ELGD-GS: ERROR-GUIDED LOCALIZATION AND GRADIENT CONSISTENCY-AWARE DENSITY CONTROL FOR EFFICIENT DYNAMIC SCENE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

4D Gaussian Splatting (4DGS) delivers high-quality dynamic scene reconstruction with real-time rendering. However, existing 4DGS methods adopt densification strategies that allocate Gaussians inefficiently. They over-populate well-reconstructed regions and ignore the influence of gradient consistency, leading to unnecessary storage overhead and suboptimal reconstruction quality. In this paper, we propose **ELGD-GS**, an **E**rror-guided **L**ocalization and **G**radient Consistency-aware **D**ensity Control of 4D Gaussian Splatting framework that enables high-quality dynamic scene reconstruction with efficient Gaussian representations. First, we introduce an error-guided Gaussian localization mechanism to identify 4D Gaussians requiring densification based on pixel-wise reconstruction errors. By recording pixel-wise photometric residuals and high-frequency structural discrepancies over time, this mechanism effectively prevents redundant Gaussian growth in well-reconstructed regions. Second, we propose a gradient consistency-aware density control strategy that measures gradient direction consistency across timestamps to regulate Gaussian growth. It facilitates the effective splitting of Gaussians affected by conflicting gradients while suppressing unnecessary densification of Gaussians with highly consistent gradient directions, thereby improving structural fidelity and representation efficiency. Experiments across multiple datasets demonstrate that ELGD-GS achieves substantial improvements in reconstruction quality and highly compact representations, outperforming the state-of-the-art method TAD-GS by a 0.42 dB PSNR gain and a 58% storage reduction on the N3DV dataset. The source code will be released at `http://github.com`.

1 INTRODUCTION

Novel view synthesis for dynamic scenes has attracted increasing attention owing to its broad applications in immersive media, virtual reality, and content creation. NeRF-based methods Mildenhall et al. (2021); Fridovich-Keil et al. (2023); Li et al. (2022) achieve impressive results by modeling scenes as neural implicit representations, but their high computational cost leads to slow rendering and limited practicality. 3D Gaussian Splatting Kerbl et al. (2023) offers an efficient alternative with explicit Gaussian primitives, enabling high-quality real-time rendering. Recent approaches extend 3D Gaussian Splatting to dynamic scenes through different temporal modeling strategies Lu et al. (2025); Yan et al. (2025); Go et al. (2025). Some methods adopt a canonical set of 3D Gaussians with a learnable deformation field that predicts per-primitive deformations and warps them to each timestep Jiawei et al. (2024); Bae et al. (2024); Guo et al. (2024); Lu et al. (2024); Wu et al. (2025). Although effective, these methods struggle with large and complex dynamic scenes: as motion magnitude grows, establishing correspondence over a broader spatiotemporal range becomes more challenging, which degrades motion estimation accuracy Li et al. (2021); Wang et al. (2025b). Other methods instead model temporal variations directly in a unified spatiotemporal space Wang et al. (2025b); Yang et al. (2024a); Duan et al. (2024); Liao et al. (2026), achieving higher visual quality and faster rendering Cho et al. (2026). Since motion is modeled directly by 4D Gaussian primitives rather than by explicit warping, appropriate densification, which introduces new 4D Gaussians at necessary locations, becomes particularly important.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

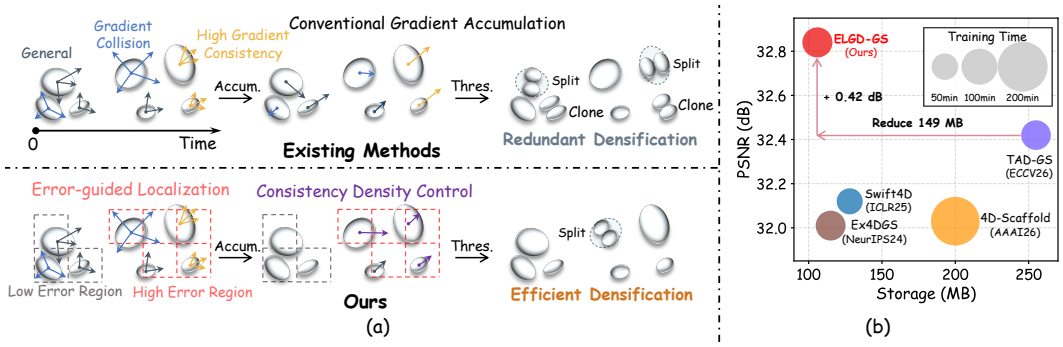

Figure 1: Comparison of existing 4DGS methods and ELGD-GS. (a) Existing methods adopt conventional gradient accumulation-based densification Wu et al. (2025); Sandu et al. (2026); Cho et al. (2026), while our method utilizes reconstruction errors over time to localize Gaussians requiring densification and leverages gradient direction consistency across timestamps for density control. *Accum.* and *Thres.* denote accumulation and thresholding operations, respectively. (b) Performance comparison on the N3DV dataset, where the marker radius indicates the training time.

Despite recent advances, 4D Gaussian Splatting methods still face several key issues that limit their efficient representation and high-quality reconstruction. Existing methods, e.g., Swift4D Wu et al. (2025), TAD-GS Sandu et al. (2026), 4D-Scaffold Cho et al. (2026), mainly rely on conventional gradient accumulation inherited from standard 3DGS Kerbl et al. (2023) to determine 4D Gaussian densification, without explicitly considering whether the corresponding regions still require additional Gaussian primitives. As shown in Fig. 1(a), regions with low reconstruction errors are already well represented by existing Gaussians, yet may still produce gradients that trigger unnecessary densification due to blindly accumulating gradients without reconstruction-error awareness. The resulting redundant primitives increase storage overhead and degrade reconstruction quality. Furthermore, these densification strategies overlook the directional behavior of pixel-wise sub-gradients. Sub-gradients with opposing directions can cancel out, reducing gradient magnitudes—a phenomenon termed gradient collision Ye et al. (2024). During dynamic reconstruction, Gaussians covering many pixels are prone to this collision, which hinders effective splitting and causes blurred structures and degraded reconstruction quality (blue arrows). Conversely, Gaussians with highly consistent gradient directions (yellow arrows) may accumulate amplified gradients and be repeatedly densified, yielding redundant primitives.

To address these issues, we propose an **E**rror-guided **L**ocalization and **G**radient Consistency-aware **D**ensity Control (**ELGD**) framework for 4D Gaussian Splatting. Our core insight is to utilize reconstruction errors to localize Gaussians requiring densification and leverage gradient direction consistency to achieve efficient densification, as shown in Fig. 1(a). Specifically, we first introduce an error-guided Gaussian localization mechanism to precisely identify 4D Gaussians that require densification. By measuring pixel-wise photometric residuals and high-frequency structural discrepancies over time, this mechanism avoids gradient accumulation in well-reconstructed regions and guides densification in under-reconstructed ones. Moreover, to address gradient collision and highly consistent gradient directions, we develop a gradient consistency-aware density control strategy that regulates 4D Gaussian densification. Since the gradient directions accumulated by a 4D Gaussian vary across timestamps, our strategy measures each Gaussian's gradient direction consistency over time to regulate density and enable more appropriate densification for dynamic regions. The primary contributions are summarized as follows:

- We develop an **error-guided Gaussian localization** mechanism that integrates photometric residuals and structural discrepancies over time to identify under-reconstructed regions, preventing redundant densification in well-reconstructed areas and enabling high-quality dynamic scene reconstruction and efficient Gaussian representations.
- We propose a **gradient consistency-aware density control** strategy that regulates 4D Gaussian densification according to the consistency of gradient direction across time, promoting the densification of Gaussians affected by conflicting gradients while suppressing unnecessary growth under highly consistent gradients.

- Extensive experiments on widely used dynamic scene reconstruction benchmarks demonstrate that our method enables significantly efficient 4D Gaussian densification. Outperforming state-of-the-art method TAD-GS by **a 0.42 dB PSNR gain** while **reducing storage by 149 MB (58%)** on the N3DV dataset, as shown in Fig. 1(b).

2 RELATED WORK

2.1 GAUSSIAN SPLATTING FOR DYNAMIC SCENE RECONSTRUCTION

3D Gaussian Splatting has been widely extended to dynamic scene reconstruction due to its fast rendering and high reconstruction quality. Existing methods can be broadly categorized into explicit 4D Gaussian representations and deformable 3D Gaussians. Explicit 4D methods Yang et al. (2024a); Duan et al. (2024); Wang et al. (2025b) directly model spatiotemporal variations in a unified four-dimensional space. For example, 4DGS Yang et al. (2024a) employs native 4D Gaussian primitives, 4D-RotorGS Duan et al. (2024) introduces anisotropic 4D Gaussians with temporal slicing, and FreeTimeGS Wang et al. (2025b) further models flexible local motions across space and time. In contrast, deformable 3D Gaussian methods maintain canonical 3D Gaussians and model temporal variations through time-dependent deformation. Representative approaches employ voxel-based deformation Wu et al. (2024), spline or analytical trajectory modeling Park et al. (2025); Lin et al. (2024); Zhang et al. (2026), and neural deformation fields Yang et al. (2024b); Shaw et al. (2024). Recent methods further improve efficiency through keyframe interpolation Lee et al. (2024), dynamic-static decomposition Wu et al. (2025); Kwak et al. (2025), or structured 4D anchors Cho et al. (2026). Despite these advances, efficient Gaussian allocation remains crucial for achieving compact and high-fidelity dynamic scene reconstruction across diverse and challenging scenes.

2.2 GAUSSIAN DENSIFICATION AND COMPACT REPRESENTATION

A substantial amount of work has focused on improving Gaussian densification in static scenes. Revising Densification Rota Bulò et al. (2024) adopts reconstruction errors to guide Gaussian growth, while Pixel-GS Zhang et al. (2024) reweights positional gradients according to pixel coverage. AbsGS Ye et al. (2024) identifies gradient collision caused by canceling pixel-wise sub-gradients and introduces homodirectional gradients to improve splitting. GDAGS Zhou et al. (2026) exploits gradient direction consistency for density control, while AD-GS Patle et al. (2025) and SteepGS Wang et al. (2025a) adopt alternating densification and optimization-guided splitting, respectively. In contrast, densification for dynamic Gaussian representations remains less explored. SaRO-GS Yan et al. (2024) adapts optimization and densification thresholds for short-lived Gaussians, while STG Li et al. (2024) introduces new Gaussians in high-error regions through guided sampling. Anchored-4DGS Li et al. (2025) adjusts anchor growth based on temporal coverage, and TAD-GS Sandu et al. (2026) employs visibility-aware gradient accumulation and temporally adaptive threshold to promote the densification of short-lived Gaussians. Despite these advances, efficient Gaussian allocation during dynamic densification remains underexplored, often leading to redundant Gaussian growth and increased storage. 4DGS-1K Yuan et al. (2026) introduces a spatial-temporal variation score to prune short-lifespan Gaussians. Pruning-based compression may compromise reconstruction fidelity, whereas our method prevents unnecessary Gaussian growth during densification and achieves higher reconstruction quality with fewer Gaussian primitives.

3 PRELIMINARIES

4D Gaussians represent a dynamic scene using a set of 4D Gaussian primitives $\mathcal{G} = \{\mathcal{G}_i\}$. Each Gaussian $\mathcal{G}_i$ is parameterized by a 4D mean $\boldsymbol{\mu}_i \in \mathbb{R}^4$, covariance $\boldsymbol{\Sigma}_i \in \mathbb{R}^{4\times4}$, opacity σ_i, and appearance feature $\mathbf{q}_i$. A 4D Gaussian is defined as:

$$\mathcal{G}_i(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right), \tag{1}$$

where $\mathbf{x} = (x, y, z, t)^T$ denotes a point in the spatiotemporal space. At timestamp t, each 4D Gaussian is conditioned into a 3D Gaussian and projected onto the image plane. Its projected covariance is given by:

$$\boldsymbol{\Sigma}_i' = \mathbf{J}\mathbf{W}\boldsymbol{\Sigma}_{i|t}\mathbf{W}^T\mathbf{J}^T, \tag{2}$$

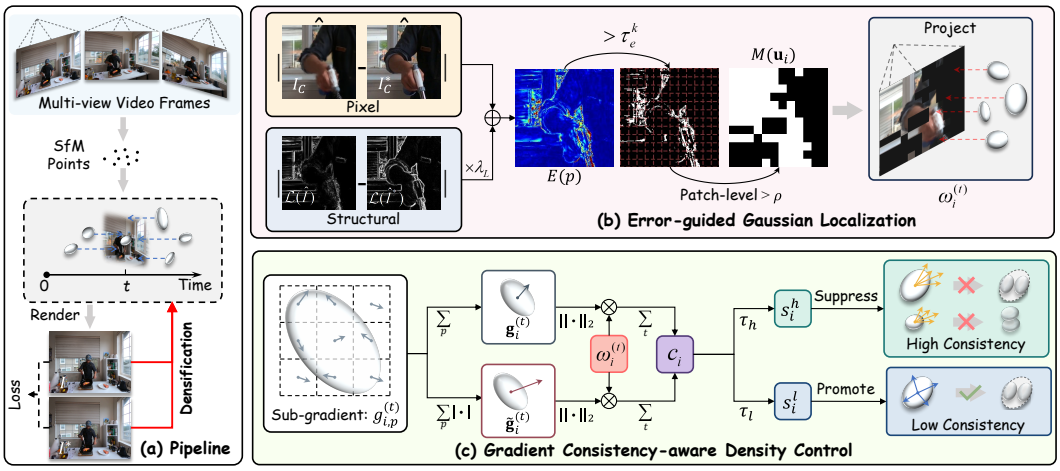

Figure 2: Overview of the proposed ELGD-GS framework. (a) ELGD-GS enables efficient densification of 4D Gaussian Splatting for dynamic scene reconstruction. (b) During densification, the error-guided Gaussian localization mechanism identifies Gaussians requiring densification according to photometric residuals and structural discrepancies, preventing redundant Gaussian growth in well-reconstructed regions. (c) The gradient consistency-aware density control strategy further regulates Gaussian growth according to gradient direction consistency, promoting the splitting of the Gaussians affected by gradient collision while suppressing unnecessary densification under highly consistent gradients.

where $\mathbf{W}$ denotes the viewing transformation, $\mathbf{J}$ is the Jacobian of the perspective projection, and $\mathbf{\Sigma}_{i|t}$ denotes the spatial covariance at time t. The pixel color is then rendered through front-to-back alpha blending:

$$\mathbf{C}(p) = \sum_{i \in \mathcal{G}_p} \mathbf{q}_i \alpha_i(p) \prod_{j<i} (1 - \alpha_j(p)) \,. \tag{3}$$

where $\mathcal{G}_p$ denotes the depth-sorted Gaussians contributing to pixel p, $\mathbf{q}_i$ is the Gaussian color, and $\alpha_i(p)$ is its opacity contribution. Following 3DGS Kerbl et al. (2023), densification is triggered according to the accumulated view-space positional gradient magnitude G_i. Let view-space positional gradient $\mathbf{g}_i = \nabla_{\boldsymbol{\mu}_i^{2D}} L$, where $\boldsymbol{\mu}_i^{2D}$ is the projected 2D mean and L^k is the loss at iteration k. A Gaussian is densified when:

$$G_i = \frac{1}{K} \sum_{k=1}^{K} \|\nabla_{\boldsymbol{\mu}_i^{2D}} L^k\|_2 > \gamma_{\text{pos}}, \tag{4}$$

where K is the accumulation interval, $\|\cdot\|_2$ is the ℓ_2 norm and γ_{pos} is the densification threshold.

4 METHOD

4.1 OVERVIEW

We propose a framework with two key components to improve the densification efficiency of 4D Gaussian Splatting, achieving higher-quality dynamic scene reconstruction with fewer Gaussian primitives, as illustrated in Fig. 2. First, we introduce an error-guided Gaussian localization strategy to identify 4D Gaussians that require densification and prevent redundant Gaussian growth in well-reconstructed regions. Second, we propose a gradient consistency-aware density control strategy to facilitate the splitting of large Gaussians affected by gradient collision while suppressing unnecessary densification caused by highly consistent gradient directions. We follow TAD-GS Sandu et al. (2026) for temporal deformation modeling.

4.2 ERROR-GUIDED GAUSSIAN LOCALIZATION

Existing densification strategies accumulate view-space positional gradients regardless of whether the corresponding image regions have already been sufficiently reconstructed. Consequently, Gaussians located in well-reconstructed regions may continuously accumulate densification signals, leading to unnecessary Gaussian growth. To address this issue, we introduce an error-guided Gaussian localization mechanism that restricts gradient accumulation according to reconstruction errors. Given a rendered image I and its reference image I^*, we first normalize their intensities and then construct a pixel-wise error map $E(p)$ by combining photometric residuals with high-frequency structural discrepancies:

$$E(p) = \sum_{C} \left| \hat{I}_C(p) - \hat{I}^*_C(p) \right| + \lambda_L \left| \mathcal{L}(\hat{I}(p)) - \mathcal{L}(\hat{I}^*(p)) \right|, \quad \hat{I} = \frac{I}{\mathrm{mean}(I) + \epsilon}, \tag{5}$$

where C denotes the color channels, $\mathcal{L}(\cdot)$ is the Laplacian operator, λ_L balances the contribution of the structural term, $\mathrm{mean}(I)$ denotes the average pixel intensity of image I, and ϵ is a small constant for numerical stability. The Laplacian term improves sensitivity to under-reconstructed edges and local structures.

To focus densification on regions with larger reconstruction errors, we adopt an increasingly strict error selection strategy during training. At early iterations, a relatively loose threshold is used to retain more candidate regions, while the criterion gradually becomes stricter as the reconstruction quality improves. Specifically, at iteration k, we determine the error threshold τ_e^k as:

$$\tau_e^k = Q_{q_k}(E(p)), \qquad q_k = q_s + (q_e - q_s)\frac{k - k_s}{k_e - k_s}, \tag{6}$$

where $Q_q(\cdot)$ denotes the q-th quantile of the error map, q_s and q_e denote the initial and final quantile levels, respectively, and k_s and k_e denote the starting and ending iterations of densification. As training proceeds, the increasing quantile raises the error threshold, allowing the model to progressively focus on under-reconstructed regions. Pixels satisfying $E(p) > \tau_e^k$ are regarded as high-error pixels. We then aggregate the binary error mask $M(p)$ at the patch level and retain only those patches whose proportion of high-error pixels exceeds a predefined threshold ρ. This patch-level filtering suppresses isolated noisy pixels and yields more reliable error regions. During gradient accumulation for densification, each visible Gaussian $\mathcal{G}_i$ is projected onto the current image plane:

$$\mathbf{u}_i = \Pi_t \left(\boldsymbol{\mu}_i^{(t)} \right), \tag{7}$$

where $\mathbf{u}_i$ is the projected 2D position of Gaussian i, and $\Pi_t(\cdot)$ denotes the projection under the current camera. A binary error-guided weight is then assigned according to the error mask:

$$\omega_i^{(t)} = M(\mathbf{u}_i). \tag{8}$$

The weight gates its positional gradient during temporal gradient accumulation. Consequently, Gaussians projected onto well-reconstructed regions are excluded from densification, while those projected onto high-error regions are retained.

4.3 GRADIENT CONSISTENCY-AWARE DENSITY CONTROL

Although reconstruction errors indicate which regions contain Gaussians that require further densification, gradient magnitude alone does not fully reflect whether a Gaussian can be effectively refined. In dynamic scene reconstruction, a large Gaussian may cover multiple pixels whose positional sub-gradients point in different directions. Opposing sub-gradients can cancel each other during aggregation, resulting in a small positional gradient despite substantial local reconstruction errors, a phenomenon known as gradient collision Ye et al. (2024). Conversely, highly aligned sub-gradients may produce persistently large gradient magnitudes and trigger unnecessary densification. We therefore incorporate a gradient consistency-aware density control strategy. For Gaussian $\mathcal{G}_i$ at observation t, let $g_{i,p}^{(t)}$ denote the view-space positional sub-gradient contributed by pixel p. We compute the signed and absolute view-space positional gradients as:

$$\mathbf{g}_i^{(t)} = \sum_{p \in P_i} g_{i,p}^{(t)}, \qquad \tilde{\mathbf{g}}_i^{(t)} = \sum_{p \in P_i} \left| g_{i,p}^{(t)} \right|, \tag{9}$$

where P_i denotes the pixels influenced by $\mathcal{G}_i$, and $|\cdot|$ denotes the element-wise absolute value. We accumulate them over temporal observations using the error mask and define the gradient direction consistency c_i as:

$$c_i = \Psi\left(\frac{\sum_{t\in\mathcal{N}} \omega_i^{(t)} \|\mathbf{g}_i^{(t)}\|_2 + \epsilon}{\sum_{t\in\mathcal{N}} \omega_i^{(t)} \|\tilde{\mathbf{g}}_i^{(t)}\|_2 + \epsilon}\right), \tag{10}$$

where $\Psi(\cdot)$ denotes the clipping operation that restricts the input value to $[0, 1]$, $\mathcal{N}$ denotes the set of observations used for temporal aggregation. A larger c_i indicates higher gradient direction consistency across different timestamps, while a smaller c_i indicates stronger directional inconsistency. For Gaussians with highly consistent gradients, we define:

$$h_i = \Psi\left(\frac{c_i - \tau_h}{1 - \tau_h}\right), \qquad s_i^h = 1 + \lambda h_i^2, \tag{11}$$

where τ_h is the high-consistency threshold and λ controls the strength of the consistency-based modulation. Accordingly, $s_i^h > 1$ when $c_i > \tau_h$, while $s_i^h = 1$ otherwise. For low-consistency Gaussians, we define:

$$l_i = \Psi\left(\frac{\tau_l - c_i}{\tau_l}\right), \qquad s_i^l = 1 + 4\lambda l_i^2, \tag{12}$$

where τ_l denotes the low-consistency threshold. Similarly, $s_i^l > 1$ when $c_i < \tau_l$, and $s_i^l = 1$ otherwise. Following TAD-GS Sandu et al. (2026), let $\bar{G}_i$ denote the opacity-weighted accumulated gradient magnitude used for densification. Based on these factors, the magnitudes used for cloning and splitting are respectively formulated as:

$$G_i^{\text{clone}} = \frac{\bar{G}_i}{s_i^h}, \qquad G_i^{\text{split}} = \bar{G}_i \frac{s_i^l}{s_i^h}. \tag{13}$$

The resulting magnitudes are then compared with the temporally adaptive threshold to determine densification Sandu et al. (2026). 4D Gaussians with highly consistent gradients over time are suppressed during densification, avoiding unnecessary Gaussian growth. In contrast, Gaussians with conflicting gradients are promoted during splitting, alleviating the effect of gradient collision and improving their ability to represent complex motion. Combined with error-guided Gaussian localization, the proposed strategy allocates Gaussians more effectively to regions with insufficient representation capacity while suppressing redundant growth in well-reconstructed regions.

5 Experiment

5.1 Implementation Details

We conduct experiments on three benchmark multi-view dynamic scene datasets: Neural 3D Video (N3DV) Li et al. (2022), InterDigital Sabater et al. (2017), and VRU Basketball Wu et al. (2025). N3DV contains six indoor scenes captured by 17–21 synchronized cameras at a resolution of 2704×2028, mainly featuring complex motions localized in small regions. We use the central camera as the test view. InterDigital consists of five indoor scenes captured by a 4×4 camera rig at 2048×1088 resolution, containing larger-range motions as well as abrupt object appearances and disappearances. We use the camera in the second row and second column for evaluation. VRU Basketball contains two real-world basketball scenes captured by 34 cameras at 1920×1080 resolution, featuring rapid human and ball motions with complex non-rigid interactions. We use the central camera for evaluation. Following TAD-GS Sandu et al. (2026), all videos are downsampled to half of their original resolution for training and evaluation. We report PSNR and SSIM to evaluate overall rendering quality and LPIPS Zhang et al. (2018) to measure perceptual similarity.

5.2 Quantitative Comparisons

We compare ELGD-GS with several existing state-of-the-art (SOTA) dynamic scene reconstruction methods, including TAD-GS Sandu et al. (2026), 4D-Scaffold Cho et al. (2026), Swift4D Wu et al. (2025), Ex4DGS Lee et al. (2024), and STG Li et al. (2024). The quantitative results on the three datasets are summarized in Table 1 and Table 2. Compared with the current SOTA method, i.e., TAD-GS, ELGD-GS improves PSNR by 0.42 dB and 0.91 dB on the N3DV and InterDigital datasets, respectively. Meanwhile, the number of Gaussian primitives is reduced by 51.3% and 52.9%, while the storage cost is reduced by 58.4% and 69.6%, respectively.

Table 1: Quantitative comparisons on the N3DV and InterDigital datasets. We report PSNR(dB), SSIM, LPIPS, the number of Gaussian primitives, training time (minutes), storage cost, and rendering FPS. * denotes the number of anchors rather than Gaussian primitives. The best and second-best results are highlighted in **bold** and underlined, respectively.

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	Number↓	Time↓	Storage↓	FPS↑
N3DV	STG Li et al. (2024)	31.40	-	-	434K	120	**62MB**	93
	Ex4DGS Lee et al. (2024)	32.01	-	-	**268K**	66	115MB	91
	Swift4D Wu et al. (2025)	32.12	0.944	0.104	354K	**48**	128MB	125
	4D-Scaffold Cho et al. (2026)	32.03	0.945	0.102	775K*	189	200MB	109
	TAD-GS Sandu et al. (2026)	32.42	0.947	0.093	712K	65	255MB	146
	ELGD-GS (Ours)	**32.84**	**0.949**	**0.089**	347K	67	106MB	**168**
Inter Digital	STG Li et al. (2024)	33.45	-	0.060	-	-	-	-
	Ex4DGS Lee et al. (2024)	32.45	0.930	0.078	-	-	213MB	-
	Swift4D Wu et al. (2025)	32.12	0.930	0.061	919K	**51**	220MB	116
	4D-Scaffold Cho et al. (2026)	33.98	0.945	0.060	793K*	194	267MB	112
	TAD-GS Sandu et al. (2026)	34.14	0.945	0.059	1604K	68	575MB	132
	ELGD-GS (Ours)	**35.05**	**0.948**	**0.057**	**756K**	66	**175MB**	**174**

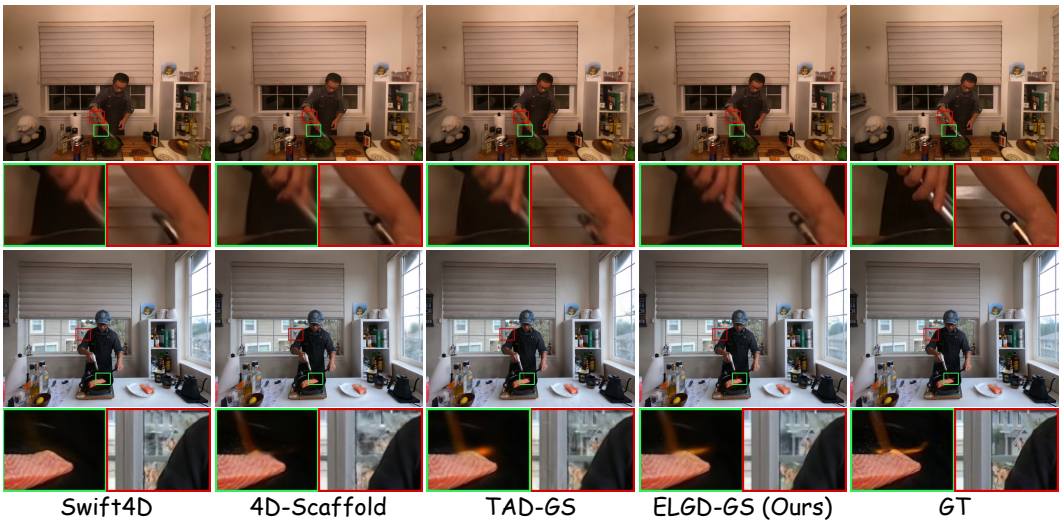

Figure 3: Qualitative comparisons on the *Cook_Spinach*, *Flame_Salmon* scenes from the N3DV dataset. ELGD-GS effectively captures complex motions and challenging background regions, yielding clearer dynamic reconstruction.

Furthermore, on the challenging VRU Basketball dataset, which contains more complex motion patterns and multiple objects, ELGD-GS achieves a 0.41 dB PSNR improvement over TAD-GS while using fewer Gaussian primitives. These results demonstrate that ELGD-GS consistently improves reconstruction quality while maintaining a substantially more compact scene representation.

Table 2: Quantitative comparisons on the VRU Basketball dataset. The best and second-best results are highlighted, respectively.

Method	PSNR↑	SSIM↑	Number↓	Time↓	Storage↓
Swift4D	25.91	0.903	1373K	66	**330MB**
4D-Scaffold	28.32	0.922	3813K*	348	1250MB
TAD-GS	28.28	0.921	1824K	67	654MB
ELGD-GS	**28.69**	**0.925**	**1096K**	**65**	388MB

5.3 Qualitative Comparisons

We qualitatively compare ELGD-GS with several state-of-the-art methods on the N3DV and InterDigital datasets in Figs. 3-4. ELGD-GS achieves better reconstruction quality in both static and dynamic regions, preserving finer details and producing cleaner object boundaries with fewer mo-

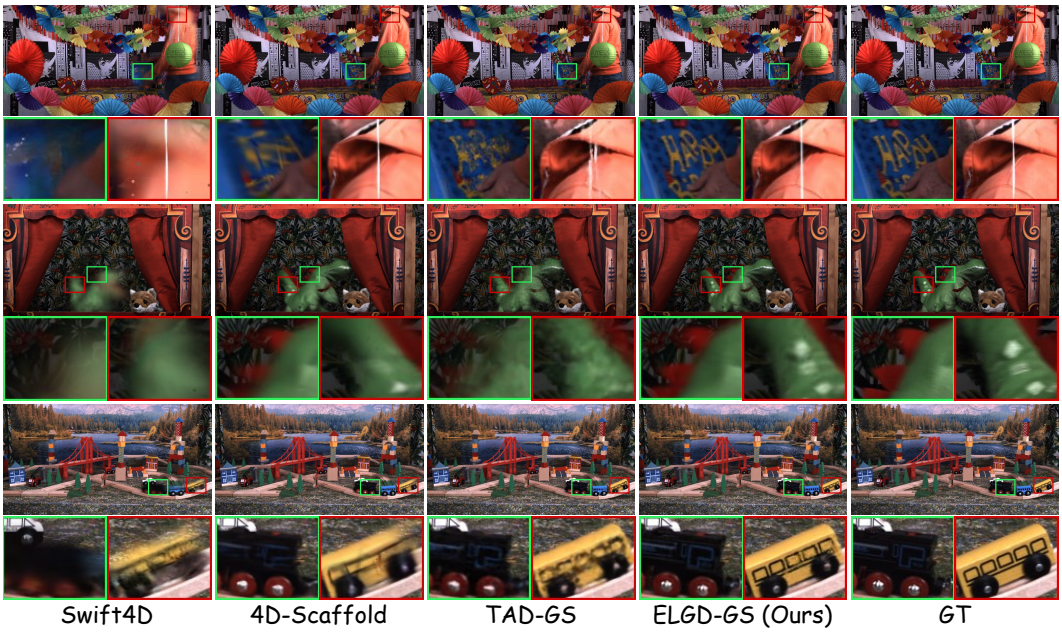

Figure 4: Qualitative comparisons on the *Birthday*, *Theater*, and *Train* scenes from the InterDigital dataset. ELGD-GS better reconstructs objects with large motions and reduces structural artifacts.

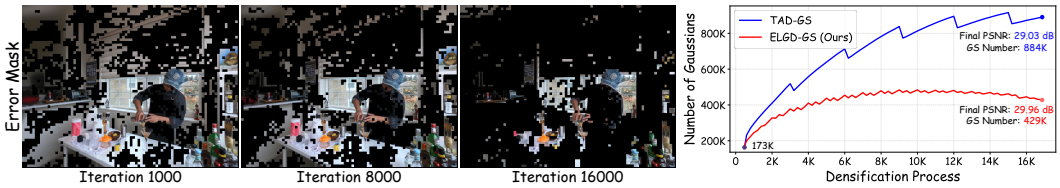

Figure 5: Visualization of the error masks and the evolution of the number of 4D Gaussians during densification on the *Coffee_Martini* scene of N3DV dataset. Our ELGD enables higher reconstruction quality with substantially fewer Gaussian primitives throughout training.

tion artifacts. These results demonstrate the effectiveness of our method in improving visual quality with a compact scene representation.

5.4 Densification Behavior Analysis

To analysis the densification behavior of ELGD model, we first visualize the evolution of the error masks during densification. As shown in Fig. 5, the error masks localize under-reconstructed regions and progressively concentrate on areas with complex motion and fine-grained structures, while filtering out well-reconstructed regions. Consequently, densification is focused on regions that require additional representation capacity, enabling higher reconstruction quality with substantially fewer Gaussian primitives throughout training. Additionally, we provide the visual results of rendered images and Gaussian ellipsoids between TAD-GS and ELGD-GS in Fig. 6. TAD-GS retains several oversized 4D Gaussians in dynamic regions, resulting in a noticeably blurred appearance. Our ELGD-GS evaluates the gradient consistency of each 4D Gaussian across different timestamps to control the densification, enabling more effective splitting of oversized 4D Gaussians that struggle to represent complex motion and thereby improving reconstruction quality in dynamic regions.

5.5 Ablation Study

To evaluate the effectiveness of our proposed design, we conduct ablation studies on the N3DV dataset, with the corresponding results reported in Table 3. (1) **Overall modules.** We separately evaluate the error-guided Gaussian localization (EL) mechanism and gradient consistency-aware

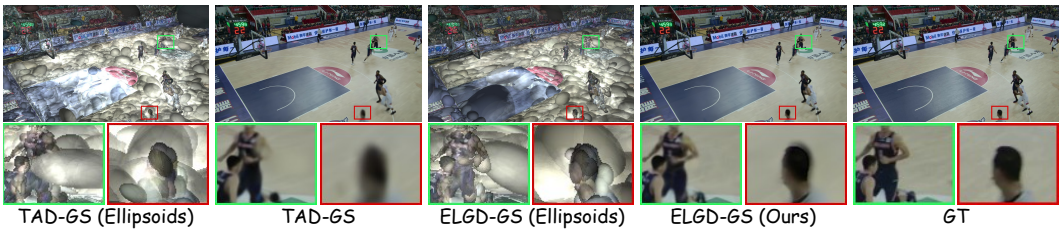

Figure 6: Visual comparison of rendered images and Gaussian ellipsoids between TAD-GS and ELGD-GS on *Basketball_gz* scene from the VRU Basketball dataset. Our method leverages gradient direction consistency to effectively split oversized Gaussians in dynamic regions, suppressing the blurred structures and improving reconstruction quality.

Table 3: Ablation study on the N3DV dataset, including (1) the overall EL and GD modules, (2) EL variants, and (3) GD variants. "Lap." denotes the Laplacian term, "Patch" patch-wise localization, "High-S." high-consistency suppression, and "Low-P." denotes low-consistency splitting promotion.

Variant	EL		GD		PSNR↑	SSIM↑	LPIPS↓	Number↓	Storage↓
	Lap.	Patch	High-S.	Low-P.					
Baseline	×	×	×	×	32.42	0.947	0.093	712K	255MB
EL only	✓	✓	×	×	32.58	0.947	0.090	384K	119MB
GD only	×	×	✓	✓	32.73	0.948	0.089	487K	168MB
w/o Lap.	×	✓	✓	✓	32.71	0.947	0.090	345K	105MB
Pixel-wise mask	✓	Pixel	✓	✓	32.54	0.946	0.090	364K	108MB
w/o High-S.	✓	✓	×	✓	32.81	0.948	0.089	387K	122MB
w/o Low-P.	✓	✓	✓	×	32.64	0.947	0.091	334K	102MB
Full model	✓	✓	✓	✓	32.84	0.949	0.088	347K	106MB

density control (GD) strategy. Introducing EL reduces Gaussian primitives by 46.1% while improving PSNR by 0.16 dB, demonstrating its effectiveness in preventing redundant Gaussian growth in well-reconstructed regions. Applying GD alone improves PSNR by 0.31 dB while reducing the number of Gaussians by 31.6%, indicating that gradient consistency-aware regulation improves reconstruction quality while suppressing unnecessary densification of Gaussians with highly consistent gradients. (2) **Variants of EL.** We further ablate the two designs within EL by removing the Laplacian-based structural term (w/o Lap.) and replacing the patch-wise error mask with a pixel-wise mask (Pixel-wise mask). Both variants degrade reconstruction quality, indicating that structural discrepancies help capture under-reconstructed details, while patch-wise localization provides more reliable error regions for densification. (3) **Variants of GD.** We further ablate the two designs within GD by disabling high-consistency suppression (w/o High-S.) and low-consistency splitting promotion (w/o Low-P.). Their corresponding results in Table 3 show that the former is important for preventing redundant densification, while the latter facilitates effective splitting under gradient collision and improves reconstruction quality.

6 CONCLUSION

In this paper, we present ELGD-GS, an error-guided localization and gradient consistency-aware density control framework for dynamic scene reconstruction. The error-guided Gaussian localization mechanism focuses on regions with insufficient representation capacity while preventing redundant densification in well-reconstructed regions. The gradient consistency-aware density control strategy regulates 4D Gaussian densification according to consistency aggregated over temporal observations, promoting the splitting of Gaussians affected by gradient collision and suppressing unnecessary growth under highly consistent gradients. Extensive experiments on multiple datasets demonstrate that ELGD-GS achieves improved reconstruction quality with lower storage cost, showing the effectiveness of our designs for compact and high-fidelity dynamic scene reconstruction.

REFERENCES

Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 321–335. Springer, 2024.

Woong Oh Cho, In Cho, Seoha Kim, Jeongmin Bae, Youngjung Uh, and Seon Joo Kim. 4d scaffold gaussian splatting with dynamic-aware anchor growing for efficient and high-fidelity dynamic scene reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3363–3371, 2026.

Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12479–12488. IEEE, 2023.

Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. Splatflow: Multi-view rectified flow model for 3d gaussian splatting synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21524–21536. IEEE, 2025.

Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(4):3119–3133, 2024.

Xu Jiawei, Fan Zexin, Yang Jian, and Xie Jin. Grid4d: 4d decomposed hash encoding for high-fidelity dynamic scene rendering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, volume 1, pp. 3, 2024.

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

Sangwoon Kwak, Joonsoo Kim, Jun Young Jeong, Won-Sik Cheong, Jihyong Oh, and Munchurl Kim. Modec-gs: Global-to-local motion decomposition and temporal interval adjustment for compact dynamic 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11338–11348. IEEE, 2025.

Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. *Advances in Neural Information Processing Systems*, 37:5384–5409, 2024.

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5511–5521. IEEE, 2022.

Yilong Li, Bo Pang, Yisong Chen, and Guoping Wang. Anchored 4d gaussian splatting for dynamic novel view synthesis. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8508–8520. IEEE, 2024.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6498–6508, 2021.

Zhanfeng Liao, Jiajun Zhang, Hanzhang Tu, Zhixi Wang, Yunqi Gao, Hongwen Zhang, and Yebin Liu. Sharptimegs: Sharp and stable dynamic gaussian splatting via lifespan modulation. *arXiv preprint arXiv:2602.02989*, 2026.

Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21136–21145. IEEE, 2024.

Yiren Lu, Yunlai Zhou, Disheng Liu, Tuo Liang, and Yu Yin. Bard-gs: Blur-aware reconstruction of dynamic scenes via gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16532–16542. IEEE, 2025.

Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8900–8910. IEEE, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26866–26875. IEEE, 2025.

Gurutva Patle, Nilay Girgaonkar, Nagabhushan Somraj, and Rajiv Soundararajan. Ad-gs: Alternating densification for sparse-input 3d gaussian splatting. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in gaussian splatting. In *European Conference on Computer Vision*, pp. 347–362. Springer, 2024.

Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, et al. Dataset and pipeline for multi-view light-field video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1743–1753. IEEE, 2017.

Vikram Sandu, Mayurdeep Pathak, and Rajiv Soundararajan. Temporally aware densification for dynamic 3d gaussian splatting. *arXiv preprint arXiv:2606.23212*, 2026.

Richard Shaw, Michal Nazarczuk, Jifei Song, Arthur Moreau, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Pérez-Pellitero. Swings: sliding windows for dynamic 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 37–54. Springer, 2024.

Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, et al. Steepest descent density control for compact 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26663–26672. IEEE, 2025a.

Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21750–21760. IEEE, 2025b.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320. IEEE, 2024.

Jiahao Wu, Rui Peng, Zhiyan Wang, Lu Xiao, Luyang Tang, Jinbo Yan, Kaiqiang Xiong, and Ronggang Wang. Swift4d: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. *arXiv preprint arXiv:2503.12307*, 2025.

Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 4d gaussian splatting with scale-aware residual field and adaptive optimization for real-time rendering of temporally complex dynamic scenes. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 7871–7880, 2024.

Jinbo Yan, Rui Peng, Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, and Ronggang Wang. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16520–16531. IEEE, 2025.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations*, volume 2024, pp. 9142–9159, 2024a.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20331–20341. IEEE, 2024b.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details in 3d gaussian splatting. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 1053–1061, 2024.

Yuheng Yuan, Qiuhong Shen, Xingyi Yang, and Xinchao Wang. 1000+ fps 4d gaussian splatting for dynamic scene rendering. *Advances in Neural Information Processing Systems*, 38:74665–74683, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Xuankai Zhang, Junjin Xiao, Shangwei Huang, Wei-shi Zheng, and Qing Zhang. Learning explicit continuous motion representation for dynamic gaussian splatting from monocular videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33291–33300, 2026.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *European conference on computer vision*, pp. 326–342. Springer, 2024.

Zheng Zhou, Yu-Jie Xiong, Jia-Chen Zhang, Chun-Ming Xia, Xihe Qiu, and Hongjian Zhan. Gradient-direction-aware density control for 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 21415–21437, 2026.

APPENDIX

The Appendix is organized as follows:

- **Appendix A: Implementation Details.** Model configurations, hyper-parameters, and 4DGS parameterization details, including the implementation of the proposed designs of ELGD framework.
- **Appendix B: Additional Experimental Results.** Additional qualitative and quantitative results on 4DGS reconstruction.
- **Appendix C: Discussions.** A detailed discussion of the rationale and trade-offs behind key design choices and current limitations and future directions.

A IMPLEMENTATION DETAILS

We use the full 300-frame sequences for the N3DV and InterDigital datasets, and 250-frame sequences for the VRU Basketball dataset. Our model is trained for 30K iterations on N3DV and 45K iterations on both InterDigital and VRU Basketball. Following the training protocol of TAD-GS Sandu et al. (2026), densification is performed every 200 iterations and terminates at iteration 17K for all datasets. The remaining densification schedule and optimization settings are kept consistent unless otherwise specified. Furthermore, for N3DV and InterDigital, we employ a 3D hash table to represent a subset of the 4D Gaussian parameters during training, further reducing the model storage cost.

We employ the Adam optimizer for Gaussian optimization. The learning rate of the Gaussian mean parameters is initialized to 2.6×10^{-4} and exponentially decayed to 2.6×10^{-6} during training. We use a batch size of 1 for all experiments, and retain the default 3DGS parameter settings for the remaining optimization parameters.

For the error-guided Gaussian localization module, the Laplacian weight λ_L is set to 0.2. The quantile level used for error-region selection is progressively increased from $q_s = 0.2$ to $q_e = 0.6$ during the densification stage. The error map is divided into patches of size 16×16, and a patch is retained only when the proportion of high-error pixels exceeds the threshold $\rho = 0.7$. This filtering suppresses isolated noisy responses and provides more reliable regions for error-guided densification.

For the gradient consistency-aware density control strategy, the consistency-aware modulation is activated after 8000 iterations to avoid unstable directional statistics during early optimization. We set the high-consistency threshold to $\tau_h = 0.8$ and the low-consistency threshold to $\tau_l = 0.2$. The modulation strength is controlled by $\lambda = 0.5$. The opacity-weighted accumulated positional gradient is used as the base densification score, and the proposed temporal consistency factors are applied to modulate the cloning and splitting signals. In particular, highly consistent gradients are suppressed to prevent unnecessary Gaussian growth, while large Gaussians affected by conflicting gradients receive enhanced splitting signals.

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 PER-SCENE QUANTITATIVE RESULTS

We provide detailed per-scene quantitative comparisons on the N3DV, InterDigital, and VRU Basketball datasets. In addition to reconstruction quality, we report the number of Gaussian primitives and storage cost to further evaluate representation efficiency.

N3DV. Table 4 reports the per-scene results on the N3DV dataset. Since dynamic regions usually occupy only a small portion of the image, global metrics may obscure reconstruction differences in moving areas. Therefore, we additionally report Masked PSNR (M-PSNR), computed using the dynamic-region masks released by TAD-GS Sandu et al. (2026), to specifically evaluate reconstruction quality in dynamic regions. ELGD-GS consistently reduces the number of Gaussian primitives

Table 4: Per-scene quantitative comparisons on the N3DV dataset. * denotes the number of anchors rather than Gaussian primitives.

Scene	Method	PSNR↑	M-PSNR↑	SSIM↑	LPIPS↓	Number↓	Storage↓
Coffee_Martini	Ex4DGS Lee et al. (2024)	28.79	21.15	-	-	340K	130MB
	Swift4D Wu et al. (2025)	29.29	23.54	0.917	0.115	428K	146MB
	4D-Scaffold Cho et al. (2026)	28.80	23.66	0.916	0.116	729K*	175MB
	TAD-GS Sandu et al. (2026)	28.85	23.78	0.920	0.109	885K	317MB
	ELGD-GS (Ours)	29.97	23.86	0.926	0.106	428K	119MB
Cook_Spinach	Ex4DGS Lee et al. (2024)	33.22	23.70	-	-	247K	118 MB
	Swift4D Wu et al. (2025)	32.89	23.09	0.950	0.098	267K	107MB
	4D-Scaffold Cho et al. (2026)	33.36	24.05	0.955	0.098	782K*	232MB
	TAD-GS Sandu et al. (2026)	33.51	24.31	0.954	0.089	662K	238MB
	ELGD-GS (Ours)	33.65	24.42	0.957	0.086	326K	102MB
Cut_Roasted_Beef	Ex4DGS Lee et al. (2024)	33.72	26.59	-	-	249K	123 MB
	Swift4D Wu et al. (2025)	33.58	26.62	0.953	0.093	265K	106MB
	4D-Scaffold Cho et al. (2026)	33.35	27.43	0.957	0.092	916K*	241MB
	TAD-GS Sandu et al. (2026)	33.89	27.68	0.959	0.088	631K	226MB
	ELGD-GS (Ours)	33.95	28.05	0.957	0.087	312K	101MB
Flame_Salmon	Ex4DGS Lee et al. (2024)	28.78	19.65	-	-	330K	128 MB
	Swift4D Wu et al. (2025)	29.62	21.03	0.922	0.115	650K	194MB
	4D-Scaffold Cho et al. (2026)	29.20	21.20	0.921	0.114	896K*	226MB
	TAD-GS Sandu et al. (2026)	29.73	21.14	0.925	0.104	944K	338MB
	ELGD-GS (Ours)	30.59	21.36	0.930	0.100	469K	127MB
Flame_Steak	Ex4DGS Lee et al. (2024)	33.90	23.01	-	-	226K	101 MB
	Swift4D Wu et al. (2025)	33.51	22.19	0.958	0.105	249K	103MB
	4D-Scaffold Cho et al. (2026)	33.43	23.28	0.962	0.103	734K*	188MB
	TAD-GS Sandu et al. (2026)	34.29	23.37	0.961	0.082	608K	218MB
	ELGD-GS (Ours)	34.39	23.31	0.962	0.079	304K	99MB
Sear_Steak	Ex4DGS Lee et al. (2024)	33.68	26.39	-	-	221K	94 MB
	Swift4D Wu et al. (2025)	33.83	26.24	0.960	0.094	263K	106MB
	4D-Scaffold Cho et al. (2026)	33.52	27.08	0.956	0.091	595K*	138MB
	TAD-GS Sandu et al. (2026)	34.36	27.23	0.963	0.081	547K	196MB
	ELGD-GS (Ours)	34.55	27.34	0.964	0.077	244K	89MB

and storage cost compared with TAD-GS across all scenes, while maintaining comparable or improved reconstruction quality in terms of both global and dynamic-region metrics.

InterDigital. Table 5 reports the per-scene quantitative results on the InterDigital dataset. As in the N3DV evaluation, we additionally report Masked PSNR (M-PSNR), computed using the dynamic-region masks released by TAD-GS Sandu et al. (2026), to specifically evaluate reconstruction quality in dynamic regions. Compared with TAD-GS, ELGD-GS reduces the average number of Gaussian primitives from 1.604M to 0.756M, corresponding to a reduction of approximately 52.8%, while decreasing the average storage cost from 575.2 MB to 175.2 MB, a reduction of approximately 69.5%. Meanwhile, ELGD-GS improves the average PSNR from 34.18 dB to 35.05 dB and the average M-PSNR from 28.64 dB to 29.69 dB, corresponding to improvements of 0.86 dB and 1.05 dB, respectively. These results demonstrate that ELGD-GS achieves a more compact Gaussian representation while improving both overall and dynamic-region reconstruction quality.

VRU Basketball. Table 6 reports the per-scene quantitative results on the VRU Basketball dataset. Compared with TAD-GS, ELGD-GS reduces the average number of Gaussian primitives from 1.824M to 1.096M, corresponding to a reduction of approximately 39.9%, while decreasing the average storage cost from 654 MB to 387.5 MB, a reduction of approximately 40.7%. Meanwhile, ELGD-GS improves the average PSNR from 28.29 dB to 28.69 dB and the average SSIM from

Table 5: Per-scene quantitative comparisons on the InterDigital dataset. * denotes the number of anchors rather than Gaussian primitives.

Scene	Method	PSNR↑	M-PSNR↑	SSIM↑	LPIPS↓	Number↓	Storage↓
Birthday	Ex4DGS Lee et al. (2024)	29.56	24.54	0.924	-	-	-
	Swift4D Wu et al. (2025)	31.07	21.87	0.930	0.031	1559K	375MB
	4D-Scaffold Cho et al. (2026)	32.43	25.89	0.960	0.026	1434K*	485MB
	TAD-GS Sandu et al. (2026)	32.05	26.99	0.957	0.025	2131K	764MB
	ELGD-GS (Ours)	32.78	27.20	0.960	0.025	941K	206MB
Painter	Ex4DGS Lee et al. (2024)	35.86	28.03	0.956	-	-	-
	Swift4D Wu et al. (2025)	36.32	23.80	0.950	0.048	399K	96MB
	4D-Scaffold Cho et al. (2026)	36.71	29.74	0.957	0.049	727K*	245MB
	TAD-GS Sandu et al. (2026)	37.93	32.72	0.958	0.047	874K	313MB
	ELGD-GS (Ours)	38.26	33.60	0.959	0.046	389K	114MB
Remy	Ex4DGS Lee et al. (2024)	35.57	29.04	0.923	-	-	-
	Swift4D Wu et al. (2025)	34.31	24.18	0.922	0.101	401K	96MB
	4D-Scaffold Cho et al. (2026)	35.98	27.84	0.926	0.110	476K*	160MB
	TAD-GS Sandu et al. (2026)	36.71	28.84	0.932	0.109	1326K	476MB
	ELGD-GS (Ours)	37.24	29.62	0.935	0.104	832K	188MB
Theater	Ex4DGS Lee et al. (2024)	30.88	25.06	0.889	-	-	-
	Swift4D Wu et al. (2025)	27.83	20.80	0.887	0.097	1096K	263MB
	4D-Scaffold Cho et al. (2026)	31.33	25.31	0.913	0.099	717K*	241MB
	TAD-GS Sandu et al. (2026)	31.39	26.79	0.913	0.098	1833K	657MB
	ELGD-GS (Ours)	32.33	27.27	0.920	0.096	822K	186MB
Train	Ex4DGS Lee et al. (2024)	30.35	24.09	0.958	-	-	-
	Swift4D Wu et al. (2025)	30.97	20.53	0.959	0.026	1140K	274MB
	4D-Scaffold Cho et al. (2026)	33.44	25.44	0.971	0.016	614K*	206MB
	TAD-GS Sandu et al. (2026)	32.83	27.88	0.964	0.018	1856K	666MB
	ELGD-GS (Ours)	34.62	30.78	0.967	0.014	798K	182MB

Table 6: Per-scene quantitative comparisons on the VRU Basketball dataset. * denotes the number of anchors rather than Gaussian primitives.

Scene	Method	PSNR↑	SSIM↑	LPIPS↓	Number↓	Storage↓
VRU_dg4	Swift4D Wu et al. (2025)	25.60	0.882	0.093	1359K	327MB
	4D-Scaffold Cho et al. (2026)	28.16	0.905	0.096	3642K*	1.2G
	TAD-GS Sandu et al. (2026)	28.13	0.904	0.092	1799K	645MB
	ELGD-GS (Ours)	28.23	0.906	0.092	1055K	374MB
VRU_gz	Swift4D Wu et al. (2025)	26.23	0.923	0.087	1386K	333MB
	4D-Scaffold Cho et al. (2026)	28.48	0.938	0.097	3983K*	1.3G
	TAD-GS Sandu et al. (2026)	28.44	0.938	0.096	1849K	663MB
	ELGD-GS (Ours)	29.15	0.945	0.088	1136K	401MB

0.921 to 0.926, while reducing the average LPIPS from 0.094 to 0.090. These results show that ELGD-GS achieves a substantially more compact representation while maintaining or improving reconstruction quality on the VRU Basketball dataset.

B.2 More Qualitative Comparisons

As shown in Fig. 7 and Fig. 8, we provide additional qualitative comparisons on the N3DV and VRU Basketball datasets. These results further demonstrate that our method can capture large motions more accurately, leading to higher-quality reconstruction of dynamic regions and more faithful dynamic scene reconstruction.



Figure 7: Qualitative comparisons on the *Cut_Roasted_Beef* scenes from the N3DV dataset.

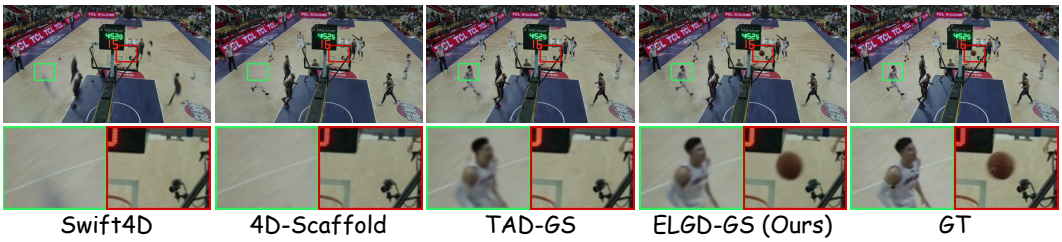

Figure 8: Qualitative comparisons on the *VRU_gz* scenes from the VRU Basketball dataset.

B.3 Hyperparameter Analysis

Table 7: Sensitivity analysis of the key hyperparameters of ELGD-GS on the N3DV dataset.

Setting	τ_l	τ_h	λ_L	PSNR↑	SSIM↑	Number↓
Default	0.2	0.8	0.2	**32.84**	**0.949**	**347K**
τ_l	0.3	0.8	0.2	32.77	0.948	341K
	0.4	0.8	0.2	32.54	0.946	336K
τ_h	0.2	0.7	0.2	32.65	0.947	361K
	0.2	0.9	0.2	32.79	0.948	346K
λ_L	0.2	0.8	0.1	32.80	0.949	377K
	0.2	0.8	0.4	32.66	0.945	384K

Furthermore, we analyze the sensitivity of ELGD-GS to three key hyperparameters, τ_l, τ_h, and λ_L, which control the low-consistency threshold, the high-consistency threshold, and the contribution of high-frequency structural discrepancies, respectively. As shown in Table 7, we evaluate multiple configurations around the default setting.

C Discussions

ELGD-GS introduce two primary designs, i.e., error-guided localization and gradient consistency-aware density control, to suppress redundant Gaussian growth and focus densification on under-reconstructed regions. These two designs enable a more compact representation while significantly improving reconstruction quality. Through extensive experiments, we identify one limitation of our framework. Several hyperparameters are manually specified rather than learned in a data-driven manner, and their settings can affect reconstruction performance.