STAB4D: 4D GAUSSIAN SPLATTING FOR DYNAMIC VIDEO STABILIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstruction-based video stabilization queries camera views that are close to, but not identical to, the observed shaky trajectory, thus creating a mismatch between observed-view optimization and stabilized-view deployment. For example, misplaced primitives can reproduce raw frames yet become visible as floaters after trajectory smoothing in Gaussian reconstruction. To address this issue, we introduce Stab4D, a local 4D Gaussian framework that shares geometry over short temporal windows while accommodating dynamic content with evidence-guided temporal lifespans. A key component of Stab4D is *Contribution-Guided Depth (CDepth)*, which uses renderer-native alpha-compositing contribution to route monocular depth evidence to rendering-relevant Gaussian primitives, with detached soft weighting and fixed-topology refinement after densification. Besides, a geometry-certified stable-view tube transfers reliable neighboring RGB-D evidence to the camera neighborhood between raw and stabilized trajectories, while unsupported exposure is conservatively suppressed. We also construct a seven-scene controlled dynamic benchmark with paired shaky/stable RGB, shared oracle geometry, and a common stable target trajectory. Under this protocol, Stab4Dachieves overall stronger stable-view reconstruction quality than competing reconstruction baselines while maintaining comparable raw-view fidelity.

1 INTRODUCTION

Video stabilization suppresses undesirable camera shake while preserving scene content and intentional motion. Image-space methods based on global transforms, meshes, flow, or learned warps are effective under moderate motion but become fragile under strong parallax, dynamic foregrounds, and disocclusion because the stabilized frame corresponds to a camera view that may not be directly observed (Matsushita et al., 2006; Liu et al., 2023; Roberto e Souza et al., 2022; Wang et al., 2023b). Explicit 3D geometry alleviates this limitation: Deep3D (Lee et al., 2021), RStab (Peng et al., 2024), GaVS (You et al., 2025), and StabiGS (Ben Mabrouk et al., 2026) all synthesize stabilized views from a geometric representation.

Unlike conventional reconstruction, however, video stabilization does not deploy the learned representation at exactly the cameras used for supervision. Trajectory smoothing intentionally changes the camera pose, so the stabilized frame at time t is effectively a nearby novel-view query, $T_t^s \neq T_t^r$. This creates a structured deployment shift from the observed shaky trajectory to a narrow neighborhood around it. Geometry that is photometrically harmless from the raw camera can therefore become directly visible after a small lateral or rotational displacement. Consequently, high raw-view reconstruction fidelity alone does not guarantee reliable stabilized-view rendering.

We argue that a *local 4D Gaussian field* is a natural representation for this problem. Per-frame 3D reconstruction discards cross-frame geometric support, whereas a globally persistent scene forces dynamic foregrounds, occlusion, and disocclusion into one static geometry. A short-horizon 4D field instead shares persistent structure while retaining time-dependent geometry, visibility, and appearance for transient content.

Efficient 4DGS models often obtain compact dynamics through shared spatiotemporal features and lightweight deformation decoders, as exemplified by 4D-GS (Wu et al., 2024). This is an effective prior for generic dynamic reconstruction, but a shared field can couple nearby primitives and smooth small independent motions, as illustrated in Fig. 1. Stabilization has a different bias: because

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.

its target camera is typically only a nearby view, preserving local sharpness and frame-to-frame appearance is as important as view transfer. We therefore retain temporal compactness with B-spline trajectories but optimize their control variables per primitive, preserving local motion freedom without an additional shared spatial deformation field.

Monocular local 4DGS still inherits sparse-view depth ambiguity: different 3D configurations can explain the same raw image, and misplaced primitives may remain hidden until a nearby stabilized view exposes them (Xiong et al., 2025). Our focus is to assign depth evidence to the rendering-relevant primitives that cause these artifacts.



Generic Wu-style 4DGS Ours

Figure 1: **Task-dependent deformation bias.** A representative efficient shared-field 4DGS (4D-GS (Wu et al., 2024)) can smooth small independent motions, whereas our primitive-wise spline deformation better preserves local motion boundaries.

We therefore introduce *Contribution-Guided Depth (CDepth)*, which uses alpha-compositing contribution to associate each Gaussian with a raw-view depth target and a detached supervision weight. CDepth starts after densification, so geometry is refined on fixed topology. Complementarily, a *geometry-certified stable-view tube* reprojects reliable neighboring RGB-D evidence to intermediate raw-to-stable poses, supervising regions supported elsewhere in the local window. Evidence-guided lifespans determine when each primitive is temporally supported.

Evaluation is also confounded by geometry frontends and trajectory smoothers. We therefore build a seven-scene dynamic Blender benchmark with paired shaky/stable views and shared geometry and target trajectories, isolating representation, temporal sharing, and nearby-view synthesis.

Our contributions are threefold:

- We formulate dynamic video stabilization as *nearby-view synthesis from a local 4D Gaussian representation*. Short-horizon temporal sharing provides geometric support across neighboring observations without imposing a globally static scene, while evidence-guided lifespans assign persistent and transient content different temporal support.
- We introduce a stabilization-oriented geometric refinement strategy for the raw-to-stabilized camera shift. Contribution-Guided Depth (CDepth) routes monocular depth evidence to rendering-relevant Gaussian primitives using detached alpha-compositing responsibility, while a geometry-certified stable-view tube transfers reliable neighboring RGB-D evidence to nearby deployment views. For regions lacking reliable cross-frame support, unsupported-exposure suppression reduces newly revealed Gaussian contributions rather than enforcing hallucinated targets.
- We build a controlled seven-scene dynamic benchmark with paired shaky/stable views and shared exact geometry, isolating representation and nearby-view synthesis from frontend and smoothing differences.

2 Related Work

Video stabilization and reconstruction-based rendering. Classical and learning-based stabilizers estimate motion, smooth trajectories, and warp or synthesize output frames (Grundmann et al., 2011; Liu et al., 2011; Goldstein & Fattal, 2012; Liu et al., 2013; Wang et al., 2013; Liu et al., 2014; 2016; Yu & Ramamoorthi, 2018; Wang et al., 2018; Zhao & Ling, 2020; Yu & Ramamoorthi, 2019; Liu et al., 2021; Shi et al., 2022; Wang et al., 2023a; Zhang et al., 2023; Zhao et al., 2023; Xu et al., 2022; Liu et al., 2026). Depth-aware methods address parallax and disocclusion (Liu et al., 2012); Deep3D and RStab use monocular geometry, GaVS and StabiGS use Gaussian representations, and VS3R (Zhu et al., 2026) combines reconstruction with generative priors. Unlike these methods,

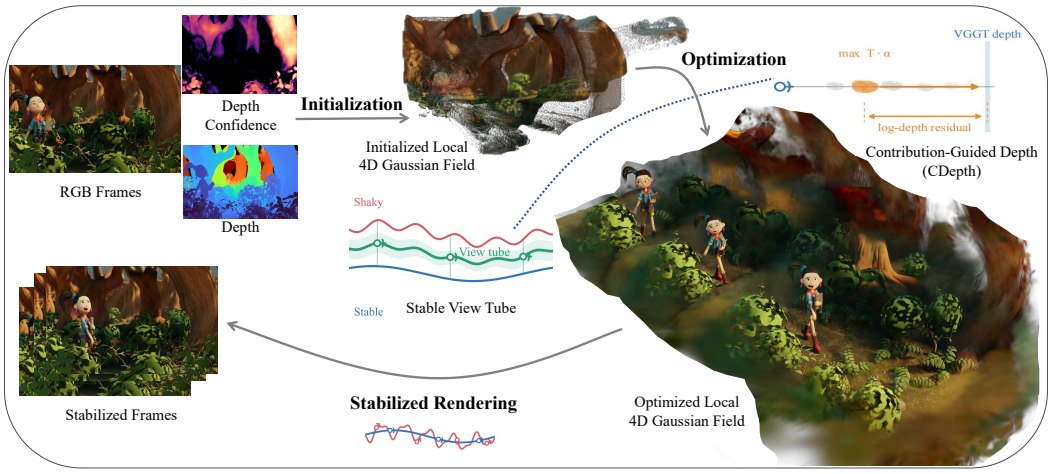


Figure 2: **Overview of Stab4D.** VGGT/VGGT4D provides RGB-D, confidence, and raw poses to initialize a local 4D Gaussian field with B-spline deformation and evidence-guided temporal lifespans. After topology search, CDepth refines rendering-relevant geometry, while the stable-view tube transfers certified neighboring RGB-D evidence to nearby deployment views. The optimized field is finally rendered along the stabilized trajectory.

we explicitly constrain the narrow raw-to-stabilized camera neighborhood using a local dynamic Gaussian field.

Dynamic Gaussian representations. 3DGS (Kerbl et al., 2023) enables differentiable real-time splatting, while dynamic extensions model deformation or temporal support (Yang et al., 2024; Wu et al., 2024; Li et al., 2024b; Wang et al., 2025b; Liao et al., 2026; Wang et al., 2026; Lee et al., 2026). These methods primarily target dynamic reconstruction or novel-view synthesis (Mildenhall et al., 2021; Pumarola et al., 2021; Park et al., 2021). We use compact B-spline motion and temporal envelopes as backbone components (Yoon et al., 2025), but retain primitive-wise motion freedom and tailor geometric supervision to the stabilization-specific raw-to-stable shift.

Geometry regularization for Gaussian splatting. Sparse-view Gaussian reconstruction is under-constrained along viewing rays. DNGaussian (Li et al., 2024a), 2DGS (Huang et al., 2024), and SparseGS (Xiong et al., 2025) use depth or surface priors to reduce this ambiguity and floater artifacts. CDepth does not redefine this failure mode; it uses native compositing responsibility to route depth evidence to individual primitives in a dynamic stabilization setting.

3 Method

3.1 Problem Formulation and Local 4D Representation

Given a monocular shaky video $\{I_t\}_{t=1}^{T}$, a geometry frontend provides intrinsics K_t, raw poses T_t^r, depth D_t, and confidence C_t; for real videos we use VGGT/VGGT4D (Wang et al., 2025a; Hu et al., 2026). We optimize one local field over each overlapping window $\mathcal{W}_k$ and obtain a stabilized trajectory $\{T_t^s\}$ with a fixed smoothing operator. The output frame is

$$\hat{I}_t^s = \mathcal{R}(\mathcal{G}(t), K_t, T_t^s), \tag{1}$$

where $\mathcal{R}$ is differentiable Gaussian rasterization; the complete pipeline is illustrated in Fig. 2.

Each canonical Gaussian is $G_i^c = (\mu_i^c, q_i^c, s_i^c, c_i^c, \alpha_i^c)$. A confidence-filtered center-frame point cloud initializes the canonical scaffold. Reliable RGB-D observations from neighboring frames are transformed into the same local reference coordinate system and used to extend geometric coverage beyond the center view. Keeping this aggregation inside a short temporal window avoids accumulating long-range pose drift and incompatible dynamic observations, while still providing substantially richer support than a frame-wise reconstruction.

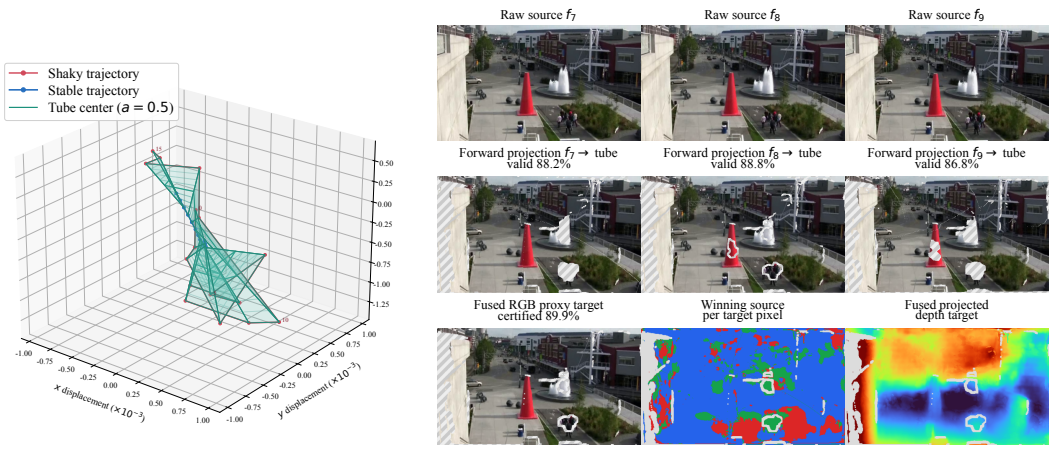


Figure 3: **Geometry-certified stable-view tube.** Left: raw and stabilized trajectories define a narrow interpolated camera tube. Right: neighboring raw RGB-D observations are projected to an intermediate tube view and fused only at certified pixels; the source map indicates which frame supports each target pixel.

Motion is parameterized by cubic B-spline controls,

$$\mu_i(t) = \mu_i^c + \sum_k B_k(t)\mathbf{c}_{ik}^{\mu}, \tag{2}$$

with analogous updates for enabled attributes. Each Gaussian has its own control variables, so the spline couples its trajectory across nearby timestamps without a shared spatial deformation decoder. This preserves local motion freedom while discouraging frame-specific deformation from absorbing camera shake. Temporal validity is handled separately by the lifespan model below.

3.2 Evidence-Guided Temporal Lifespan

To avoid both full-window persistence and nearly frame-wise representations, each Gaussian has continuous temporal support

$$s_i(\hat{t}) = \exp\left[-\exp(p_i^\tau)(\hat{t} - \mu_i^\tau)^2\right], \qquad \tilde{\alpha}_i(\hat{t}) = \alpha_i(\hat{t})s_i(\hat{t}). \tag{3}$$

Cross-frame visibility and depth consistency initialize and regularize this support: reliable agreement encourages persistence, reliable contradiction suppresses it, and occluded/out-of-view/low-confidence frames are treated as unknown rather than negative evidence. The temporal envelope is frozen during early topology search and later optimized with a weak anti-collapse constraint. Additional lifespan details and statistics are given in Appendix A.

3.3 Contribution-Guided Depth Regularization

A rendered pixel is compositional, so an image-space depth residual alone does not identify which Gaussian should move. For Gaussian i at raw frame t, we use its standard front-to-back contribution

$$w_{it}(p) = T_{it}(p)\alpha_{it}(p), \qquad p_{it}^* = \arg\max_p w_{it}(p), \qquad w_{it}^{\max} = \max_p w_{it}(p), \tag{4}$$

where $T_{it}(p)$ is the accumulated transmittance before Gaussian i. If $w_{it}^{\max} > 0$ and the selected pixel passes the valid depth/confidence mask, we define

$$\begin{aligned} d_{it}^* &= D_t(p_{it}^*), \\ \bar{r}_{it} &= \left[\left|\log\frac{z_{it}+\epsilon}{d_{it}^*+\epsilon}\right| - \log 1.03\right]_+, \\ \eta_{it} &= \mathrm{sg}\left(\frac{w_{it}^{\max}}{w_{it}^{\max}+0.01}\right), \end{aligned} \tag{5}$$

where z_{it} is the current camera-space Gaussian depth and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The primitive-level objective is

$$\mathcal{L}_{\mathrm{cdepth}} = \frac{\sum_i m_{it}\eta_{it}\rho(\bar{r}_{it})}{\sum_i m_{it}\eta_{it} + \epsilon}, \tag{6}$$

with Smooth-L1 ρ and validity indicator m_{it}. Routing is recomputed online but detached, preventing opacity reduction from escaping the penalty. Unlike rendered-depth supervision, which constrains only composited pixel depth, CDepth assigns depth evidence directly to rendering-responsible primitives. It is activated after the final topology update with $\lambda_{\mathrm{cdepth}} = 0.05$.

3.4 Geometry-Certified Stable-View Tube Supervision

CDepth refines geometry supported by observed raw views, whereas stabilization queries nearby cameras that can expose regions seen only in neighboring frames. We therefore sample a narrow tube between raw and stabilized poses,

$$T_t^{(\lambda)} = \mathrm{Interp}(T_t^r, T_t^s; \lambda), \qquad \lambda \in \Lambda \subset (0, 1]. \tag{7}$$

For each virtual tube view, reliable RGB-D pixels from nearby raw frames are unprojected to 3D and reprojected into $T_t^{(\lambda)}$. Confidence, positive-depth, image-boundary, visibility, and depth-consistency tests reject unreliable candidates; a z-buffer resolves collisions. This produces a partial proxy target $(\widetilde{I}, \widetilde{D}, M)$ and

$$\mathcal{L}_{\mathrm{tube}} = \lambda_I \frac{\sum_p M(p)\|\hat{I}(p) - \widetilde{I}(p)\|_1}{\sum_p M(p) + \epsilon} + \lambda_D \frac{\sum_p M(p)\rho(\hat{D}(p) - \widetilde{D}(p))}{\sum_p M(p) + \epsilon} + \lambda_\alpha \mathcal{L}_{\mathrm{cover}}. \tag{8}$$

Unsupported pixels are not forced to match hallucinated RGB-D; Fig. 3 illustrates the certified tube construction. For regions without reliable neighboring observations, we instead suppress newly exposed unsupported Gaussian contributions, as illustrated in Fig. 4. Let $e_i^v = [c_i^v - c_i^r]_+$ denote newly exposed contribution and g_i its cross-frame support:

$$\mathcal{L}_{\mathrm{expose}} = \frac{\sum_{v,i} e_i^v(1 - g_i)}{\sum_{v,i} e_i^v + \epsilon}. \tag{9}$$



w/o Expose Ours

Figure 4: **Unsupported exposure.** Newly revealed Gaussian contribution without reliable local support is conservatively suppressed.

3.5 Optimization and Rendering

The local field is trained with raw-view photometric reconstruction, temporal regularization, CDepth, tube supervision, and unsupported-exposure suppression. Topology updates are confined to an early structure-search stage; CDepth is activated after the final update and the remaining optimization uses fixed Gaussian topology. Each local field is then rendered along the stabilized path and adjacent windows are fused over their overlap. Complete schedules and weights are reported in Appendix F.

4 Controlled Dynamic Stabilization Benchmark

We construct seven 120-frame dynamic Blender sequences at 24 fps and 640×360, with paired shaky/stable RGB and exact camera/depth geometry at each timestamp. Synthetic shake is calibrated from 21 VGGT4D trajectory windows from 11 NUS videos (1,344 poses) using category-balanced second-order Gauss–Markov residual models. The benchmark uses the 35th-percentile operating point of the estimated real-motion distribution (0.348° rotational RMS and 0.158% depth-normalized translation); full calibration and scale normalization are in Appendix D.

Table 1: Quantitative comparison on NUS. We report crop ratio (C), distortion (D), and stability (S) for five subsets and the average. Higher is better.

Method	Regular			Quick Rotation			Zooming		
	C	D	S	C	D	S	C	D	S
Yu *et al.* (Yu & Ramamoorthi, 2020)	0.949	0.975	0.843	0.766	0.762	0.902	0.897	0.863	0.891
DIFRINT (Choi & Kweon, 2020)	1.000	0.932	0.817	1.000	0.881	0.865	1.000	0.853	0.875
DUT (Xu et al., 2022)	0.949	0.960	0.872	0.761	0.694	0.920	0.899	0.861	0.922
LightStab (Liu et al., 2026)	0.935	0.978	0.875	0.752	0.708	0.918	0.885	0.872	0.924
Deep3D (Lee et al., 2021)	0.792	0.992	0.832	0.589	0.910	0.909	0.643	0.978	0.898
RStab (Peng et al., 2024)	1.000	0.995	0.872	1.000	0.963	0.910	1.000	0.980	0.912
GaVS (You et al., 2025)	1.000	0.981	0.887	1.000	0.953	0.913	1.000	0.962	0.910
Stab4D	1.000	0.991	0.932	1.000	0.974	0.930	1.000	0.973	0.923

Method	Crowd			Parallax			Average		
	C	D	S	C	D	S	C	D	S
Yu *et al.* (Yu & Ramamoorthi, 2020)	0.908	0.919	0.865	0.921	0.922	0.889	0.888	0.888	0.878
DIFRINT (Choi & Kweon, 2020)	1.000	0.821	0.846	1.000	0.882	0.860	1.000	0.873	0.853
DUT (Xu et al., 2022)	0.899	0.862	0.904	0.925	0.877	0.913	0.887	0.851	0.906
LightStab (Liu et al., 2026)	0.853	0.854	0.911	0.914	0.880	0.916	0.868	0.858	0.909
Deep3D (Lee et al., 2021)	0.702	0.963	0.912	0.779	0.970	0.888	0.701	0.963	0.888
RStab (Peng et al., 2024)	1.000	0.976	0.912	1.000	0.972	0.884	1.000	0.977	0.898
GaVS (You et al., 2025)	1.000	0.976	0.917	1.000	0.927	0.876	1.000	0.960	0.901
Stab4D	1.000	0.982	0.919	1.000	0.974	0.898	1.000	0.979	0.920

Controlled geometry and trajectory protocol. Real videos lack exact paired pose/depth and stable-view ground truth, while existing stabilizers also use different geometry frontends and smoothing rules. To remove these confounds, every reconstruction-based baseline receives the same Blender $(K_t^*, T_t^{*,r}, D_t^{*,r})$ and is evaluated at the same stable target camera $T_t^{*,s}$, bypassing its native pose/depth estimation and trajectory smoothing in this controlled comparison. This protocol is therefore not intended as an end-to-end pose or depth benchmark; it is a controlled test of how different representations use the same geometric evidence over time and transfer it from observed to stabilized views.

5 Experiments

Setup. We evaluate on NUS and the controlled dynamic benchmark. On NUS, we compare the native stabilization pipelines of Yu *et al.* (Yu & Ramamoorthi, 2020), DIFRINT (Choi & Kweon, 2020), DUT (Xu et al., 2022), LightStab (Liu et al., 2026), Deep3D, RStab, and GaVS; the controlled benchmark uses Deep3D, RStab, and GaVS under the shared geometry/trajectory protocol. We use 16-frame local windows and optimize each window for 2,000 steps. CDepth is enabled after the final densification/pruning update with $\lambda_{\text{cdepth}} = 0.05$; tube supervision uses three neighboring source frames. Complete settings are listed in Appendix F.

5.1 Main Results

NUS. Table 1 reports quantitative results on all five NUS subsets, while Fig. 5 provides qualitative comparisons. The highlighted regions further show that competing methods can introduce ghosting, local geometric distortion, or erroneous color filling after stabilization, whereas Stab4Dpreserves more coherent scene structure. Stab4Dachieves the best average stability while preserving full-frame crop ratio and high distortion score. This combination indicates that the stability improvement is not obtained through aggressive cropping or excessive geometric deformation.

Controlled dynamic benchmark. Table 2 and Fig. 6 compare all reconstruction-based methods at identical Blender shaky and stable cameras. Stable-view quality therefore measures transfer from observed to deployment views without confounding pose/depth estimation or trajectory smoothing.

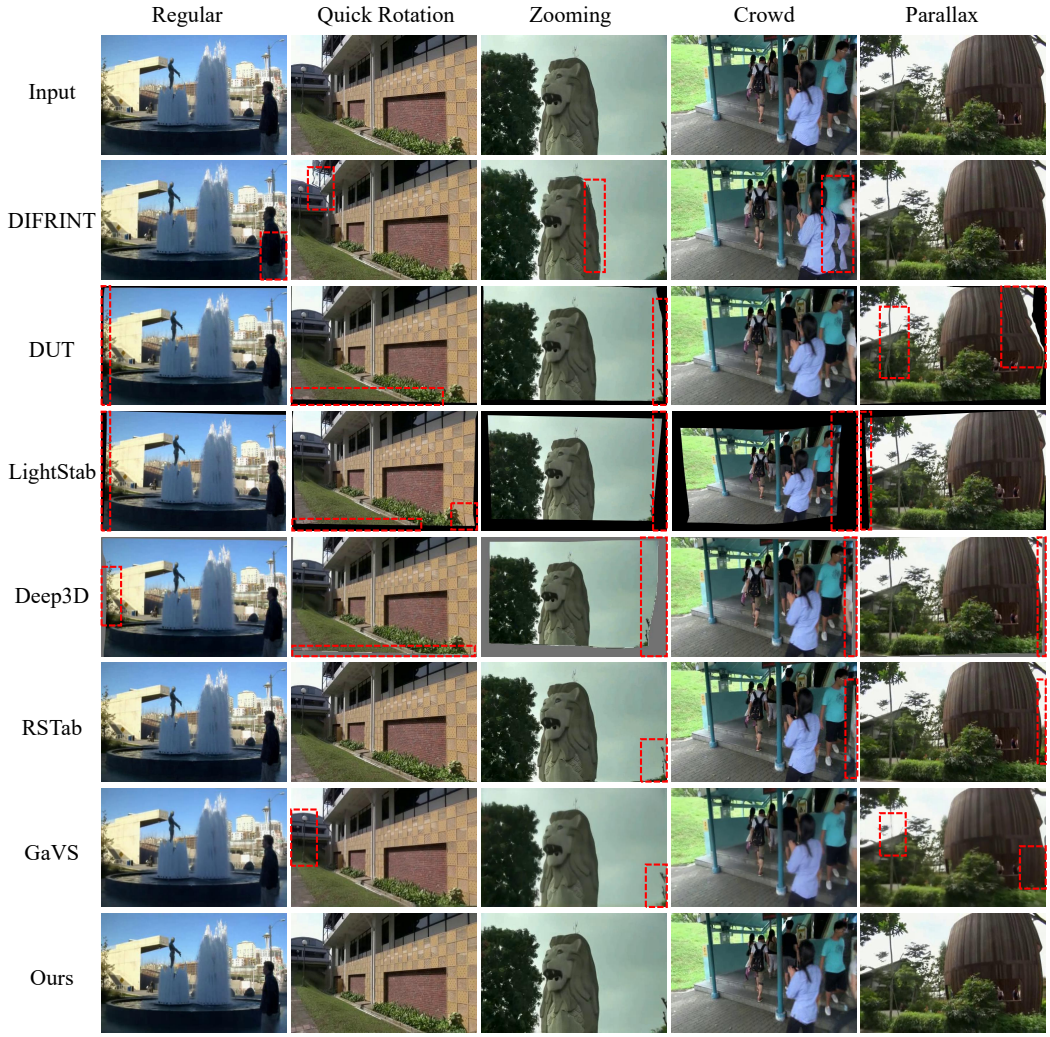


Figure 5: **Qualitative comparison on NUS.** Columns correspond to Regular, Quick Rotation, Zooming, Crowd, and Parallax; rows show input and competing stabilizers. Boxes highlight visible stabilization artifacts, such as ghosting, geometric distortion, and erroneous color filling.

Table 2: **Controlled benchmark with shared geometry and target trajectory.** Higher is better for PSNR, SSIM (Wang et al., 2004), S, and D; lower is better for LPIPS (Zhang et al., 2018).

Method	GT shaky view			GT stable view			$S\uparrow$	$D\uparrow$
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓		
Deep3D (Lee et al., 2021)	29.250	0.796	0.317	28.313	0.747	0.189	0.857	0.912
RStab (Peng et al., 2024)	38.485	0.965	0.014	30.385	0.781	0.098	0.858	0.965
GaVS (You et al., 2025)	28.527	0.873	0.139	25.066	0.723	0.220	0.856	0.929
Stab4D	38.765	0.989	0.023	32.696	0.854	0.170	0.876	0.978

A particularly informative trend is the different behavior at raw and stabilized views. RStab and Stab4Dobtain similar raw-view PSNR (38.485 vs. 38.765 dB), yet at the paired stable cameras Stab4Dimproves PSNR from 30.385 to 32.696 dB and also obtains higher SSIM. RStab obtains lower LPIPS, reflecting different behavior across complementary fidelity metrics. The raw/stable gap supports our central premise: fitting the observed trajectory is insufficient when hidden geometric errors become exposed by stabilization.

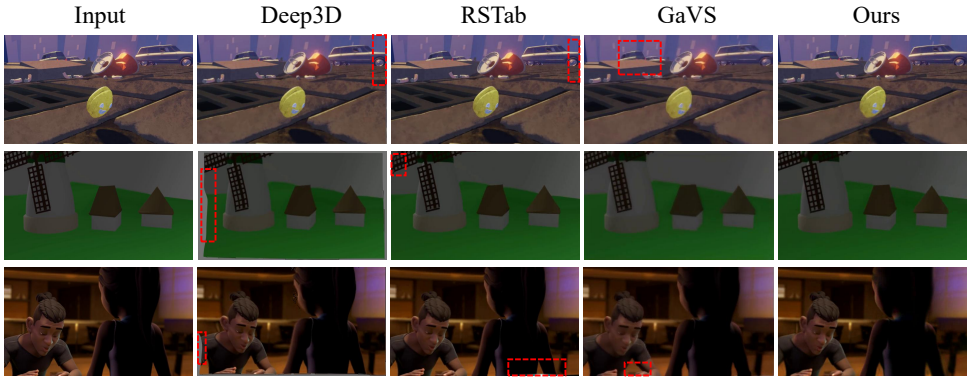


Figure 6: **Qualitative comparison on the controlled dynamic benchmark.** All reconstruction-based methods use the same Blender pose/depth inputs and stable target cameras. Boxes highlight visible stabilization artifacts, such as ghosting, geometric distortion, and erroneous color filling.

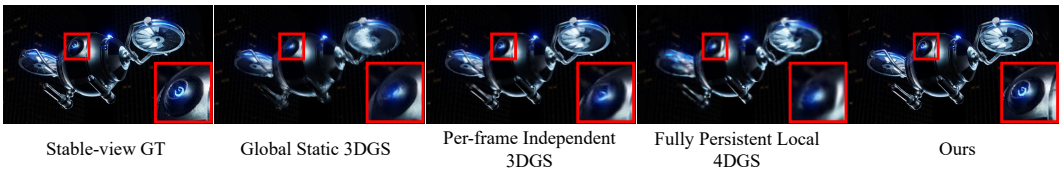


Figure 7: **Temporal-support ablation.** From left: stable-view GT, static 3DGS, frame-wise 3DGS, local 4DGS with full-window support, and our evidence-guided lifespan. Adaptive temporal support improves shifted-view reconstruction without over-sharing dynamic content.

5.2 Component Analysis

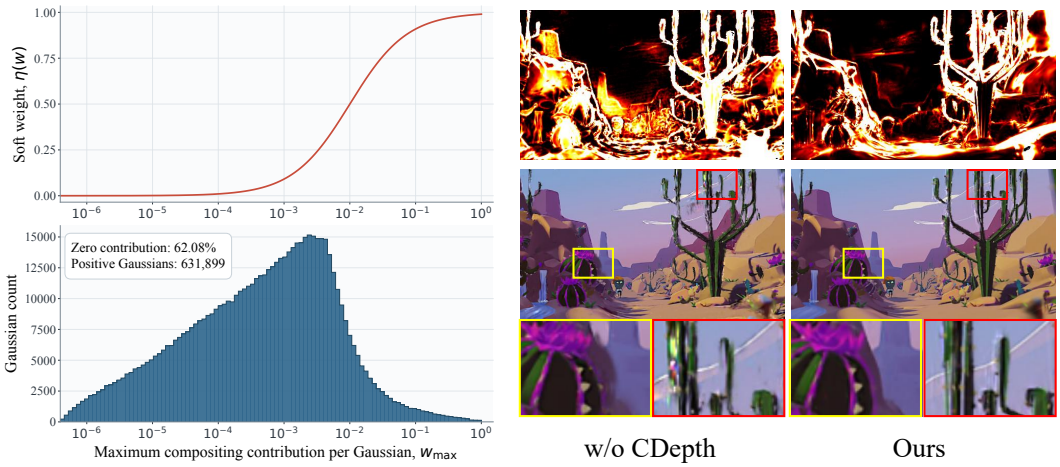
Temporal lifespan. Figure 7 compares several temporal-sharing strategies at the same stabilized target view. Frame-wise geometry provides insufficient reusable support after the camera moves, while forcing full-window persistence can over-share inconsistent dynamic observations. Evidence-guided lifespan provides an adaptive intermediate solution. Across the seven controlled scenes, pooled FWHM quartiles are $6.02/12.84/18.86$ frames; only 3.98% of primitives have support shorter than four frames, while 35.70% remain effectively persistent in a 16-frame window. Full per-scene statistics are provided in Appendix A.

CDepth. Figure 8 validates the contribution weighting and its geometric effect. Positive $w_{it}^{\max}$ spans several orders of magnitude, supporting continuous weighting rather than a hard cutoff. CDepth suppresses misplaced far-depth contribution while preserving raw-view fidelity; residuals remain concentrated near ambiguous depth boundaries. To separate primitive-level routing from generic depth regularization, Table 3 also replaces CDepth with a conventional rendered-depth loss under the same validity mask and loss scale; its definition is given in Appendix B.

Table 3 reveals a complementary progression across temporal support, observed-view geometry, and deployment-view supervision. Evidence-guided lifespan provides the largest initial gain, showing that dynamic monocular video benefits from sharing geometry without forcing full-window persistence. Generic rendered-depth supervision changes Stable PSNR only from 30.449 to 30.487 dB, whereas responsibility-routed CDepth reaches 31.925 dB with essentially unchanged Raw PSNR (38.765 vs. 38.766 dB). This separation indicates that the gain is not explained by additional raw-view fitting or depth supervision alone, but by assigning geometric correction to rendering-relevant primitives. Tube supervision then reaches 32.465 dB by constraining the nearby deployment views, while unsupported-exposure suppression gives the final 32.696 dB. Notably, these latter gains occur with essentially unchanged raw-view PSNR, indicating that they specifically improve transfer to the stabilized views rather than further fitting the observed cameras. Together, these results show that temporal sharing, primitive-level geometry refinement, and deployment-view supervision address complementary aspects of the raw-to-stable view-transfer problem.

Table 3: **Core-component ablation on the controlled benchmark.** Rendered depth is an alternative to CDepth; subsequent rows are cumulative.

Variant	Raw PSNR↑	Stable PSNR↑	Stable SSIM↑	Stable LPIPS↓
Local 4DGS, fixed full-window support	26.795	25.221	0.725	0.187
+ evidence-guided lifespan	38.765	30.449	0.782	0.175
Lifespan + rendered-depth loss	38.763	30.487	0.766	0.177
Lifespan + CDepth	**38.766**	31.925	0.847	0.172
+ stable-view tube	38.765	32.465	0.847	0.171
+ unsupported-exposure (Full Stab4D)	38.765	**32.696**	**0.854**	**0.170**



(a) CDepth weighting and contribution distribution.

(b) Controlled CDepth ablation on Lunte.

Figure 8: **CDepth analysis.** (a) Per-Gaussian maximum compositing contribution and soft weight $\eta(w) = w/(w + 0.01)$. (b) Lunte ablation showing reduced far-tail response and improved lateral novel-view rendering, with residual errors near ambiguous depth boundaries.

Configuration sensitivity. We use $W = 16$ and terminate topology updates in the early optimization stage. Longer windows increase Gaussian population without improving stable-view quality, while late densification repeatedly perturbs an already fitted field. Complete window-length, densification, frame-sampling, learning-rate, and budget sweeps are reported in Appendix E.

6 Conclusion

We presented Stab4D, a local 4D Gaussian framework for dynamic video stabilization that treats trajectory smoothing as a structured nearby-view deployment problem. The central observation is that a representation optimized on shaky input cameras is ultimately queried from nearby but systematically shifted stabilized views, so raw-view reconstruction quality alone does not fully characterize stabilization performance.

Stab4Daddresses this shift through complementary components within a local 4D representation. Evidence-guided temporal lifespans provide cross-frame geometric support while preserving dynamic content; CDepth routes monocular depth evidence to rendering-relevant Gaussian primitives; and the geometry-certified stable-view tube extends supervision toward the camera neighborhood reached after smoothing. Unsupported-exposure suppression further complements this process in newly revealed regions.

Experiments on real videos and the controlled dynamic benchmark show that these design choices improve stabilized-view reconstruction while preserving strong raw-view fidelity. The controlled analysis further demonstrates that similar raw-view reconstruction can lead to substantially different stable-view quality. Overall, our results highlight the importance of explicitly optimizing the representation for the nearby deployment views induced by trajectory smoothing.

AI USE STATEMENT

Generative AI tools were used only to assist with language editing, improving the clarity and organization of the manuscript, and literature retrieval and discovery.

ETHICS STATEMENT

This work introduces a video stabilization method and a synthetic dynamic benchmark built from Blender scenes. No new human-subject data are collected for the synthetic benchmark. Experiments on real videos use existing public research datasets under their respective terms.

REPRODUCIBILITY STATEMENT

The paper describes the local 4D representation, evidence-guided temporal lifespan, CDepth, stable-view tube supervision, controlled oracle-geometry benchmark protocol, and the default optimization stages. The supplementary material provides detailed thresholds, loss weights, densification schedules, per-scene statistics, and additional diagnostics.

REFERENCES

Souheib Ben Mabrouk, Jean-Emmanuel Deschaud, Eva Coupeté, Thomas Derbanne, and Nicolas Rahmouni. Stabigs: Video stabilization through rendering-aware trajectory optimization in 3dgs-reconstructed scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8481–8491, 2026.

Jinsoo Choi and In So Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM Transactions on Graphics (TOG)*, 39(1):1–9, 2020.

Amit Goldstein and Raanan Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics (TOG)*, 31(5):1–10, 2012.

Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR 2011*, pp. 225–232. IEEE, 2011.

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 414–424, 2026.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 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.

Lucas Yunkyu Lee, Soonho Kim, Youngwook Kim, Sangmin Kim, and Jaesik Park. Freetimegs++: Secrets of dynamic gaussian splatting and their principles. *arXiv preprint arXiv:2605.03337*, 2026.

Yao-Chih Lee, Kuan-Wei Tseng, Yu-Ta Chen, Chien-Cheng Chen, Chu-Song Chen, and Yi-Ping Hung. 3d video stabilization with depth estimation by cnn-based optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10621–10630, 2021.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785. IEEE, 2024a.

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, 2024b.

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.

Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):1–10, 2011.

Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3d video stabilization. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 631–639. 2023.

Shuaicheng Liu, Yinting Wang, Lu Yuan, Jiajun Bu, Ping Tan, and Jian Sun. Video stabilization with a depth camera. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 89–95. IEEE, 2012.

Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM transactions on graphics (TOG)*, 32(4):1–10, 2013.

Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4209–4216, 2014.

Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *European Conference on Computer Vision*, pp. 800–815. Springer, 2016.

Tao Liu, Kan Ren, Gang Wan, and Shibo Wen. No labels, no look-ahead: Unsupervised online video stabilization with classical priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6868–6877, 2026.

Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Hybrid neural fusion for full-frame video stabilization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2299–2308, 2021.

Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on pattern analysis and Machine Intelligence*, 28(7):1150–1163, 2006.

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.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.

Zhan Peng, Xinyi Ye, Weiyue Zhao, Tianqi Liu, Huiqiang Sun, Baopu Li, and Zhiguo Cao. 3d multi-frame fusion for video stabilization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7507–7516. IEEE, 2024.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10313–10322. IEEE, 2021.

Marcos Roberto e Souza, Helena de Almeida Maia, and Helio Pedrini. Survey on digital video stabilization: Concepts, methods, and challenges. *ACM Computing Surveys (CSUR)*, 55(3):1–37, 2022.

Zhenmei Shi, Fuhao Shi, Wei-Sheng Lai, Chia-Kai Liang, and Yingyu Liang. Deep online fused video stabilization. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1250–1258, 2022.

Jian Wang, Qiang Ling, and Peiyan Li. Robust video stabilization based on motion decomposition. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(5):1–24, 2023a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Miao Wang, Guo-Ye Yang, Jin-Kun Lin, Song-Hai Zhang, Ariel Shamir, Shao-Ping Lu, and Shi-Min Hu. Deep online video stabilization with multi-grid warping transformation learning. *IEEE Transactions on Image Processing*, 28(5):2283–2292, 2018.

Xuezhen Wang, Li Ma, Yulin Shen, Zeyu Wang, and Pedro V Sander. Retimegs: Continuous-time reconstruction of 4d gaussian splatting. *arXiv preprint arXiv:2603.13783*, 2026.

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.

Yiming Wang, Qian Huang, Chuanxu Jiang, Jiwen Liu, Mingzhou Shang, and Zhuang Miao. Video stabilization: A comprehensive survey. *Neurocomputing*, 516:205–230, 2023b.

Yu-Shuen Wang, Feng Liu, Pu-Sheng Hsu, and Tong-Yee Lee. Spatially and temporally optimized video stabilization. *IEEE transactions on visualization and computer graphics*, 19(8):1354–1361, 2013.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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.

Haolin Xiong, Sairisheek Muttukuru, Hanyuan Xiao, Rishi Upadhyay, Pradyumna Chari, Yajie Zhao, and Achuta Kadambi. Sparsegs: Sparse view synthesis using 3d gaussian splatting. In *2025 International Conference on 3D Vision (3DV)*, pp. 1032–1041. IEEE, 2025.

Yufei Xu, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Dut: Learning video stabilization by simply watching unstable videos. *IEEE Transactions on Image Processing*, 31:4306–4320, 2022.

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, 2024.

Jihwan Yoon, Sangbeom Han, Jaeseok Oh, and Minsik Lee. Splinegs: learning smooth trajectories in gaussian splatting for dynamic scene reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 42886–42903, 2025.

Zinuo You, Stamatios Georgoulis, Anpei Chen, Siyu Tang, and Dengxin Dai. Gavs: 3d-grounded video stabilization via temporally-consistent local reconstruction and rendering. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2025.

Jiyang Yu and Ravi Ramamoorthi. Selfie video stabilization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 551–566, 2018.

Jiyang Yu and Ravi Ramamoorthi. Robust video stabilization by optimization in cnn weight space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3800–3808, 2019.

Jiyang Yu and Ravi Ramamoorthi. Learning video stabilization using optical flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8159–8167, 2020.

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.

Zhuofan Zhang, Zhen Liu, Ping Tan, Bing Zeng, and Shuaicheng Liu. Minimum latency deep online video stabilization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 23030–23039, 2023.

Minda Zhao and Qiang Ling. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE Transactions on Image Processing*, 29:3582–3595, 2020.

Weiyue Zhao, Xin Li, Zhan Peng, Xianrui Luo, Xinyi Ye, Hao Lu, and Zhiguo Cao. Fast full-frame video stabilization with iterative optimization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 23534–23544, 2023.

Muhua Zhu, Xinhao Jin, Xinping Wang, Yu Zhang, Yifei Xue, Tie Ji, and Yizhen Lao. Vs3r: Robust full-frame video stabilization via deep 3d reconstruction. *arXiv preprint arXiv:2603.05851*, 2026.

A Additional Temporal-Lifespan Analysis

The local 4D representation uses evidence-guided temporal lifespan allocation to avoid two undesirable extremes: forcing every Gaussian to remain active throughout the entire local window, and allowing the representation to collapse toward nearly independent frame-wise Gaussian snapshots. We use the following effective-support statistic

$$N_i^{\text{eff}} = \sum_{t \in \mathcal{W}_k} s_i(t). \tag{10}$$

This appendix provides additional statistics and qualitative examples. In particular, we report the full distribution of N_i^{eff}, temporal FWHM statistics, and representative short-, medium-, and long-lived primitives. A healthy solution should exhibit a heterogeneous lifespan distribution rather than concentrating all Gaussians near either one frame or the full local window.

Table 4 summarizes the learned lifespan statistics across the seven controlled scenes. FWHM is computed from the continuous temporal envelope and is not clipped to the 16-frame observation window. The analyzed 16-frame windows contain 4.05M Gaussians, with pooled FWHM quartiles of $6.02/12.84/18.86$ frames. Only 3.98% have FWHM below four frames, while 35.70% remain effectively persistent over the local window.

B Additional CDepth Analysis

CDepth uses the native renderer contribution to associate individual Gaussian primitives with observed depth targets. Unlike a hard visibility threshold, the soft weighting in Eq. equation 5 retains every primitive with a positive compositing contribution while smoothly reducing weak contributors. For the rendered-depth control in Table 3, we instead supervise the composited expected depth

$$\begin{aligned} \hat{D}_t(p) &= \frac{\sum_i w_{it}(p) z_{it}}{\sum_i w_{it}(p) + \epsilon}, \\ \mathcal{L}_{\text{rdepth}} &= \frac{\sum_p M_t(p) \rho\Big(\Big[\Big| \log \frac{\hat{D}_t(p)+\epsilon}{D_t(p)+\epsilon} \Big| - \log 1.03 \Big]_+ \Big)}{\sum_p M_t(p) + \epsilon}. \end{aligned} \tag{11}$$

It uses the same valid mask, robust residual, and loss weight as CDepth, but does not assign a residual to individual rendering-responsible primitives.

For additional analysis, we provide the diagnostic definition used for the controlled lateral-view comparison in Fig. 8. This isolates geometric correction from changes caused by Gaussian population, densification, or other regularization terms. We additionally provide controlled lateral-view comparisons in which all training settings are held fixed except for the CDepth objective. These experiments distinguish geometric correction from changes caused by Gaussian population, densification, or other regularization terms.

For the lateral diagnostic used in Fig. 8, the RGB comparison is rendered at lateral-sweep index 15 with $\Delta x = -0.00327 = -0.375b$ and $b = 0.00872$ in scene coordinates. Index 24 coincides with the original raw camera pose ($\Delta x = 0$). We define the far-tail diagnostic as

$$e_i = \left[\frac{z_i - \hat{D}(p_i)}{\hat{D}(p_i)} - 0.05 \right]_+, \qquad F(p) = \sum_i T_{i,p} \alpha_{i,p} e_i, \tag{12}$$

where $\hat{D}$ is the scale-aligned VGGT reference depth. Hence $F(p)$ is not a conventional depth map; it weights excessive far-depth error by actual alpha-compositing contribution.

C Additional Stable-View Tube Analysis

The stable-view tube constrains the narrow camera neighborhood between the raw trajectory and the desired stabilized trajectory. Certified pixels receive RGB-D supervision obtained by reprojecting

Table 4: **Distribution of learned Gaussian lifespans in the first 16-frame local window.** FWHM is measured in frames; values ≥ 16 indicate near-persistent support within the current window.

Scene	# Gaussians	P25 / Med. / P75	< 4	4–8	8–16	≥ 16
Ellie	915,748	5.07 / 10.29 / 17.95	4.47%	37.60%	25.94%	31.98%
Lunte	1,666,285	7.51 / 12.68 / 16.63	0.52%	26.31%	44.63%	28.53%
Pip City	463,434	8.17 / 13.13 / 19.76	3.49%	20.70%	39.39%	36.42%
Security Drone	217,429	6.36 / 19.26 / 28.58	11.92%	14.72%	9.96%	63.40%
Snow Barscene	201,019	5.21 / 16.97 / 30.43	14.42%	18.62%	15.02%	51.93%
Windmill	198,662	12.43 / 25.12 / 30.75	6.29%	9.54%	15.28%	68.90%
Worker Robot	387,385	5.16 / 10.27 / 19.76	7.23%	34.45%	24.87%	33.44%
Overall	**4,049,962**	**6.02 / 12.84 / 18.86**	**3.98%**	**27.17%**	**33.15%**	**35.70%**

reliable neighboring observations, while regions without a reliable pseudo target are not forced to match hallucinated geometry. Instead, unsupported Gaussian contribution that becomes newly exposed under a small lateral camera displacement is suppressed.

We distinguish three cases:

- regions already visible in the current raw frame;
- regions not visible in the current frame but observed reliably in neighboring frames;
- regions unsupported by any reliable local observation.

The first case is mainly handled by raw-view reconstruction and CDepth. The second case can receive certified tube reprojection supervision. The third case receives only the conservative unsupported-exposure constraint.

D Controlled Benchmark and Motion-Calibration Details

D.1 Real Handheld-Trajectory Statistics

The motion-calibration data are obtained from VGGT4D camera trajectories estimated on real NUS handheld videos. The statistics contain 21 windows of 64 frames from 11 independent videos, covering Crowd, Parallax, QuickRotation, Regular, and Zooming, for a total of 1,344 estimated camera-to-world poses.

Adjacent windows from the same video contain one overlapping frame and therefore are not treated as statistically independent. During model fitting, each motion category receives equal total weight. Confidence intervals are computed using 1,000 video-level cluster-bootstrap resamples so that categories with more extracted windows do not dominate the uncertainty estimate.

For the smooth/residual decomposition, positions and sign-continuous unit quaternions are smoothed using a third-order Savitzky–Golay local polynomial with 0.8 s support. Half of the smoothing support is discarded at each sequence boundary, and the residual trajectories are resampled to the benchmark frame rate of 24 fps.

Table 5 summarizes the category-balanced residual statistics under this main decomposition.

The video-level cluster-bootstrap 95% confidence interval for rotational RMS is $[0.437°, 0.747°]$. The corresponding interval for translational RMS is $[0.208\%, 0.539\%]$ of scene depth.

D.2 AR(2) Motion Model and Sequence Generation

Rotation and translation are modeled independently using the category-balanced AR(2) process

$$u_t = a_1 u_{t-1} + a_2 u_{t-2} + L\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I_3), \tag{13}$$

Table 5: **Estimated real handheld-motion statistics under the 0.8-s smooth/residual decomposition.** Translation is expressed relative to median scene depth.

Metric	Rotation	Translation / depth
Residual magnitude RMS	0.479°	0.284%
Median magnitude	0.284°	0.131%
95th percentile	0.974°	0.539%
Median-energy frequency f_{50}	2.089 Hz	2.984 Hz
80%-energy frequency f_{80}	3.102 Hz	6.985 Hz
Energy in 5–12 Hz	5.29%	29.84%

Table 6: **Reference depth used for cross-scene translation normalization.** Values are expressed in the native Blender units of each scene.

Scene	Reference depth d_s
Ellie	4.034
Lunte Desert	25.323
Pip City	47.207
Snow Bar Scene	2.846
Windmill	26.783
Worker Robot	72.914
Security Drone	26.456

where weighted least squares estimates the autoregressive coefficients and the Cholesky factor of the innovation covariance defines L. At 24 fps, the fitted coefficients are

$$\begin{aligned} a_1^R &= 1.2072214, & a_2^R &= -0.6436997, \\ a_1^T &= 0.4729358, & a_2^T &= -0.2421420. \end{aligned} \tag{14}$$

The rotational process has a complex-conjugate pole radius of approximately 0.802 and a characteristic frequency near 2.75 Hz, producing a damped, temporally correlated random motion rather than frame-independent white noise.

Each 120-frame synthetic sequence is preceded by a 120-frame warm-up period to remove the zero-initialization transient. The finite realization is then demeaned before severity normalization. The reported benchmark operating point is the 35th percentile of the estimated real-window RMS distribution, corresponding to 0.348° rotational RMS and 0.158% depth-normalized translational RMS.

For robustness, the radial magnitude of each generated residual is bounded by the empirical real-motion 99th percentile:

$$\|r_t\|_2 \le 1.380^\circ, \qquad \|q_t\|_2 \le 0.984\%. \tag{15}$$

None of the seven final benchmark trajectories reaches these clipping bounds.

D.3 Cross-Scene Scale Normalization

A single Blender translation unit does not correspond to the same apparent camera motion across scenes with different geometric scales. We therefore estimate a scene reference depth d_s for each stable trajectory. Nine frames are sampled uniformly along the path, and 15 rays are cast through the camera frustum at each sampled frame. The median valid hit distance defines d_s.

Table 6 lists the resulting values.

Although these scene scales differ by more than an order of magnitude, the normalized translation q_t is converted back into scene units using the scene reference depth d_s. As verified in Fig. 9, all seven benchmark sequences realize essentially the same configured relative perturbation severity.

Across the seven final trajectories, rotational RMS ranges only from 0.3477607° to 0.3477614°, while translational RMS ranges from 0.1583692% to 0.1583717% of scene depth. The remaining

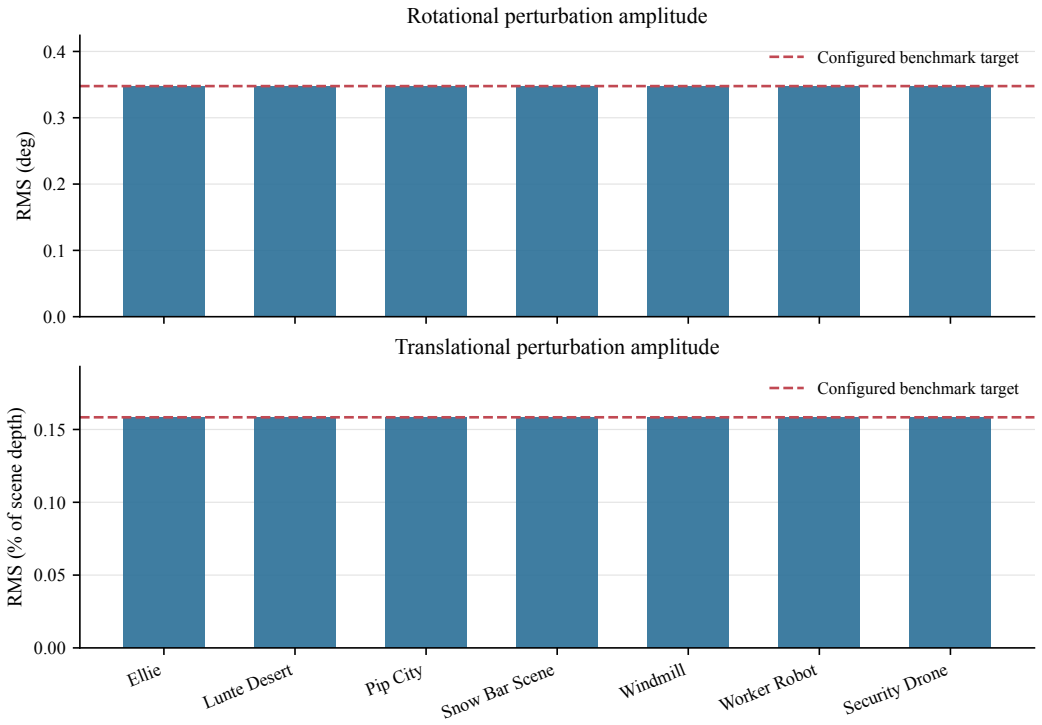
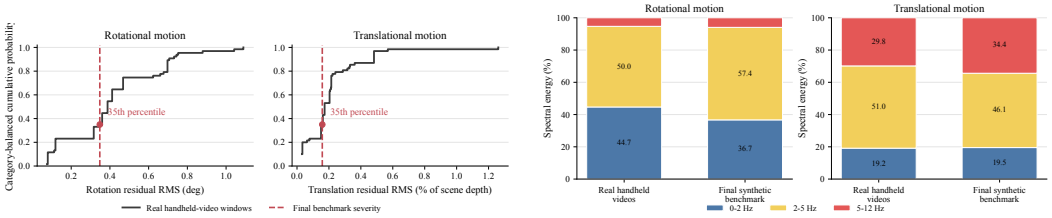


Figure 9: **Cross-scene verification of perturbation normalization.** Despite substantially different native Blender scales, all seven scenes realize the configured rotational and depth-normalized translational RMS targets. This plot verifies the scale-normalization procedure rather than representing an independently optimized result.



(a) Selected severity within the real-motion distribution.

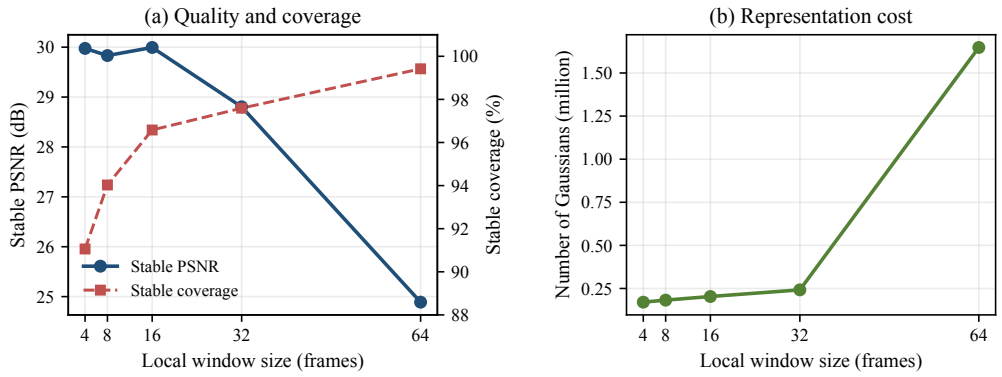
(b) Real/synthetic frequency composition.

Figure 10: **Real-motion calibration of the synthetic benchmark.** The final operating point uses the category-balanced 35th-percentile residual magnitude while preserving the dominant temporal-frequency composition of the estimated handheld motion.

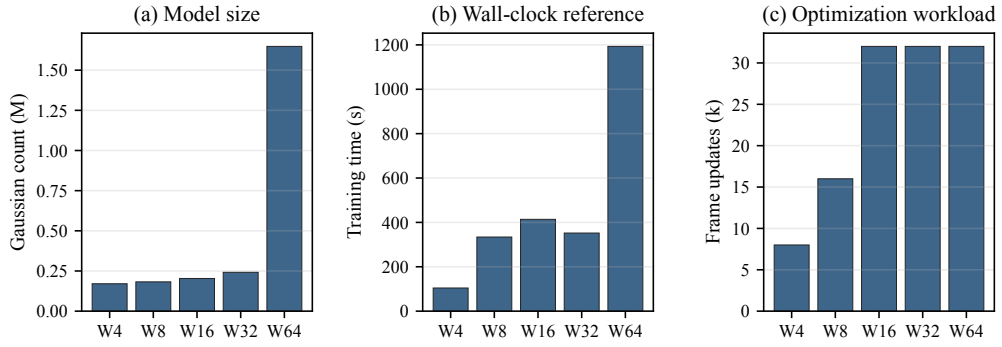
differences are numerical. Figure 10 further compares the selected severity and temporal-frequency composition with the real-motion statistics.

E Additional Parameter Analysis

This appendix provides the complete window-length, densification, sampling, learning-rate, and optimization-budget sensitivity analyses used to select the default configuration.



(a) Stable-view quality, coverage, and wall-clock reference.



(b) Representation size and optimization workload.

Figure 11: **Detailed local-window sweep.** Increasing the window initially improves stable-view coverage, but excessively long windows increase representation cost and reduce stable-view fidelity. These measurements support the default choice of $W = 16$.

E.1 Detailed Local-Window Analysis

We vary the local window length $W \in \{4, 8, 16, 32, 64\}$ on the Security Drone sequence while fixing the optimizer to 2,000 steps and using `frames_per_step=16`; Fig. 11 and Table 7 summarize the corresponding quality–cost trade-off. Stable PSNR measures rendering fidelity at the stabilized reference view, while stable coverage denotes the fraction of valid pixels covered by the Gaussian representation.

Increasing the temporal context from $W = 4$ to $W = 16$ improves stable-view coverage from 91.1% to 96.6% while maintaining approximately 25 dB Stable PSNR. Further increasing the window provides only modest additional coverage but degrades stable-view quality and increases representation size. At $W = 64$, coverage reaches 99.4%, but Stable PSNR falls to 19.89 dB and the representation grows to 1.65M Gaussians.

Because the number of available frames is smaller than `frames_per_step` for $W = 4$ and $W = 8$, this experiment should be interpreted as a fixed-optimizer-step window sensitivity study rather than a strictly compute-matched comparison.

E.2 Post-Densification Recovery

Topology updates temporarily disturb the optimized Gaussian representation. Figure 12 illustrates this effect on the first 16 frames of the Ellie Blender-GT experiment.

After the last densification step at iteration 1,500, raw PSNR is 27.53 dB. With no further topology update, only 250 additional optimization steps increase PSNR to 31.65 dB. Over the same interval,

Table 7: Detailed local-window sweep on Security Drone. Wall-clock time is reported as a hardware-dependent reference rather than a strictly compute-matched efficiency metric.

W	Stable PSNR↑	Coverage↑	Gaussians (M)↓	Time (s)↓
4	24.97	91.1%	0.171	104.4
8	24.83	94.0%	0.182	334.1
16	24.99	96.6%	0.203	413.8
32	23.80	97.6%	0.242	352.0
64	19.89	99.4%	1.648	1193.3

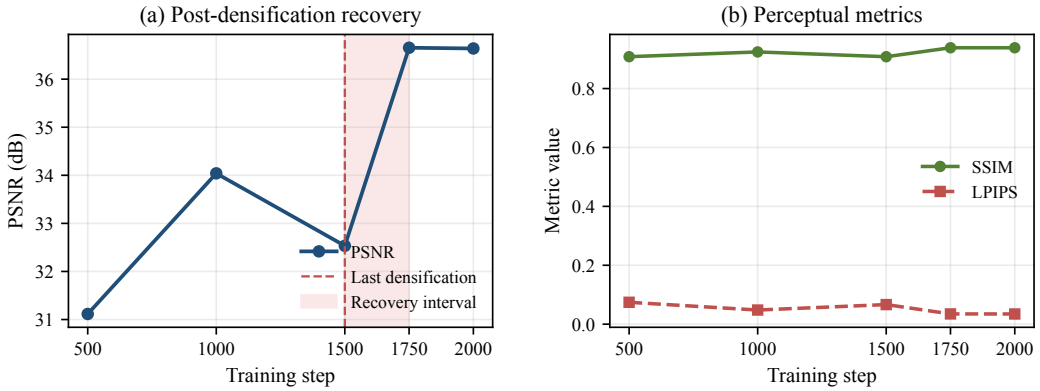


Figure 12: **Post-densification recovery.** The last topology update temporarily reduces reconstruction quality. Once densification is stopped, a short fixed-topology optimization interval rapidly recovers PSNR, SSIM, and LPIPS, after which the result saturates.

SSIM improves from 0.908 to 0.939, while LPIPS decreases from 0.0665 to 0.0348. The results at 1,750 and 2,000 steps are nearly identical, indicating that the representation rapidly saturates after this recovery stage.

This behavior motivates separating topology search from final convergence: densification is useful while the Gaussian structure is being established, but continuing structural updates late in optimization repeatedly perturbs an already well-fitted representation.

E.3 Detailed Densification-Schedule Sweep

We additionally vary the densification interval i, pruning interval p, and densification stop step u. This diagnostic uses a longer optimization budget than the final 2k-step configuration to isolate the effect of prolonged topology updates. Early-stop configurations with $u =$ 3k–9k consistently reach approximately 36.1–36.3 dB final raw PSNR, whereas extending topology updates to 14k–18k iterations provides no additional gain and can reduce reconstruction quality.

As summarized in Table 8, among the tested early-stop schedules, `i500/p1000/u7k` reaches the highest final raw PSNR of 36.33 dB. The nearby configuration `i500/p1000/u5k` obtains 36.19 dB with lower wall-clock cost. These results indicate that topology updates should be treated as a finite structure-search stage rather than being continued throughout the full optimization.

Substantially later stop settings provide no consistent gain; their lower final quality further supports early termination of topology updates.

E.4 Frames per Optimization Step

We vary `frames_per_step`$\in \{6, 8, 12, 16\}$ while fixing the number of optimizer steps to 5,000. As shown in Fig. 13, increasing the number of sampled frames improves raw reconstruction PSNR,

Table 8: Detailed early-stop densification sweep. i, p, and u denote the densification interval, pruning interval, and densification stop step, respectively.

Schedule	Raw PSNR↑	Gaussians (k)↓	Time (s)↓
i250/p500/u3k	36.13	204.7	1349
i250/p500/u5k	36.10	201.4	1329
i250/p500/u7k	36.18	197.7	1332
i250/p500/u9k	36.17	196.5	1330
i500/p1000/u3k	36.17	201.4	1245
i500/p1000/u5k	36.19	198.9	**1244**
i500/p1000/u7k	**36.33**	200.9	1305
i500/p1000/u9k	36.13	199.1	1386

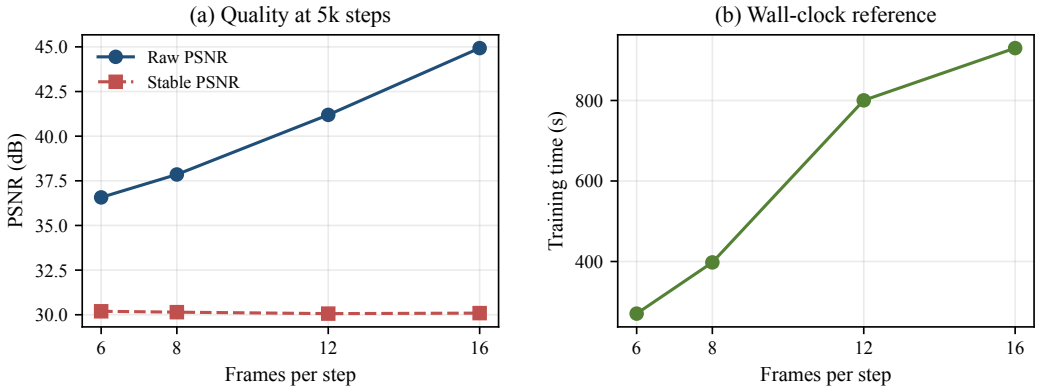


Figure 13: **Effect of the number of frames sampled per optimizer step.** Increasing the sampled-frame count improves raw-view reconstruction under a fixed number of optimizer steps, while stable-view PSNR changes comparatively little. The right panel reports the corresponding wall-clock trend.

while Stable PSNR changes only slightly and the additional cost grows approximately with the number of frame updates.

Because different settings correspond to different total numbers of frame updates, this experiment is intended to characterize the throughput–quality trend rather than provide a strictly equal-compute comparison. We use `frames_per_step=16` in the final implementation because it provides the most complete temporal sampling at the selected local window size.

E.5 Learning-Rate Schedule

We compare several learning-rate scaling and warm-down configurations at both short and long optimization budgets. As shown in Fig. 14, the resulting Stable PSNR values remain within a relatively narrow range. The same qualitative ranking is also largely preserved when optimization is extended from 4k to 20k steps.

This indicates that the exact learning-rate schedule is not the dominant bottleneck of the current system. In contrast, the local-window length and the timing of topology updates have much larger effects on stable-view quality and representation behavior. We therefore use the selected learning-rate configuration in all final experiments.

E.6 Optimization Budget

The post-densification experiment in Fig. 12 shows that reconstruction quality rapidly saturates once topology updates are terminated. The final implementation therefore separates the total training budget into an early structural optimization stage and a later fixed-topology recovery stage.

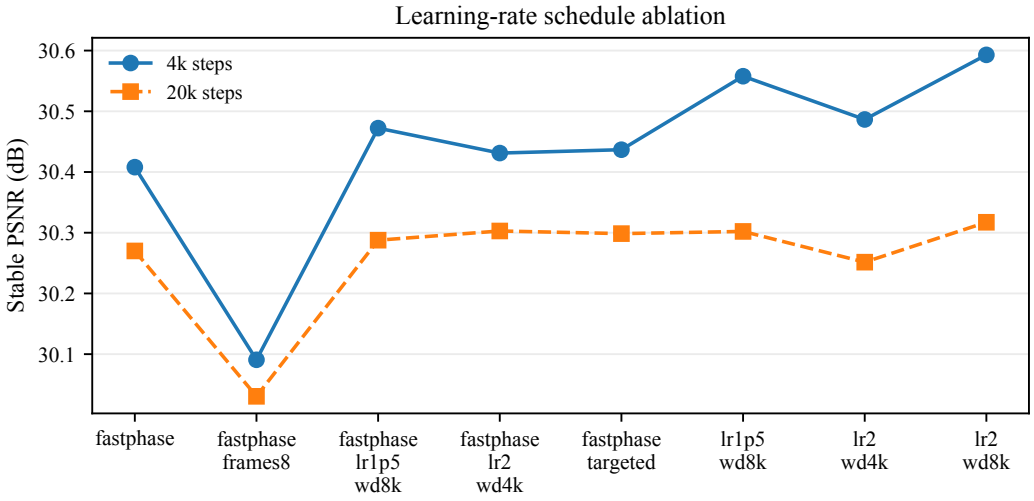


Figure 14: **Learning-rate schedule ablation.** Stable-view quality varies only modestly over the tested learning-rate scales and warm-down settings at both 4k and 20k optimization steps.

Table 9: Final default configuration of Stab4D.

Parameter	Setting
Local window length W	16
Frames per optimization step	16
Total optimization steps / window	2000
Densification interval	500
Pruning interval	1000
Densification start step	500
Densification stop step	1500
Minimum temporal span	2 frames
Positive evidence threshold	$\lvert z - D\rvert \leq \max(3s_z,\ 0.05z + 0.01)$
Negative evidence threshold	In-view but depth-inconsistent
λ_{evid}	0.01
λ_{span}	0.002
λ_{cdepth}	0.05
CDepth activation	After densification stops
Tube sampling probability	0.5
λ_I	0.03
λ_α	0.005
λ_{expose}	0.05
Gaussian population cap	6×10^6

F Final Hyperparameter Configuration

Table 9 summarizes the final default configuration used in our experiments.

G Additional Qualitative Results

The supplementary material provides additional stabilized videos and frame-level comparisons on both NUS and the controlled dynamic benchmark. For the controlled benchmark, all reconstruction-based methods use the same Blender ground-truth pose/depth and stable target trajectory under the protocol in Sec. D.

We additionally include lateral camera sweeps and Gaussian PLY files for representative examples. These visualizations are particularly useful for identifying geometry failures that are difficult to diagnose from the final stabilized RGB image alone, including depth-separated surface fragments, unsupported floaters, and geometry that becomes visible only under lateral camera displacement.