SNAPSHOT-TO-FLOW: RESIDUAL DYNAMICS FOR ADAPTING PRETRAINED 4D SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward 4D reconstruction supplies a reusable scene prior, but individual videos retain motion and appearance errors. We introduce Snapshot-to-Flow, a scene-level adapter that learns residual dynamics around pretrained trajectories. A shared velocity field integrates source-referenced position residuals; local appearance and scene-wide color controls address complementary rendering errors before an unchanged rasterizer. Complete sequence records with dynamic-branch updates show mean novel-frame PSNR gains of 1.29, 2.25 and 1.12 dB on STAG4D, HyperNeRF and Neural 3D Video. These validation gains retain all evaluated sequences, including negative-gain cases. An eight-scene motion-query study reports a mean gain of 1.31 dB beyond the conditioning interval. Component and scene-attribute diagnostics characterize the complete adaptation procedure; separate cross-scene controls distinguish its benefits from integration-specific superiority.

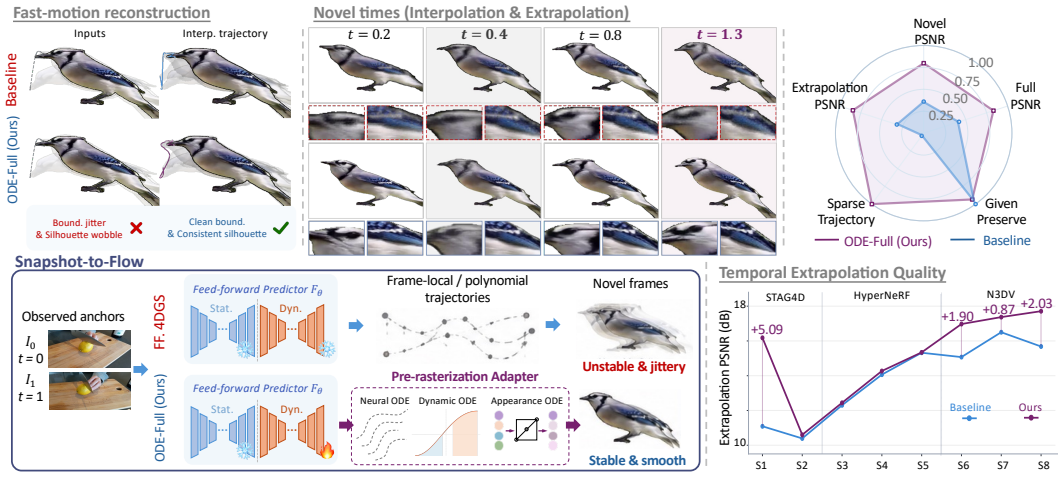


Figure 1: **Sequence adaptation and motion queries.** Motion and appearance controls specialize the scene; observed-frame preservation is an objective, not an exact constraint. Beyond-pair scores use motion queries with appearance controls disabled, not verified forecasting beyond training. ODE: ordinary differential equation; PSNR: peak signal-to-noise ratio; Interp.: interpolation; ODE-Full: controls with dynamic-branch adaptation.

1 INTRODUCTION

Feed-forward reconstruction makes dynamic scenes available from sparse visual observations. Gaussian representations support efficient rendering (Kerbl et al., 2023; Luiten et al., 2024; Li et al., 2024), while learned predictors transfer scene priors across videos (Wu et al., 2026; Liang et al., 2025; Ma et al., 2025). Reconstruction can therefore begin with a predicted scene and correct video-specific errors. Repeated rendering can amortize this additional preparation cost.

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.

The adaptation problem is not simply a smaller version of training a reconstruction model. A pretrained predictor already supplies geometry, appearance and an initial account of motion. Relearning all of these quantities from a limited sequence can discard useful structure; correcting rendered frames separately does not provide a shared scene-level description of the correction. The design question is therefore where to place adaptation capacity so that a single fitted representation can improve reconstruction across time.

We take predicted trajectories as the reference for adaptation. A Gaussian is not merely an unstructured feature: its position, opacity and color determine how a moving scene element contributes to an image. These quantities have different roles. Position corrections alter its path, local appearance corrections change its contribution to the rendering, and shared color controls capture changes common to the scene. Organizing adaptation around these explicit attributes makes its intervention inspectable at both the trajectory and image levels.

Our central idea is to learn residual dynamics around the pretrained scene rather than replace its motion model. A shared velocity field is evaluated along predicted paths and integrated relative to each primitive's source time. The same fitted field defines corrections across temporal queries, with the reference path retained as the starting point. Integration organizes temporal corrections; it does not guarantee physical motion accuracy or superiority over alternative parameterizations.

We introduce *Snapshot-to-Flow*, which combines source-referenced motion residuals with local appearance and scene-wide color controls at the prediction–rendering interface. The renderer is unchanged, and the controls are fitted once per sequence. A compact configuration freezes the predictor; an extended configuration additionally updates its dynamic branch. This separation distinguishes the adaptation representation from the amount of trainable capacity. Our implementation is evaluated with one pretrained backbone; its interface does not by itself establish transfer to all reconstruction models.

Complete STAG4D, HyperNeRF and Neural 3D Video records show mean novel-frame PSNR gains of 1.29, 2.25 and 1.12 dB with dynamic updates. These validation gains include negative-gain sequences and match the component study's full-configuration row. Per-scene results, temporal slices and attribute diagnostics describe the fitted corrections. Beyond-pair motion queries and cross-scene direct-residual controls use separately specified evaluation and training settings.

Our contributions are:

- A trajectory-space formulation of sequence adaptation that retains a pretrained 4D scene as a reference and learns its residual evolution.
- A scene-level implementation combining source-referenced motion correction with local and shared appearance controls before an unchanged renderer.
- A multi-dataset evaluation of reconstruction gains, adaptation capacity and temporal behavior, with per-scene outcomes and a separately identified continuation study.

2 Related Work

Feed-forward reconstruction and scene adaptation. Dynamic radiance fields and Gaussians represent evolving scenes (Park et al., 2021; Li et al., 2022; Kerbl et al., 2023). Feed-forward models include pixelSplat and AnySplat for static scenes (Charatan et al., 2024; Jiang et al., 2025), and BTimer, STORM, NutWorld, StreamSplat, 4D-LRM and MoVieS for dynamic reconstruction (Liang et al., 2025; Yang et al., 2025; Shen et al., 2025; Wu et al., 2026; Ma et al., 2025; Lin et al., 2026). Fuse-and-Refine and ReSplat refine predicted scenes (Wang et al., 2025b; Xu et al., 2026); TokenGS adapts Gaussian tokens and LoRA3D uses low-rank geometric self-calibration (Ren et al., 2026; Lu et al., 2025). These complement parameter-efficient adaptation (Houlsby et al., 2019; Hu et al., 2022). Our focus is the representation of sequence-specific corrections at the scene interface, rather than learning another reconstruction backbone.

Dynamic scene structure and temporal models. Dynamic 3D Gaussians uses persistent primitives and local-rigidity constraints (Luiten et al., 2024); MoSca organizes deformation with motion scaffolds (Lei et al., 2025). GaussianFlow models primitive flow, FreeTimeGS learns temporal support, and ODE-GS learns latent dynamics from reconstructed Gaussian histories (Lin et al., 2024; Wang et al., 2025a; 2026). These methods establish the importance of temporal scene structure. We

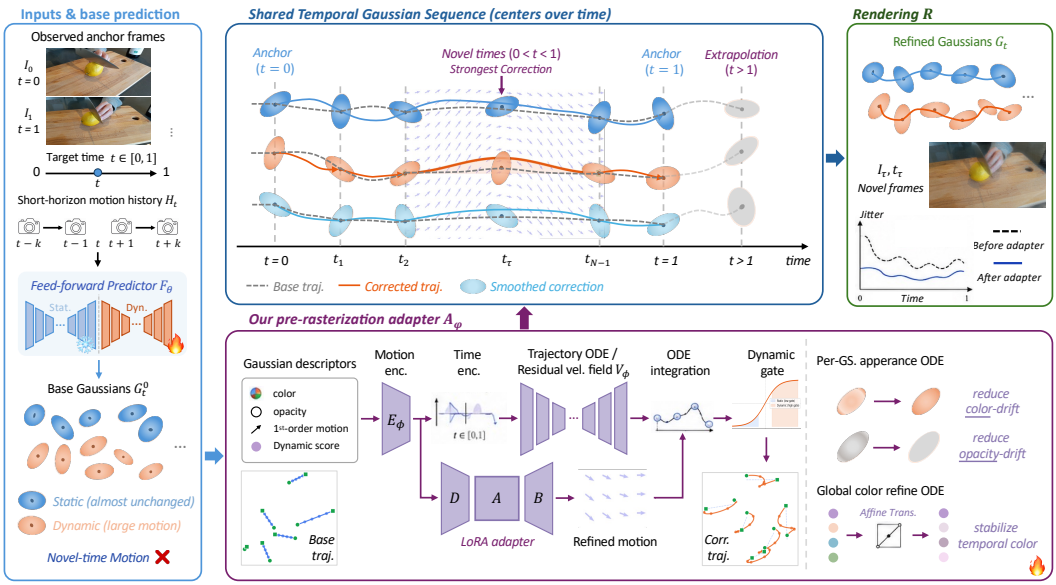


Figure 2: **Trajectory-space adaptation.** A pretrained predictor supplies reference paths; motion and appearance controls specialize the scene before rendering. ODE: ordinary differential equation; GS: Gaussian splatting; LoRA: low-rank adaptation. Adapter-only freezes the predictor; ODE-full also adapts its dynamic branch.

use neural differential equations (Chen et al., 2018) to parameterize corrections along existing predicted trajectories, not to infer an interaction graph or replace the complete scene dynamics. Shared motion and appearance controls organize the residual task around explicit scene attributes.

Temporal interpolation and rendering quality. Softmax splatting, RIFE, FILM and AMT synthesize intermediate images (Niklaus & Liu, 2020; Huang et al., 2022; Reda et al., 2022; Li et al., 2023). They address an important part of the practical objective: accurate frames at unobserved times. Our method instead modifies a dynamic Gaussian representation and exposes the corresponding motion and appearance corrections. This distinction motivates scene-level diagnostics, but does not remove image interpolation as a relevant comparison or make image quality sufficient evidence of accurate 3D geometry.

3 Method

3.1 Residual Adaptation of a Scene Prior

The predictor supplies reference paths and conditioning statistics; the adapter specializes motion and appearance before rendering:

$$G^0(t) = F_\theta(I_0, I_1, t),$$
$$G(t) = A_\phi(G^0(t), \mathcal{H}, t), \qquad \hat{I}(t) = R(G(t)). \tag{1}$$

Here I_0, I_1 are endpoint images at normalized times zero and one, t is a query time, F_θ is the pretrained predictor, G^0 its Gaussian output, A_ϕ the correction module, $\mathcal{H}$ conditioning statistics of the base prediction, and R the unchanged splatting renderer. The corrected scene G renders to $\hat{I}$. Its N primitives store centers, scales, rotations, opacities and colors in predictor coordinates, not necessarily metric-world coordinates.

Adapter-only fits ϕ with predictor parameters θ frozen. ODE-full additionally updates the dynamic branch $\psi \subseteq \theta$. The compact and extended configurations have different capacities; Appendix B records their settings.

We instantiate this interface with StreamSplat (Wu et al., 2026), conditioned on two images and estimated depths; adaptation uses additional supervision. Primitive identities and source times are fixed within local intervals. We evaluate this instantiated scene interface, without claiming cross-fragment correspondence or universal backbone transfer.

3.2 Motion and Appearance Residual Dynamics

The motion branch evaluates a shared velocity field along reference paths. Integration accumulates each correction relative to its primitive's source time:

$$\begin{aligned} x_i^0(t) &= x_{i,s}^0 + \sum_{k=1}^{K} c_{i,k}^0 (t - a_i)^k, \\ v_i(s) &= \lambda_v V_\phi(x_i^0(s), f_i, \gamma(s)), \\ \delta_i(t) &= \int_{a_i}^{t} v_i(s)\, ds, \qquad x_i(t) = x_i^0(t) + g_i \delta_i(t). \end{aligned} \tag{2}$$

Here i indexes a Gaussian, $x_{i,s}^0 \in \mathbb{R}^3$ is its own-anchor center, $c_{i,k}^0 \in \mathbb{R}^3$ is its degree-k temporal coefficient, K is the trajectory degree and $a_i \in \{0, 1\}$ identifies the source endpoint. The velocity network V_ϕ uses integration time s, base position $x_i^0(s)$, descriptor f_i and sinusoidal time encoding $\gamma(s)$. The scalar λ_v scales velocity; δ_i is integrated displacement, x_i the corrected center, and $g_i = |\tanh(\kappa \rho_i)|$ an opacity-derived weight from decoder scalar ρ_i with sharpness κ.

Descriptors encode base color, anchor opacity, first-order motion and dynamic opacity in 32 channels. The width-64 velocity network uses four time-frequency pairs, $\lambda_v = 0.05$ and $\kappa = 5$. Velocity depends on the base path, giving a state-independent ODE. Four Simpson substeps implement the state-independent Runge–Kutta update with identical midpoint evaluations. Both reference and corrected paths support continuous queries; integration is the chosen residual parameterization.

Endpoint support and a temporal envelope weight the color and opacity corrections:

$$\begin{aligned} p_i &= \mathbf{1}[\alpha_i^0(0) > \tau_p]\mathbf{1}[\alpha_i^0(1) > \tau_p], \\ w_i &= p_i + (1 - p_i)\beta, \qquad \eta(t) = \eta_{\min} + (1 - \eta_{\min})\, 4t(1 - t). \end{aligned} \tag{3}$$

Here α_i^0 is base opacity, $\tau_p = 0.05$ the threshold, $\mathbf{1}$ the indicator, p_i the endpoint-persistence proxy, w_i the appearance weight, and $\beta = 0.25$ the transient-primitive weight. The envelope η is defined here for $t \in [0, 1]$; $\eta_{\min}$ controls its endpoint floor. These heuristic weights summarize endpoint support rather than calibrated uncertainty or intervening visibility.

A width-48 second-order latent ODE predicts local color and opacity residuals:

$$\dot{h}_i(s) = u_i^h(s), \qquad \dot{u}_i^h(s) = f_\xi(h_i(s), u_i^h(s), h_i(0), s). \tag{4}$$

Here h_i is a state initialized from primitive statistics, u_i^h its latent velocity, f_ξ the acceleration network with parameters ξ, and s integration time. Four substeps yield local color and opacity-logit residuals u_i, z_i. A width-32 scene-level ODE with the same form supplies shared RGB scale $s(t)$ and bias $b(t)$. The local and shared controls act through

$$\begin{aligned} r_i^{\text{loc}}(t) &= \text{clip}_{[0,1]}(r_i^0(t) + \eta(t) w_i u_i(t)), \\ \alpha_i(t) &= \text{sigmoid}(\text{logit}(\alpha_i^0(t)) + \eta(t) w_i z_i(t)), \\ r_i(t) &= \text{clip}_{[0,1]}(r_i^{\text{loc}}(t) \odot (1 + s(t)) + b(t)). \end{aligned} \tag{5}$$

Here r_i^0 is base RGB, r_i^{loc} locally corrected RGB, r_i final RGB, α_i corrected opacity and $\odot$ componentwise multiplication. Reference output bounds are 0.03 and 0.05 for u_i, z_i, with global scale 0.03; cumulative variants use the scales in Appendix B. This appearance envelope applies to interpolation. The beyond-pair experiment disables appearance controls and evaluates motion queries separately.

Source-time behavior. Motion residuals vanish at their own source time, not both endpoints. The floor $\eta_{\min} = 0.20$ permits endpoint appearance changes, and dynamic updates can alter the base. Zero output heads suppress initial residuals, but nonzero low-rank initialization makes the complete initial mapping only approximately unchanged.

3.3 Sequence Adaptation and Querying

Sequence supervision fits the active controls once; subsequent temporal queries reuse their parameters. Novel denotes validation frames whose RGB targets are excluded from adaptation training. We use AdamW with learning rate 5×10^{-5}, 25 warmup iterations and 6–8 epochs. Image, mask, perceptual and relative-depth terms fit rendered observations, while spatial, temporal and attribute regularizers constrain corrections. Relative depth supplies a training prior, not metric ground truth. Checkpoint selection balances Novel validation quality against anchor and full-frame degradation through soft penalties. Appendix B records the objective and selection rule. Preparation cost, query cost and the scope of motion diagnostics are distinguished in Section 4.

4 Experiments

4.1 Evaluation Setting

We evaluate reconstruction, components and temporal behavior on STAG4D Video-to-4D, HyperNeRF and Neural 3D Video (N3DV), with unchanged official pretrained StreamSplat as reference. Adapter-only freezes it; ODE-full also updates its dynamic branch. Aggregate claims use complete records: Table 2's ODE-Full row and Appendix C.1.

Novel comprises validation frames excluded from the loss but used for checkpoint selection. Given denotes input frames; Full is their evaluated union with Novel. Complete-record scores use original images. Table 1's archived Given column instead uses input-video Anchor scores. Additional training frames supervise adaptation. PSNR denotes peak signal-to-noise ratio (dB); gains subtract unchanged-base scores.

Preparation and rendering. Sequence adaptation takes 273–998 seconds, excluding checkpoint evaluation. This preparation fits reusable controls; it is not zero-shot inference. Querying also incurs correction cost. Appendix C.4 profiles a separate continuation configuration, not the throughput of these sequence-specific runs.

4.2 Adapting the Scene Prior

Sequence-specific adaptation improves mean validation-frame PSNR by 1.29, 2.25, and 1.12 dB on STAG4D, HyperNeRF, and Neural 3D Video, respectively. These complete-cohort results support the effectiveness of the combined motion and appearance adaptation procedure. Table 2 reports these means; Appendix C.1 provides per-scene results. Table 1 retains archived summaries with the captioned cohort and metric differences.

Table 1: **Archived sequence-adaptation summaries.** PSNR: peak signal-to-noise ratio (dB); Δ: gain; Seq.: sequence count; ODE: ordinary differential equation; N3DV: Neural 3D Video. Novel: validation; Given: input-video Anchor; Full: input/validation union. Adapter freezes the predictor; ODE-full updates its dynamic branch. The extended STAG4D subset excludes negative gains; its absolute PSNR and the HyperNeRF subset remain unreconciled (Appendix C.1).

Setting	Dataset	Seq.	Novel	Novel Δ	Full Δ	Given Δ
Adapter	STAG4D	31	23.25	+0.60	+0.38	−0.15
	HyperNeRF	17	16.44	+0.44	+0.24	−0.31
	N3DV	6	30.28	+0.14	+0.01	−0.11
ODE-full	STAG4D	21	24.59	+1.94	+1.41	+0.15
	HyperNeRF	15	18.27	+2.27	+1.37	−0.36
	N3DV	6	31.26	+1.12	+0.71	+0.06

From image gains to scene corrections. Figures 3 and 4 locate rendering improvements in images and temporal slices. Table 2 examines cumulative control contributions. Appendix C.4 compares integration with direct residual prediction under separate cross-scene training; the gains above evaluate the complete sequence-adaptation procedure.

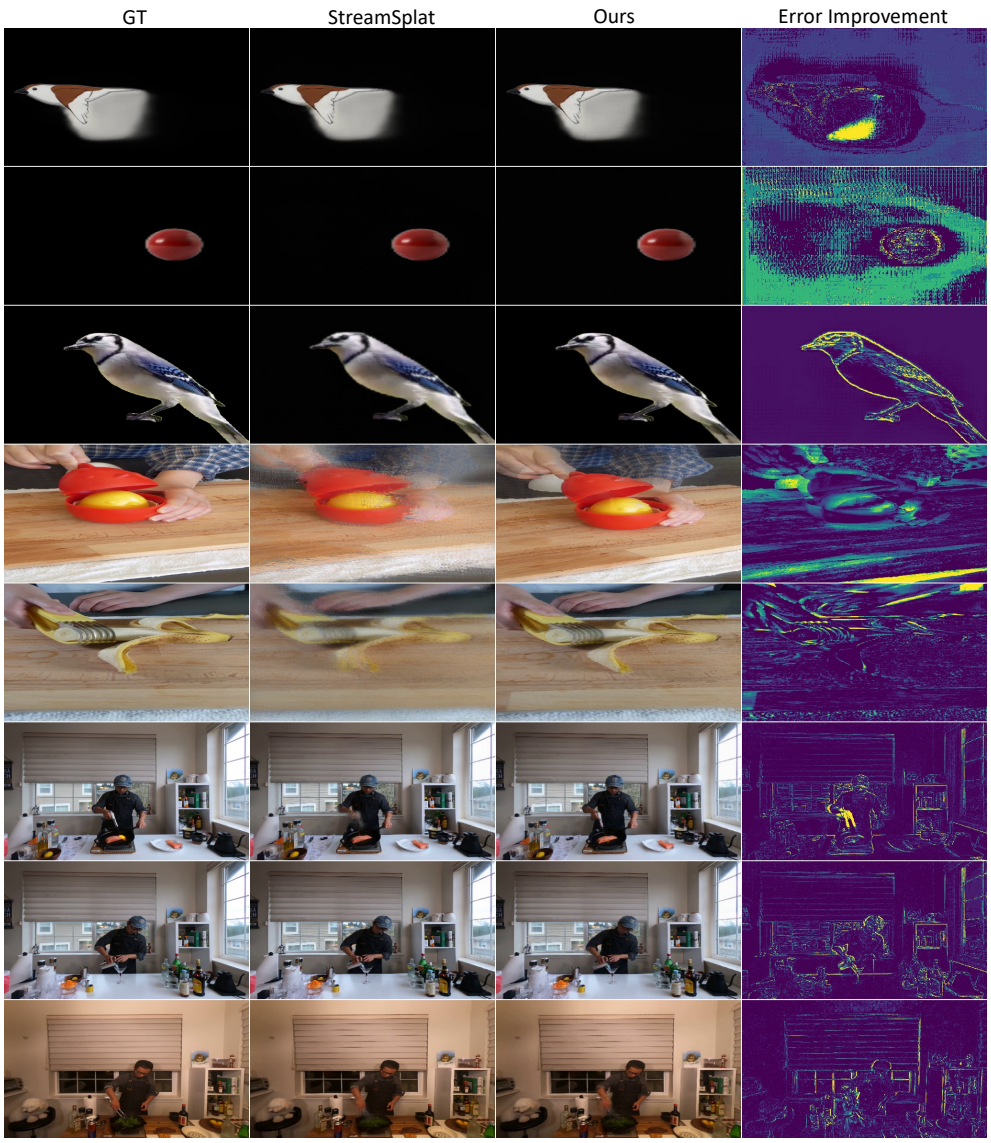


Figure 3: **Scene-level reconstruction.** Ground truth (GT), StreamSplat, adapted results and error improvement on STAG4D, HyperNeRF and Neural 3D Video (N3DV).

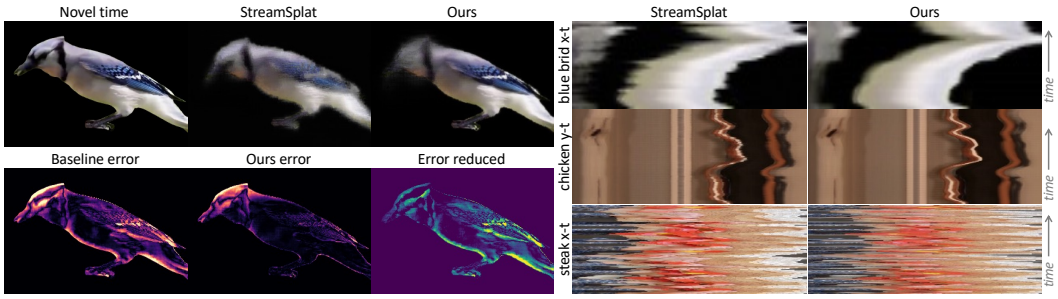
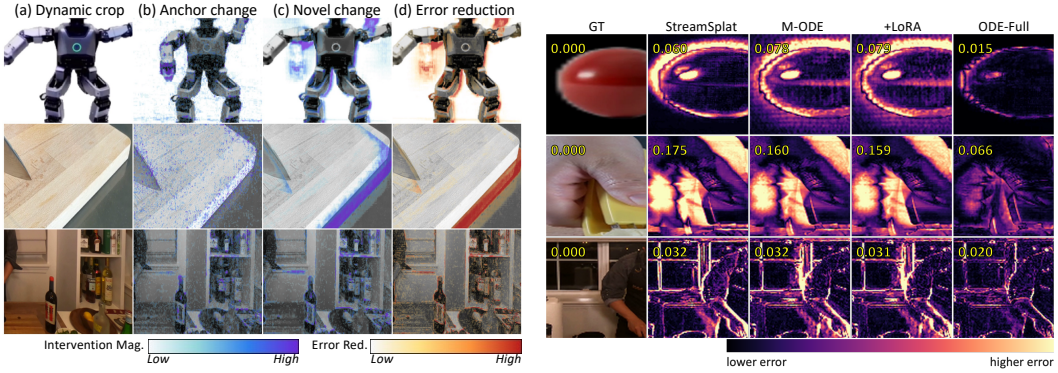


Figure 4: **Spatial and temporal corrections.** Novel-frame comparisons and error maps (left); image-row or image-column slices over time (right). One axis follows time; the other follows a fixed image row or column.

Complementary motion and appearance controls. Adding appearance controls improves all three datasets over motion and dynamic LoRA (Table 2). Decoder adaptation and full dynamic updates yield further gains. This cumulative pattern supports joint motion–appearance adaptation, with the strongest results obtained when dynamic-branch parameters are also updated.

Table 2: **Component contributions.** Novel validation PSNR (peak signal-to-noise ratio) gains, dB. M-ODE: motion ordinary differential equation; +LoRA: dynamic low-rank adaptation; +App: local appearance/global color; LW-Full: decoder LoRA; ODE-Full: dynamic updates. Its complete-cohort means match Appendix C.1. N3DV: Neural 3D Video; Params: archived counts; M: million. Count reconciliation: Appendix B.

Setting	STAG4D	HyperNeRF	N3DV	Params
M-ODE	+0.15	+0.34	−0.04	0.265M
+LoRA	+0.15	+0.35	+0.00	0.294M
+App	+0.53	+0.46	+0.08	0.304M
LW-Full	+0.57	+0.48	+0.21	0.304M
ODE-Full	+1.29	+2.25	+1.12	full



(a) Intervention maps **(b) Component effects**

Figure 5: **Where the controls act.** (a) Dynamic crops, anchor changes, hidden-frame changes and error reduction. Blue–purple: intervention; orange–red: error reduction. This separates changes to observed frames from corrections at hidden times. (b) Component error maps: GT, ground truth; M-ODE, motion-only ordinary differential equation; LoRA, low-rank adaptation; ODE-Full, continuous-time controls with dynamic-branch adaptation.

4.3 Temporal Behavior of the Adapted Scene

Table 3 characterizes scene corrections, not motion accuracy. Persistence is the fraction with both endpoint opacities above 0.05. Motion and Leak are maximum infinity-norm residual displacements of persistent primitives at Novel and endpoint times. Vel. is maximum sampled infinity-norm residual velocity; Id. err. is mean absolute attribute change at time zero. These are sequence means; rounded zeros are not exact constraints, and smaller motion need not mean better geometry.

Table 3: **Trajectory diagnostics.** Persist.: persistent fraction; Motion: novel-frame displacement; Leak: endpoint leakage; Vel.: sampled maximum velocity; Id. err.: identity error; ODE: ordinary differential equation; N3DV: Neural 3D Video.

Setting	Dataset	Persist.	Motion	Leak	Vel.	Id. err.
Adapter	STAG4D	0.244	0.025	0.000	0.181	0.042
	HyperNeRF	0.267	0.032	0.000	0.188	0.054
	N3DV	0.315	0.033	0.000	0.200	0.056
ODE-full	STAG4D	0.827	0.002	0.000	0.138	0.004
	HyperNeRF	0.367	0.006	0.000	0.209	0.010
	N3DV	0.812	0.010	0.000	0.152	0.017

Table 4: **Archived tracks, not model accuracy.** Normalized image-plane distances; Red.: archived reduction; N3DV: Neural 3D Video. Original labels are retained, but constructed/fallback tracks invalidate the first three comparisons and their aggregate (Appendix C.2).

Scene	StreamSplat	ODE-full	Red.
STAG4D / jump	0.214	0.000	100%
HyperNeRF / slice-banana	0.063	0.000	100%
HyperNeRF / cross-hands	0.106	0.000	100%
N3DV / flame-steak	0.042	0.016	63%
Mean	0.106	0.004	96%

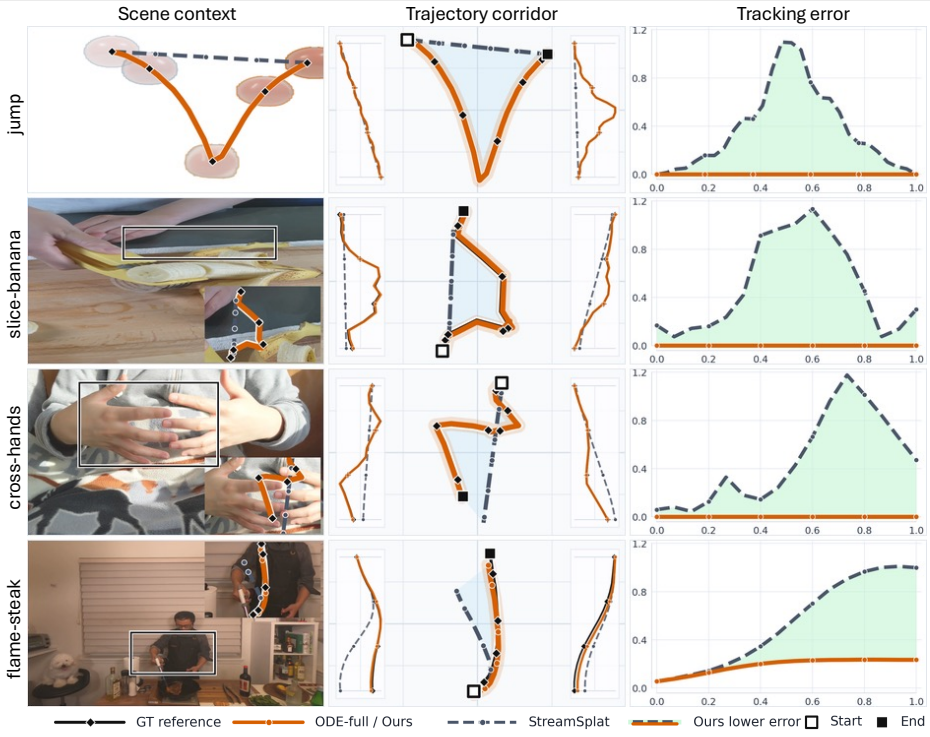


Figure 6: **Archived trajectory visualization.** Scene context, projected curves and deviations. Constructed/fallback tracks in the first three panels preclude model-accuracy conclusions (Appendix C.2).

Residual magnitude and endpoint support. The attribute diagnostics in Table 3 quantify how the adapted representation changes. For N3DV, Motion decreases from 0.033 to 0.010 while persistence increases from 0.315 to 0.812. The extended configuration therefore combines smaller measured residual displacements with a larger fraction of primitives supported at both endpoints. Residuals are measured relative to each configuration's base prediction, which can itself change under dynamic updates.

Querying beyond the conditioning interval. With conditioning frames 0 and 3, queries use $t = j/3$. Interpolation averages frames 1–2; extrapolation averages frames 4–9, excluding anchors. Gains are 2.56 and 1.31 dB (Table 5). This tests queries beyond the conditioning pair, not verified forecasting beyond all training observations. Appendix C.2 specifies the motion-query evaluator.

Table 5: **Beyond-pair queries.** PSNR: peak signal-to-noise ratio (dB); BL: StreamSplat; Ours: adapted motion-query configuration; Δ: gain; N3DV: Neural 3D Video. Interpolation: $t = 1/3, 2/3$; extrapolation: $t = j/3$, $j = 4, \ldots, 9$. Means exclude anchors.

		Interpolation			Extrapolation		
Dataset	Scene	BL	Ours	Δ	BL	Ours	Δ
STAG4D	jump	32.36	34.40	+2.04	11.09	16.18	+5.09
	blue_bird	23.18	24.32	+1.14	10.39	10.60	+0.21
HyperNeRF	chickchicken	18.21	19.76	+1.54	12.30	12.44	+0.14
	cut_lemon	18.51	19.14	+0.63	14.06	14.27	+0.22
	slice_banana	20.32	21.53	+1.21	15.32	15.34	+0.03
N3DV	coffee_martini	24.02	28.87	+4.85	15.07	16.97	+1.90
	flame_steak	30.15	32.82	+2.67	16.50	17.37	+0.87
	sear_steak	27.08	33.50	+6.42	15.68	17.71	+2.03
Mean		24.23	26.79	+2.56	13.80	15.11	+1.31

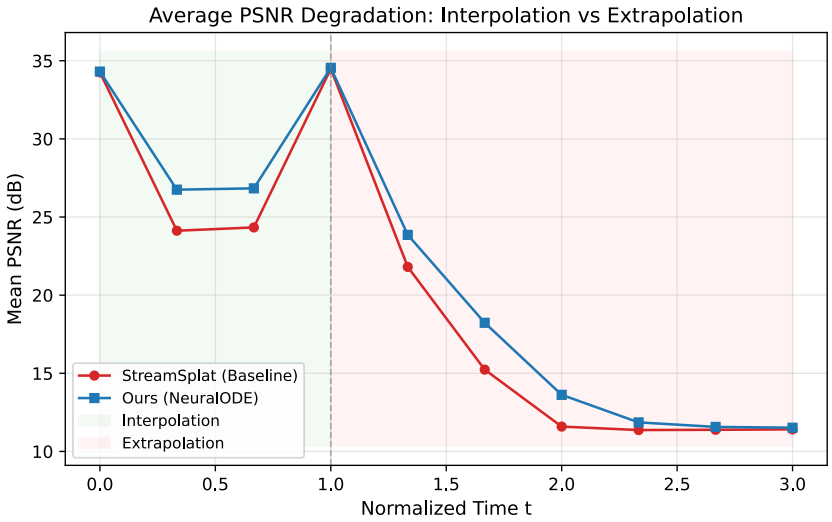


Figure 7: **Beyond-pair motion queries.** Peak signal-to-noise ratio (PSNR, dB) for Table 5's eight scenes. Appearance controls are disabled; $t = 1$ is the conditioning endpoint, not a verified training cutoff.

Beyond-pair gains are positive in all eight evaluated sequences, from 0.03 dB on `slice_banana` to 5.09 dB on `jump`. This rendering advantage persists beyond the conditioning pair, while declining absolute quality characterizes the temporal limits of both methods (Figure 7).

What the evidence establishes. Complete-cohort validation gains support sequence-specific scene adaptation. The cumulative component study motivates joint motion–appearance controls; attribute diagnostics and motion queries characterize the adapted representation. Appendix C.6 consolidates these conclusions and the separately evaluated continuation controls.

5 Conclusion

Snapshot-to-Flow specializes pretrained 4D scenes through residual dynamics. Source-referenced trajectory corrections and local and shared appearance controls improve sequence-specific rendering while retaining the predicted scene as a reference. Experiments across three datasets document gains, capacity trade-offs and temporal behavior. The contribution is an explicit scene-level adaptation formulation; integration-specific superiority and independently verified 3D accuracy are outside the conclusions established here.

AI USE STATEMENT

Writing assistance. We used generative AI tools to assist with manuscript revision and language polishing, including improving clarity, organization, and readability.

REPRODUCIBILITY STATEMENT

Section 3 describes sequence adaptation; Appendix B records its objective, modules and validation selection. The complete per-scene records in Appendix C.1 support the aggregate Novel gains in the abstract and Table 2, and distinguish them from the archived summaries in Table 1. Appendix C.2 specifies beyond-pair motion queries. Cross-scene implementation, controls and costs are separately recorded in Appendices A.1, B.1 and C.4. Appendix C.6 maps these records to the claims they support.

REFERENCES

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pp. 6571–6583, 2018.

Facebook Research. 3D Video Asset. GitHub release v1.0, 2022. URL `https://github.com/facebookresearch/Neural_3D_Video/releases/tag/v1.0`. Released 18 March 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *European Conference on Computer Vision*, pp. 624–642, 2022. doi: 10.1007/978-3-031-19781-9_36.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):257:1–257:16, 2025. doi: 10.1145/3763326.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Jiahui Lei, Yijia Weng, Adam W. Harley, Leonidas Guibas, and Kostas Daniilidis. MoSca: Dynamic Gaussian fusion from casual videos via 4D motion scaffolds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6165–6177, 2025.

Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D video synthesis from multi-view video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5521–5531, 2022.

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

Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. AMT: All-pairs multi-field transforms for efficient frame interpolation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9801–9810, 2023.

Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, and Jiahui Huang. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. In *Advances in Neural Information Processing Systems*, volume 38, pp. 151718–151746, 2025. doi: 10.52202/085713-5074. URL https://research.nvidia.com/labs/toronto-ai/bullet-timer/.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. MoVieS: Motion-aware 4D dynamic view synthesis in one second. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–306, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_MoVieS_Motion-Aware_4D_Dynamic_View_Synthesis_in_One_Second_CVPR_2026_paper.html.

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

Ziqi Lu, Heng Yang, Danfei Xu, Boyi Li, Boris Ivanovic, Marco Pavone, and Yue Wang. LoRA3D: Low-rank self-calibration of 3D geometric foundation models. In *International Conference on Learning Representations*, pp. 44356–44378, 2025. URL https://openreview.net/forum?id=LSp4KBhAom.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision*, pp. 800–809, 2024. doi: 10.1109/3DV62453.2024.00044.

Ziqiao Ma, Xuweiyi Chen, Shoubin Yu, Sai Bi, Kai Zhang, Ziwen Chen, Sihan Xu, Jianing Yang, Zexiang Xu, Kalyan Sunkavalli, Mohit Bansal, Joyce Chai, and Hao Tan. 4D-LRM: Large space-time reconstruction model from and to any view at any time. In *Advances in Neural Information Processing Systems*, volume 38, pp. 81712–81742, 2025. doi: 10.52202/085713-2730.

Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5437–5446, 2020.

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. *ACM Transactions on Graphics*, 40(6): 238:1–238:12, 2021. doi: 10.1145/3478513.3480487.

Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame interpolation for large motion. In *European Conference on Computer Vision*, pp. 250–266, 2022. doi: 10.1007/978-3-031-20071-7_15.

Jiawei Ren, Michal Jan Tyszkiewicz, Jiahui Huang, and Zan Gojcic. TokenGS: Decoupling 3D Gaussian prediction from pixels with learnable tokens. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15365–15375, 2026. URL https://research.nvidia.com/labs/toronto-ai/tokengs/.

Qiuhong Shen, Xuanyu Yi, Mingbao Lin, Hanwang Zhang, Shuicheng Yan, and Xinchao Wang. Seeing world dynamics in a nutshell. *arXiv preprint arXiv:2502.03465*, 2025. URL https://arxiv.org/abs/2502.03465.

Daniel Wang, Patrick Rim, Tian Tian, Dong Lao, Alex Wong, and Ganesh Sundaramoorthi. ODE-GS: Latent ODEs for dynamic scene extrapolation with 3D Gaussian Splatting. In *International Conference on Learning Representations*, pp. 131975–131997, 2026. URL https://openreview.net/forum?id=XlRbpFj3lJ.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21750–21760, 2025a. URL https://zju3dv.github.io/freetimegs/.

Yiming Wang, Lucy Chai, Xuan Luo, Michael Niemeyer, Manuel Lagunas, Stephen Lombardi, Siyu Tang, and Tiancheng Sun. Learning efficient fuse-and-refine for feed-forward 3D Gaussian Splatting. In *Advances in Neural Information Processing Systems*, volume 38, pp. 76311–76341, 2025b. doi: 10.52202/085713-2563.

Zike Wu, Qi Yan, Xuanyu Yi, Lele Wang, and Renjie Liao. StreamSplat: Towards online dynamic 3D reconstruction from uncalibrated video streams. In *International Conference on Learning Representations*, pp. 123114–123137, 2026. URL https://streamsplat3d.github.io/.

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. ReSplat: Learning recurrent Gaussian Splatting. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2510.08575.

Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Maximilian Igl, Apoorva Sharma, Peter Karkus, Danfei Xu, Boris Ivanovic, Yue Wang, and Marco Pavone. STORM: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *International Conference on Learning Representations*, pp. 50446–50465, 2025. URL https://openreview.net/pdf?id=M2NFWRPMUd.