RESTREAM: STATIC TOKEN REUSE FOR FAST STREAMING VOLUMETRIC VIDEO RECONSTRUCTION

Anonymous authors
Paper under double-blind review

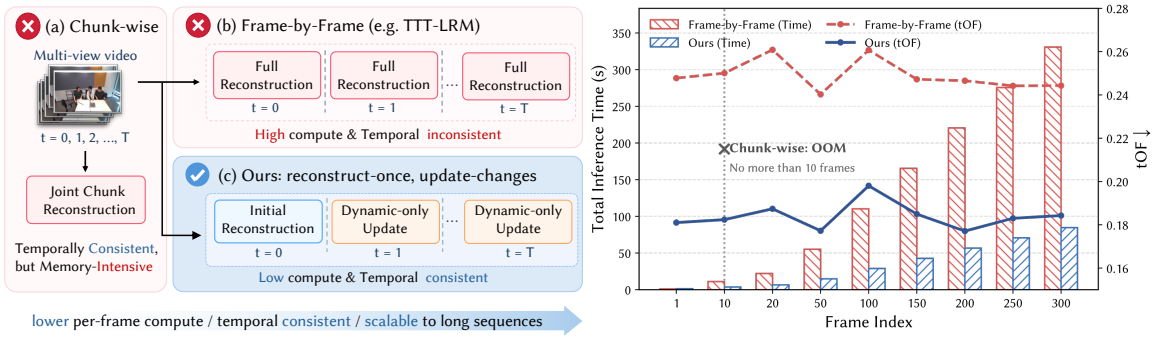


Figure 1: **Left:** Comparison of volumetric video reconstruction paradigms. (a) Chunk-wise methods jointly reconstruct multiple timestamps. (b) Frame-by-frame methods independently reconstruct each timestamp. (c) ReStream reuses static representations from a reference frame and selectively updates dynamic tokens for efficient streaming reconstruction. **Right:** Inference time across sequence lengths and temporal consistency measured by tOF (↓), using six input views at 960×720 resolution.

ABSTRACT

Volumetric video reconstruction recovers dynamic 3D scenes from synchronized multi-view videos. Although Large Reconstruction Models (LRMs) enable feed-forward reconstruction, their frame-by-frame application repeatedly processes unchanged scene content, incurring substantial redundant computation and hindering low-latency streaming reconstruction. We present ReStream, a training-free framework that exploits temporal redundancy through **static token reuse**. Our key observation is that complementary subsets of image tokens can be processed independently and their outputs assembled with negligible degradation in reconstruction quality, revealing substantial decomposability of multi-view reconstruction at the token level. Building on this observation, ReStream performs full reconstruction only at the reference timestamp and selectively reuses the resulting token representations across subsequent frames. For each new timestamp, dynamic tokens are identified by comparing current and reference patch embeddings and updated together with a small subset of static tokens that provide spatial context, while the remaining static representations are directly reused from the reference cache. This design concentrates computation on scene changes rather than repeatedly processing the entire scene. ReStream serves as a plug-and-play strategy for pretrained attention- and TTT-based reconstruction models, requiring neither retraining nor additional temporal modules. Experiments across diverse multi-view video benchmarks demonstrate substantial inference acceleration while maintaining comparable reconstruction quality and improved temporal consistency.

1 INTRODUCTION

Volumetric video reconstruction recovers dynamic 3D scenes from synchronized multi-view videos, typically captured by fixed camera rigs, enabling applications such as immersive telepresence, sports broadcasting, and virtual reality Burdea & Coiffet (2003); Jin et al. (2023); Rauschnabel et al. (2022);

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.

Speicher et al. (2019). Many existing methods rely on iterative per-scene optimization of neural radiance fields or 3D Gaussian representations Kerbl et al. (2023). The resulting computational cost poses a major bottleneck for low-latency streaming, particularly as image resolution and camera count increase. In this work, we investigate **fast streaming reconstruction** by extending pretrained feed-forward 3D reconstruction models to continuous multi-view video.

Recent large reconstruction models (LRMs), including DUSt3R Wang et al. (2024), VGGT Wang et al. (2025a), LRM Hong et al. (2023), LVSM Jin et al. (2024), ZipMap Jin et al. (2026), and DepthSplat Xu et al. (2025a), have made substantial progress in feed-forward 3D reconstruction. These models enable end-to-end reconstruction and generalize to unseen scenes. **However, most LRMs are designed for static scenes. Extending them to dynamic scenes remains non-trivial.** A straightforward approach is *frame-by-frame reconstruction*: the full network independently processes the multi-view observations at each timestamp (Fig. 1(b)). This approach ignores temporal redundancy and repeatedly reconstructs unchanged scene content, accumulating substantial computational overhead over long sequences. Alternatively, methods such as Nopo4D Balice et al. (2026) incorporate temporal modeling to jointly reconstruct multiple timestamps within a chunk (Fig. 1(a)). Such *chunk-wise reconstruction* promotes temporal consistency, but its memory requirements increase with the number of jointly processed timestamps. The available GPU memory therefore limits the temporal extent of each chunk, preventing joint reconstruction over arbitrarily long sequences.

Our key observation is that feed-forward multi-view reconstruction exhibits substantial redundancy at the patch level. As illustrated in Fig. 2, we partition image tokens into complementary subsets, process them independently, and assemble their outputs for rendering. Across different partition ratios, this procedure yields reconstruction quality comparable to full-token inference. This **patch-level decomposability** shows that reconstruction can be performed over selective token subsets rather than requiring all image tokens to interact simultaneously, opening up opportunities for more efficient and parallel reconstruction.

This observation motivates selective token updates for volumetric video reconstruction. With camera viewpoints fixed, static regions remain largely unchanged across frames, allowing their reconstructed features to be cached and reused. Subsequent frames therefore require recomputation only for dynamic tokens, supplemented by a small subset of static tokens as spatial anchors. These anchors provide context for dynamic reconstruction, while cached features preserve the static scene representation. This strategy reduces per-frame computation by restricting token interaction to dynamic regions and their supporting anchors.

Based on this insight, we introduce ReStream, a training-free framework for **Static Token Reuse for Fast Streaming Volumetric Video Reconstruction**. As shown in Fig. 1(c), ReStream processes all tokens at a reference timestamp and caches the resulting features. For each subsequent timestamp, differences between current and reference patch embeddings identify the tokens requiring updates. These dynamic tokens are processed jointly with a small subset of static tokens that serve as spatial anchors. The updated dynamic features are then combined with cached static features to reconstruct the current scene, avoiding repeated interaction over the full token set.

Our static token reuse strategy supports both intra-view and cross-view interactions in attention- and TTT-based reconstruction models. ReStream therefore serves as a training-free, plug-and-play extension to pretrained feed-forward reconstruction models. Extensive experiments across diverse benchmarks and reconstruction backbones demonstrate substantial inference speedups while maintaining comparable reconstruction quality. Our contributions are summarized as follows:

(1) We identify and analyze **patch-level decomposability** in feed-forward multi-view reconstruction, providing an empirical basis for static token reuse.

(2) Building on this observation, we propose ReStream, a fast streaming reconstruction framework that exploits patch-level decomposability along the temporal dimension through **static reuse, dynamic token updates, and static anchoring**.

(3) We demonstrate the plug-and-play applicability of ReStream across attention- and TTT-based reconstruction backbones, achieving substantial inference speedups while maintaining comparable reconstruction quality on diverse multi-view video benchmarks.

2 Related Works

2.1 Volumetric Video Reconstruction

Volumetric video reconstructs dynamic 3D scenes from synchronized multi-view observations, enabling free-viewpoint rendering over time. Typical capture systems employ multiple calibrated cameras at fixed viewpoints to continuously record scene dynamics. Early approaches rely on geometric priors such as shape-from-silhouette Ahmed et al. (2008), deformable templates Carranza et al. (2003), or depth-assisted fusion Newcombe et al. (2015), while subsequent neural methods improve reconstruction and rendering quality through learned scene representations Bozic et al. (2020); Lombardi et al. (2019). Recent neural and Gaussian representations further advance dynamic scene reconstruction by explicitly modeling temporal variations. Methods such as HexPlane Cao & Johnson (2023), K-Planes Fridovich-Keil et al. (2023), Deformable-3DGS Yang et al. (2024), SCGS Huang et al. (2023), 4DGaussian Wu et al. (2024), and 4DGS Yang et al. (2023) represent scene dynamics through deformation or spatio-temporal representations. For synchronized multi-view capture, SpaceTimeGS Li et al. (2024), Swift4D Wu et al. (2025b), LongVolCap Xu et al. (2024), and FreeTimeGS Wang et al. (2025c) further exploit dense multi-view observations for high-quality volumetric video reconstruction. Despite impressive rendering quality, these approaches generally rely on scene-specific optimization, motivating recent efforts toward direct feed-forward reconstruction.

2.2 Feed-Forward Reconstruction

Feed-forward reconstruction has emerged as an efficient alternative to conventional per-scene optimization. Geometry-oriented models such as DUSt3R Wang et al. (2024), VGGT Wang et al. (2025a), Stream3R Lan et al. (2025), Fast3R Yang et al. (2025), TTT3R Chen et al. (2025b), StreamVGGT Zhuo et al. (2026), π^3 Wang et al. (2025d), and DepthAnything3 Lin et al. (2025) directly infer 3D geometry from images using learned geometric priors. In parallel, feed-forward novel-view synthesis methods, including GS-LRM Zhang et al. (2024b), PixelSplat Charatan et al. (2024), MVSplat Chen et al. (2024), DepthSplat Xu et al. (2025a), LVSM Jin et al. (2024), Long-LRM Ziwen et al. (2025), TTT-LRM Wang et al. (2026), Rayzer Jiang et al. (2025); Zhao et al. (2026), and ZipMap Jin et al. (2026), directly predict renderable scene representations from multi-view observations. Despite architectural differences, these models typically rely on intensive token interaction, through attention or TTT-style mechanisms, to aggregate spatial and multi-view information.

Recent works further extend feed-forward reconstruction to dynamic scenes, including MonST3R Zhang et al. (2024a), CUT3R Wang et al. (2025b), PAGE-4D Zhou et al. (2025), VGGT-4D Hu et al. (2025), 4DGT Xu et al. (2025b), DVGGT Chen et al. (2025a), L4GM Ren et al. (2024), and UFO-4D Hur et al. (2026). Many of these methods address reconstruction from monocular observations, while feed-forward reconstruction of synchronized multi-view videos remains relatively underexplored. Among methods targeting multi-view video, SplatVoxel Wang et al. (2025e) enables history-aware streaming by propagating and fusing historical Gaussian representations with current observations, without requiring training on dynamic scenes, whereas Nopo4D Balice et al. (2026) jointly reconstructs multiple timestamps through explicit temporal modeling. Our work explores a complementary direction: reducing redundant computation within pretrained reconstruction backbones through token-level temporal reuse. Specifically, under fixed-view capture, we cache intermediate representations of temporally stable patches and selectively recompute dynamic tokens together with sparse static anchors. This strategy supports attention- and TTT-based backbones without introducing additional temporal networks or requiring video-specific retraining.

3 Method

3.1 Problem Formulation and Observation

Following the standard fixed-view capture setting widely adopted in volumetric video reconstruction Wang et al. (2023); Wu et al. (2025a); Xu et al. (2023a; 2024), we consider a dynamic scene captured by V synchronized cameras with fixed viewpoints over T timestamps, denoted as $\mathcal{I} = \{\mathbf{I}_t^v\}_{t=0,v=1}^{T-1,V}$. Such a capture setup provides natural spatial alignment across timestamps,



Figure 2: **Patch-level reconstruction observation.** Independently processing complementary token subsets (A and B) largely preserves reconstruction quality compared with full-token interaction. Left: qualitative comparison at $r = 0.8$. Right: PSNR versus the Part-A ratio r.

allowing temporally stable scene content to be identified and reused. Given a pretrained feed-forward multi-view reconstruction model, ReStream extends its frame-wise reconstruction capability to efficient volumetric video inference through temporal representation reuse.

Patch-level Reconstruction Observation. We observe that multi-view reconstruction can operate at a finer, patch-level granularity without requiring *all image tokens* to interact simultaneously. To verify this, let $\mathcal{X}$ denote the complete set of image tokens. We partition $\mathcal{X}$ into two complementary subsets $\mathcal{X}_A$ and $\mathcal{X}_B$ according to a ratio r, such that $\mathcal{X} = \mathcal{X}_A \cup \mathcal{X}_B$, $\mathcal{X}_A \cap \mathcal{X}_B = \varnothing$, and $|\mathcal{X}_A| = r|\mathcal{X}|$. Instead of jointly processing all tokens as $\mathcal{B}(\mathcal{X})$, we independently process the two subsets and assemble their outputs:

$$\hat{\mathcal{X}} = \mathrm{Assemble}(\mathcal{B}(\mathcal{X}_A), \mathcal{B}(\mathcal{X}_B)) \,. \tag{1}$$

As shown in Fig. 2, the resulting reconstruction remains highly consistent with full-token interaction across different partition ratios, with only marginal PSNR degradation. This *patch-level decomposability* suggests that accurate reconstruction can be achieved through selective interaction among token subsets. For volumetric video, we exploit this property temporally by reusing established static representations and reconstructing only dynamic tokens, with a small number of static tokens serving as temporary spatial anchors.

Beyond the temporal reuse explored in this work, patch-level decomposability may enable more general forms of selective computation for multi-view reconstruction. For example, different patch subsets could be processed independently or in parallel and subsequently assembled, potentially reducing the latency and computational cost of full-scene reconstruction.

3.2 Static-Dynamic Token Decoupling

We first determine which patch representations can be reused and which require an update. Since the cameras remain fixed, corresponding image regions are naturally aligned over time, allowing temporal changes to be identified directly in the patch embedding space. Let $\mathcal{E}$ denote the patch embedding layer and $\mathbf{X}_t^v = \mathcal{E}(\mathbf{I}_t^v) \in \mathbb{R}^{P \times C}$ denote the patch embeddings of view v at timestamp t. For each patch, we measure its variation relative to the reference timestamp and obtain a binary dynamic mask through thresholding:

$$\Delta_{t,p}^v = \left\| \mathbf{x}_{t,p}^v - \mathbf{x}_{0,p}^v \right\|_2 , \qquad m_{t,p}^v = \mathbb{1}\left[\Delta_{t,p}^v > \tau\right] . \tag{2}$$

where p denotes the patch index, $\mathbf{x}_{0,p}^v$ denotes the reference-frame token at patch p in view v, and $\tau = 0.02$ is used by default. According to the binary mask, patches with $m_{t,p}^v = 1$ form the dynamic token subset $\mathcal{X}_{D,t}$, while the remaining patches form the static token subset $\mathcal{X}_{S,t}$. This partition converts temporal redundancy into explicit computational reuse. Tokens in $\mathcal{X}_{S,t}$ correspond to temporally stable content whose representations have already been established at the reference timestamp and can therefore be directly reused, whereas tokens in $\mathcal{X}_{D,t}$ capture current scene changes and are forwarded for dynamic reconstruction.

3.3 Static Reuse and Dynamic Token Update

As shown in Fig. 3, our streaming inference operates at the patch level. Instead of repeatedly performing full interaction over all tokens, each subsequent timestamp reuses static representations from the reference frame and selectively updates only dynamic content.

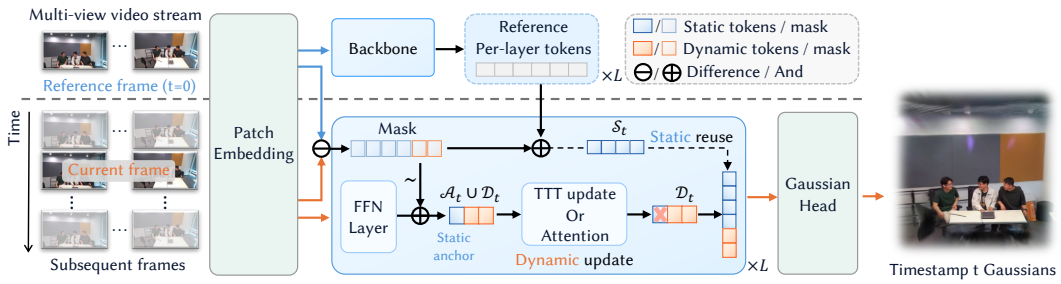


Figure 3: **Overview of ReStream**. We derive the static–dynamic mask from patch-level feature differences between the reference and current frames. Static patches directly reuse reference-frame representations, while dynamic tokens are selectively updated together with static anchors, enabling efficient and temporally consistent streaming reconstruction.

Static Anchoring. Although only dynamic tokens require new representations, **processing them in isolation may provide insufficient spatial context for accurate reconstruction**. We therefore randomly sample a small subset $\mathcal{X}_{0,A} \subset \mathcal{X}_{0,S}$ as *static anchors*, where $|\mathcal{X}_{0,A}| = \eta|\mathcal{X}_{0,S}|$ and $\eta = 0.1$ by default. These anchors interact with dynamic tokens to provide stable geometric references inherited from the reference reconstruction. Importantly, they serve only as temporary context: their updated outputs are discarded after interaction, while the corresponding static representations continue to reuse the cached reference features. This introduces only limited additional computation while preserving the established static scene representation. See Appendix Tab. 6 for the static anchoring ablation.

Incremental Token Interaction. At the reference timestamp, all tokens pass through the complete interaction block, and the resulting representations are cached for subsequent frames. For each timestamp $t > 0$, the static tokens $\mathcal{X}_{t,S}$ directly reuse their corresponding reference representations $\mathcal{X}_{0,S}$, while the dynamic tokens $\mathcal{X}_{t,D}$ are processed together with the sampled static anchors $\mathcal{X}_{t,A}$. Formally,

$$\mathcal{X}_t = \text{Assemble}\Big(\mathcal{X}_{0,S} \,,\, \big[\mathcal{B}\big(\mathcal{X}_{t,D} \,,\, \mathcal{X}_{0,A} \big)\big]_{\mathcal{X}_{t,D}} \Big). \tag{3}$$

Dynamic Update · Static Anchoring · Skip & Reuse

Here, $\mathcal{B}$ denotes a generic token interaction block, and $[\cdot]_{\mathcal{X}_{t,D}}$ retains only the outputs corresponding to tokens in $\mathcal{X}_{t,D}$, such that the anchor outputs are discarded after interaction. The resulting representation combines cached static features with newly updated dynamic features. In practice, static tokens typically constitute around 80% of all tokens (As shown in Appendix Fig.8), allowing ReStream to bypass most redundant computation and focus computation on scene changes.

Interaction Mechanisms. The same patch-level computation strategy applies to different interaction operators. For **attention-based blocks**, attention is evaluated for dynamic tokens together with static anchors, while the remaining static outputs are reused from the reference timestamp. For **TTT-style blocks**, dynamic tokens and static anchors participate in fast-weight adaptation, restricting the expensive *Update* operation to the effective update set before *Apply*. Thus, both mechanisms implement the same streaming principle: **reuse established static representations and allocate new computation to scene changes**. More deatils in Appendix-Sec.A.1.

4 Experiments

We conduct extensive experiments to evaluate the effectiveness, efficiency, and generality of ReStream. Our experiments are designed to answer four key questions: **(1)** How well does ReStream preserve reconstruction quality under streaming inference? **(2)** Does selective reconstruction preserve temporal consistency over long sequences? **(3)** How much acceleration does ReStream achieve, and where does the efficiency gain come from? **(4)** Can the proposed streaming strategy generalize across

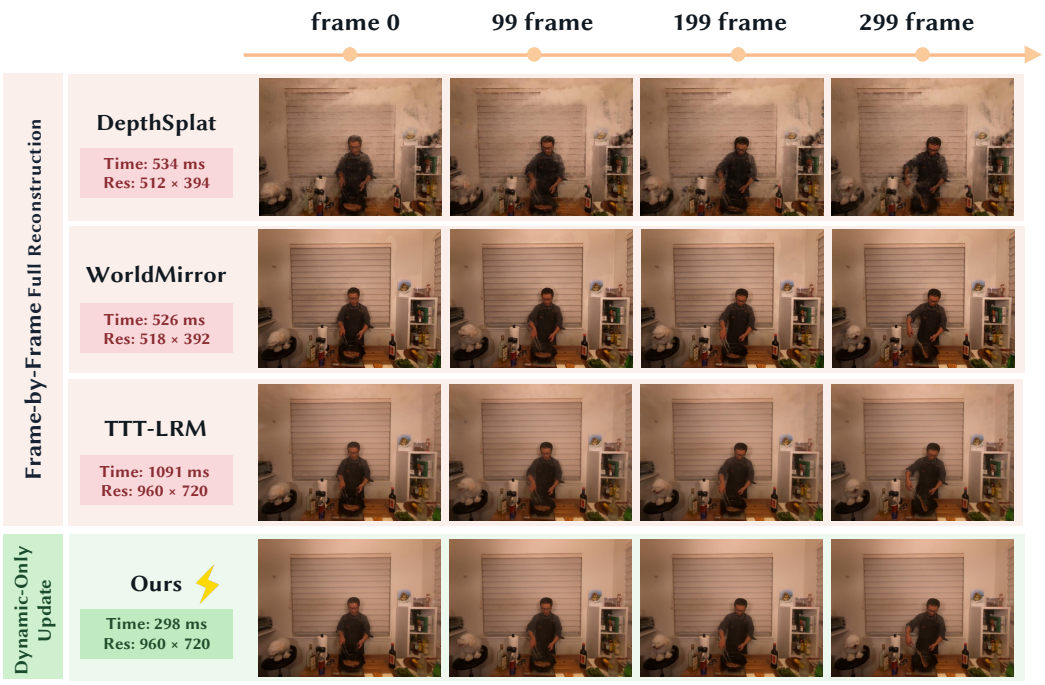


Figure 4: **Qualitative comparison.** Our approach achieves inference acceleration with no noticeable degradation in visual quality. More visual results are shown in the supplementary video.

Table 1: **Quantitative comparison on different datasets.** We report PSNR (↑), SSIM (↑), LPIPS (↓), Time (↓), and Speedup (↑). Six uniformly spaced camera views are used as input for all methods, and all metrics are averaged over all frames.

Dataset	TTTLRM			Ours(TTT backbone)			WorldMirror			DepthSplat		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM ↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ZJU-SelfCap	20.72	0.715	0.270	20.61	0.707	0.280	19.17	0.603	0.300	17.07	0.562	0.323
MeetRoom	30.14	0.932	0.187	30.06	0.928	0.195	26.91	0.906	0.137	20.04	0.719	0.345
ENeRF	23.99	0.528	0.325	23.85	0.524	0.334	25.11	0.660	0.272	10.79	0.162	0.637
N3DV	29.23	0.896	0.157	29.12	0.895	0.158	27.81	0.873	0.153	22.46	0.730	0.323
Mean	26.02	0.768	0.235	25.91	0.764	0.242	24.75	0.760	0.216	17.59	0.543	0.407
Resolution↑	960×720			960×720			518×392			512×394		
tOF ↓	0.282			0.192			0.292			1.471		
Time(s) ↓	$1.1_{\times 1}$			$0.3_{\times 3.6\downarrow}$			0.5			0.5		

different feed-forward reconstruction backbones and interaction mechanisms? We further conduct comprehensive ablations to analyze the key components and design choices of ReStream.

4.1 Experimental Setup

We evaluate ReStream on four synchronized multi-view dynamic benchmarks: N3DV Li et al. (2022b), MeetRoom Li et al. (2022a), SelfVolCap Xu et al. (2023b), and ENeRF Lin et al. (2022), covering diverse scene characteristics and motion patterns. Unless otherwise specified, we use six input views at a resolution of 960 × 720 and adopt TTT-LRM Wang et al. (2026) as the default backbone. To evaluate the generality of ReStream, we further integrate it into an attention-based reconstruction model Liu et al. (2025). All pretrained model parameters remain unchanged, with the original models performing standard frame-wise inference as baselines. We report PSNR, SSIM, and LPIPS for reconstruction quality, tOF Chu et al. (2020) for temporal consistency, and per-frame latency for inference efficiency. Runtime is averaged over multiple runs after GPU warm-up, excluding data loading. All experiments are conducted on a single NVIDIA A100 GPU with 80 GB of memory.

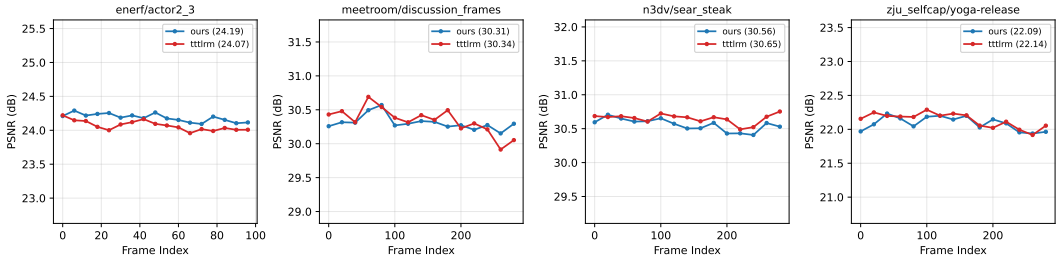


Figure 5: **Temporal reconstruction quality on four dynamic scenes.** Compared with the TTT-LRM backbone, ReStream maintains comparable and stable PSNR over long video sequences while achieving a 3.6× inference speedup (Tab.1). The above results use frame 0 as the sole reference frame for all subsequent 299 frames.

4.2 Main Experiment

Q1: How Well Does ReStream Preserve *Quality*

We compare ReStream with full frame-by-frame inference using the same pretrained backbone and input configuration. As shown in Tab. 1, ReStream substantially improves inference efficiency while preserving comparable reconstruction quality across diverse dynamic scenes. These results demonstrate that selective dynamic reconstruction effectively eliminates redundant computation without sacrificing reconstruction fidelity.

Qualitative Comparison. Fig. 4 presents qualitative comparisons on representative dynamic scenes. ReStreampreserves fine scene structures and appearance details while maintaining reconstruction quality in both static and dynamic regions. In particular, the reconstructed dynamic content remains consistent with full frame-by-frame inference despite only a subset of tokens being recomputed at subsequent timestamps.

Q2: Does ReStream Preserve *Temporal Consistency*?

Beyond per-frame reconstruction quality, we evaluate whether ReStream preserves temporal consistency over long sequences. Following TecoGAN Chu et al. (2020), we adopt temporal optical flow error (tOF), which measures the discrepancy between the inter-frame motion of reconstructed and ground-truth sequences:

$$\text{tOF} = \frac{1}{|\mathcal{P}||\Omega|} \sum_{t \in \mathcal{P}} \sum_{\mathbf{x} \in \Omega} \left\| \mathcal{F}(\hat{I}_{t-1}, \hat{I}_t)(\mathbf{x}) - \mathcal{F}(I_{t-1}, I_t)(\mathbf{x}) \right\|_2, \tag{4}$$

where $\mathcal{P}$ denotes the evaluated consecutive-frame pairs and Ω denotes the image domain. A lower tOF indicates better agreement with the ground-truth temporal motion.



(a) Input renderings (b) Ours (b) baseline

Figure 6: **Temporal consistency comparison using two-frame residual maps.** Lower residuals indicate better temporal consistency. Compared with TTT-LRM, ReStream produces substantially lower residuals and fewer temporal artifacts across different scenes.

As shown in Fig. 5, ReStream maintains stable reconstruction quality over long sequences without progressive degradation. Fig. 6 further shows reduced frame-to-frame fluctuations, particularly in temporally stable regions. Consistently, ReStream achieves substantially lower tOF than frame-by-frame reconstruction in Tab. 1, demonstrating better preservation of ground-truth temporal motion. These results show that reusing established static representations improves temporal stability while selective reconstruction preserves dynamic scene changes.

Q3: How Much *Acceleration* Does ReStream Achieve?

We next analyze the computational benefits of incremental streaming reconstruction. Unlike **full** frame-by-frame inference, whose computation repeatedly covers the complete token set, the cost of ReStream after reference initialization is primarily determined by the amount of scene content that changes.

Overall Acceleration. Tab. 1 reports the inference efficiency across different datasets. ReStreamconsistently reduces per-frame latency and improves throughput over **full** frame-by-frame reconstruction. The acceleration is particularly pronounced when large portions of the scene remain temporally stable, since their representations can be directly reused without repeated token interaction.

Where Does the Speedup Come From? To identify the source of the computational gain, we profile the reconstruction pipeline at the block level. As shown in Tab. 2, the major reduction originates from token interaction blocks, where computation over temporally stable tokens is avoided. Network components unaffected by our streaming strategy remain nearly unchanged, while dynamic-region identification and token indexing introduce only limited overhead. This confirms that the acceleration primarily results from eliminating redundant token interaction rather than modifying the reconstruction head or reducing the input resolution.

Table 2: Runtime comparison with TTT-LRM on the N3DV dataset. Runtime includes only the optimized components, while other unchanged components are excluded.

	TTT-LRM	**Ours**
Frame-Attention	368.3	114.8
Fast Weight	613.0	153.2
Overall (ms/frame)	981.3	268.0
Speedup	1.00×	**3.66**×

Q4: Does ReStream *Generalize* across Reconstruction Backbones?

A key property of ReStream is that its selective reconstruction principle is independent of a specific token interaction operator. We therefore integrate our streaming strategy into representative feed-forward reconstruction backbones using different interaction mechanisms while keeping all pretrained parameters unchanged. As shown in Tab. 3, ReStream consistently improves inference efficiency across different reconstruction architectures while preserving reconstruction quality. In particular, the same static-reuse and dynamic-update principle applies to both conventional attention and TTT-style interaction. For attention-based blocks, interaction is evaluated only for dynamically reconstructed tokens together with temporary static anchors, while the remaining static representations are reused. For TTT-style blocks, the expensive fast-weight update is similarly restricted to dynamic tokens and static anchors. Details in Appendix Sec.A.1.

Table 3: Generalization of ReStream to WorldMirror and TTT-LRM on N3DV, averaged over 300 frames.

Method	PSNR↑	SSIM↑	LPIPS↓	Time(s)↓	Speedup↑
WorldMirror	27.81	0.873	0.153	0.51	1.00×
WorldMirror + Ours	27.73	0.872	0.154	0.14	3.64×
TTT-LRM	29.23	0.896	0.157	1.10	1.00×
TTT-LRM + Ours	29.12	0.895	0.158	0.30	3.67×

Table 4: Effect of reference frame refresh interval on a 300-frame sequence. Runtime includes initialization and reference refreshes (Conducted on N3DV *coffee_martini*).

Refresh interval	Reconstruction quality			Runtime	
	PSNR ↑	SSIM ↑	LPIPS ↓	Total (s) ↓	Avg. (s/frame) ↓
1	26.26	0.858	0.181	327.4	1.09
5	26.26	0.858	0.183	135.8	0.45
10	26.24	0.858	0.183	111.3	0.37
20	26.22	0.857	0.183	99.7	0.33
300	26.18	0.856	0.185	88.7	0.30

E.g., interval = 1: full frame-wise inference; interval = 20: refresh every 20 frames.

4.3 Ablation Studies

Reference Frame Refresh Interval. We study how frequently the reference representation needs to be refreshed. A shorter refresh interval introduces more complete forward passes, whereas a longer interval maximizes temporal reuse. As shown in Tab. 4, ReStream maintains stable reconstruction quality over long temporal windows, demonstrating that the reference representation can be reused effectively across many subsequent timestamps. The refresh interval can therefore be adjusted according to the desired quality–efficiency operating point.

Impact of Threshold τ. We study how the threshold τ controls the trade-off between static token reuse and current-frame computation. As shown in Fig. 7, (1) increasing τ reduces the number of patches classified as dynamic, thereby improving computational efficiency; however, an excessively large τ may misclassify genuinely dynamic patches as static, leading to noticeable reconstruction quality degradation for dynamic scene reconstruction. Conversely, (2) decreasing τ preserves more patches as dynamic, which better maintains reconstruction quality but increases inference time. Within a moderate range of $\tau \in [0.01, 0.1]$, ReStream achieves a favorable quality–efficiency trade-off, maintaining high reconstruction quality while providing multi-fold inference acceleration.

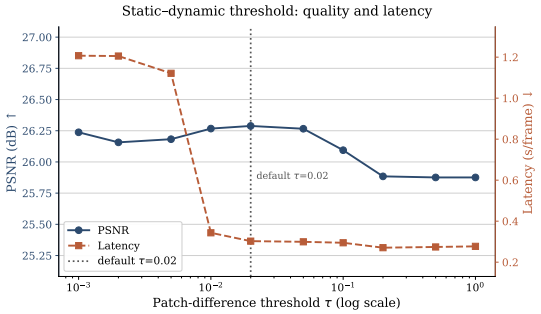


Figure 7: Effect of the static–dynamic threshold τ on reconstruction quality and inference latency. (Conducted on *coffee_martini* and *discussion* scene)

5 Discussion and Limitations

Our study reveals that multi-view reconstruction can operate at a finer, patch-level granularity without requiring all image tokens to interact simultaneously. We exploit this property temporally by reusing established static representations and reconstructing only dynamic tokens, enabling efficient and temporally consistent streaming inference while largely preserving the performance of the original model. Beyond temporal reuse, this patch-level decomposability suggests broader opportunities for selective or parallel token processing, potentially inspiring more efficient feed-forward multi-view reconstruction architectures.

The computational benefit of ReStream depends on the amount of temporal redundancy. As dynamic content increases, more patches require current-frame reconstruction, and the computational cost gradually approaches full frame-by-frame inference. As shown in Fig. 7, an excessively small threshold (e.g., $\tau = 0.001$) classifies most patches as dynamic, substantially increasing inference cost without providing a corresponding improvement in reconstruction quality. Detailed complexity analysis is provided in the supplementary material (Appendix Sec.A.2).

6 Conclusion

We introduced ReStream, a training-free framework for efficient feed-forward reconstruction of multi-view dynamic scenes. Instead of repeatedly processing all scene content at every timestamp, our method decouples temporal computation into static reuse and dynamic updates: representations of stable regions are reused across time, while only dynamic tokens are selectively recomputed. This design reduces redundant computation without additional temporal modeling or retraining, and is compatible with both attention-based and TTT-style reconstruction backbones. Experiments on diverse dynamic benchmarks demonstrate substantial acceleration with minimal degradation in reconstruction quality. Our results suggest that temporal computation reuse is a simple yet effective principle for converting static reconstruction models into efficient streaming reconstructors.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We did not use generative AI tools for any other tasks requiring disclosure, including developing the research methodology, designing experiments, implementing the proposed method, or interpreting results; the remaining required disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools to assist with drafting and editing parts of the manuscript, improving its readability, and writing a small number of utility functions. The core method and implementation were developed by the authors. All AI-assisted text and code were reviewed and verified by the authors. We take full responsibility for the final content of this work, including text and code produced with the aid of generative AI.

REFERENCES

Naveed Ahmed, Christian Theobalt, Christian Rossl, Sebastian Thrun, and Hans-Peter Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2008. 3

Matteo Balice, Yanik Kunzi, Chenyangguang Zhang, Matteo Matteucci, Marc Pollefeys, and Sunghwan Hong. No pose, no problem in 4d: Feed-forward dynamic gaussians from unposed multi-view videos, 2026. URL https://arxiv.org/abs/2605.22190. 2, 3

Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7002–7012, 2020. 3

Grigore C Burdea and Philippe Coiffet. *Virtual reality technology*. John Wiley & Sons, 2003. 1

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 130–141, 2023. 3

Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003. 3

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19457–19467, 2024. 3

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, BING WANG, Guang Chen, et al. Feedforward 4d reconstruction for dynamic driving scenes using unposed images. 2025a. 3

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Ttt3r: 3d reconstruction as test-time training. *arXiv preprint arXiv:2509.26645*, 2025b. 3

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386. Springer, 2024. 3

Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020. 6, 7

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12479–12488, 2023. 3

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 2

Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction, 2025. URL https://arxiv.org/abs/2511.19971. 3

Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *arXiv preprint arXiv:2312.14937*, 2023. 3

Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. Ufo-4d: Unposed feedforward 4d reconstruction from two images. *arXiv preprint arXiv:2602.24290*, 2026. 3

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. *arXiv preprint arXiv:2505.00702*, 2025. 3

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024. 2, 3

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Holynski. Zipmap: Linear-time stateful 3d reconstruction via test-time training, 2026. URL https://arxiv.org/abs/2603.04385. 2, 3

Yili Jin, Kaiyuan Hu, Junhua Liu, Fangxin Wang, and Xue Liu. From capture to display: A survey on volumetric video. *arXiv preprint arXiv:2309.05658*, 2023. 1

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, 2023. 2

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer. *arXiv preprint arXiv:2508.10893*, 2025. 3

Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35:13485–13498, 2022a. 6

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5521–5531, 2022b. 6

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8508–8520, 2024. 3

Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia Conference Proceedings*, 2022. 6

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 3

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025. 6

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 3

Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 343–352, 2015. 3

Philipp A Rauschnabel, Reto Felix, Chris Hinsch, Hamza Shahab, and Florian Alt. What is xr? towards a framework for augmented and virtual reality. *Computers in human behavior*, 133:107289, 2022. 1

Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung W Kim, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024. 3

Maximilian Speicher, Brian D Hall, and Michael Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pp. 1–15, 2019. 2

Chen Wang, Hao Tan, Wang Yifan, Zhiqin Chen, Yuheng Liu, Kalyan Sunkavalli, Sai Bi, Lingjie Liu, and Yiwei Hu. tttlrm: Test-time training for long context and autoregressive 3d reconstruction. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2026. 3, 6

Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19706–19716, 2023. 3

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proc. of CVPR*, 2025a. 2, 3

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proc. of CVPR*, pp. 10510–10522, 2025b. 3

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proc. of CVPR*, pp. 20697–20709, 2024. 2, 3

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 *CVPR*, 2025c. URL https://zju3dv.github.io/freetimegs. 3

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025d. 3

Yiming Wang, Lucy Chai, Xuan Luo, Michael Niemeyer, Manuel Lagunas, Stephen Lombardi, Siyu Tang, and Tiancheng Sun. Splatvoxel: History-aware novel view streaming without temporal training, 2025e. URL https://arxiv.org/abs/2503.14698. 3

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20310–20320, 2024. 3

Jiahao Wu, Rui Peng, Jianbo Jiao, Jiayu Yang, Luyang Tang, Kaiqiang Xiong, Jie Liang, Jinbo Yan, Runling Liu, and Ronggang Wang. Localdygs: Multi-view global dynamic scene modeling via adaptive local implicit feature decoupling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9519–9529, 2025a. 3

Jiahao Wu, Rui Peng, Zhiyan Wang, Lu Xiao, Luyang Tang, Jinbo Yan, Kaiqiang Xiong, and Ronggang Wang. Swift4d: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=c1RhJVTPwT. 3

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16453–16463, 2025a. 2, 3

Zhen Xu, Sida Peng, Haotong Lin, Guangzhao He, Jiaming Sun, Yujun Shen, Hujun Bao, and Xiaowei Zhou. 4k4d: Real-time 4d view synthesis at 4k resolution. 2023a. 3

Zhen Xu, Tao Xie, Sida Peng, Haotong Lin, Qing Shuai, Zhiyuan Yu, Guangzhao He, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Easyvolcap: Accelerating neural volumetric video research. 2023b. 6

Zhen Xu, Yinghao Xu, Zhiyuan Yu, Sida Peng, Jiaming Sun, Hujun Bao, and Xiaowei Zhou. Representing long volumetric video with temporal gaussian hierarchy. *ACM Transactions on Graphics*, 43(6), November 2024. URL https://zju3dv.github.io/longvolcap. 3

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. *Advances in neural information processing systems*, 2025b. 3

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21924–21935, 2025. 3

Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 3

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

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. *arXiv preprint arxiv:2410.03825*, 2024a. 3

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024b. 3

Qitao Zhao, Hao Tan, Qianqian Wang, Sai Bi, Kai Zhang, Kalyan Sunkavalli, Shubham Tulsiani, and Hanwen Jiang. E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training. In *CVPR*, 2026. 3

Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. Page-4d: Vggt-4d perception via disentangled pose and geometry estimation. In *The Fourteenth International Conference on Learning Representations*, 2025. 3

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer, 2026. URL https://arxiv.org/abs/2507.11539. 3

Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 3

A Appendix

A.1 Acceleration for Different Token Interaction Mechanisms

Modern 3D foundation models typically consist of multiple token interaction stages, including intra-frame attention and global cross-view interaction implemented with either standard attention or TTT-style fast weights. Although these mechanisms differ in their interaction scope and implementation, ReStream applies the same principle of *static reuse and dynamic update* to each of them. We provide implementation details for the three common interaction mechanisms below.

Frame Attention. Frame attention performs token interaction independently within each input view. For the reference timestamp, we perform standard full attention and cache the key and value features of the static patches. At a subsequent timestamp t, we compute new queries, keys, and values only for the dynamic tokens $\mathcal{X}_{t,D}$. The cached static keys and values are then concatenated with the current dynamic keys and values to construct the attention context,

$$\mathbf{K}_t^{\text{stream}} = [\mathbf{K}_{0,S}; \mathbf{K}_{t,D}], \qquad \mathbf{V}_t^{\text{stream}} = [\mathbf{V}_{0,S}; \mathbf{V}_{t,D}]. \tag{5}$$

Only the current dynamic queries attend to this combined context:

$$\mathcal{X}_t = \text{Assemble}(\mathcal{X}_{t,D}, \mathcal{X}_{0,S}), \quad \mathcal{X}_{t,D} = \text{Attn}\left(\mathbf{Q}_{t,D}, \mathbf{K}_t^{\text{stream}}, \mathbf{V}_t^{\text{stream}}\right). \tag{6}$$

The static query outputs are skipped entirely and directly reuse their cached reference representations. Therefore, although dynamic queries can still access the complete static and dynamic context within the view, attention outputs are evaluated only for the much smaller set of dynamic queries. In our experiments, dynamic patches typically account for only around 10%–20% of all patches, substantially reducing the computation of frame attention.

Global Attention. Global attention aggregates tokens across all input views and becomes substantially more expensive as the number of image tokens increases. Unlike frame attention, retaining all static tokens in the interaction would still introduce considerable global computation. We therefore employ the static anchoring strategy described in the main paper. Specifically, we randomly sample a small subset of static tokens $\mathcal{X}_{0,A} \subset \mathcal{X}_{0,S}$, where $|\mathcal{X}_{0,A}| = \eta|\mathcal{X}_{0,S}|$ and $\eta = 0.1$ by default. The sampled static anchors are combined with the current dynamic tokens, and global attention is performed only over $\mathcal{X}_{t,D} \cup \mathcal{X}_{0,A}$:

$$\left[\widetilde{\mathcal{X}}_{t,D}, \widetilde{\mathcal{X}}_{t,A}\right] = \text{Attn}\left(\mathcal{X}_{t,D} \cup \mathcal{X}_{0,A}\right). \tag{7}$$

The static anchors provide geometric context from the reference reconstruction, while the dynamic tokens capture the current scene changes. After interaction, only the updated dynamic representations $\widetilde{\mathcal{X}}_{t,D}$ are retained, whereas $\widetilde{\mathcal{X}}_{t,A}$ is discarded. The complete representation at timestamp t is then assembled as

$$\mathcal{X}_t = \text{Assemble}\left(\mathcal{X}_{0,S}, \widetilde{\mathcal{X}}_{t,D}\right), \tag{8}$$

where all static positions, including the sampled anchor positions, directly reuse their cached reference representations. This substantially reduces the number of tokens participating in expensive global interaction while preserving static geometric context.

TTT Fast-Weight Interaction. For TTT-based reconstruction models, global token interaction is implemented through fast-weight adaptation rather than explicit pairwise attention. We apply the same selective interaction principle by restricting the expensive `Update` operation to the current dynamic tokens and sampled static anchors. Specifically, we sample $\mathcal{X}_{0,A} \subset \mathcal{X}_{0,S}$, with $|\mathcal{X}_{0,A}| = \eta|\mathcal{X}_{0,S}|$ and $\eta = 0.1$ by default, and perform fast-weight adaptation only over $\mathcal{X}_{t,D} \cup \mathcal{X}_{0,A}$:

$$\mathbf{W}_t = \texttt{Update}\left(\mathbf{W}_{\text{init}}, \mathcal{X}_{t,D} \cup \mathcal{X}_{0,A}\right). \tag{9}$$

The adapted fast weights are then applied to the same token subset to obtain the corresponding representations:

$$\left[\widetilde{\mathcal{X}}_{t,D}, \widetilde{\mathcal{X}}_{t,A}\right] = \texttt{Apply}\left(\mathbf{W}_t, \mathcal{X}_{t,D} \cup \mathcal{X}_{0,A}\right). \tag{10}$$

Only the updated dynamic representations $\widetilde{\mathcal{X}}_{t,D}$ are retained, while the outputs associated with the static anchors are discarded. The complete representation is subsequently assembled as

$$\mathcal{X}_t = \text{Assemble}\left(\mathcal{X}_{0,S}, \widetilde{\mathcal{X}}_{t,D}\right). \tag{11}$$

Therefore, static anchors participate only temporarily in fast-weight adaptation and application to provide geometric context, without introducing persistent updates to the cached static representations. Since both fast-weight adaptation and subsequent token processing operate on only a small fraction of the original tokens, ReStream substantially reduces the cost of TTT-style global interaction.

Summary. The three interaction mechanisms therefore exploit temporal redundancy in complementary ways. For frame attention, we preserve the complete within-view context by caching static keys and values, while evaluating queries only for dynamic patches. For global attention and TTT-style interaction, where processing all static tokens remains expensive, we retain all dynamic tokens but replace the complete static set with a small set of sampled static anchors. Despite their different implementations, all three cases follow the same principle: *reuse established static representations and allocate new computation primarily to scene changes*.

A.2 Complexity Analysis

Let N_S and N_D denote the numbers of static and dynamic patch tokens at each timestamp, respectively. The total number of tokens is therefore $N = N_S + N_D$. For simplicity, we omit the small computational overhead introduced by static anchors in the following analysis. Conventional frame-by-frame reconstruction processes all tokens independently at every timestamp, regardless of whether the corresponding scene content has changed. In contrast, ReStream reuses the representations of static patches and performs current-frame computation primarily on dynamic patches.

For attention-based interaction, conventional full-token processing has a quadratic complexity of

$$\mathcal{O}\big((N_S + N_D)^2\big). \tag{12}$$

By restricting current-frame interaction to dynamic patches, ReStream reduces the complexity to

$$\mathcal{O}\big(N_D^2\big) \tag{13}$$

for global attention. For intra-view frame attention, we retain the complete static context by caching static keys and values while evaluating new queries only for dynamic patches. The corresponding complexity is therefore reduced from $\mathcal{O}((N_S + N_D)^2)$ to

$$\mathcal{O}(N_D(N_S + N_D)). \tag{14}$$

For TTT-style interaction, the fast-weight adaptation scales approximately linearly with the number of participating tokens. The original model therefore requires

$$\mathcal{O}(N_S + N_D), \tag{15}$$

while ReStream reduces the current-frame adaptation to

$$\mathcal{O}(N_D). \tag{16}$$

Therefore, the computational cost of ReStream at subsequent timestamps is governed primarily by the number of dynamic patches rather than the total number of image patches. When static content dominates the scene, i.e., $N_D \ll N_S$, a substantial fraction of redundant computation can be eliminated. This analysis also explains why the acceleration of ReStream increases as the proportion of temporally stable content grows.

A.3 More experiment

Scaling with Sequence Length. We further evaluate sequences of increasing temporal length to study the scalability of ReStream. Full frame-by-frame inference repeatedly incurs complete reconstruction cost at every timestamp, whereas ReStream performs full reconstruction only for reference initialization and amortizes this cost over subsequent frames. As shown in Tab. 5, the initialization overhead becomes increasingly negligible as the sequence grows, while subsequent streaming inference maintains substantially lower per-frame computation. Consequently, the computational advantage of ReStream becomes more pronounced for long volumetric videos.

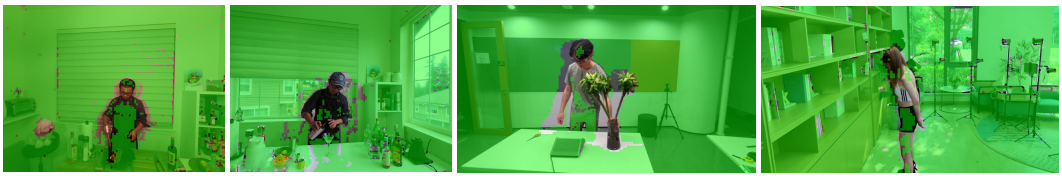


Figure 8: visualization of static and dynamic patches

Table 5: **Scaling with sequence length.** Runtime and speedup of ReStream compared with frame-by-frame **full** reconstruction (baseline: TTTLRM) over sequences of increasing length.

	Total time (s)↓		Streaming efficiency	
Frames	TTT-LRM	Ours	Avg. (s/frame)↓	Speedup↑
1	1.09	1.09	1.091	1.00×
5	5.46	2.29	0.458	2.38×
10	10.91	3.69	0.369	2.95×
20	21.83	6.56	0.328	3.32×
50	54.57	15.59	0.311	3.49×
100	109.13	29.99	0.299	3.63×
300	327.39	88.69	0.295	3.69×

Impact of Static Anchoring. We further investigate the role of static anchors by varying the proportion of static tokens participating in current-frame interaction. As shown in Tab. 6, processing dynamic tokens alone (0%) leads to degraded reconstruction quality, indicating that dynamic tokens still benefit from spatial context provided by the established static scene. Introducing only a small fraction of static anchors substantially recovers the reconstruction quality, while further increasing the anchor ratio provides diminishing improvements but incurs additional computation. This demonstrates that a sparse set of static tokens is sufficient to provide effective geometric context for dynamic reconstruction. **We therefore use a 10% static anchor ratio by default, providing a favorable balance between reconstruction quality and inference efficiency.**

A.4 Evaluation Protocol and Implementation Details

Dataset coverage. We evaluate ReStream on N3DV, MeetRoom, ZJU-SelfCap, and ENeRF. For each dataset, evaluation includes every frame of every scene, rather than a selected subset of scenes or timestamps. The same evaluation coverage is used when comparing ReStream with its corresponding full frame-by-frame reconstruction baseline.

Input and evaluation views. For every scene, we use camera view 0 as the target view for novel-view synthesis evaluation. Six uniformly sampled camera views provide the synchronized multi-view inputs at each timestamp. Input images are cropped and resized to a resolution of 960 × 720 before being passed to the reconstruction model. Rendered results are evaluated against the ground-truth observations from the designated target camera.

Shared parameter configuration. We use a single, fixed ReStream parameter configuration across all scenes and datasets. No scene-specific or dataset-specific parameter tuning is performed. This shared configuration is used throughout the evaluation to assess the method under a consistent operating setting.

Comparison with the original backbone. To isolate the effect of selective token updates and static representation reuse, ReStream and its corresponding full frame-by-frame baseline use the same pretrained backbone, input views, image preprocessing, and evaluation frames. The pretrained model parameters remain unchanged.

Table 6: **Ablation on the static anchor ratio.** A small fraction of static anchors provides sufficient spatial context for dynamic token reconstruction, while larger ratios introduce additional computation with diminishing quality gains. Results are evaluated on the N3DV dataset.

Anchor Ratio	PSNR ↑	SSIM ↑	LPIPS ↓	Time (s/frame) ↓
0%	28.48	0.882	0.167	**0.29**
5%	28.96	0.892	0.160	0.30
10%	29.12	0.895	0.158	0.30
20%	29.17	0.895	0.158	0.35
100%	**29.23**	**0.896**	**0.157**	1.10