PF-SGS: Pose-Free Streaming 3D Gaussian Splatting for Online Scene Reconstruction

Wenjie Mu[1,*] **Ziniu Liu**[1,*] **Tong Wu**[1] **Zhan Li**[1] **Chuanzhou Su**[2] **Xuanyi Shen**[3]
Fan Lu[1] **Junqiao Zhao**[1,†] **Tiantian Feng**[1] **Chen Ye**[1] **Guang Chen**[1,4]
[1]Tongji University [2]The University of Hong Kong
[3]Columbia University [4]Shanghai Innovation Institute

Abstract

Extending feed-forward 3D Gaussian Splatting (3DGS) to long sequences remains challenging. Existing offline reconstruction methods require accessing and jointly encoding all views in advance, making them unsuitable for dynamically growing scenes; while streaming-based online reconstruction methods offer superior scalability, their inherent forgetting behaviors and unidirectional dependencies limit reconstruction quality. In this paper, we propose PF-SGS, a pose-free Streaming feed-forward 3DGS framework. PF-SGS leverages a persistent hidden state as scene memory to integrate streaming inputs frame-by-frame, progressively recovering scene geometry and camera poses in a self-supervised manner. At its core lie two targeted designs: Adaptive State Gating, which suppresses long-term memory degradation by dynamically regulating update magnitudes to ensure stable state evolution; and Delayed Gaussian Modeling, which introduces bounded look-ahead observations to compensate for insufficient local geometric constraints under streaming inputs, thereby improving geometric consistency. Experiments demonstrate that PF-SGS processes sequences of varying lengths with linear time complexity and maintains reliable pose estimation and reconstruction quality, matching or outperforming existing offline and online reconstruction methods.

1 Introduction

Feed-forward 3D Gaussian Splatting (3DGS) [1, 2, 3] enables real-time novel view synthesis in a single forward pass, emerging as an efficient paradigm for 3D reconstruction. However, extending feed-forward 3DGS to long sequences remains challenging [4, 5]; compared with sparse-view reconstruction, long sequences involve longer camera trajectories and denser observations, demanding efficient inference while preserving stable global consistency.

Recent methods have explored long-sequence modeling. For instance, GS-LRM [6] and Long-LRM [7] leverage global attention and a hybrid Mamba-Transformer, to model cross-view geometric relationships, thereby obtaining geometrically consistent scene representations; AnySplat [8] distills pose and geometry priors from VGGT [9] to enable fast reconstruction over long input sequences. However, above methods typically follow a offline reconstruction paradigm—all images are assumed to be available beforehand and jointly encoded in a single forward pass for holistic reconstruction, which not only struggles to adapt to dynamically growing scenes, but also incurs prohibitive memory consumption and inference latency on long sequences.

Another line of work explores streaming-based online reconstruction paradigms that aim to achieve incremental modeling over long-range trajectories. For example, FreeSplat [10] and LongSplat [11] achieve dynamic expansion over long sequences by decoding Gaussian parameters frame by frame;

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

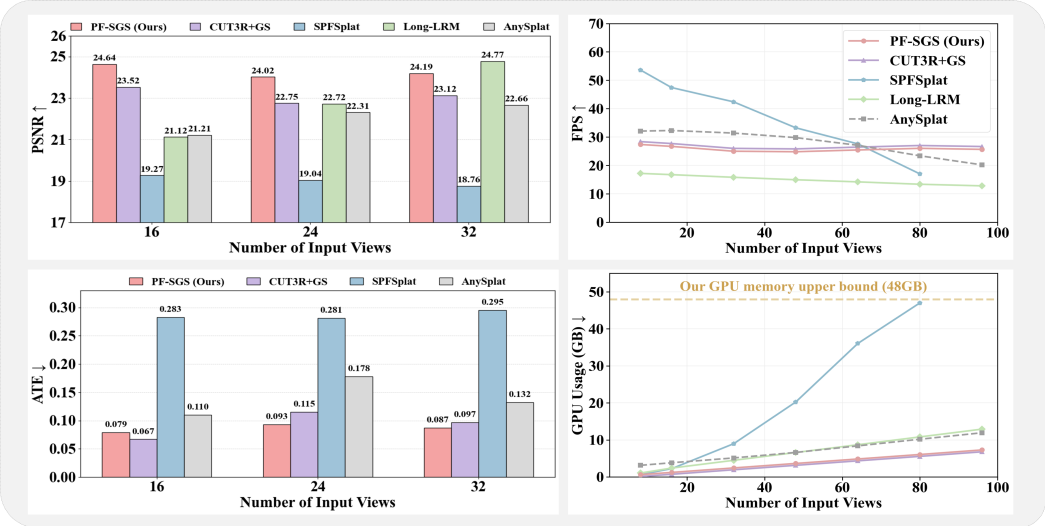

Figure 1: **The Performance Comparison of Baselines and PF-SGS.** PF-SGS achieves rendering and pose estimation performance comparable to methods that rely on ground-truth poses, without requiring any prior, while maintaining linear time complexity with respect to sequence length.

StreamGS [12] leverages a frozen DUSt3R [13] to provide initial geometry information, thereby supporting frame-wise Gaussian aggregation. However, these methods heavily rely on accurate pose and geometric priors, limiting their applicability in real-world scenarios. In contrast, CUT3R [14] learns recurrent state modeling under the supervision of ground-truth poses and 3D point maps, enabling continuous estimation of camera motion and scene geometry from unposed image streams. Despite this capability, it remains constrained by two inherent limitations: **1)** recurrent updates continuously overwrite historical memory with instantaneous observations, leading to progressive memory degradation over long sequences; **2)** the unidirectional streaming inference restricts the model to relying only on historical memory and the current frame, resulting in insufficient local geometric constraints. Crucially, in the absence of ground-truth pose supervision, these issues are further amplified during the long-sequence modeling.

To improve the stability of long-sequence modeling, we propose PF-SGS, a pose-free streaming feed-forward 3DGS framework PF-SGS maintains a persistent hidden state as scene memory, encoding input images frame-by-frame and synchronously updating this memory under self-supervised constraints, enabling continuous estimation of camera poses and 3D Gaussian representations without relying on any priors. Specifically, to suppress memory degradation in long-sequence modeling, we introduce Adaptive State Gating, which adaptively regulates the memory update magnitude based on the current observation and historical state, thereby stabilizing geometry and pose estimation; meanwhile, Delayed Gaussian Modeling leverages bounded look-ahead observations to supplement geometric cues for current-frame modeling, alleviating the deficiency of geometric constraints under streaming inputs. As illustrated in Fig. 1, extensive experiments demonstrate that PF-SGS delivers performance comparable to pose-required baselines without relying on any priors, and thus provides a viable solution for long-sequence modeling.

2 Related Work

2.1 Feed-forward 3D Gaussian Splatting

Feed-forward 3DGS has emerged as one of the dominant paradigms in 3D reconstruction, owing to its strong generalization capability and instant reconstruction speed [1, 2, 15, 16, 17]. Early representative methods such as PixelSplat [18], MVSplat [19], and DepthSplat [20] achieve high-quality scene reconstruction via multi-view feature aggregation. However, their performance is tightly coupled with accurate camera pose priors. To address this, subsequent works gradually alleviate this dependency, GGRt [21] and PF3plat [22] employ joint optimization of poses and Gaussian representations, whereas SelfSplat [23] adopts a self-supervised approach for pose and

Gaussian prediction. Other methods incorporate mature geometric backbones to strengthen geometric reasoning: NoPoSplat [24] builds upon MASt3R [25], while YoNoSplat [26] incorporates π^3 [27] for more reliable geometric representations. Nevertheless, these methods [28, 29, 30, 31] primarily focus on sparse-view scene modeling and lack mechanisms for handling long-range context, making them difficult to maintain stable reconstruction performance over long sequences.

2.2 Long-sequence Scene Reconstruction

Offline reconstruction method Such methods typically assume that all images of a scene are available at inference time, and perform joint encoding for holistic reconstruction in a single forward pass [7, 9, 8]. GS-LRM [6] and LVT [32] leverage Transformers to model long-range dependencies across global observations; Long-LRM [7] adopts a hybrid Mamba-Transformer architecture, incorporating lightweight token merging and gaussian pruning to balance quality and efficiency. In contrast, LVSM [33] and RayZer [34] forgo explicit 3D representation construction in favor of broader view coverage, instead relying on global latent representations and cross-view attention for end-to-end novel view synthesis. Although these methods [35, 36] demonstrate strong scene modeling capabilities, their inference paradigm struggles with dynamically growing scenes and incurs excessive memory consumption and latency as input views increase.

Online reconstruction method These methods process image sequences frame-by-frame and incrementally expand the scene representation [37, 38, 39, 40, 41]. FreeSplat [10] strengthens inter-view geometric constraints by introducing cost volumes, validating the feasibility of long-sequence online reconstruction in indoor scenes; LongSplat [11] incorporates a streaming 3DGS update mechanism that selectively compresses redundant historical gaussians to maintain scalability across prolonged sequences; To enhance modeling efficiency, CUT3R [14] introduces persistent hidden states to compress scene information, updating scene representations through continuous interaction with current observations. Critically, these methods rely on accurate camera poses—whether as inputs or as supervisory signals—which can be difficult to obtain or maintain in complex environments. In contrast, PF-SGS inherits the advantages of hidden state-based modeling while eliminating dependency on ground-truth camera poses, providing a more competitive solution.

3 Preliminaries

3.1 Pose-free Feed-forward 3DGS

Pose-free feed-forward 3DGS aims to learn a direct mapping from multi-view images to an explicit 3D Gaussian representation without ground-truth pose inputs [42, 43]. Given N input images $\{I_i\}_{i=1}^N$, where $I_i \in \mathbb{R}^{H\times W\times 3}$, and corresponding intrinsics $\{Z_i\}_{i=1}^N$, it is formulated as:

$$f_\Theta : \{(I_i; Z_i)\}_{i=1}^N \mapsto \{(\mu_{i,j}, o_{i,j}, \Sigma_{i,j}, c_{i,j})\}_{i=1,\dots,N}^{j=1,\dots,H\times W} \tag{1}$$

where Θ denotes the learnable parameters of the feed-forward network. Each Gaussian primitive $\mathcal{G}_i$ is jointly defined by its center position μ, opacity o, covariance matrix Σ, and spherical harmonic coefficients c. It is important to note that all Gaussian centers μ are defined in a canonical space referenced to the camera coordinate system of the first frame [13, 25]. For a given target viewpoint, the 3D Gaussian primitives in canonical space are projected onto the image plane and aggregated into per-pixel color $\hat{I}_i$ and depth $\hat{D}_i$ via a differentiable Gaussian rasterizer [44].

3.2 Persistent State Representation

To facilitate online reconstruction under streaming inputs, CUT3R [14] maintains a persistent hidden state, which is modeled as a set of learnable tokens $S_i \in \mathbb{R}^{L\times C}$ to dynamically encode the geometric and appearance characteristics of the scene. Specifically, given the image features $F_i = \mathcal{E}(I_i)$ of the i-th frame, the interaction between the current observation and historical memory is decomposed into a state-update and a state-readout process. First, the current observation F_i is written into the previous state S_{i-1} to update the scene memory:

$$S_i = \Phi_{\text{update}}(S_{i-1}, F_i) \tag{2}$$

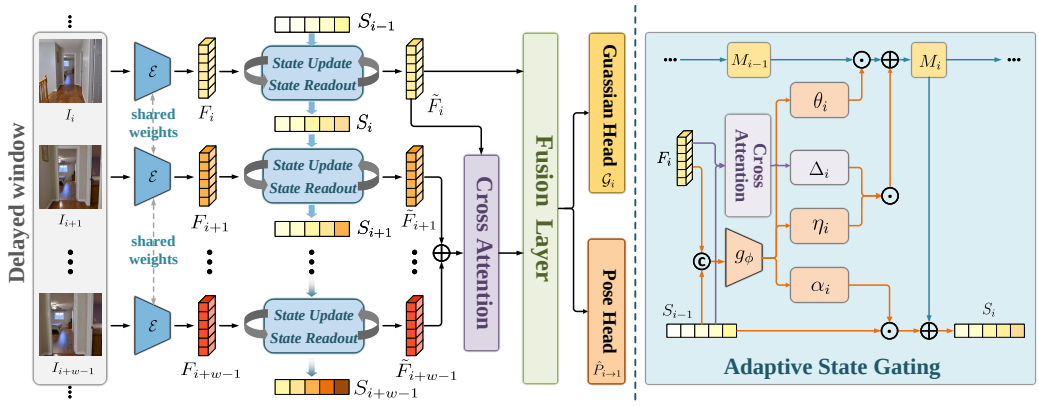

Figure 2: **Overview of our method.** PF-SGS first extracts frame-wise features $\{F_i, \ldots, F_{i+w-1}\}$ within a delayed window of length w using a shared encoder $\mathcal{E}$. Next, F_i interacts with the previous state S_{i-1} through a dual process: State Readout injects historical context into F_i to produce $\tilde{F}_i$, while State Update leverages Adaptive State Gating **(right)** to dynamically regulate the update magnitude from F_i to S_{i-1}, yielding S_i. Finally, $\tilde{F}_i$ aggregates local geometric constraints from look-ahead features $\{\tilde{F}_{i+1}, \ldots, \tilde{F}_{i+w-1}\}$ via cross-attention, resulting in a fused representation that is subsequently decoded into Gaussian primitives $\mathcal{G}_i$ and relative pose $\hat{P}_{i\to 1}$.

where S_i denotes the updated hidden state. Then, the model reads historical context from the updated state S_i and injects it into the current image feature F_i, yielding the enhanced feature $\tilde{F}_i$:

$$\tilde{F}_i = \Phi_{\text{read}}(F_i, S_i) \tag{3}$$

Both Φ_{update} and Φ_{read} are performed through cross-attention, enabling the model to accumulate long-term geometric memory under streaming inputs without explicitly caching all historical frames.

4 Method

Fig. 2 presents an overview of our method. PF-SGS maintains a persistent hidden state S_i as scene memory, integrating streaming inputs recurrently to continuously estimate camera poses and Gaussian primitives. To preserve geometric consistency and training stability during long-sequence modeling, we introduce Adaptive State Gating (Sec. 4.1) and Delayed Gaussian Modeling (Sec. 4.2): the former dynamically regulates the magnitude of memory updates according to the historical state and current observation, thereby stabilizing state updates; the latter aggregates local look-ahead observations to provide richer geometric cues for current-frame modeling. Driven by self-supervised losses (Sec. 4.3), PF-SGS achieves robust online scene reconstruction without any priors.

4.1 Adaptive State Gating

In recurrent state-based online reconstruction frameworks, the current observation is continuously integrated into a fixed-capacity hidden state via cross-attention, and the standard state update in Eq. 2 can be formulated as:

$$\Delta_i = CrossAttn(S_{i-1}, F_i) = Softmax\left(\frac{Q_{S_{i-1}} K_{F_i}^{\top}}{\sqrt{d}}\right) V_{F_i} \tag{4}$$

$$S_i = S_{i-1} + \Delta_i \tag{5}$$

where Δ_i denotes the instantaneous update residual introduced by F_i. In essence, the hidden state in pose-free streaming reconstruction serves as a recurrently evolving implicit canonical scene representation shared across views. Under the pose-free setting, the hidden state is required not only to accumulate scene appearance, but also to maintain stable geometric consistency throughout recurrent updates. Consequently, stable state evolution becomes crucial for long-sequence modeling. However, standard additive state updates lack an explicit mechanism to adaptively regulate how

current observations modify the historical geometric state. As recurrent geometric inference proceeds, noisy or geometrically inconsistent observations may gradually perturb the latent canonical structure, causing accumulated geometric drift and degraded reconstruction consistency. This issue becomes particularly severe under self-supervised pose-free settings, where the absence of explicit pose constraints makes recursive alignment errors more likely to accumulate over long sequences.

Recent studies in long-horizon recurrent modeling [45, 46, 47] suggest that information contributions across different time steps should be adaptively regulated during recurrent inference, rather than being subjected to uniform normalization. Inspired by this, we propose an Adaptive State Gating (ASG). Unlike conventional direct additive updates, ASG conditions on both the historical hidden state and the current observation to dynamically generate gating coefficients, thereby adaptively regulating state evolution to support robust long-sequence modeling.

Specifically, we first perform global pooling over the current features F_i and the previous hidden state S_{i-1}, producing the statistical vectors $\bar{f}_i$ and $\bar{s}_{i-1}$, respectively. These vectors are then concatenated and fed into a lightweight MLP g_ϕ to predict a set of multi-dimensional gating variables $\{\alpha_i, \theta_i, \eta_i\}$:

$$[\alpha_i, \theta_i, \eta_i] = \sigma\left(g_\phi\big([\bar{f}_i; \bar{s}_{i-1}]\big)\right), \quad \alpha_i, \theta_i, \eta_i \in \mathbb{R}^{L\times 1} \tag{6}$$

where $\sigma(\cdot)$ is the Sigmoid function. Subsequently, we incorporate these gating variables into an auxiliary correction term M_i (with $M_0 = 0$). This term recursively combines the correction from the previous step with the current update residual, such that the state update is no longer a fixed-form direct additive update. Accordingly, Eq. 5 can be reformulated as:

$$M_i = \eta_i \odot M_{i-1} + \theta_i \odot \Delta_i \tag{7}$$

$$S_i = \alpha_i \odot S_{i-1} + M_i \tag{8}$$

where $\odot$ denotes element-wise multiplication. In this formulation, α_i, θ_i, and η_i modulate the contributions of the historical state S_{i-1}, the previous correction term M_{i-1}, and the instantaneous residual Δ_i, respectively. Therefore, the state update is no longer a direct additive process, but a data-driven gated recurrent process that jointly adjusts the contributions of the historical memory, accumulated correction, and current observation. Note that the conventional additive update can be viewed as a degenerate case of our formulation when $\alpha_i = \mathbf{1}$, $\theta_i = \mathbf{1}$, and $\eta_i = \mathbf{0}$.

4.2 Delayed Gaussian Modeling

Unlike offline reconstruction methods that have access to complete observations, streaming-based reconstruction must perform recursive geometric inference using only historical states and the current observation under strictly causal constraints. As a result, the current frame is often geometrically under-constrained due to insufficient cross-view overlap and incomplete triangulation cues, especially under large viewpoint changes.

To address this, we introduce Delayed Gaussian Modeling (DGM), which moderately relaxes the strict unidirectionality constraint of streaming inference by aggregating local look-ahead observations to provide richer geometric cues for current frame modeling. This design improves geometric consistency while preserving the streaming paradigm. Specifically, let w denote a fixed delayed window length. For the i-th frame I_i, Gaussian primitive prediction is not performed immediately upon input; instead, it is deferred until the sequence progresses to the $(i{+}w)$-th frame. We then use $\tilde{F}_i$ as the Query, and the concatenated subsequent features within the local window $\mathcal{C}_i = \{\tilde{F}_{i+1}, \tilde{F}_{i+2}, \ldots, \tilde{F}_{i+w-1}\}$ as Keys & Values to extract cross-frame geometric cues via cross-attention:

$$H_i = CrossAttn\Big(\tilde{F}_i, \mathcal{C}_i\Big), \quad Q = \tilde{F}_i, K \,\&\, V = \mathcal{C}_i \tag{9}$$

The resulting contextual feature H_i is then fused with $\tilde{F}_i$ to obtain the final enhanced representation $\bar{F}_i = \mathrm{Fuse}(\tilde{F}_i, H_i)$, which is fed into the Gaussian and Pose head to predict the corresponding Gaussian primitives $\mathcal{G}_i$ and the relative pose $P_{i\to 1}$ from the i-th frame to the canonical coordinate of the first frame [13]:

$$\mathcal{G}_i = \mathrm{Head}_{\mathrm{gauss}}(\bar{F}_i), \qquad P_{i\to 1} = \mathrm{Head}_{\mathrm{pose}}(\bar{F}_i). \tag{10}$$

For boundary cases with insufficient look-ahead frames, zero-vector padding is applied for alignment. Notably, the modeling target of DGM is strictly confined to the current frame I_i, while subsequent frames $\{I_{i+1}, \ldots, I_{i+w-1}\}$ are used solely to provide local geometric cues to enhance the cross-view constraints of I_i. As a result, the inference latency and additional computational overhead introduced by this design are inherently bounded.

4.3 Training Loss

Our total loss comprises rendering, reprojection, and photometric terms:

$$\mathcal{L}_{total} = \mathcal{L}_{render} + \mathcal{L}_{repro} + \mathcal{L}_{photo} \tag{11}$$

Rendering Loss To supervise the quality of synthesized views, we compute the pixel-wise error and perceptual error between the rendered image $\hat{I}_i$ and the ground-truth image I_i:

$$\mathcal{L}_{render} = ||I_i - \hat{I}_i||_2 + \gamma_1 \mathrm{LPIPS}(I_i, \hat{I}_i) \tag{12}$$

Reprojection Loss We introduce a reprojection loss [48] that imposes explicit geometric constraints on the 3D Gaussian centers in canonical space. For each pixel $q_{i,j}$ in the i-th frame I_i, we project its corresponding canonical Gaussian center $\mu_{i,j}$ onto the image plane of view i using the predicted relative pose $P_{i\to 1}$, and minimize the Euclidean distance to the pixel coordinate:

$$\mathcal{L}_{repro} = \sum_{i=1}^{N} \sum_{j=1}^{H\times W} \left\| q_{i,j} - \Pi\big(Z_i, P_{i\to 1}^{-1}, \mu_{i,j}\big) \right\|_2 \tag{13}$$

where $\Pi(\cdot)$ denotes the camera projection operation.

Photometric Loss This loss term [49, 50] aims to further enforce geometric consistency between adjacent frames, thereby improving pose estimation accuracy under long sequences:

$$\mathcal{L}_{photo} = \gamma_2 \frac{1 - \mathrm{SSIM}(I_i', I_i)}{2} + (1-\gamma_2) \left\| I_i' - I_i \right\|_1 \tag{14}$$

where I_i' denotes the image warped from I_r ($r < i$) to the viewpoint of I_i using the predicted depth $\hat{D}_i$ and relative pose $P_{i\to r} = P_{r\to 1}^{-1} P_{i\to 1}$.

5 Experiments

5.1 Datasets and Baselines

We adopt RE10K [51] and ACID [52] for training and follow prior split protocols [53, 54]. During testing, we evaluate in-domain performance on RE10K and zero-shot generalization on DL3DV-140 [55]. We compare with offline reconstruction baselines, including Long-LRM [7], AnySplat [8], and SPFSplat [56]. We additionally compare with CUT3R+GS for online reconstruction, which extends CUT3R [14] with a Gaussian prediction head. For novel view synthesis, we use PSNR, SSIM, and LPIPS. For pose estimation, we report ATE, RPE_t, and RPE_r for global trajectory accuracy and relative translational & rotational errors.

5.2 Experiment Settings

We employ the AdamW optimizer with a learning rate of 1×10^{-5} and a batch size of 6. Training is conducted for 250K iterations on four NVIDIA H800 ($\sim$ 5 days), and all evaluations are performed on a single NVIDIA A6000. The input sequence length is set to $N \in \{16, 24, 32\}$, and the sampling interval between frames is dynamically determined by the total frame count Sum and N (i.e., $interval = Sum/N$). We set the image resolution to 256×256, the delayed window size to $w = 4$, and the loss weights to $\gamma_1 = 0.05$ and $\gamma_2 = 0.85$.

5.3 Qualitative and Quantitative Comparison

Rendering Quality Comparison Table 1 and Fig. 3 respectively present the quantitative and qualitative comparisons. PF-SGS operates without any priors, yet consistently achieves leading performance across most settings, matching or even surpassing Long-LRM [7], a offline reconstruction baseline that requires ground-truth pose inputs. This demonstrates that online reconstruction can effectively accumulate cross-frame geometric cues, thereby enabling high-quality scene reconstruction. Compared with the ground-truth pose supervised CUT3R+GS [14], PF-SGS exhibits stronger

Table 1: **Quantitative comparison of rendering quality on RE10K and DL3DV.**

Method	Type	Views	RE10K			DL3DV		
			PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Long-LRM (ICCV 2025)	Offline Pose-input	16	21.12	0.789	0.243	18.27	**0.627**	0.371
		24	22.72	**0.813**	0.230	**20.05**	**0.643**	**0.341**
		32	**24.77**	**0.836**	0.201	**21.51**	**0.658**	**0.323**
CUT3R+GS (CVPR 2025)	Online Pose-sup	16	23.52	0.781	0.214	20.09	0.603	0.359
		24	22.75	0.754	0.240	19.33	0.569	0.391
		32	23.12	0.767	0.227	18.33	0.524	0.412
SPFSplat (ICCV 2025)	Offline Pose-free	16	19.27	0.610	0.329	14.64	0.332	0.570
		24	19.04	0.604	0.338	14.39	0.328	0.577
		32	18.76	0.595	0.352	14.27	0.320	0.580
AnySplat (TOG 2025)	Offline Pose-free	16	21.21	0.697	0.259	16.28	0.405	0.458
		24	22.31	0.730	0.234	16.71	0.425	0.457
		32	22.66	0.744	0.227	17.20	0.456	0.435
PF-SGS (Ours)	Online Pose-free	16	**24.64**	**0.817**	**0.168**	**20.61**	0.625	**0.310**
		24	**24.02**	0.801	**0.182**	20.01	0.592	0.325
		32	24.19	0.806	**0.179**	19.20	0.566	0.353

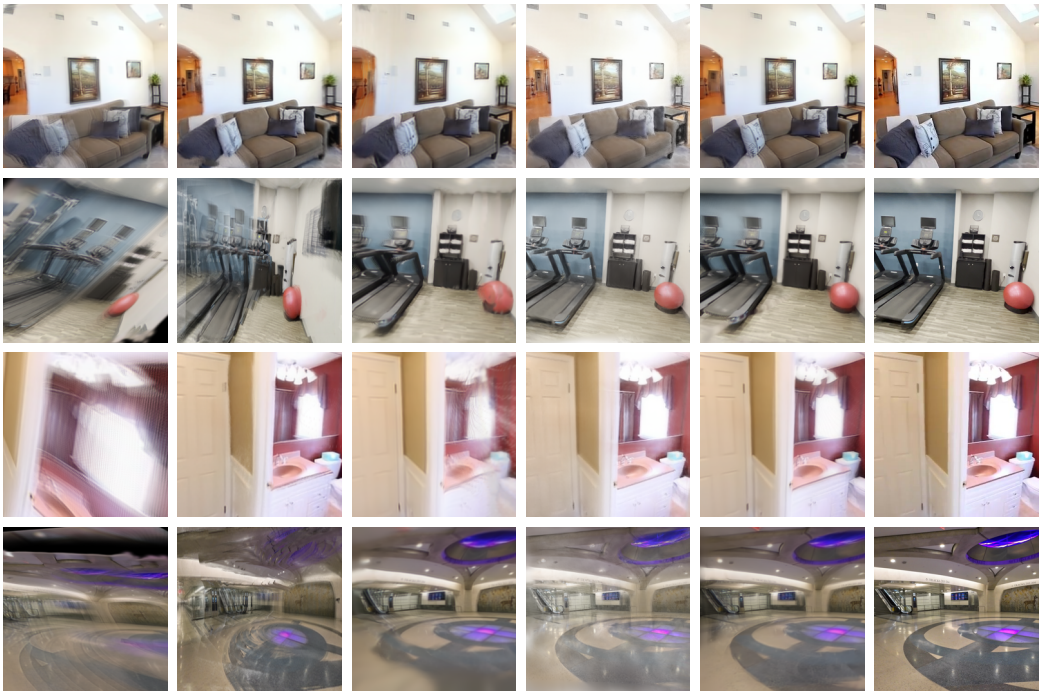

Figure 3: **Rendering quality comparison on RE10K and DL3DV.**

geometric consistency on long sequences, validating the effectiveness of the proposed method. Moreover, SPFSplat [56] suffers from a tight coupling between its modeling and specific target views. Due to the lack of a unified, observation-independent scene representation, it struggles to maintain stable rendering under the standard target pose-only protocol. Although similarly pose-free, AnySplat [8] is limited by the in-domain bias of its distillation source. In contrast, PF-SGS maintains robust performance in cross-domain evaluation, highlighting its robustness under distribution shifts.

Pose Estimation Comparison As shown in Table 2 and Fig. 4, PF-SGS can recover reliable camera motion from unposed streaming inputs. Among other pose-free methods, both SPFSplat [56] and AnySplat [8] exhibit inferior pose accuracy. SPFSplat [56] implicitly couples pose estimation with target-view features and prioritizes self-consistent alignment between poses and Gaussians representation, rather than long-range trajectory accuracy. AnySplat [8] heavily depends on the accuracy of pseudo labels distilled from a frozen VGGT [9], whose inaccuracies degrade its performance under complex motion. Compared with the ground-truth pose-supervised CUT3R+GS [14], PF-SGS achieves comparable overall performance and exhibits stronger robustness as the sequence length

Table 2: **Quantitative comparison of pose estimation on RE10K and DL3DV.**

Method	Type	Views	RE10K			DL3DV		
			ATE ↓	RPE_t ↓	RPE_r ↓	ATE ↓	RPE_t ↓	RPE_r ↓
CUT3R+GS (CVPR 2025)	Online Pose-sup	16	**0.067**	**0.013**	1.361	**0.043**	0.016	1.246
		24	0.115	0.041	1.411	0.067	0.019	1.391
		32	0.097	0.033	1.393	0.086	0.022	1.420
SPFSplat (ICCV 2025)	Offline Pose-free	16	0.283	0.285	5.370	0.527	0.496	4.677
		24	0.281	0.302	5.396	0.524	0.606	3.302
		32	0.295	0.293	5.345	0.497	0.592	1.881
AnySplat (TOG 2025)	Offline Pose-free	16	0.110	0.049	**1.091**	0.258	0.268	1.383
		24	0.178	0.104	**1.048**	0.242	0.221	**1.202**
		32	0.132	0.067	**1.057**	0.184	0.178	**1.072**
PF-SGS (Ours)	Online Pose-free	16	0.079	0.017	1.383	0.046	**0.015**	**1.225**
		24	**0.093**	**0.035**	1.405	**0.060**	**0.017**	1.341
		32	**0.087**	**0.027**	1.382	**0.075**	**0.019**	1.352

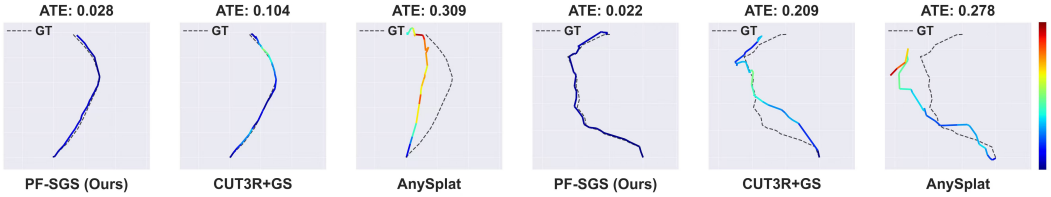

Figure 4: **Trajectory Comparison on the RE10K and DL3DV.**

increases. Unlike the direct state update in CUT3R+GS [14], PF-SGS dynamically modulates the update magnitude of current observations to the hidden state, effectively reducing the impact of transient noise on historical memory and alleviating long-sequence pose drift.

Computational Overhead Comparison Fig. 1 shows the computational overhead of various methods. Both PF-SGS and CUT3R+GS [14] process streaming inputs with linear time complexity, maintaining relatively stable FPS on long sequences while exhibiting approximately linear growth in memory usage. This benefits from the online reconstruction paradigm and the compact encoding of scene geometry via the hidden state, effectively avoiding the substantial overhead of processing long sequences. In contrast, offline reconstruction methods such as Long-LRM [7] and AnySplat [8] require joint modeling of all input views, limiting their inference efficiency; SPFSplat [56] is efficient on short sequences, but its computational cost grows rapidly with the number of views, limiting its practicality for long sequences. Overall, PF-SGS achieves a better balance between reconstruction quality and efficiency, supporting effective online scene reconstruction.

5.4 Ablation Study

Adaptive State Gating Analysis We conduct a detailed analysis of the effect of Adaptive State Gating (ASG). As shown in Table 3 and Fig. 5, under the setting without ground-truth pose supervision, removing ASG leads to evident accumulation of pose errors, accompanied by noticeable structural distortions in 3D reconstruction. In contrast, PF-SGS maintains substantially more stable hidden state evolution and significantly slows the growth trend of ATE. This highlights the critical role of ASG in suppressing memory degradation and preserving long-horizon geometric consistency. To rule out the possibility that the performance gains stem from increased parameters, we replace the adaptive gates with input-independent learnable parameters, referred to as Static Gates. Since Static Gates are independent of the historical state and current observation, they can only learn a fixed update bias during training, which disrupts stable memory evolution and can even underperform simple additive updates. This confirms that ASG benefits from dynamic modulation conditioned

Table 3: **Quantitative comparison of ablation components.**

Variant	PSNR ↑	SSIM ↑	LPIPS ↓	ATE ↓
Ours	**24.19**	**0.806**	**0.179**	**0.087**
w/o ASG	23.04	0.764	0.215	0.124
w/o DGM	23.58	0.786	0.195	0.095
w/o M_i	23.25	0.777	0.207	0.118
w/ Static Gates	22.78	0.758	0.226	0.129
w/o DGM & ASG	22.34	0.738	0.245	0.137



Figure 5: **Effect of Adaptive State Gating on pose estimation and reconstruction quality.**



Baseline | w/ Static Gates | w/o ASG | w/o M_i | w/o DGM | Ours

Figure 6: **Component ablation comparison.** Baseline denotes the variant without DGM and ASG.

on both inputs, rather than from the gating parameters themselves. Finally, we ablate the recursive correction term M_i, retaining only a single-step gated update for the historical state and current residual. This simplified variant still remains inferior to the full model, indicating that single-step gating is insufficient to stabilize state evolution over long sequences without accumulated correction.

Delayed Gaussian Modeling Analysis We further evaluate the effectiveness of Delayed Gaussian Modeling (DGM). As shown in Table 3 and Fig. 6, removing DGM causes noticeable blurring and detail loss, suggesting that the historical state and current observation alone cannot provide sufficient local geometric constraints during long-sequence modeling. By introducing bounded look-ahead observations, the model can incorporate additional cross-view geometric cues, thereby improving modeling accuracy. We further investigate the effect of the look-ahead window size w. Results in Table 4 show that rendering quality improves with w and stabilizes after $w = 4$, suggesting that introducing a few look-ahead observations is sufficient to provide effective geometric cues. Meanwhile, increasing the window size incurs little computational overhead: memory usage stays largely stable and FPS fluctuates only within a narrow range. This is because the subsequent frames serve purely as geometric context, not as additional reconstruction targets.

Table 4: **Ablation on delayed window.**

w	PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑	Mem ↓
1	23.58	0.786	0.195	22.5	2328 MB
2	24.02	0.797	0.180	21.4	2376 MB
3	24.16	0.804	0.179	21.9	2451 MB
4	**24.19**	**0.806**	**0.177**	21.6	2478 MB
5	24.18	0.806	0.178	21.5	2556 MB
6	24.19	0.805	0.177	21.9	2654 MB

6 Conclusion and Limitations

We propose PF-SGS, which maintains a persistent hidden state as scene memory and performs online reconstruction through frame-by-frame updates. To address memory degradation and geometric under-constraining in long-sequence modeling, we introduce Adaptive State Gating to dynamically regulate update magnitudes for stable memory evolution, and incorporate Delayed Gaussian Modeling to supplement local geometric cues for improved reconstruction consistency. However, PF-SGS still has several limitations: **1)** scene information is compressed into a fixed-capacity hidden state that cannot scale dynamically with scene size, risking scene degradation; **2)** historical frames are stored solely through implicit encoding, preventing retrospective correction via global bundle adjustment. As a result, frame-level errors gradually accumulate during loop updates, causing performance to degrade as the sequence increases; this remains a common challenge faced by current online reconstruction methods. In the future, we will explore a scalable explicit-implicit hybrid memory architecture, aiming to achieve backtrackable optimization that balances efficiency and global consistency

Acknowledgments

This work was supported by the Explorers Program of Shanghai (Basic Research Funding) (26TS1404200), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM312), and the Tongji University "Medicine+X" Cultivation Program (2026-0648-YB-05).

References

[1] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European conference on computer vision*, pages 338–355. Springer, 2024.

[2] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *Advances in Neural Information Processing Systems*, 37:107064–107086, 2024.

[3] Heng Jia, Linchao Zhu, and Na Zhao. H3r: Hybrid multi-view correspondence for generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7655–7665, 2025.

[4] Yanqi Bao, Tianyu Ding, Jing Huo, Yaoli Liu, Yuxin Li, Wenbin Li, Yang Gao, and Jiebo Luo. 3d gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(7):6832–6852, 2025.

[5] Shuting He, Peilin Ji, Yitong Yang, Changshuo Wang, Jiayi Ji, Yinglin Wang, and Henghui Ding. A Survey on 3D Gaussian Splatting Applications: Segmentation, Editing, and Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2026. Early Access.

[6] 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*, pages 1–19. Springer, 2024.

[7] 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*, pages 4349–4359, 2025.

[8] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics (TOG)*, 44(6):1–16, 2025.

[9] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[10] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. *Advances in Neural Information Processing Systems*, 37:107326–107349, 2024.

[11] Guichen Huang, Ruoyu Wang, Xiangjun Gao, Che Sun, Yuwei Wu, Shenghua Gao, and Yunde Jia. Longsplat: Online generalizable 3d gaussian splatting from long sequence images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4994–5002, 2026.

[12] Yang Li, Jinglu Wang, Lei Chu, Xiao Li, Shiu-hong Kao, Ying-Cong Chen, and Yan Lu. Streamgs: Online generalizable gaussian splatting reconstruction for unposed image streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25841–25850, 2025.

[13] 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*, pages 20697–20709, 2024.

[14] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[15] Yifan Liu, Keyu Fan, Weihao Yu, Chenxin Li, Hao Lu, and Yixuan Yuan. Monosplat: Generalizable 3d gaussian splatting from monocular depth foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21570–21579, 2025.

[16] Zhicong Wu, Hongbin Xu, Gang Xu, Ping Nie, Zhixin Yan, Jinkai Zheng, Liangqiong Qu, Ming Li, and Liqiang Nie. Textsplat: Text-guided semantic fusion for generalizable gaussian splatting. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 8478–8487, 2025.

[17] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9869–9877, 2025.

[18] 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*, pages 19457–19467, 2024.

[19] 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*, pages 370–386. Springer, 2024.

[20] 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*, pages 16453–16463, 2025.

[21] Hao Li, Yuanyuan Gao, Chenming Wu, Dingwen Zhang, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards pose-free generalizable 3d gaussian splatting in real-time. In *European Conference on Computer Vision*, pages 325–341. Springer, 2024.

[22] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2410.22128*, 2024.

[23] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22012–22022, 2025.

[24] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No Pose, No Problem: Surprisingly Simple 3D Gaussian Splats from Sparse Unposed Images. In *International Conference on Learning Representations*, volume 2025, pages 54009–54033, 2025.

[25] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

[26] Botao Ye, Boqi Chen, Haofei Xu, Daniel Barath, and Marc Pollefeys. Yonosplat: You only need one model for feedforward 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pages 39852–39871, 2026.

[27] 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 *International Conference on Learning Representations*, volume 2026, pages 10481–10497, 2026.

[28] Muyu Xu, Fangneng Zhan, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Musasplat: Efficient sparse-view 3d gaussian splats via lightweight multi-scale adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11343–11351, 2026.

[29] Jiahui Ren, Mochu Xiang, Jiajun Zhu, and Yuchao Dai. Panosplatt3r: Leveraging perspective pretraining for generalized unposed wide-baseline panorama reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28959–28969, 2025.

[30] Yuki Fujimura, Takahiro Kushida, Kazuya Kitano, Takuya Funatomi, and Yasuhiro Mukaigawa. Ufv-splatter: Pose-free feed-forward 3d gaussian splatting adapted to unfavorable views. *arXiv preprint arXiv:2507.22342*, 2025.

[31] Jiale Xu, Shenghua Gao, and Ying Shan. Freesplatter: Pose-free gaussian splatting for sparse-view 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25442–25452, 2025.

[32] Tooba Imtiaz, Lucy Chai, Kathryn Heal, Xuan Luo, Jungyeon Park, Jennifer Dy, and John Flynn. Lvt: Large-scale scene reconstruction via local view transformers. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.

[33] 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.

[34] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4929, 2025.

[35] Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. ilrm: An iterative large 3d reconstruction model. *arXiv preprint arXiv:2507.23277*, 2025.

[36] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21936–21947, 2025.

[37] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025.

[38] Mark He Huang, Lin Geng Foo, Christian Theobalt, Ying Sun, and De Wen Soh. Onlinesplatter: Pose-free online 3d reconstruction for free-moving objects. *arXiv preprint arXiv:2510.20605*, 2025.

[39] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.

[40] Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. Long3r: Long sequence streaming 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5273–5284, 2025.

[41] 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. *arXiv preprint arXiv:2602.20160*, 2026.

[42] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024.

[43] 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.

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

[45] Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems*, 36:33202–33221, 2023.

[46] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to Memorize at Test Time. In *Advances in Neural Information Processing Systems*, volume 38, pages 113506–113543. Curran Associates, Inc., 2025.

[47] Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. Liger: Linearizing large language models to gated recurrent structures. *arXiv preprint arXiv:2503.01496*, 2025.

[48] Ranran Huang and Krystian Mikolajczyk. Spfsplatv2: Efficient self-supervised pose-free 3d gaussian splatting from sparse views. *arXiv preprint arXiv:2509.17246*, 2025.

[49] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, Zilong Dong, and Ping Tan. Dro: Deep recurrent optimizer for video to depth. *IEEE Robotics and Automation Letters*, 8(5):2844–2851, 2023.

[50] Xiaogang Song, Haoyue Hu, Li Liang, Weiwei Shi, Guo Xie, Xiaofeng Lu, and Xinhong Hei. Unsupervised monocular estimation of depth and visual odometry using attention and depth-pose consistency loss. *IEEE Transactions on Multimedia*, 26:3517–3529, 2023.

[51] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021.

[52] 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.

[53] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023.

[54] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20050, 2024.

[55] 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*, pages 22160–22169, 2024.

[56] Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27947–27957, 2025.

[57] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. In *Advances in Neural Information Processing Systems*, volume 37, pages 21875–21911. Curran Associates, Inc., 2024.

A Appendix

A.1 Canonical Space Definition

In pose-free feed-forward 3DGS, the absence of ground-truth camera poses in the input sequence prevents the model from relying on a predefined world coordinate system to fuse 3D representations across different viewpoints. Consequently, a unified reference frame must be established for all predicted 3D Gaussians. Following the canonical space modeling in , we define the camera coordinate system of the first frame as the canonical coordinate and require all Gaussian centers predicted from subsequent frames to be expressed in this coordinate system. Specifically, the first frame's camera coordinate system is anchored as the baseline for the canonical space, with its relative pose serving as the Identity Transformation. For the i-th frame, the model predicts its relative pose $\hat{P}_{i\to 1}$ with respect to this canonical frame. Concurrently, the Gaussian centers μ_i are directly defined in the canonical space rather than the current frame's local coordinate system. In the reprojection loss (Eq. 13), the inverse pose $\hat{P}_{i\to 1}^{-1}$ is utilized to map the Gaussian centers μ_i back onto the image plane of the current frame for pixel-wise supervision. This approach ensures that Gaussian primitives from different frames are naturally aligned in a unified coordinate system, obviating the risk of error accumulation associated with explicit fusion of local predictions.

A.2 Training Details

To achieve stable training on long sequences of 32 frames, we adopt a Truncated Backpropagation Through Time strategy. Specifically, the complete 32 frame sequence is partitioned into multiple consecutive sub-sequences. After each sub-sequence, the gradient flow is truncated, while the hidden state and adaptive state gating variables are preserved and propagated forward. This design ensures that the model can learn long-range geometric consistency while avoiding the computational explosion associated with persistent hidden state unfolding.

We follow the progressive training paradigm validated in CUT3R [14], adopting a three-stage strategy to stabilize geometric representation learning and recurrent state modeling. The first 100K iteration short-sequence ($N = 4$) stage serves as a warm-up phase, where the model learns basic multi-view geometry and spatial perception from short and stable inputs. This provides a reliable initialization and avoids exposing the model too early to severe long-horizon error accumulation. The following 100K iteration long-sequence ($N = 32$) stage then focuses on learning stable state evolution over extended streaming inputs. With the Adaptive State Gating module introduced at this stage, the model learns to regulate memory updates under accumulated history and current observations, thereby improving long-range geometric consistency. The final 50K iteration fine-tuning stage further refines the prediction heads and gating behavior while keeping the learned backbone representation stable. This helps improve reconstruction quality and reduces overfitting to a fixed training length, leading to better robustness on variable-length sequences.

A.3 More Settings

Following the common practice in feed-forward 3DGS of initializing with a mature geometric backbone [24, 57, 8], PF-SGS and CUT3R+GS inherit the pretrained weights of CUT3R [14]. All comparison baselines are trained from scratch under the same resolution and training protocol to ensure a fair evaluation of cross-domain generalization.

A.4 More Ablation Analyses

We further conduct an ablation analysis on the key components of PF-SGS, with results summarized in Table 5. Both self-supervised losses contribute positively to scene modeling, with the reprojection loss having the most pronounced impact. Since the model is required to align Gaussian primitives from different frames into a unified canonical space, the reprojection loss explicitly enforces consistency between projected Gaussian centers and their corresponding pixel coordinates across views, providing crucial geometric supervision. Without this term, appearance-based supervision alone is insufficient to effectively suppress long-range drift.

Removing the progressive training strategy leads to a significant performance degradation, and the model even struggles to maintain stable reconstruction results. This is because error accumulation is inevitable during long-range recurrent inference. If all modules are jointly optimized on long sequences, errors introduced during the update process can be continuously propagated and amplified, making the model converge to suboptimal local solutions. In contrast, the progressive training strategy introduces a curriculum from short sequences to long sequences, providing a more stable initialization for state evolution and ensuring convergence to a consistent geometric solution.

Table 5: **Additional ablation studies.** TS denotes the progressive training strategy.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓	ATE ↓
Ours	**24.19**	**0.806**	**0.179**	**0.087**
w/o α	23.36	0.771	0.211	0.117
w/o θ	22.87	0.751	0.222	0.129
w/o η	23.73	0.780	0.191	0.101
w/o $\mathcal{L}_{repro}$	20.04	0.672	0.335	0.205
w/o $\mathcal{L}_{photo}$	23.84	0.791	0.187	0.096
w/o TS	16.38	0.545	0.490	0.365

As shown in Table 6, the effectiveness of DGM is correlated with the sampling density of the input sequence. With a fixed total frame count per scene, a shorter sequence length (e.g., 16 frames) is associated with larger inter-frame viewpoint variations, where correspondences between adjacent frames become relatively sparse and the model is more prone to insufficient geometric constraints. DGM mitigates this by incorporating bounded look-ahead observations, thereby supplying additional cross-view geometric cues to supplement the current frame's inference. As the sampling density increases (e.g., 64 frames), the dense observations already provide sufficient geometric constraints, and the marginal contribution of DGM correspondingly decreases.

Table 6: **Ablation of Delayed Gaussian Modeling across sequence lengths.**

Views	Ours			w/o DGM		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
16	24.64	0.797	0.170	23.46	0.785	0.200
32	24.19	0.806	0.179	23.58	0.786	0.195
64	23.77	0.793	0.188	23.43	0.782	0.199

A.5 Longer Sequence Evaluation

We further extend the evaluation to longer input sequences to examine the modeling stability of various methods. As shown in Table 7, Long-LRM [7] maintains stable and leading reconstruction quality on long sequences. By relying on ground-truth poses for precise alignment, it benefits from increasingly reliable multi-view constraints as more input views are provided, enabling a denser and more consistent scene representation. This indicates that accurate pose priors play a crucial role in preserving long-range spatial consistency. In contrast, pose-free methods exhibit varying degrees of performance degradation as the number of views increases. Although CUT3R+GS [14] introduces ground-truth pose supervision, its standard additive update scheme forces the hidden state to overfit the current observation, causing historical memory to be continuously overwritten; as the sequence length increases, structural distortions and trajectory drift become increasingly evident. Benefiting from the proposed Adaptive State Gating, PF-SGS dynamically adjusts the state update magnitude based on the historical state and the current observation, effectively alleviating memory degradation and pose drift during long-sequence modeling, and exhibiting stronger robustness.

Table 7: **Quantitative comparison on longer input sequences.**

Method	Views	PSNR ↑	SSIM ↑	LPIPS ↓	ATE ↓
Long-LRM	48	**24.67**	**0.855**	0.202	/
	64	**24.84**	**0.857**	0.197	/
	96	**24.68**	**0.855**	**0.200**	/
CUT3R+GS	48	23.32	0.774	0.221	0.127
	64	22.69	0.751	0.244	0.165
	96	22.55	0.744	0.249	0.171
SPFSplat	48	18.85	0.598	0.347	0.291
	64	18.42	0.584	0.372	0.305
	96	OOM			
AnySplat	48	22.57	0.740	0.232	0.137
	64	22.68	0.749	0.227	0.152
	96	22.34	0.731	0.240	0.165
PF-SGS	48	24.11	0.805	**0.181**	**0.086**
	64	23.77	0.793	**0.188**	**0.101**
	96	23.39	0.781	0.202	**0.113**

A.6 Limitation Analysis

Although PF-SGS demonstrates strong performance, it still has several limitations: **1)** The fixed-capacity hidden state cannot continuously encode an indefinitely growing scene. When the scene scale exceeds the capacity of the hidden state, the scene geometry accumulated in earlier stages is inevitably overwritten by subsequent observations. **2)** Historical frames are stored only through implicit encoding, preventing retrospective correction via global bundle adjustment and causing

frame-wise errors to accumulate during recurrent updates. Although ASG significantly suppresses this error accumulation, pose drift ultimately persists, remaining a fundamental challenge in online reconstruction. As shown in Fig. 7, pose estimation maintains relatively high accuracy in the early stage of the sequence, but the error gradually diverges as the sequence progresses and cannot be retrospectively corrected by subsequent observations.

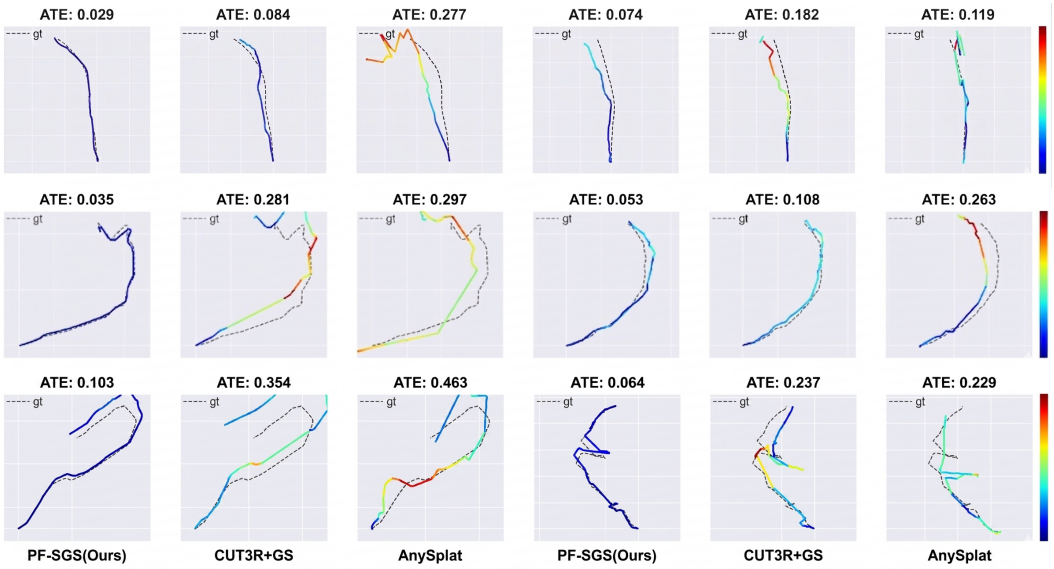

Figure 7: PF-SGS maintains relatively accurate pose estimation in the early stage, while the trajectory error gradually diverges as the sequence progresses due to the lack of explicit global correction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions and scope of the research, and the related claims are supported by the experiments in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 and Appendix A.6 discuss the limitations of our method, including practical constraints and potential directions for improvement.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is primarily an algorithmic and empirical contribution focused on neural rendering architecture and does not contain new theoretical theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the methodology, training strategies, and experimental settings required to reproduce the main experiments in Sections 4 and 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code in the supplementary materials and commit to releasing it as open source upon paper acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Section 5 describes the experimental settings, including the RE10K and DL3DV dataset splits, the choice of optimizer, and the hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not report error bars or statistical significance tests, as the main comparisons follow standard evaluation protocols in prior 3D reconstruction works.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We reports the computational resources used for the experiments, including the hardware and implementation environment needed for reproduction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research uses publicly available datasets and adheres to the ethical guidelines of the NeurIPS community.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work focuses on fundamental 3D reconstruction research, and we do not identify direct societal impacts beyond the general applications of 3D perception.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or datasets that require additional safeguards for responsible release.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All utilized datasets (e.g., RE10K, DL3DV) and baseline models are properly cited and used according to their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release new datasets, benchmarks, or other standalone assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.