S^{2}4D: GENERALIZABLE S*TREAMING RECONSTRUCTION FOR S*PARSE-VIEW DYNAMIC SCENES

Anonymous authors
Paper under double-blind review

ABSTRACT

High-quality 4D reconstruction from multi-view videos is essential to immersive volumetric video. However, existing dynamic novel-view synthesis methods often require densely sampled camera arrays and scene-specific optimization, resulting in costly capture and substantial reconstruction latency. We propose S^{2}4D, a generalizable sparse-view streaming framework for low-latency dynamic 4D reconstruction. Its Sparse-view Anchor-driven Motion Network (SAM-Net) extracts sparse-view 2D motion features and lifts them to compact 3D Gaussian anchors via visibility-depth aware aggregation. Interleaved Refinement Streaming performs feed-forward motion propagation on intermediate frames and periodically applies lightweight local refinement. Max-points bounded recalibration controls Gaussian growth and helps limit sparse-view overfitting. A sparse-view flow consistency (SFC) loss regularizes view-specific motion through cross-view geometry to improve sparse-view generalization without inference cost. Experiments on in-domain and cross-domain datasets demonstrate state-of-the-art PSNR among the compared streaming methods, an average per-frame reconstruction time of 0.62 seconds on N3DV, and effective generalization to unseen scenes under the evaluated sparse-camera settings and a fixed motion network at deployment.

1 INTRODUCTION

Learned view synthesis supports free-viewpoint and immersive media: Stereo Magnification introduced multiplane synthesis (Zhou et al., 2018), and Immersive Light Field Video demonstrated volumetric playback (Broxton et al., 2020); broader XR and streaming requirements are surveyed by Martin et al. (2022) and Bentaleb et al. (2025). 3DGS enables real-time explicit rendering (Kerbl et al., 2023), with later work refining primitive ordering (Wang et al., 2026d). Dynamic extensions model deformation (Wu et al., 2024a), spacetime Gaussians (Li et al., 2024), or high-resolution performances (Jiang et al., 2024; Xu et al., 2024), but typically require **dense synchronized cameras** and **scene-specific optimization** for each newly captured dynamic scene.

SparseCam4D targets fewer cameras (Pan et al., 2026), 4C4D addresses the resulting geometry–appearance imbalance (Zhou et al., 2026), and SynCamMaster highlights synchronization as a separate challenge (Bai et al., 2024). With few views, geometry, appearance, and motion are jointly under-constrained. Existing approaches introduce generative priors (Pan et al., 2026) or structured motion (Chao & Isler, 2026), yet remain offline or require per-scene optimization (Zhou et al., 2026; Chao & Isler, 2026), even when the available input views are sparse.

Full-sequence models optimize a collected spacetime volume (Fridovich-Keil et al., 2023; Cao & Johnson, 2023). Streaming Radiance Fields updates online (Li et al., 2022a), while 3DGStream performs per-frame Gaussian fitting (Sun et al., 2024). IGS learns motion transfer (Yan et al., 2025); Recon-GS and ClipGStream use compact residuals and clip updates (Fu et al., 2025; Liang et al., 2026); and stateful models retain geometric context (Chen et al., 2026a; Jin et al., 2026). These systems do not jointly address sparse synchronized views, where cross-view flow disagreement, unreliable lifting, and sparse photometric supervision cause overfitting and long-horizon drift.

We study **sparse-view streaming 4D reconstruction**: causal updates from few cameras while preserving long-sequence novel-view quality. As shown in Figure 1, **S^{2}4D** adapts IGS's generalizable Gaussian motion transfer to sparse views. (i) To aggregate multi-view motion under few cameras, **SAM-Net** uses WAFT optical-flow features, summarizes dynamics with farthest-point-sampling

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

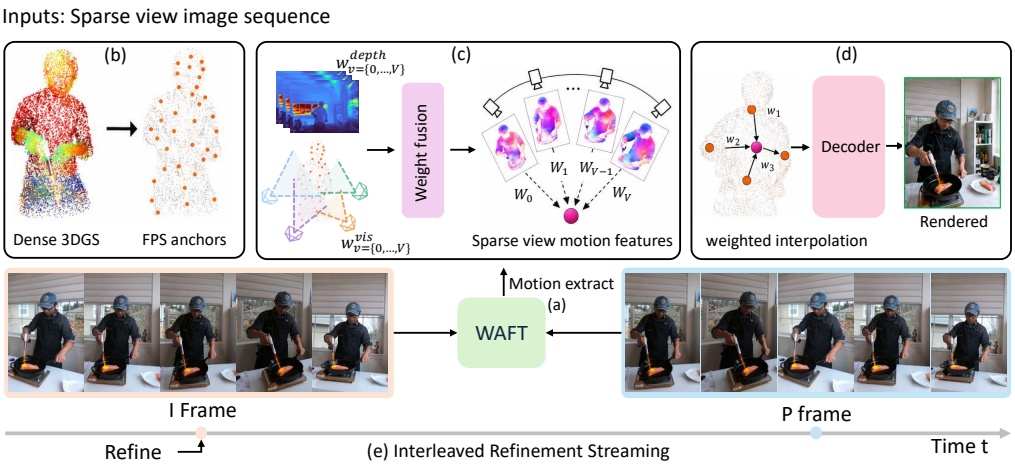

Figure 1: The overall pipeline of S^{2}4D. SAM-Net (a) extracts 2D motion features from the refined I-frame to the P-frame, (b) samples N_a anchors from the I-frame, and (c) lifts image features to anchors via visibility-depth weighted projection; (d) each Gaussian then aggregates neighboring-anchor descriptors and decodes inter-frame residual motion. (e) Interleaved refinement streaming infers successive P-frames until the next I-frame, where max-point bounded recalibration is applied.

(FPS) anchors, and fuses them via visibility- and depth-aware weights to predict per-Gaussian residual motion in one pass. (ii) **Interleaved Refinement Streaming** retains IGS's periodic reference updates and **Max-points Bounded Recalibration**, using sparse-view photometric optimization every t frames to correct appearance and topology changes while capping Gaussian growth. (iii) A **Sparse-view Flow Consistency (SFC)** loss regularizes view-specific motion through calibrated reprojection and depth, improving sparse-view generalization.

Experiments on in-domain and cross-domain dynamic benchmarks show that S^{2}4D achieves state-of-the-art novel-view quality among streaming methods while operating at substantially lower per-frame reconstruction cost than strong offline sparse-view baselines. The learned motion model also generalizes effectively to unseen capture domains and scene types. In summary, the major contributions of our work are summarized below for sparse-view dynamic scene reconstruction:

- We propose SAM-Net, which adapts anchor-driven Gaussian motion transfer to sparse cameras through WAFT features and visibility-depth weighted fusion, with SFC regularization for sparse-view generalization at no inference overhead.
- We integrate this motion predictor with IGS's bounded key-frame refinement, using sparse-view Gaussian optimization to correct the propagated scene state. The resulting system combines fast motion transfer with periodic appearance and geometry updates.
- Extensive experiments on sparse-view dynamic benchmarks show that our method achieves state-of-the-art novel-view quality among streaming approaches, while matching or approaching strong offline sparse-4D baselines at much lower per-frame reconstruction cost, with favorable cross-domain transfer across the evaluated capture environments.

2 Related Work

Optimization-based reconstruction. Yoon et al. (2020) enforces coherent monocular depth; later systems encode Gaussian deformation (Wu et al., 2024a), spacetime primitives (Li et al., 2024), compact performances (Jiang et al., 2024), or 4K content (Xu et al., 2024). Streaming Radiance Fields updates online (Li et al., 2022a); 3DGStream fits each frame (Sun et al., 2024); QUEEN compresses residuals (Girish et al., 2024); and ClipGStream updates clips (Liang et al., 2026). All remain scene-specific and require fitting their representations to the observed target sequence.

Feed-forward reconstruction. pixelSplat and MVSplat regress Gaussians from multi-view features (Charatan et al., 2024; Chen et al., 2025c); DL3DV supplies scale (Ling et al., 2024), and DPT supplies depth priors (Ranftl et al., 2021). Newer models learn equivariant/self-supervised geometry

(Wang et al., 2026b; Zhao et al., 2026) or unified worlds (Team HY-World, 2026). Dynamic variants include L4GM (Ren et al., 2024), DGGT (Chen et al., 2025b), and 4D-LRM (Ma et al., 2025). Stateful methods use test-time training (Wang et al., 2026a; Jin et al., 2026), causal transformers (Zhuo et al., 2026; LAN et al., 2026), or fuse-and-refine updates (Wang et al., 2026c). We target synchronized multi-view streams under sparse camera coverage and causal reconstruction constraints, where each update uses only the available observations.

Sparse-view reconstruction. RegNeRF regularizes unseen rays (Niemeyer et al., 2022); FSGS and CoR-GS use few-shot synthesis and co-regularization (Zhu et al., 2024; Zhang et al., 2024a). GPS-Gaussian+ targets sparse human scenes (Zhou et al., 2024), while geometry–appearance consistency and perceptual densification refine sparse 3DGS (Xiong et al., 2026; Deng et al., 2026). Dynamic methods use generation (Pan et al., 2026), rebalancing (Zhou et al., 2026), monocular priors (Wang et al., 2025c), or skeletons (Chao & Isler, 2026). AVS and EasyVolcap supply data/tooling (AVS, 2024; Xu et al., 2023); VGG features support LPIPS (Simonyan & Zisserman, 2015; Zhang et al., 2018), whereas VBench evaluates temporal video quality (Huang et al., 2024b). Appendix I provides further coverage of these reconstruction paradigms and their practical capture requirements.

3 METHOD

3.1 SCENE STATE AND SPARSE-VIEW UPDATES

Given synchronized images from V calibrated cameras, S^{2}4D transfers a reconstructed scene state to each incoming frame. Figure 1 shows the information path: image-pair motion is aggregated at 3D anchors, distributed to Gaussian primitives, and periodically corrected by photometric refinement. We build on the anchor-driven motion and key-frame streaming framework of IGS (Yan et al., 2025). Our sparse-view adaptation combines WAFT motion features, explicit visibility-depth weighting, SFC regularization, and sparse-view Gaussian refinement.

We represent the reference scene by $\mathcal{G} = \{g_i\}_{i=1}^{N_g}$, with $g_i = (\mathbf{x}_i, \mathbf{s}_i, \mathbf{q}_i, o_i, \mathbf{h}_i)$. Here $\mathbf{x}_i \in \mathbb{R}^3$ is the center, $\mathbf{s}_i$ and unit quaternion $\mathbf{q}_i$ specify scale and orientation, o_i is opacity, and $\mathbf{h}_i$ contains spherical-harmonic color coefficients. Following 3DGS (Kerbl et al., 2023), a pixel $\mathbf{u}$ in camera v is rendered as the following front-to-back composition of projected Gaussian contributions:

$$\hat{I}_v(\mathbf{u}) = \sum_{i \in \mathcal{R}_v(\mathbf{u})} T_{i,v}(\mathbf{u})\, \alpha_{i,v}(\mathbf{u})\, \mathbf{c}_{i,v}, \qquad T_{i,v}(\mathbf{u}) = \prod_{\substack{\ell \in \mathcal{R}_v(\mathbf{u}) \\ \ell \prec_v i}} \big(1 - \alpha_{\ell,v}(\mathbf{u})\big), \tag{1}$$

where $\mathcal{R}_v(\mathbf{u})$ contains the contributing Gaussians, $\ell \prec_v i$ denotes front-to-back order, $\alpha_{i,v}$ is projected opacity, and $\mathbf{c}_{i,v}$ is the view-dependent RGB color evaluated from $\mathbf{h}_i$. Unlike canonical time-conditioned deformation fields (Yang et al., 2024b; Wu et al., 2024a), our state update is conditioned on observed image pairs; Appendix J reviews the canonical formulation.

3.2 SPARSE-VIEW ANCHOR-DRIVEN MOTION NETWORK

Motion evidence and spatial support. SAM-Net takes corresponding reference and target images $\{(I_v, I'_v)\}_{v=1}^{V}$. We retain the optical-flow encoding and geometry-conditioned modulation design of IGS, using pretrained WAFT (Wang & Deng, 2026) for the motion features. With optical-flow encoder $\mathcal{E}_{\text{flow}}$ and modulation operator $\mathcal{M}$, the per-view computation is

$$E_v = \mathcal{E}_{\text{flow}}(I_v, I'_v), \qquad F_v = \mathcal{M}(E_v, D_v, \mathbf{r}_v). \tag{2}$$

Here E_v denotes intermediate motion embeddings, D_v is reference depth and $\mathbf{r}_v$ encodes Plücker rays. Both training frames have precomputed depths; inference uses precomputed frame-0 depth and rendered Gaussian depths thereafter. Scale-shift modulation (Peebles & Xie, 2022) yields reference-aligned motion maps $F_v \in \mathbb{R}^{C \times H \times W}$ for cross-view feature lifting.

The scene geometry supplies a compact spatial support for fusing these maps. We sample anchor positions $\mathbf{a}_j$ from Gaussian centers inside the scene bounding box $\mathcal{B}$:

$$\mathcal{A} = \{\mathbf{a}_j\}_{j=1}^{N_a} = \mathrm{FPS}_{N_a}\big(\{\mathbf{x}_i : \mathbf{x}_i \in \mathcal{B}\}\big). \tag{3}$$

Here N_g counts Gaussian primitives and N_a counts anchors; $N_a = 8192$ unless ablated. Anchors provide motion support, while Gaussians retain the rendering attributes. Throughout this section, i indexes Gaussians, j indexes anchors, and v, w index cameras.

Visibility-depth weighted lifting. Let π_v project a world point into camera v, and let $\Psi(X, \mathbf{u})$ bilinearly sample map X. At each anchor we collect a motion feature and reference depth:

$$\mathbf{f}_{j,v} = \Psi(F_v, \pi_v(\mathbf{a}_j)), \qquad d_{j,v} = \Psi(D_v, \pi_v(\mathbf{a}_j)). \tag{4}$$

The binary mask $b_{j,v}$ marks positive camera-space depth and an in-frustum projection. Inverse depth then biases fusion toward nearer observations. With a positive depth floor $d_{\min}$, the weights are

$$c_{j,v} = \frac{1}{\max(d_{j,v}, d_{\min})}, \qquad w_{j,v} = b_{j,v}\,\frac{c_{j,v}}{\max_{w=1,\dots,V} c_{j,w}}. \tag{5}$$

This is a geometric validity mask with a depth-based weighting heuristic; it does not test surface occlusion. The implementation uses $d_{\min} = 0.05$ and clamps the aggregation denominator to avoid amplifying weakly supported features at sparsely observed anchors:

$$\mathbf{f}^a_j = \frac{\sum_{v=1}^{V} w_{j,v}\mathbf{f}_{j,v}}{\max(1, \sum_{v=1}^{V} w_{j,v})}, \qquad \{\mathbf{m}^a_j\}_{j=1}^{N_a} = \mathrm{Transformer}\big(\{\mathbf{f}^a_j\}_{j=1}^{N_a}\big). \tag{6}$$

Thus $\mathbf{f}^a_j$ denotes the fused feature and $\mathbf{m}^a_j$ the contextualized anchor motion descriptor. An anchor without valid projections receives a zero fused feature before contextualization.

From anchors to Gaussian updates. For Gaussian g_i, let $\mathcal{N}_K(i)$ contain the indices of its K nearest anchors, with $K = 8$. We explicitly distinguish the Gaussian–anchor distance $\rho_{ij} = \|\mathbf{x}_i - \mathbf{a}_j\|_2$ from the sampled image depth $d_{j,v}$. The Gaussian motion descriptor is

$$\mathbf{m}^g_i = \sum_{j \in \mathcal{N}_K(i)} p_{ij}\mathbf{m}^a_j, \qquad p_{ij} = \frac{\exp(-\beta\rho_{ij})}{\sum_{\ell \in \mathcal{N}_K(i)} \exp(-\beta\rho_{i\ell})}, \quad \beta = 10. \tag{7}$$

Both indices in ρ_{ij} and p_{ij} refer to the same Gaussian–anchor pair, and $\sum_{j \in \mathcal{N}_K(i)} p_{ij} = 1$. The neighbor count K controls local interpolation support independently of the total anchor count N_a. Retaining IGS's linear motion heads, we decode the per-primitive motion residuals:

$$\delta\mathbf{x}_i = W_x\mathbf{m}^g_i + \mathbf{b}_x, \qquad \delta\mathbf{q}_i = W_q\mathbf{m}^g_i + \mathbf{b}_q, \tag{8}$$

where $(W_x, \mathbf{b}_x)$ and $(W_q, \mathbf{b}_q)$ are learned translation and rotation parameters. The descriptors are interpolated before decoding; each Gaussian receives its own motion even when it shares anchors with nearby primitives. The target attributes are then

$$\mathbf{x}'_i = \mathbf{x}_i + \delta\mathbf{x}_i, \qquad \mathbf{q}'_i = \mathrm{unit}\big(\mathbf{q}_i \otimes \delta\mathbf{q}_i\big). \tag{9}$$

Here $\otimes$ is the Hamilton product and unit normalizes the quaternion for rendering. The relative rotation is initialized near the identity. Scale, opacity, and color are copied from the reference state; primitives outside $\mathcal{B}$ remain unchanged during motion propagation.

3.3 Learning with Cross-view Motion Constraints

Rendered target images supervise the predicted Gaussian updates through

$$\mathcal{L}_{\mathrm{rgb}} = (1-\lambda)\|\hat{I} - I'\|_1 + \lambda\big(1 - \mathrm{SSIM}(\hat{I}, I')\big). \tag{10}$$

We average this loss over the training views; λ weights the structural similarity term (Wang et al., 2004). Sparse photometric supervision alone leaves motion ambiguous outside the observed cameras. We therefore add *Sparse-view Flow Consistency* (SFC), which regularizes view-specific motion through calibrated reprojection (Figure 2) using shared scene geometry.

Let Δ_v denote the backbone's forward flow. For a reference pixel $\mathbf{u}$, the training-pair depth supplies 3D lifts $\mathbf{P}_v(\mathbf{u})$ and $\mathbf{P}'_v(\mathbf{u})$ of the source and displaced pixels. Their projections into camera w define

$$\Delta_{v\to w}(\mathbf{u}) = \pi_w(\mathbf{P}'_v(\mathbf{u})) - \pi_w(\mathbf{P}_v(\mathbf{u})), \qquad \mathbf{u}_w = \pi_w(\mathbf{P}_v(\mathbf{u})). \tag{11}$$

We penalize disagreement with the flow estimated in view w:

$$\mathcal{L}_{v\to w} = \frac{1}{|\Omega_{v,w}|} \sum_{\mathbf{u} \in \Omega_{v,w}} \left\|\Delta_w(\mathbf{u}_w) - \Delta_{v\to w}(\mathbf{u})\right\|_1. \tag{12}$$

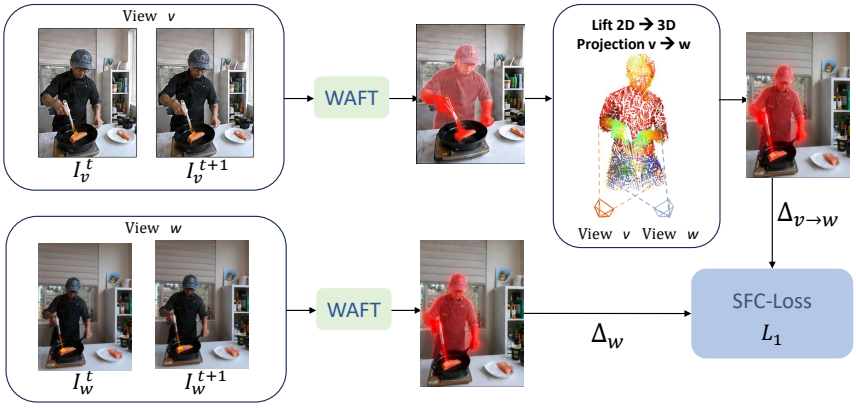

Figure 2: Sparse-view flow consistency loss. Independently estimated 2D flows are aligned using the training-pair depth and reprojection across calibrated cameras.

Here $\Omega_{v,w}$ contains valid in-frustum samples. For the selected camera pairs $\mathcal{P}$, the regularizer and total training objective are defined jointly for learning the scene motion:

$$\mathcal{L}_{\text{sfc}} = \frac{1}{|\mathcal{P}|} \sum_{(v,w)\in\mathcal{P}} \mathcal{L}_{v\to w}, \qquad \mathcal{L} = \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{sfc}}\mathcal{L}_{\text{sfc}}. \tag{13}$$

The coefficients λ_{rgb} and λ_{sfc} balance rendering supervision and cross-view regularization. SFC encourages a shared geometric explanation of the observed motion and adds no inference step. Its effect depends on the depth-based lifting approximation; Appendix C.2 specifies those lifts and the camera pairs used in training with sparse calibrated observations.

3.4 Streaming with Bounded Scene Refinement

The motion predictor transfers geometry while holding the remaining attributes fixed. Appearance changes and newly exposed regions therefore require occasional updates to the scene representation itself. We retain IGS's key-frame propagation and bounded refinement strategy (Yan et al., 2025), adapting its local optimization to sparse observations through the Co-Adaptation-of-3DGS rasterizer (Chen et al., 2025a) during the periodic Gaussian-state updates.

Interleaved Refinement Streaming. Frame 0 is reconstructed first. With refinement interval t, the refined states at frames $0, t, 2t, \ldots$ serve as references. Within an interval, each incoming frame is predicted directly from the latest refined reference and its images. At the next boundary, we refine the predicted Gaussians and replace the reference state. Intermediate frames need only a SAM-Net forward pass; they do not recursively pass prediction errors to one another. Periodic photometric correction addresses residual errors across intervals, while keeping the learned SAM-Net weights fixed at deployment across all target scenes and evaluation sequences.

Specifically, let $r(n) = t\lfloor (n-1)/t \rfloor$ denote the reference for incoming frame $n \geq 1$, and let $\mathcal{S}_\theta$ denote SAM-Net followed by Gaussian motion application. The streaming update is

$$\widetilde{\mathcal{G}}^{\,n} = \mathcal{S}_\theta(\mathcal{G}^{r(n)}, \mathbf{I}^{r(n)}, \mathbf{I}^n), \qquad \mathcal{G}^n = \begin{cases} \text{Refine}(\widetilde{\mathcal{G}}^{\,n}, \mathbf{I}^n), & n \bmod t = 0, \\ \widetilde{\mathcal{G}}^{\,n}, & \text{otherwise.} \end{cases} \tag{14}$$

Here $\mathbf{I}^n$ collects the sparse input views at frame n, and superscripts denote time. Refinement updates the Gaussian attributes against current observations while keeping the learned network parameters θ fixed throughout deployment for both in-domain and cross-domain evaluation.

Max-points Bounded Recalibration. Each refinement optimizes all Gaussian attributes using the available sparse views and permits 3DGS cloning, splitting, and pruning (Kerbl et al., 2023). Following IGS, gradient-based priority restricts densification to a fixed primitive budget. This bounds storage and rendering cost as the scene evolves. For sparse-view deployment, the budget also limits redundant primitives fitted to the few observed cameras. We use $t = 5$ and 50 refinement iterations; the complete reconstruction time includes initialization, propagation, and these periodic updates.

4 EXPERIMENTS

4.1 IMPLEMENTATION DETAILS

Training Data. N3DV (Li et al., 2022b) contains six dynamic scenes recorded by 21 cameras at 2704×2028 resolution, each with 300 frames. We train on {*coffee_martini*, *cook_spinach*, *flame_salmon*, *flame_steak*}. For all frames, we construct 3D Gaussians and generate forward and backward motions over five frames, yielding 12,000 training pairs.

Training Procedure. We initialize the optical-flow encoder with WAFT (Wang & Deng, 2026) and fine-tune it jointly with SAM-Net. Although SAM-Net supports an arbitrary number of views, we use $V = 3$ during training, following the sparse-view setting in (Pan et al., 2026). Each view produces a 64-channel motion map at 256×256 resolution. We sample $N_a = 8192$ anchors from the Gaussian points and employ a four-layer Transformer to produce 128-dimensional anchor motion descriptors. For interleaved refinement, we adopt a sparse Gaussian splatting rasterizer from Co-Adaptation-of-3DGS (Chen et al., 2025a) for accurate depth and geometry rendering. The three input views also provide photometric supervision. Training is performed for 30 epochs on four 48 GB L40 GPUs, with the same network configuration across all training scenes.

Evaluation Protocol. We distinguish in-domain scene generalization from cross-domain transfer. Within N3DV, the held-out *sear_steak* and *cut_roasted_beef* sequences depict different cooking actions, object interactions, and motion trajectories, but share the training scenes' kitchen setting and acquisition conditions. Table 7 evaluates these unseen sequences separately; Table 1 retains the six-scene average (four training and two held-out) for comparison with prior reports, rather than as an independent generalization test. Cross-domain evaluation on Meeting Room (Li et al., 2022a), ENeRF (Lin et al., 2022), and PKU-MVHumans (Zheng et al., 2024) tests transfer beyond the kitchen domain to different environments, human motions, and capture configurations. These indoor, outdoor, and human-centric datasets use 13, 18, and 56/60 cameras, respectively; we take three views as input and reserve the rest for evaluation. SAM-Net weights remain fixed at test time, while each target scene still undergoes frame-0 Gaussian initialization and periodic Gaussian refinement. We use $t = 5$ (60 reference frames per 300-frame sequence, comprising frame 0 and 59 subsequent refinement frames) and 50 iterations per refinement, with densification/pruning every 20 iterations. Metrics include frame 0 and are averaged over frames and then scenes. Storage includes frame-0/refinement Gaussians and propagation residuals; Time denotes the average per-frame reconstruction cost. Further details are in Appendix C.

4.2 COMPARISONS

Baselines. We compare with offline methods LocalDyGS (Wu et al., 2025a), MonoFusion (Wang et al., 2025c), SparseCam4D (Pan et al., 2026), and 4C4D (Zhou et al., 2026); online methods 3DGStream (Sun et al., 2024), QUEEN (Girish et al., 2024), and Recon-GS (Fu et al., 2025); and feed-forward methods IGS (Yan et al., 2025), TTTLRM (Wang et al., 2026a), AnySplat (Jiang et al., 2025b), WorldMirror (Liu et al., 2026), and DGGT (Chen et al., 2025b).

In-domain evaluation. Table 1 reports the six-scene protocol, while Appendix D isolates the two held-out scenes. We use the same frame-0 Gaussians and rasterizer for the online baselines and $S^2$4D-H. At 1352×1014, $S^2$4D-H obtains **23.12** dB in 0.62 seconds per frame. At the feed-forward baselines' matched 518×350 resolution, $S^2$4D-L reaches **23.56** dB (1.05 dB above IGS), **0.16** seconds, 234 FPS, and 4.5 MB. We report the settings separately because resolution changes both observed detail and pixel-space metrics. The qualitative comparison in Figure 3 further demonstrates these advantages. Our method more faithfully preserves the contours and fine structures of the kitchen appliance (red boxes in the first row) and the appearance and geometry of the cat (red boxes in the second row). It also reconstructs the moving hands and body boundaries (green boxes) with less blurring and fewer artifacts, closely matching the ground truth.

Cross-domain evaluation. To assess generalization, we evaluate the N3DV-trained model on Meeting Room (Li et al., 2022a), ENeRF (Lin et al., 2022), and PKU-MVHumans (Zheng et al., 2024) without per-scene retraining. Results are in Tables 2 and 3. On Meeting Room, matched-resolution $S^2$4D-L obtains the best PSNR/SSIM (**21.26/0.849**) and time (**0.16**s), with 0.192 LPIPS and 1.3 MB. At 1280×720, $S^2$4D-H obtains 20.27/0.841/0.196, 0.59 seconds, and 1.6 MB; 4C4D retains the best LPIPS (0.185) among the methods evaluated at this higher resolution.

Table 1: N3DV six-scene averages. H: 1352×1014 (ours and optimization); L: 518×350 (ours and feed-forward baselines).

Method	PSNR↑ (dB)	Time↓ (s)	Render↑ (FPS)	Size↓ (MB)
Offline optimization				
LocalDyGS	20.12	1.65	105	100
MonoFusion	18.43	>10	69	67
SparseCam4D	21.91	>10	98	91
4C4D	22.29	4.52	110	231
Online optimization				
3DGStream	19.52	1.78	215	7.8
QUEEN	17.16	0.71	248	7.8
Recon-GS	21.37	5.68	**250**	**0.4**
Feed-forward				
IGS	22.51	2.45	201	6.7
TTTLRM	16.92	0.92	225	19
AnySplat	15.53	0.41	219	25
WorldMirror	17.09	0.17	222	22
DGGT	16.32	0.66	118	60
$S^2$4D-H (Ours)	**23.12**	0.62	204	5.2
$S^2$4D-L (Ours)	**23.56**	**0.16**	234	4.5

Table 2: Meeting Room comparison. H: 1280×720 (ours and optimization); L: 518×350 (ours and feed-forward baselines).

Method	PSNR↑ (dB)	SSIM↑	LPIPS↓	Time↓ (s)	Size↓ (MB)
Offline optimization					
LocalDyGS	18.26	0.735	0.291	1.14	90
MonoFusion	17.14	0.697	0.294	>10	55
SparseCam4D	20.13	0.745	0.237	>10	85
4C4D	19.83	0.795	**0.185**	4.31	210
Online optimization					
3DGStream	18.23	0.736	0.285	1.78	7.59
QUEEN	16.32	0.703	0.298	0.70	7.5
Recon-GS	19.59	0.809	0.235	3.50	**0.3**
Feed-forward					
IGS	19.45	0.802	0.222	2.35	6.1
TTTLRM	13.47	0.492	0.533	0.92	19
AnySplat	14.37	0.570	0.422	0.41	25
WorldMirror	18.32	0.711	0.244	0.17	22
DGGT	17.25	0.656	0.288	0.66	60
$S^2$4D-H (Ours)	**20.27**	**0.841**	0.196	0.59	1.6
$S^2$4D-L (Ours)	**21.26**	**0.849**	0.192	**0.16**	1.3



Figure 3: Qualitative comparison on the N3DV dataset.

On ENeRF and PKU-MVHumans, $S^2$4D achieves the best quality–efficiency trade-off, running at **0.71**s per frame while obtaining the highest PSNR/SSIM on both datasets (**24.16/0.797** and **28.47/0.903**), as well as the best ENeRF LPIPS (**0.168**). Its second-best t-LPIPS on ENeRF (0.0238) and PKU-MVHumans (0.0234) further indicates strong local temporal consistency over the evaluated long sequences. These results demonstrate robust generalization across unseen indoor, outdoor, and human-centric scenes. Figure 6 compares per-frame PSNR on the Meeting Room *trimming* sequence. $S^2$4D maintains more stable reconstruction quality over time than 3DGStream and Recon-GS, whose PSNR progressively declines. The smaller quality degradation and widening gap toward later frames support the effectiveness of our streaming strategy in reducing long-horizon drift and limiting accumulated reconstruction errors. Besides, Figure 4 further shows that our method reconstructs the rapidly moving hands and arms, and the structures of the interacted plant, with sharper boundaries. Figure 5(b) shows per-frame reconstruction time compared with online meth-

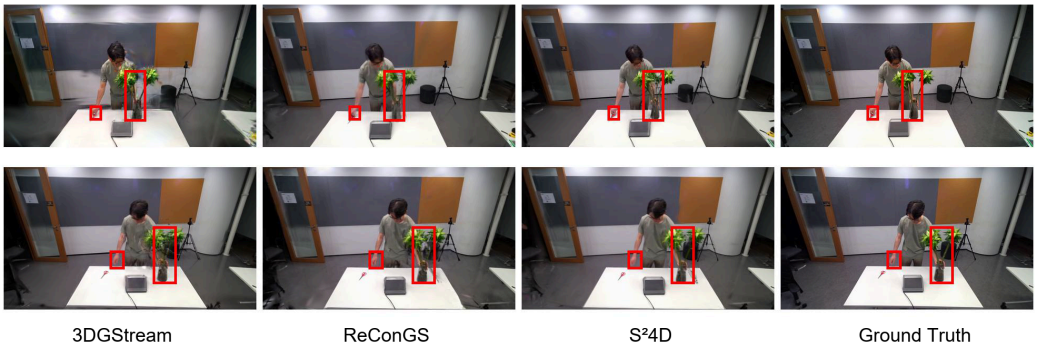

Figure 4: Qualitative comparison on the Meeting Room dataset (cross-domain).

Table 3: Cross-domain quantitative comparison on ENeRF and PKU-MVHumans. Best results are in bold and second-best results are underlined within each metric.

Method	Type	Time	ENeRF				PKU-MVHumans			
		(s/frame) ↓	PSNR ↑	SSIM ↑	LPIPS ↓	t-LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	t-LPIPS ↓
LocalDyGS	Offline	1.99	20.12	<u>0.796</u>	0.261	0.0272	18.26	0.735	0.291	0.0254
MonoFusion	Offline	>10	21.91	0.759	0.258	0.0311	23.34	0.781	0.233	0.0299
SparseCam4D	Offline	>10	22.61	0.751	0.201	**0.0233**	24.44	0.782	0.179	**0.0216**
4C4D	Offline	6.55	<u>23.98</u>	0.782	<u>0.181</u>	0.0244	<u>25.66</u>	<u>0.822</u>	0.167	0.0244
3DGStream	**Online**	3.11	20.65	0.731	0.261	0.0344	20.84	0.761	0.278	0.0313
QUEEN	**Online**	<u>1.27</u>	19.11	0.652	0.252	0.0261	23.03	0.723	<u>0.146</u>	0.0251
Recon-GS	**Online**	6.31	17.62	0.529	0.447	0.0248	20.05	0.755	**0.134**	0.0249
$S^2$4D (Ours)	**Online**	**0.71**	**24.16**	**0.797**	**0.168**	<u>0.0238</u>	**28.47**	**0.903**	0.159	<u>0.0234</u>

ods. Additional free-viewpoint rendering videos and qualitative comparisons are provided in the supplementary material for the evaluated scenes and camera trajectories.

4.3 Ablation Study

Pretrained optical-flow prior in SAM-Net. We replace the pretrained optical-flow encoder with a randomly initialized four-layer UNet and train it jointly with the remaining modules. PSNR drops from 23.12 to 20.23 dB (Table 4), supporting the value of transferable image-space motion knowledge for recovering 3D deformation from sparse observations.

Visibility-depth aware sparse-view 3D motion projection. Replacing visibility-depth aware fusion of the projected multi-view features with an unweighted mean reduces PSNR from 23.12 to 21.97 dB (Table 4). Weighting observations by their geometric reliability improves the resulting novel-view reconstruction quality under sparse multi-view supervision.

Sparse-view Flow Consistency (SFC) loss. Removing $\mathcal{L}_{\text{sfc}}$ reduces PSNR from 23.12 to 22.45 dB (Table 4). This supports using calibrated reprojection and precomputed reference-frame depth to regularize camera-specific motion cues during training. SFC improves novel-view reconstruction quality and is omitted at inference, so it adds no reconstruction overhead.

Interleaved Refinement Streaming. Figure 5(a) confirms the quality benefit of periodic key-frame refinement; Figure 6 provides long-sequence evidence for reduced drift. Removing Max-points Bounded Recalibration lowers PSNR to 22.68 dB and increases storage from 5.21 to 70.26 MB (Table 4), showing that a bounded Gaussian budget controls densification and helps preserve novel-view quality under sparse supervision throughout the streaming sequence.

Refinement interval selection. Table 5 compares $t \in \{1, 5, 10\}$. Refining every frame ($t = 1$) increases cost and yields lower quality, consistent with overfitting to sparse views, whereas $t = 10$ extends propagation too far between corrections, reducing quality under large motions. We use $t = 5$ to balance quality, time, and storage under sparse multi-view supervision.

Table 4: Component ablations on N3DV.

Method	PSNR↑ (dB)	Time↓ (s)	Size↓ (MB)
w/o optical-flow pretraining	20.23	0.67	**4.69**
w/o visibility-depth fusion	21.97	**0.61**	5.21
w/o bounded recalibration	22.68	1.02	70.26
w/o SFC loss	22.45	0.63	5.21
Ours	**23.12**	0.62	5.21

Table 5: Refinement interval.

t	PSNR↑ (dB)	Time↓ (s)	Size↓ (MB)
1	22.89	5.51	26.04
5	**23.12**	0.62	5.21
10	21.31	0.48	**1.08**

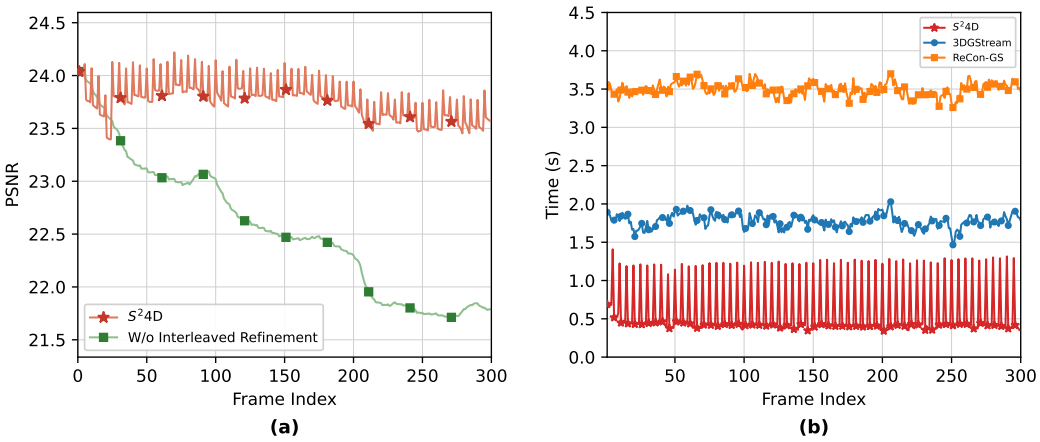

Figure 5: (a) Ablation on key-frame refinement. (b) Per-frame reconstruction time.

Number of anchor points. Figure 7 varies the anchor count from 1,024 to 16,384. Increasing from 8,192 to 16,384 raises PSNR from 23.12 to 23.35 dB, but increases per-frame reconstruction time by 50% (0.62 to 0.93 s). We retain 8,192 as a practical quality–efficiency trade-off. The additional memory required by larger anchor feature sets and attention computations further motivates this choice under our GPU memory budget for sparse-view streaming applications.

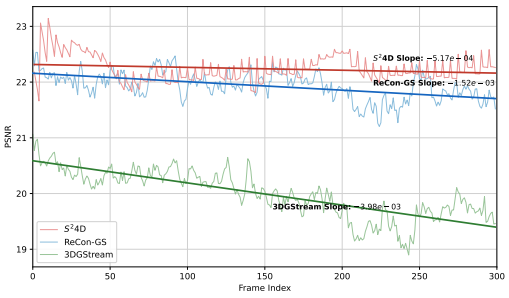

Figure 6: Per-frame PSNR on the Meeting Room *trimming* video sequence.

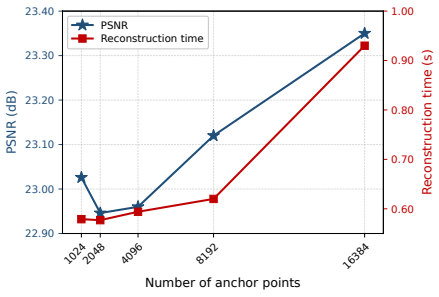

Figure 7: Anchor-count ablation of reconstruction quality and per-frame time.

5 Conclusion

We present S^{2}4D, a sparse-view streaming framework for dynamic 3D Gaussian reconstruction. SAM-Net lifts sparse-view features into anchor-conditioned 3D motion through visibility-depth-aware aggregation, while SFC regularizes camera-specific motion cues to improve sparse-view generalization without inference overhead. Interleaved Refinement Streaming with Max-points Bounded Recalibration limits long-horizon drift and densification overfitting. S^{2}4D achieves state-of-the-art novel-view quality among streaming methods, approaches strong offline sparse-4D baselines at substantially lower per-frame cost, and generalizes across domains. The reported gains concern the evaluated synchronized and calibrated benchmarks; broader deployment across uncontrolled camera layouts, severe occlusion, and more diverse motion distributions remains an important direction for future validation under varied real-world capture conditions.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing and LaTeX formatting during the preparation of this submission.

ETHICS STATEMENT

This work evaluates dynamic-scene reconstruction on established multi-view benchmarks cited in Section 4.1. The submission reports no new data collection or human-subject study.

REPRODUCIBILITY STATEMENT

Training and evaluation settings are described in Section 4.1 and Appendix C. Metrics, including storage and per-frame reconstruction time, are defined in Appendix B. Per-sequence results and additional ablations appear in Appendices D and F.

REFERENCES

AVS. AVS dataset, 2024. URL https://www.avs.org.cn/. Dataset website.

Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Xiao Fu, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di Zhang. SynCamMaster: Synchronizing multi-camera video generation from diverse viewpoints. *arXiv preprint arXiv:2412.07760*, 2024.

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

Abdelhak Bentaleb, May Lim, Sarra Hammoudi, Saad Harous, and Roger Zimmermann. Solutions, challenges, and opportunities in volumetric video streaming: An architectural perspective. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 21(7), 2025. doi: 10.1145/3705321.

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics*, 39(4):86:1–86:15, 2020.

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.

Jun-Jee Chao and Volkan Isler. Sv-gs: Sparse view 4d reconstruction with skeleton-driven gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5027–5037, June 2026.

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.

Kangjie Chen, Yingji Zhong, Zhihao Li, Jiaqi Lin, Youyu Chen, Minghan Qin, and Haoqian Wang. Quantifying and alleviating co-adaptation in sparse-view 3d gaussian splatting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=GrPo8NTtzK.

Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. Geometric context transformer for streaming 3d reconstruction. *arXiv preprint arXiv:2604.14141*, 2026a.

Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Hongyang Li, Ya-Qin Zhang, and Hao Zhao. Dggt: Feedforward 4d reconstruction of dynamic driving scenes using unposed images, 2025b. URL https://arxiv.org/abs/2512.03004.

Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=aMs6FtNaY5.

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, 2025c.

Yang Deng, Zhanke Wang, Jiahao Wu, Jie Liang, Jingui Ma, Yang Hu, and Ronggang Wang. Pano-GS: Perception-aware gaussian optimization with gradient consistency and multi-criteria densification for high-quality rendering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3560–3568, 2026.

Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds, 2024.

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

Jiaye Fu, Qiankun Gao, Chengxiang Wen, Yanmin Wu, Siwei Ma, Jiaqi Zhang, and Jian Zhang. Recon-gs: Continuum-preserved gaussian streaming for fast and compact reconstruction of dynamic scenes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Sharath Girish, Tianye Li, Amrita Mazumdar, Abhinav Shrivastava, David Luebke, and Shalini De Mello. QUEEN: QUantized efficient ENcoding for streaming free-viewpoint videos. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://research.nvidia.com/labs/amri/projects/queen/.

Tingxuan Huang, Haowei Zhu, Jun-hai Yong, Hao Pan, and Bin Wang. Mango-gs: Enhancing spatio-temporal consistency in dynamic scenes reconstruction using multi-frame node-guided 4d gaussian splatting. In *International Conference on Learning Representations*, 2026.

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. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4220–4230, 2024a. doi: 10.1109/CVPR52733.2024.00404.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024b.

Junhwa Hur, Charles Herrmann, Songyou Peng, Philipp Henzler, Zeyu Ma, Todd Zickler, and Deqing Sun. UFO-4d: Unposed feedforward 4d reconstruction from two images. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8gDDWqO59H.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

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*, 2025a.

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 (TOG)*, 44(6):1–16, 2025b.

Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. HiFi4G: High-fidelity human performance rendering via compact gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19734–19745, 2024.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QQBPWtvtcn.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Yushi LAN, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. STream3r: Scalable sequential 3d reconstruction with causal transformer. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RTTYGeC2Io.

Wonjoon Lee, Sungmin Woo, Donghyeong Kim, Jungho Lee, Sangheon Park, and Sangyoun Lee. Morgs: Efficient per-gaussian motion reasoning for streamable dynamic 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41044–41053, June 2026.

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.

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.

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 (CVPR)*, pp. 8508–8520, June 2024.

Jie Liang, Jiahao Wu, Chao Wang, Jiayu Yang, Xiaoyun Zheng, Kaiqiang Xiong, Zhanke Wang, Jinbo Yan, Feng Gao, and Ronggang Wang. ClipGStream: Clip-stream gaussian splatting for any length and any motion multi-view dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41022–41032, 2026.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2026.

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.

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, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

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. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=HFNJOpXHfm.

Ziqiao Ma, Xuweiyi Chen, Shoubin Yu, Sai Bi, Kai Zhang, Ziwen Chen, Sihan Xu, Jianing Yang, Zexiang Xu, Kalyan Sunkavalli, et al. 4D-LRM: Large space-time reconstruction model from and to any view at any time. In *Advances in Neural Information Processing Systems*, 2025.

Daniel Martin, Sandra Malpica, Diego Gutierrez, Belen Masia, and Ana Serrano. Multimodality in VR: A survey. *ACM Computing Surveys*, 54(10s):1–36, 2022.

Lukas Mehl, Azin Jahedi, Jenny Schmalfuss, and Andrés Bruhn. M-FUSE: Multi-frame fusion for scene flow estimation. In *Proc. Winter Conference on Applications of Computer Vision (WACV)*, 2023.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. ISSN 0001-0782.

Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5480–5490, 2022.

Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. CoherentGS: Sparse novel view synthesis with coherent 3d gaussians. In *European Conference on Computer Vision (ECCV)*, 2024.

Weihong Pan, Xiaoyu Zhang, Zhuang Zhang, Zhichao Ye, Nan Wang, Haomin Liu, and Guofeng Zhang. Sparsecam4d: Spatio-temporally consistent 4d reconstruction from sparse cameras. In *CVPR*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021.

Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, and Huan Ling. L4gm: Large 4d gaussian reconstruction model. In *Proceedings of Neural Information Processing Systems(NeurIPS)*, Dec 2024.

Changyue Shi, Chuxiao Yang, Xinyuan Hu, Minghao Chen, Wenwen Pan, Yan Yang, Jiajun Ding, Zhou Yu, and Jun Yu. Sparse4dgs: 4d gaussian splatting for sparse-frame dynamic scene reconstruction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40:8933–8941, 03 2026. doi: 10.1609/aaai.v40i11.37848.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20675–20685, 2024.

Team HY-World. HY-World 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint*, 2026.

Zachary Teed and Jia Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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. *CVPR*, 2026a.

Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9065–9076, 2023a.

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

Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 76–87, 2023b.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=DTQIjngDta.

Yihan Wang and Jia Deng. WAFT: Warping-alone field transforms for optical flow. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HTqGE0KcuF.

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. *Advances in Neural Information Processing Systems*, 38:76311–76341, 2026c.

Zhanke Wang, Zhiyan Wang, Kaiqiang Xiong, Jiahao Wu, Yang Deng, and Ronggang Wang. SAP: Exact sorting in splatting via screen-aligned primitives. *Advances in Neural Information Processing Systems*, 38:48975–49001, 2026d.

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

Zihan Wang, Jeff Tan, Tarasha Khurana, Neehar Peri, and Deva Ramanan. Monofusion: Sparse-view 4d reconstruction via monocular fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8252–8263, October 2025c.

Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. Accflow: Backward accumulation for long-range optical flow, 2023.

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 (CVPR)*, pp. 20310–20320, June 2024a.

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.

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 *International Conference on Learning Representations (ICLR)*, 2025b.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024b.

Haolin Xiong, Sairisheek Muttukuru, Rishi Upadhyay, Pradyumna Chari, and Achuta Kadambi. SparseGS: Real-time 360° sparse view synthesis using gaussian splatting. In *International Conference on 3D Vision (3DV)*, 2025.

Kaiqiang Xiong, Rui Peng, Jiahao Wu, Zhanke Wang, Jie Liang, Xiaoyun Zheng, Feng Gao, and Ronggang Wang. Intrinsic geometry-appearance consistency optimization for sparse-view gaussian splatting. *arXiv preprint arXiv:2603.02893*, 2026.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

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. In *SIGGRAPH Asia 2023 Technical Communications*, 2023.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4DGT: Learning a 4d gaussian transformer using real-world monocular videos. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=qMRFNxioPC.

Jinbo Yan, Rui Peng, Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, and Ronggang Wang. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16520–16531, 06 2025. doi: 10.1109/CVPR52734.2025.01540.

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

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

Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ImRhA9xmay.

Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5336–5345, 2020.

Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. CoR-GS: Sparse-view 3d gaussian splatting via co-regularization. In *European Conference on Computer Vision (ECCV)*, 2024a.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3d gaussian splatting, 2024b. URL https://arxiv.org/abs/2404.19702.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Xiaoyun Zheng, Liwei Liao, Xufeng Li, Jianbo Jiao, Rongjie Wang, Feng Gao, Shiqi Wang, and Ronggang Wang. Pku-dymvhumans: A multi-view video benchmark for high-fidelity dynamic human modeling. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Boyao Zhou, Shunyuan Zheng, Hanzhang Tu, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. GPS-Gaussian+: Generalizable pixel-wise 3d gaussian splatting for real-time human-scene rendering from sparse views. *arXiv preprint arXiv:2411.11363*, 2024.

Junsheng Zhou, Zhifan Yang, Liang Han, Wenyuan Zhang, Kanle Shi, Shenkun Xu, and Yushen Liu. 4c4d: 4 camera 4d gaussian splatting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=5APgTKsnx8.

A Overview of the Appendix

This appendix provides additional details that complement the main paper:

- Details of metrics calculation in Appendix B.
- The SAM-Net architecture and additional implementation settings in Appendix C.
- Per-sequence results for the two N3DV test scenes in Appendix D.
- Practical deployment under different camera counts in Appendix E.
- Additional ablation studies in Appendix F.
- Additional discussion in Appendix G.
- Additional limitations and future work in Appendix H.
- Extended related work in Appendix I.
- Dynamic Gaussian background in Appendix J.

Anonymous code (`S^2 4D_code.zip`) and a free-viewpoint demo (`S^2 4D_freeview.mp4`) are included in the supplementary materials. The complete code, pretrained weights, and the training dataset we constructed will be released as open-source after the review process.

B Details of the Metrics Calculation

As mentioned in Section 4 of the main paper, frame-based metrics are averaged over the full 300-frame sequence (taking N3DV as an example), including frame 0, following previous methods (Sun et al., 2024; Li et al., 2022a). Temporal LPIPS instead averages the 299 consecutive-frame transitions, as defined below. Specifically:

Temporal LPIPS (t-LPIPS). Following the frame-difference evaluation described by Mango-GS (Huang et al., 2026), we measure the perceptual discrepancy between temporal changes in rendered and ground-truth images. Let $\hat{I}_v^n$ and I_v^n be the rendered and ground-truth RGB images at frame n in evaluation view v. For a sequence of T frames, define

$$\delta\hat{I}_v^n = \hat{I}_v^{n+1} - \hat{I}_v^n, \qquad \delta I_v^n = I_v^{n+1} - I_v^n, \tag{15}$$

$$\text{t-LPIPS}_v = \frac{1}{T-1}\sum_{n=0}^{T-2} d_{\text{LPIPS}}\left(\delta\hat{I}_v^n, \delta I_v^n\right), \tag{16}$$

where d_{LPIPS} is the learned perceptual distance (Zhang et al., 2018). RGB values are in $[0, 1]$, so the signed frame differences lie in $[-1, 1]$. Each pair uses the same evaluation camera at consecutive timestamps, without optical-flow warping. We average over evaluation views and then scenes, and report unscaled values; lower is better. For 300 frames, this gives 299 temporal pairs, including the transition from frame 0 to frame 1. This metric evaluates local temporal fidelity relative to the observed changes; long-horizon behavior is assessed separately in Figure 6.

Storage. The storage required for S^{2}4D includes the Gaussian primitives for frame 0 and each refinement frame, as well as the residuals for each propagation frame. Since each propagation frame is generated by applying motion from the previous refinement frame using SAM-Net, we only need to store the corresponding displacement ($\delta\mathbf{x}_i$) and relative quaternion ($\delta\mathbf{q}_i$), along with the mask of points with motion. We report the average storage requirements over the 300 frames.

Time. In line with previous methods (Sun et al., 2024; Li et al., 2022a), Time denotes the average per-frame reconstruction time for constructing a free-viewpoint video from a multi-view video sequence. This includes the time for constructing the Gaussian primitives for frame 0, generating propagation frames using SAM-Net, and performing Max-points Bounded Recalibration on the refinement frames. The total time is averaged over all 300 frames, which corresponds to our per-frame reconstruction time. For offline methods, we define the per-frame reconstruction time as the total reconstruction time divided by the number of frames.

C More Implementation Details

C.1 SAM-Net Architecture and Relation to IGS

Architecture. SAM-Net follows an encoder–projection–decoder architecture. A pretrained optical-flow encoder extracts per-view inter-frame motion features, which are modulated by reference depth and Plücker-ray camera information. FPS anchors sample these maps through camera projection; visibility-depth-aware fusion and a Transformer then produce 3D anchor descriptors. Neighboring-anchor interpolation and a lightweight decoder predict each Gaussian's translation and rotation residuals, while its other attributes are propagated unchanged.

Depth availability during training and inference. Each training example contains a reference–target image pair together with the precomputed depth maps for both frames. These depths are available from the prepared per-frame reconstruction data. SAM-Net uses the reference-frame depth D_v for geometry conditioning and anchor-feature fusion. At inference, precomputed depth is available only for frame 0. Depth for subsequent reference frames is rendered from the reconstructed Gaussian states in the input cameras and supplied to later motion predictions. SFC is evaluated only during training, using the available training-pair depths; deployment does not require externally supplied target-frame depth maps.

Relation to IGS. Our implementation builds on the public IGS codebase (Yan et al., 2025). IGS already uses optical-flow features, viewpoint/depth modulation, FPS anchors, a Transformer, neighboring-anchor interpolation, and residual motion decoding. It also introduces key-frame propagation with max-point bounded refinement. We retain these components and adapt them to sparse cameras using WAFT features, explicit visibility-depth weighted fusion, SFC regularization, and the Co-Adaptation-of-3DGS rasterizer for refinement. Thus, optical-flow encoding, modulation, and bounded key-frame refinement are inherited designs, while our evaluation examines their sparse-view adaptation.

Notation and numerical settings. Gaussian g_i has center $\mathbf{x}_i$ and orientation $\mathbf{q}_i$; anchor j has position $\mathbf{a}_j$. The counts are N_g and N_a, respectively, with $N_a = 8192$ by default. Anchor descriptors $\mathbf{m}_j^a$ are interpolated into Gaussian descriptors $\mathbf{m}_i^g$ using $K = 8$ neighbors and weights proportional to $\exp(-10\|\mathbf{x}_i - \mathbf{a}_j\|_2)$. FPS is restricted to the scene bounding box; if fewer than N_a centers are available, sampled indices are repeated to preserve the input size. In depth weighting, the sampled depth is floored at 0.05, inverse depths are normalized across views, and the weighted-sum denominator is clamped to at least one. The translation head is initialized to zero and the relative-quaternion head to $(1, 0.01, 0.01, 0.01)$, a nonzero near-identity rotation; quaternion normalization is applied for rendering.

Table 6 summarizes first-frame Gaussian reconstruction quality, time, storage, and point count; Time in this table refers to frame-0 reconstruction. For the N3DV scenes, we set the SH degree to 3, whereas for Meeting Room it was set to 1 to mitigate overfitting caused by sparse viewpoints. During the Max-points Bounded Recalibration process, all scenarios used the same learning rate settings. Specifically, the learning rate for position and rotation was set to ten times that in 3DGS, while the other parameters were kept consistent with 3DGS.

The maximum points number N_{max} was determined based on the number of Gaussians in the initial frame of each scene. Specifically, N_{max} was set to 200,000 for N3DV and 60,000 for the Meeting Room dataset.

C.2 Detailed Sparse-view Flow Consistency Loss

In the sparse-view setting (e.g., three input cameras), supervision from only a few views can encourage the motion network to overfit camera-specific 2D cues. As shown in Figure 2, we therefore introduce a *Sparse-view Flow Consistency* (SFC) loss that uses calibrated cameras and the precomputed training-pair depths to impose a shared geometric interpretation across views, regularizing the learned motion representation for better sparse-view generalization.

For each view $v \in \{v_0, v_1, v_2\}$, the backbone outputs a forward flow $\Delta_v \in \mathbb{R}^{2\times H\times W}$. In the lift below, $d_v(\mathbf{u}) = D_v(\mathbf{u})$ is the precomputed reference-frame depth. Given intrinsics $\mathbf{K}_v$ and pose $\mathbf{T}_v \in \mathrm{SE}(3)$, $\pi_v^{-1}(\mathbf{u}, d)$ back-projects a pixel with depth d from view v to a world-space point

Table 6: Reconstruction results of Gaussian models for the first frame in each scenario.

Scene	PSNR↑ (dB)	Time ↓ (s)	Storage↓ (MB)	Points num
N3DV (Li et al., 2022b)				
cut roasted beef	28.57	1.22	13.2	58435
sear steak	28.60	1.38	12.1	53568
Meeting Room (Li et al., 2022a)				
trimming	25.06	1.05	7.0	31205
vrheadset	24.35	0.99	7.4	32833

$\mathbf{P}_v(\mathbf{u})$, including the camera-to-world transform $\mathbf{T}_v$. Its inverse projection π_v includes $\mathbf{T}_v^{-1}$ and the intrinsics:

$$\mathbf{P}_v(\mathbf{u}) = \pi_v^{-1}(\mathbf{u}, d_v(\mathbf{u})), \qquad \mathbf{P}'_v(\mathbf{u}) = \pi_v^{-1}(\mathbf{u} + \Delta_v(\mathbf{u}), d_v(\mathbf{u})). \tag{17}$$

This expression reuses the reference depth for the displaced pixel and therefore imposes a constant-depth approximation along the source camera's depth axis. Availability of a target-frame depth map does not imply its use in this particular formula. The resulting consistency constraint is a depth-conditioned motion proxy; it does not independently recover general 3D scene flow or resolve occluded correspondences. We back-project and re-project both points into a second view $w \neq v$ to obtain the geometry-consistent flow

$$\Delta_{v\to w}(\mathbf{u}) = \pi_w(\mathbf{P}'_v(\mathbf{u})) - \pi_w(\mathbf{P}_v(\mathbf{u}))\,. \tag{18}$$

Let $\mathbf{u}_w = \pi_w(\mathbf{P}_v(\mathbf{u}))$. The pairwise loss is

$$\mathcal{L}_{v\to w} = \frac{1}{|\Omega_{v,w}|} \sum_{\mathbf{u}\in\Omega_{v,w}} \|\Delta_w(\mathbf{u}_w) - \Delta_{v\to w}(\mathbf{u})\|_1\,, \tag{19}$$

where $\Omega_{v,w}$ contains valid samples with positive depth, in-frustum reprojection, and in-bounds coordinates. Aggregating over all view pairs $\mathcal{P} = \{(v_0, v_1), (v_0, v_2), (v_1, v_2)\}$ gives

$$\mathcal{L}_{\text{sfc}} = \frac{1}{|\mathcal{P}|} \sum_{(v,w)\in\mathcal{P}} \mathcal{L}_{v\to w}. \tag{20}$$

SFC is combined with the standard Gaussian Splatting rendering losses in Eq. (10):

$$\mathcal{L} = \lambda_{\text{rgb}}\,\mathcal{L}_{\text{rgb}} + \lambda_{\text{sfc}}\,\mathcal{L}_{\text{sfc}}. \tag{21}$$

Here $\mathcal{L}_{\text{rgb}}$ supervises the deformed Gaussian renderings against target-frame ground-truth images in the reference–target training pairs on the training views. Δ_v is taken from the final flow estimate after iterative refinement in the optical-flow backbone. By constraining view-specific predictions only during training, $\mathcal{L}_{\text{sfc}}$ promotes transfer beyond the observed cameras and is omitted at inference without additional runtime.

D Per-sequence Quantitative Results

Table 1 in the main paper follows the six-scene N3DV protocol commonly reported by prior work. Although our model is trained on four of those scenes and evaluated on the other two, many published baselines provide only their aggregate results over all six scenes. We therefore retain the six-scene averages in the main table to enable comparison with those reported numbers. Table 7 instead isolates the two held-out scenes, *sear_steak* and *cut_roasted_beef*. For this stricter held-out protocol, we reproduced the corresponding methods and report their per-scene rendering results below.

S^{2}4D yields the best result on both held-out scenes for all three metrics. On *sear_steak*, it improves over the strongest reproduced baseline for each metric by 0.11 dB PSNR and 0.001 SSIM, while reducing LPIPS from 0.225 to 0.168. On *cut_roasted_beef*, it improves the strongest baseline for each metric by 0.73 dB PSNR and 0.019 SSIM, and reduces the best baseline LPIPS from 0.181 to 0.179. These held-out results complement rather than numerically decompose the six-scene aggregate in Table 1, because the latter is retained to match the reporting protocol available for prior work.

Table 7: Reproduced per-scene results on the two held-out N3DV scenes.

Method	Type	sear_steak			cut_roasted_beef		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Optimization-Based							
LocalDyGS	Offline	23.52	0.796	0.231	20.09	0.740	0.251
MonoFusion	Offline	21.71	0.739	0.268	19.98	0.666	0.287
SparseCam4D	Offline	23.07	0.812	0.229	21.79	0.752	0.251
4C4D	Offline	23.70	0.830	0.251	21.50	0.827	0.187
3DGStream	Online	21.05	0.766	0.261	20.34	0.741	0.288
QUEEN	Online	19.11	0.652	0.252	18.11	0.603	0.273
Recon-GS	Online	23.34	0.830	0.447	21.69	0.826	0.181
Feed-Forward							
IGS	Feed-forward	23.05	0.817	0.225	21.94	0.767	0.253
TTTLRM	Feed-forward	17.60	0.788	0.401	16.81	0.645	0.444
AnySplat	Feed-forward	16.11	0.758	0.452	15.27	0.666	0.502
WorldMirror	Feed-forward	17.49	0.796	0.441	16.91	0.739	0.390
DGGT	Feed-forward	17.01	0.771	0.375	16.01	0.751	0.401
$S^2$4D (Ours)	Feed-forward	**23.81**	**0.831**	**0.168**	**22.67**	**0.846**	**0.179**

E Practical Deployment under Different Camera Counts

The number and placement of cameras directly determine the capture cost and the geometric constraints available to a practical 4D reconstruction system. SparseCam4D (Pan et al., 2026) likewise evaluates reconstruction with different numbers of input views. Its default sparse setting selects the smallest camera subset that sufficiently covers the observed scene while maintaining overlap between adjacent views, which enables its video diffusion model to interpolate intermediate viewpoints reliably. Following this deployment criterion, we use $V = 3$ as the minimum subset for Meeting Room (Li et al., 2022a): the selected views cover the scene while retaining adjacent-view overlap, and all remaining cameras are reserved for novel-view evaluation.

To study the quality–capture-cost trade-off, we follow the camera-count protocol of SparseCam4D and define a 3-, 6-, and 9-view evaluation on Meeting Room. All settings use the same scenes and output resolution. For each input-camera count, we evaluate all remaining cameras that are excluded from reconstruction and report the mean over these unseen viewpoints, followed by averaging over frames and scenes. The evaluation-camera set therefore depends on the selected input cameras in each setting.

Table 8: Performance on Meeting Room with different numbers of input camera views. Each setting averages results over its remaining unseen viewpoints, then over frames and the *trimming* and *vrheadset* sequences at 1280×720.

	3 views			6 views			9 views		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
4DGaussian	15.44	0.413	0.574	18.51	0.559	0.459	21.02	0.611	0.421
SparseCam4D	20.13	0.745	0.237	22.45	0.768	0.222	24.01	0.791	0.198
$S^2$4D-H (Ours)	**20.27**	**0.841**	**0.196**	**22.95**	**0.777**	**0.215**	**24.97**	**0.802**	**0.189**

Increasing the camera count consistently raises PSNR for all three methods. $S^2$4D remains best at every camera count, exceeding SparseCam4D by 0.14, 0.50, and 0.96 dB with 3, 6, and 9 views, respectively. It also achieves the best SSIM and LPIPS in every setting. Although its perceptual metrics are not strictly monotonic from three to six views, the nine-view setting reaches its highest PSNR and lowest LPIPS, whereas the three-view setting has the highest SSIM. The strong three-view result supports the practical sparse-camera deployment choice.

F More Ablation Study

In our experiments, we also explored incorporating additional modules into SAM-Net. However, the results showed that these modules did not achieve the expected improvements. The ablation studies are detailed as follows.

Pose-conditioned multi-view attention. Rather than aggregating per-view motion features with the visibility–depth weighted mean used in the main paper, we tested a learned alternative. For each anchor, the sampled feature from every camera is concatenated with a pose embedding of that view; the resulting N_v tokens are passed through self-attention and softmax to obtain fusion weights. As reported in Table 9, this design brings no gain on N3DV test scenes. We conjecture that the dataset's forward-facing layout induces only mild viewpoint discrepancy, so explicit attention is unnecessary; the idea may still be useful for wide-baseline or 360° capture.

Point Transformer V3 for anchor interaction. After lifting multi-view features onto anchors, the main pipeline contextualizes the N_a descriptors with a standard Transformer. To reduce the quadratic cost of dense self-attention over thousands of anchors, we replaced this block with Point Transformer V3 (Wu et al., 2024b), which models local geometric neighborhoods among anchors more efficiently. Table 9 shows that the substitution does not improve reconstruction quality in our setting. We hypothesize that the compact anchor set already admits sufficient context from a lightweight Transformer, while the inductive bias of local point operators adds little for this motion-propagation task and may discard useful long-range couplings.

Table 9: Additional ablation study results.

Method	PSNR (dB)↑
+ Pose-cond. multi-view attn.	21.44
+ Point Transformer V3 (anchor relation)	21.10
Ours	**23.12**

G Additional Discussion

G.1 Temporal Flicker in Background Regions

The supplementary video shows that temporal flicker under our pipeline concentrates on largely static backgrounds. Local photometric updates at refinement timestamps tend to perturb these regions, while consecutive SAM-Net propagation steps stay comparatively smooth over time. We attribute this to refinement-driven displacement of background primitives, which becomes more pronounced when the initial sparse-view reconstruction includes floating artifacts. By contrast, dynamic foreground motion is regularized by SAM-Net, which yields smoother inter-frame point updates and avoids the flicker patterns reported for 3DGStream. Mitigating this issue may involve separating foreground from background and restricting refinement with the resulting mask, together with improving the quality of the first-frame Gaussian reconstruction under sparse cameras.

H More Limitations and Future Work

We further outline remaining bottlenecks of $S^2$4D and corresponding directions for improvement.

First, causal streaming updates remain sensitive to the fidelity of the static Gaussian map built at frame 0 (Sun et al., 2024). Spurious floaters around moving regions in this initialization can propagate into SAM-Net and hurt later frames. Designing better static reconstructors is not our contribution here; nevertheless, a cleaner starting representation would strengthen the overall dynamic pipeline.

Second, due to limited computational resources, we train the generalizable motion model on only four N3DV scenes. Although the cross-dataset results demonstrate transfer to the evaluated benchmarks, this small and relatively homogeneous training set cannot establish robust generalization to more complex unseen scenes, motions, appearances, or topologies. Expanding training to broader

and more diverse multi-view videos is therefore an important direction for improving cross-domain generalization. Since our supervision is based primarily on photometric view-synthesis objectives, such scaling does not require ground-truth motion fields.

Third, our experiments use established benchmarks with synchronized, calibrated cameras and do not yet constitute deployment tests in uncontrolled sparse-camera environments. Practical usability should be validated on real captures and analyzed under a more detailed taxonomy of scene and acquisition conditions, including camera count, baseline and spatial coverage, forward-facing versus surrounding layouts, motion and occlusion complexity, and calibration or synchronization noise. Evaluating these factors separately would clarify where the method remains reliable and where additional adaptation is required.

Fourth, geometry awareness is injected by conditioning an optical-flow encoder with depth and viewing rays. Adopting stronger long-range flow models (Wu et al., 2023) or dedicated scene-flow networks (Mehl et al., 2023; Teed & Deng, 2021) in place of this backbone is expected to refine motion estimates and, in turn, reconstruction quality.

I Extended Related Work

I.1 Optimization-based Dynamic Reconstruction

Dynamic novel-view synthesis has progressed from spacetime radiance fields (Fridovich-Keil et al., 2023; Cao & Johnson, 2023) to explicit 3D Gaussians (Kerbl et al., 2023). Optimization-based 4D methods mainly follow two practical paradigms. *Chunk-based* approaches fit a single spatiotemporal model after collecting the full sequence (Wu et al., 2024a; Yang et al., 2024b; Li et al., 2024; Yang et al., 2024a; Huang et al., 2024a; Wu et al., 2025b;a), but must wait for the complete video and cannot support live free-viewpoint services. *Streaming* approaches refresh the representation as frames arrive (Li et al., 2022a; Wang et al., 2023b; Sun et al., 2024; Girish et al., 2024; Fu et al., 2025; Lee et al., 2026), enabling causal updates yet still relying on costly per-frame optimization with high latency and long-horizon drift (Sun et al., 2024; Yan et al., 2025). Since interactive free-viewpoint viewing requires continuous causal reconstruction (Li et al., 2022a; Sun et al., 2024), we retain the streaming formulation. Despite strong rendering quality, these methods remain scene-specific optimization pipelines with limited cross-scene generalization. **Our method instead learns a feed-forward generalizable motion prior for cross-scene streaming inference, substantially accelerating causal reconstruction over conventional optimization-based streaming systems** (Sun et al., 2024; Girish et al., 2024).

I.2 Feed-forward Dynamic Reconstruction

Feed-forward models learn rich cross-scene priors and reconstruct unseen scenes without heavy per-scene optimization. Early Gaussian generalizable models regress multi-view features into per-pixel Gaussians (Charatan et al., 2024; Chen et al., 2025c; Xu et al., 2025a; Zhang et al., 2024b), while DUSt3R and VGGT scale joint geometry–camera inference through large-scale pretraining (Wang et al., 2024; 2025a). Follow-ups further strengthen geometric priors and Gaussian-based view synthesis (Wang et al., 2026b; Lin et al., 2025; Jiang et al., 2025b; Ye et al., 2026; Liu et al., 2026; Jin et al., 2025; Jiang et al., 2025a). Dynamic extensions predict time-varying Gaussians from monocular or multi-view inputs (Ren et al., 2024; Lin et al., 2026; Xu et al., 2025b; Chen et al., 2025b; Hur et al., 2026; Balice et al., 2026; Yan et al., 2025). However, many process fixed sequences offline or decode timestamps independently at moderate resolutions, so naive frame-by-frame application struggles to guarantee high-resolution, temporally coherent free-viewpoint video. Recent online state-update models make streaming reconstruction more practical (Wang et al., 2025b; Chen et al., 2026b; Wang et al., 2026a; Zhuo et al., 2026; LAN et al., 2026), yet mainly target monocular videos or multi-view image collections rather than synchronized multi-view streams, and typically operate only at low or moderate resolutions. **In contrast, our method is a streaming feed-forward framework that, most importantly, enables high-resolution 4D reconstruction for immersive viewing while retaining cross-scene generalizable motion transfer.**

I.3 Sparse-View Static and Dynamic Reconstruction

Outside controlled studios, dense synchronized camera arrays are rarely available (Pan et al., 2026; Zhou et al., 2026). Static sparse-view NeRFs mitigate under-constrained optimization with semantic or geometric regularization (Jain et al., 2021; Niemeyer et al., 2022; Wang et al., 2023a), but remain slow to optimize. Sparse-view 3DGS further reduces floaters and overfitting via depth-guided densification, floater pruning, and co-regularization (Zhu et al., 2024; Xiong et al., 2025; Zhang et al., 2024a; Paliwal et al., 2024), yet still requires costly scene-specific fitting. Feed-forward multi-view models provide fast initialization (Charatan et al., 2024; Chen et al., 2025c; Fan et al., 2024), but are mainly designed for static scenes. Sparse dynamic reconstruction is harder because geometry, appearance, and motion must be estimated jointly; recent sparse-camera/frame 4D systems add visibility-aware opacity decay, generative completion, or geometric/skeleton priors (Zhou et al., 2026; Pan et al., 2026; Wang et al., 2025c; Chao & Isler, 2026; Shi et al., 2026), yet largely remain offline scene-specific optimizations. **We instead target generalizable streaming 4D reconstruction under sparse multi-view deployment**.

J Dynamic Gaussian Splatting Background

To capture temporal dynamics, recent approaches (Yang et al., 2024b; Wu et al., 2024a) attach a time-conditioned deformation field to the canonical representation. Using the Gaussian attributes defined in Section 3.1, a representative additive deformation model predicts residuals at timestamp τ:

$$(\delta\mathbf{x}_i, \boldsymbol{\eta}_i^q, \delta\mathbf{s}_i) = \mathcal{F}_\theta(\gamma(\mathrm{sg}(\mathbf{x}_i)), \gamma(\tau)) \,. \tag{22}$$

Here $\mathcal{F}_\theta$ denotes the deformation network, $\gamma(\cdot)$ is positional encoding (Mildenhall et al., 2021), and $\mathrm{sg}(\cdot)$ stops gradients through canonical coordinates. The warped attributes are $(\mathbf{x}_i + \delta\mathbf{x}_i, \mathbf{s}_i + \delta\mathbf{s}_i, \mathrm{unit}(\mathbf{q}_i + \boldsymbol{\eta}_i^q), o_i, \mathbf{h}_i)$ and are rasterized with Eq. (1). Here $\boldsymbol{\eta}_i^q$ is an additive quaternion residual, distinct from SAM-Net's relative quaternion $\delta\mathbf{q}_i$ composed by the Hamilton product. Both the canonical Gaussians and $\mathcal{F}_\theta$ are trained by minimizing a photometric objective:

$$\mathcal{L}_{\mathrm{rgb}} = (1-\lambda)\|\hat{I} - I\|_1 + \lambda\big(1 - \mathrm{SSIM}(\hat{I}, I)\big). \tag{23}$$