3D Gaussian Splatting for Efficient Retrospective Dynamic Scene Novel View Synthesis with a Standardized Benchmark

Supplementary Material

7. More Detail on Blender API

While Neural Radiance Fields (NeRF) and 3D Gaussian Splatting (3DGS) have achieved remarkable progress in novel view synthesis , preparing multi-view datasets across different methods remains a bottleneck due to varying format requirements and coordinate system conventions To alleviate the extensive manual setup and debugging typically required for dataset conversion, we introduce our comprehensive Blender add-on designed to streamline the generation of synthetic datasets.

It facilitates the extraction of consistent camera parameters and rendered images over time, bridging the gap between synthetic scene creation in Blender and ready-to-train pipelines for various 3DGS and NeRF variants in just a few clicks It is particularly well-suited for dynamic scene evaluation and complex camera virtualization tasks, such as those involving sports broadcasting

7.1. Key Features

Our proposed tool significantly accelerates the data preparation and empirical evaluation processes through the following core functionalities:

- **Versatile Camera Generation & Stadium Layout:** The add-on supports standard sampling layouts, including upper hemisphere, full sphere, and elliptical rings Crucially for large-scale and sports-focused applications, it introduces a specialized *Stadium Layout* designed for soccer and football stadiums This feature allows users to define field dimensions, generate grid-based camera tiles , and automatically position goal-line cameras , ensuring comprehensive multi-view coverage for methods like TACV
- **Flexible Dataset Export Modes:** To support modern point-based rendering techniques, RS Studio exports point clouds using either a 3DGS COLMAP surface mode (sampled directly from the mesh surface) or a 3DGS depth mode (generated via ray intersections from RS Studio cameras)
- **Dynamic Camera Tracking:** For downstream evaluation and instant render testing of dynamic models , the tool exports per-frame camera tracks with intrinsics and extrinsics formatted as JSON files Supported tracking modes include keyframed animation, automated orbit paths, linear interpolation, and target following
- **Dataset Import and Pose Validation:** Users can import existing datasets in standard NeRF synthetic or custom TACV formats to visually verify camera poses within the 3D environment or reuse existing configurations for new data generation
- **Geographically-Aware Lighting:** Through integration with physical solar models, the tool provides adjustable indoor and outdoor lighting driven by specific geographic coordinates (latitude and longitude) and time , enabling robust illumination augmentation for synthetic scenes.

By open-sourcing RS Studio, we aim to provide the community with a robust pipeline that lowers the barrier to entry for constructing complex, temporal, and large-scale synthetic datasets.

8. Additional Ablation Studies

Here, we provide additional analyses supporting the two central claims of the paper: (i) our time-archival 3DGS formulation preserves the practical efficiency of Gaussian splatting for retrospective rendering, and (ii) warm-start training primarily exploits local temporal smoothness rather than any implicit temporal coupling in the objective. Unless otherwise stated, all evaluations follow the same camera calibration, train/val/test split protocol, and training implementation described in the main paper.

Table 5. Real-time rendering comparison. TA-3DGS preserves the native rendering efficiency of Gaussian splatting while operating at full HD resolution (1920×1080). On a single A40 GPU, our method achieves real-time rendering across three TACV datasets, reaching 65.8 FPS on DWS, 55.9 FPS on SPK, and 59.9 FPS on SMP. Prior dynamic Gaussian methods such as 4DGS and FreeTimeGS also report real-time rendering, although the reported numbers are measured on different GPUs and resolutions.

Method	GPU	Rendering Resolution	FPS ↑
4DGS [51]	RTX 3090	800×800	82
FreeTimeGS [50]	RTX 4090	1920×1080	450
TA-3DGS (ours, DWS)	A40	1920×1080	65.8
TA-3DGS (ours, SPK)	A40	1920×1080	55.9
TA-3DGS (ours, SMP)	A40	1920×1080	59.9

8.1. Real-Time Rendering Throughput at Full HD

Table 5 reports rendering throughput at full HD (1920×1080) on a single NVIDIA A40 GPU. TA-3DGS achieves real-time performance across all three TACV sequences, reaching 65.8 FPS on DWS, 55.9 FPS on SPK, and 59.9 FPS on SMP. These results indicate that our time-archival representation retains the native runtime advantages of Gaussian splatting despite operating at high resolution.

Resolution-normalized throughput. Since prior dynamic Gaussian approaches report FPS under different resolutions and GPU configurations, direct FPS comparisons are not strictly controlled. For additional context, we report a resolution-normalized throughput in megapixels per second (MPix/s), computed as $\mathrm{MPix/s} = \mathrm{FPS} \cdot H \cdot W / 10^6$. Under this metric, TA-3DGS achieves 136.4 MPix/s (DWS), 115.9 MPix/s (SPK), and 124.2 MPix/s (SMP) at 1920×1080. By comparison, 4DGS reports 52.5 MPix/s at 800×800, while FreeTimeGS [50] reports substantially higher throughput at 1920×1080 on an RTX 4090. We emphasize that these numbers remain indicative only, as GPU architecture and implementation details differ across works.

8.2. Warm-Start Test

A key interpretability question is whether warm-starting yields benefits because the optimization implicitly encodes a temporal constraint (e.g., through hidden coupling in the loss), or simply because adjacent frames are typically locally smooth. To isolate this effect, we introduce an **"distant warm-start"** baseline on DWS (Table 6). This baseline keeps the dataset, loss, and training pipeline fixed, but replaces the standard adjacent-frame initialization with an initialization from a temporally distant checkpoint; the result is reported on frames 40–49.

The distant warm-start baseline exhibits a **"large quality degradation"**, dropping PSNR from 42.50 to 26.80 (absolute drop: 15.70 dB; relative: -36.94%) and increasing LPIPS from 0.0111 to 0.0246 (absolute increase: 0.0135; relative: $+121.99\%$). This controlled perturbation supports the interpretation that warm-start primarily leverages local temporal continuity and a favorable basin of attraction for optimization, rather than relying on any implicit temporal coupling embedded in the objective formulation.

8.3. Extended Warm-Start Analysis

The main paper reports warm-start ablations over a short temporal window i.e., 5 frame window. Here, we extend the evaluation to 15 time instances for the dynamic component (Table 7), comparing: **S1** (Warm + Densify), **S2** (Warm + NoDensify, ours), and **S3** (GT Init per frame). This extended analysis clarifies two practical properties: quality robustness over longer warm-start chains and the effect of densification on apparent quality gains.

Quality trends. Across 15 frames, S1 achieves consistently higher PSNR and lower LPIPS than S2 and S3; however, this improvement is expected because densification increases model capacity over time, which is precisely the scalability issue highlighted in the main paper. In contrast, S2 maintains a fixed Gaussian budget and therefore prioritizes stability in representation size and speed. Despite this constraint, S2 remains competitive with the per-frame GT initialization baseline S3. Averaged across the 15 frames, S2 attains 28.21 dB PSNR and 0.3998 LPIPS, while S3 attains 28.37 dB PSNR and 0.3945 LPIPS. The small gap indicates that warm-starting with fixed capacity is able to preserve most of the per-frame reinitialization quality while avoiding its computational overhead and impractical sensing requirements.

Table 6. **Effect of abnormal distant warm-start on DWS.** Compared with the standard warm-start setting, initializing a target segment from a temporally distant checkpoint causes a substantial quality drop on DWS. This supports the interpretation that warm-start primarily benefits from *local temporal smoothness*, rather than any hidden temporal coupling in the loss design.

Setting	**PSNR↑**	**LPIPS↓**
Standard warm-start	42.5027	0.0111
Distant warm-start baseline	26.8031	0.0246
Absolute drop / increase	-15.6996	+0.0135
Relative change	-36.94%	+121.99%

Note: The distant warm-start baseline keeps the same dataset and training pipeline, but replaces normal adjacent-frame initialization with an abnormal initialization from a temporally distant checkpoint. The reported distant warm-start result is measured on frames 40–49 of DWS.

Interpretation. The extended results reinforce the paper's central claim: under synchronized multi-view constraints, a temporally chained optimization strategy can remain stable over time even without explicit deformation constraints. Meanwhile, the superior quality of S1 should be interpreted in light of its expanding model complexity (enabled by densification), rather than as evidence that explicit temporal coupling is necessary.

8.4. Per-Frame Training Throughput Under Fixed-Capacity Warm-Start

Table 8 reports the per-iteration time for S2 across the same 15-frame window. Even with a fixed number of Gaussians, the iteration time increases from 12.34 ms (frame 1) to 34.50 ms (frame 15), with an average of 24.91 ms per iteration. This increase is consistent with the fact that the rasterization workload of Gaussian splatting depends not only on the *count* of Gaussians but also on scene-dependent visibility patterns and screen-space coverage, which can change over time for dynamic sequences.

Practical implication. Despite this variability, S2 maintains predictable memory and representation size (no densification) while sustaining practical training throughput across time. This observation complements the real-time rendering results (Section 8.1) and supports the overall deployment motivation of time-archival retrospective rendering in long dynamic events.

Table 7. **Extended Analysis of Warm Start Initializations.** We provide a detailed comparison of S1 (Warm+Densify), S2 (Warm+NoDensify), and S3 (GT Init) across 15 time instances. These extended results for the dynamic part of the scene further support the robustness of our approach (S2) beyond the 5 frames presented in the main paper.

	PSNR ↑			LPIPS ↓		
Frame	S1	S2	S3	S1	S2	S3
1	28.386	27.229	28.356	0.3948	0.4257	0.3954
2	29.117	27.727	28.394	0.3733	0.4118	0.3942
3	29.867	27.980	28.343	0.3532	0.4058	0.3948
4	30.595	28.103	28.345	0.3350	0.4024	0.3952
5	31.253	28.221	28.367	0.3246	0.3998	0.3951
6	31.856	28.255	28.320	0.3163	0.3982	0.3943
7	32.222	28.270	28.351	0.3108	0.3972	0.3943
8	32.643	28.317	28.345	0.3071	0.3959	0.3942
9	32.839	28.375	28.388	0.3035	0.3953	0.3940
10	33.235	28.381	28.342	0.3005	0.3948	0.3946
11	33.356	28.424	28.439	0.2990	0.3947	0.3948
12	33.519	28.412	28.388	0.2975	0.3941	0.3949
13	33.707	28.431	28.442	0.2964	0.3944	0.3940
14	33.773	28.491	28.421	0.2950	0.3938	0.3940
15	33.818	28.484	28.361	0.2944	0.3934	0.3938

Table 8. **Per-frame Training Speed of S2.** Detailed iteration time (ms) for our proposed S2 setting across the extended sequence.

Frame	Iteration Time (ms)
1	12.340
2	14.581
3	16.810
4	18.900
5	20.887
6	22.770
7	24.506
8	25.809
9	27.194
10	28.618
11	29.888
12	31.100
13	32.348
14	33.395
15	34.498