Temporally-Aware Pruning for Compact 4D Gaussian Splatting Models

Enming Zhang, Mingzhe Zhang
University of Toronto

Motivation

- 4D Gaussian Splatting (4DGS) enables real-time, high-fidelity rendering of dynamic scenes, but the models are extremely redundant, which contains millions of Gaussians, leading to large storage, slow rendering, and heavy computation. Existing pruning methods mostly use static heuristics (opacity, scale) and ignore the temporal behavior in dynamic scenes.
- Many Gaussians show simple or predictable motion, or remain nearly static across time. These Gaussians contribute little to temporal fidelity and can be safely merged. Conversely, a small subset with complex motion, high curvature, or temporally varying appearance must be preserved.
- This motivates a dynamics-aware pruning framework that evaluates each Gaussian's spatio-temporal importance, using statistics such as trajectory magnitude, motion curvature, and opacity variation.
- By combining these with static cues (coverage, anisotropy), we can selectively keep Gaussians that drive dynamic complexity while aggressively removing redundant ones.
- Our goal is to achieve 40-60% Gaussian reduction and 1.5–2× rendering speedup, with minimal loss in temporal quality (tLPIPS / PSNR / SSIM), through a post-hoc, architecture-free method.

Related Work

Foundations in Static Scenes

- **3D Gaussian Splatting [3]** revolutionized radiance fields by enabling high-fidelity, real-time rendering for static scenes using explicit Gaussian representations.

Evolution to Dynamic Scenes

- Early neural approaches like **Neural 3D Video [2]** achieved impressive quality for dynamic scenes but relied on implicit representations, limiting real-time performance.
- Recently, **4D Gaussian Splatting [1]** extended the explicit representation to the temporal dimension, achieving real-time rendering for dynamic scenes.

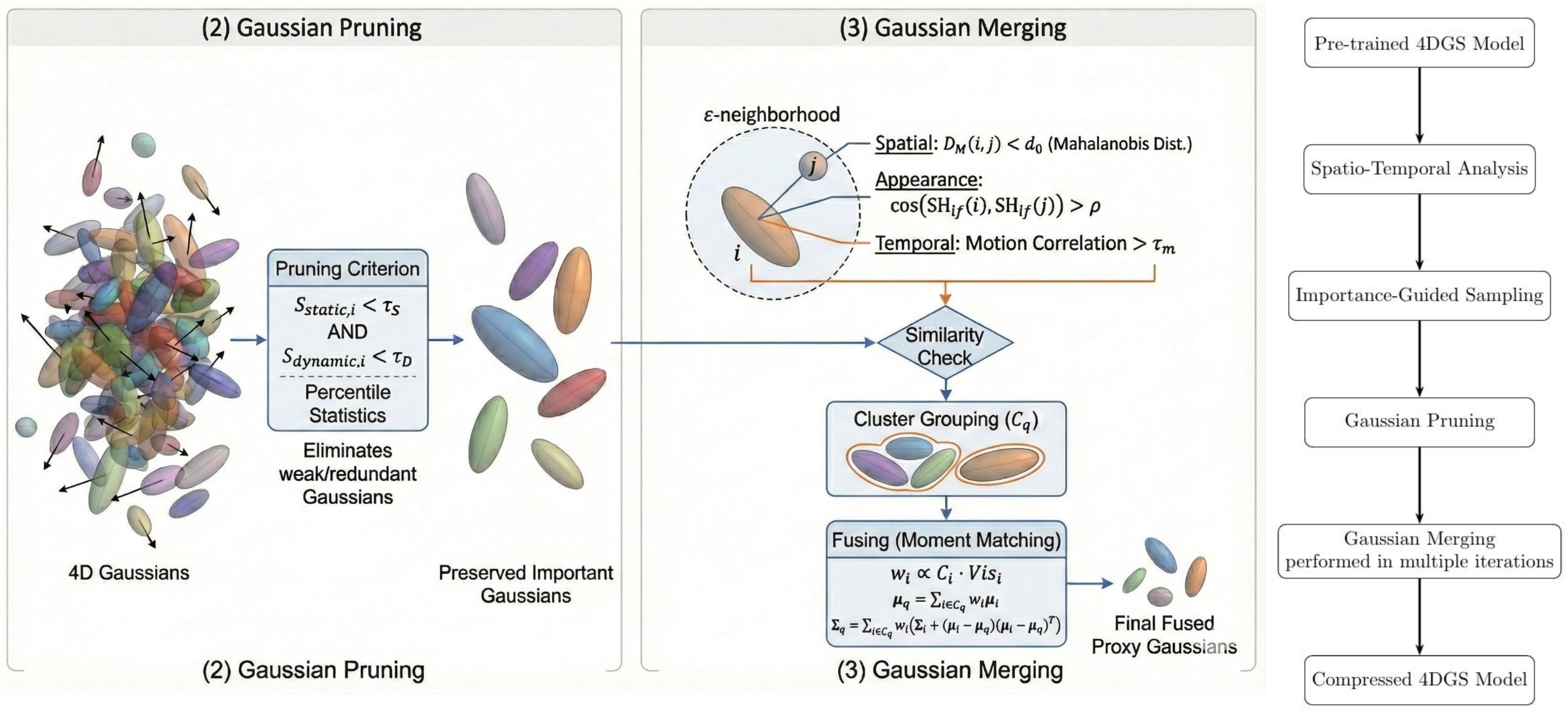
Limitations & Our Approach

- Despite its speed, the 4DGS model **[1]** introduces significant redundancy and high storage costs. Our work focuses on optimizing the 4DGS model by introducing a dynamics-aware pruning strategy to reduce model size while maintaining visual fidelity.

References

[1] Wu et al., "4D Gaussian Splatting for Real-Time Dynamic Scene Rendering," CVPR, 2024.
[2] Li et al., "Neural 3D Video Synthesis from Multi-View Video," CVPR, 2022.
[3] Kerbl et al. "3D Gaussian Splatting for Real-Time Radiance Field Rendering". SIGGRAPH 2023.

New Technique



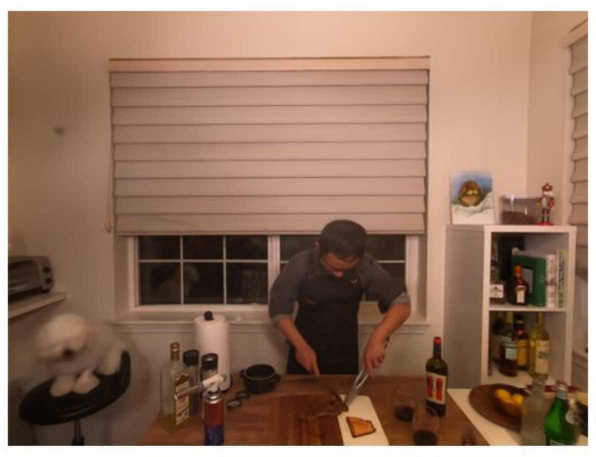
Overview of the Dynamics-Aware Gaussian Compression Pipeline.
Our method proceeds in two post-processing stages; additionally, we introduce a lightweight importance-guided sampling step to pre-filter candidates before pruning. Unlike prior static-only pruning or retraining-based compression approaches (e.g., temporal smoothing, opacity heuristics, or grid-based architectures), our method directly evaluates spatio-temporal importance from a pre-trained 4DGS model without modifying the rendering pipeline.

Gaussian Pruning. In this phase, we filter the pre-trained 4D Gaussians based on calculated hybrid importance scores. By jointly evaluating static saliency (S_{static}) and dynamic contribution (S_{dynamic}), we apply adaptive pruning thresholds τ_s and τ_d. This mechanism effectively identifies and eliminates Gaussians that are visually insignificant (e.g., low opacity) or temporally redundant (e.g., static or predictable motion), while preserving structures that drive complex dynamics. Compared to related work that applies only opacity/scale pruning or temporal smoothing, our pruning specifically preserves Gaussians with high motion complexity, improving temporal fidelity under the same compression ratio.

Gaussian Merging. To further reduce redundancy, we perform similarity-based merging on the preserved Gaussians. We search within an ε-neighborhood and enforce a strict triple-consistency check: spatial Mahalanobis distance (D_M), appearance similarity via Spherical Harmonics (cosine similarity), and Temporal Motion Correlation. Valid clusters (C_q) are mathematically fused using Moment Matching, aggregating weights and covariance to generate a single "Proxy Gaussian," thereby reducing storage overhead while maintaining geometric and appearance fidelity. This step complements pruning by collapsing groups of spatially and temporally consistent Gaussians, further reducing model size without introducing artifacts.

Experimental Results

Ground Truth



Not Pruned
Gaussians: 110K
FPS: 105

Pruned
Gaussians: 73K
FPS: 116

- **Qualitative**: As shown in the visual comparison, our pruned model (Right) maintains almost the same rendering quality compared to the unpruned baseline (Middle) and Ground Truth (Left), effectively preserving fine details and dynamic motions.
- **Quantitative**: The table below reports the per-scene metrics. Despite a significant reduction in model size, our model achieves competitive PSNR and SSIM scores comparable to the full 4DGS model, demonstrating an excellent trade-off between efficiency and visual fidelity.

Table 1: Per-scene results on synthetic datasets. We compare our method with 3D-GS, K-Planes, HexPlane, TiNeuVox, and 4D-GS (Real-Time).

Method	Bouncing Balls			Hellwarrior			Hook			Jumpingjacks		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3D-GS	23.20	0.9591	0.0600	24.53	0.9336	0.0580	21.71	0.8876	0.1034	23.20	0.9591	0.0600
K-Planes	40.05	0.9934	0.0322	24.58	0.9520	0.0824	28.12	0.9489	0.0662	31.11	0.9708	0.0468
HexPlane	39.86	0.9915	0.0323	24.55	0.9443	0.0732	28.63	0.9572	0.0505	31.31	0.9729	0.0398
TiNeuVox	40.23	0.9926	0.0416	27.10	0.9638	0.0768	28.63	0.9433	0.0636	33.49	0.9771	0.0408
4D-GS (RT)	**40.62**	**0.9942**	**0.0155**	**28.71**	**0.9733**	**0.0369**	**32.73**	**0.9760**	**0.0272**	**35.42**	**0.9857**	**0.0128**
Ours	39.36	0.9928	0.0212	25.77	0.9590	0.0537	30.03	0.9641	0.0380	33.13	0.9800	0.0258

Method	Lego			Mutant			Standup			Trex		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3D-GS	23.06	0.9290	0.0642	20.64	0.9297	0.0828	21.91	0.9301	0.0785	21.93	0.9539	0.0487
K-Planes	**25.49**	**0.9483**	**0.0331**	32.50	0.9713	0.0362	33.10	0.9793	0.0310	30.43	0.9737	0.0343
HexPlane	25.10	0.9388	0.0437	33.67	0.9802	0.0261	34.40	0.9839	0.0204	30.67	0.9749	0.0273
TiNeuVox	24.65	0.9063	0.0648	30.87	0.9607	0.0474	34.61	0.9797	0.0326	31.25	0.9666	0.0478
4D-GS (RT)	25.03	0.9376	0.0382	**37.59**	**0.9880**	**0.0167**	**38.11**	**0.9898**	**0.0074**	**34.23**	**0.9850**	**0.0131**
Ours	24.58	0.9211	0.0731	32.34	0.9740	0.0286	34.26	0.9833	0.0201	31.33	0.9760	0.0305