05	21.14	27.32	32.84	23.41	24.42
Mip-Splatting	42.21	32.31	18.72	28.12	59.24	53.88	43.57	38.28	40.68	32.87	39.80	35.08	29.84
Ours	4.04	3.98	4.80	4.71	5.34	7.70	5.43	5.64	6.05	8.60	9.77	6.36	6.93

Table 8. Sensitivity to λ

$\lambda =$	**0.2**	**0.7**	**1.0**	**0.5** (Ours)
PSNR ↑	27.406	27.346	27.370	**27.423**
SSIM ↑	0.813	0.812	0.813	**0.814**
LPIPS ↓	**0.221**	**0.221**	**0.221**	**0.221**

Table 9. Comparison on different budget schedules.

	linear	**beizer**	**cosine**	**exp**	**log**	**Ours**
PSNR ↑	27.093	27.055	27.098	27.159	27.126	**27.423**
SSIM ↑	0.798	0.797	0.800	0.801	0.801	**0.813**
LPIPS ↓	0.253	0.253	0.250	0.249	0.249	**0.221**

Table 10. Impact of position-appearance based densification

	Only Position	Position + Color (Ours)
PSNR ↑	27.285	**27.423**
SSIM ↑	0.804	**0.814**
LPIPS ↓	0.238	**0.221**

Table 11. Original Image resolutions of scenes in the MipNeRF-360 dataset.

Scene	Width	Height
bicycle	4946	3286
flower	5025	3312
garden	5187	3361
stump	4978	3300
treehill	5068	3326

C.7. Memory Consumption

In the main paper we show the timebreakdown for different resolution rendering. Here we plot the memory consumption for different resolutions in Fig. 9.

C.8. Comparison on MipNeRF-360 4K dataset.

MipNeRF-360 [1] contains nine scenes but only has five scenes with $\geq 4K$ resolution. We describe these five scenes in Tab. 11. All these scenes are outdoor scenes. Tab. 12 compares our method and recent baselines on this challenging high resolution dataset.

Table 12. Quantitative comparison with Gaussian Splatting baselines for 4K scenes. We compare PSNR, SSIM, and LPIPS for quality. For resource efficiency, we report training time, memory and, where applicable and peak number (Peak #G) of Gaussians. '-accel' means accelerated with optimized CUDA implementation.

	MipNeRF-360 4K [1]					
	SSIM↑	PSNR↑	LPIPS↓	Train time ↓	Memory ↓	Peak #G ↓
3DGS [6]	0.797	26.75	0.410	113 m	412 MB	**1.74 M**
3DGS-accel [6]	0.791	26.52	0.364	40 m	383 MB	1.62 M
SpeedySplat [4]	0.678	23.23	0.467	97 m	**65.6 MB**	1.79 M
Turbo-3DGS (ours)	**0.805**	**26.80**	**0.327**	**24 m**	680 MB	2.80 M

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 1, 2, 3, 4

[2] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. *European Conference on Computer Vision*, 2024. 2, 3

[3] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. *European Conference on Computer Vision*, 2024. 2, 3

[4] Alex Hanson, Allen Tu, Geng Lin, Vasu Singla, Matthias Zwicker, and Tom Goldstein. Speedy-splat: Fast 3d gaussian splatting with sparse pixels and sparse primitives. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 21537–21546, 2025. 4

[5] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018. 1, 2, 3, 4

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 4

[7] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene

reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017. 1, 2, 3, 4

[8] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2, 3

[9] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[10] Linning Xu, Vasu Agrawal, William Laney, Tony Garcia, Aayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kontschieder, Aljaž Božič, Dahua Lin, Michael Zollhöfer, and Christian Richardt. VR-NeRF: High-fidelity virtualized walkable spaces. In *SIGGRAPH Asia Conference Proceedings*, 2023. 3

[11] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2023. 2, 3