Fast Spatial Memory with Scalable Elastic Test-Time Training

Ziqiao Ma*[1,2], Xueyang Yu*[3], Haoyu Zhen[3], Yuncong Yang[3]
Joyce Chai[2], and Chuang Gan[1,3]

[1]MIT-IBM Watson AI Lab [2]University of Michigan
[3]University of Massachusetts Amherst

https://fast-spatial-memory.github.io/

Fig. 1: Fast Spatial Memory (FSM) is an efficient, scalable 4D reconstruction model that learns spatiotemporal representations from long sequences to render novel views at novel times. The model is powered by Large Chunk Elastic Test-Time Training (LaCET) blocks and is compatible with a range of rendering decoders, including LRM-style and LVSM-style decoders.

Abstract. Large Chunk Test-Time Training (LaCT) has shown strong performance on long-context 3D reconstruction, but its fully plastic inference-time updates remain vulnerable to catastrophic forgetting and overfitting. As a result, LaCT is typically instantiated with a single large chunk spanning the full input sequence, falling short of the broader goal of handling arbitrarily long sequences in a single pass. We propose Elastic Test-Time Training inspired by elastic weight consolidation, that stabilizes LaCT fast-weight updates with a Fisher-weighted elastic prior around a maintained anchor state. The anchor evolves as an exponential moving average of past fast weights to balance stability and plasticity. Based on this updated architecture, we introduce Fast Spatial Memory (FSM), an efficient and scalable model for 4D reconstruction that learns spatiotemporal representations from long observation sequences and renders novel view-time combinations. We pre-trained FSM on large-scale curated 3D/4D data to capture the dynamics and semantics of complex spatial environments. Extensive experiments show that FSM supports fast adaptation over long sequences and delivers high-quality 3D/4D reconstruction with smaller chunks and mitigates the camera-interpolation shortcut. Overall, we hope to advance LaCT beyond the bounded single-chunk setting toward robust multi-chunk adaptation, a necessary step

* Equal contribution.

for generalization to genuinely longer sequences, while substantially alleviating the activation-memory bottleneck.

Keywords: Test Time Training · Novel View Synthesis · Large Reconstruction Model

1 Introduction

Building a spatial memory would require learning to compress visual observations across viewpoints and time into a unified 4D representation that preserves both spatial structure and temporal dynamics. This capability would advance applications in 4D asset generation [35,54] for video games, film production, and AR/VR, as well as world modeling [19,70] for embodied AI and robotics. Especially, reconstructing dynamic scenes from temporally extended and dynamically sampled observations (e.g., long videos captured by moving cameras) remains a central challenge.

Recent advances in Large Reconstruction Models (LRMs) [12,66] and Large View Synthesis Models (LVSM) [15, 20] offer promising rendering-based alternatives to efficient and high-quality 3D/4D reconstruction. Typically built on Transformer-based sequence modeling, these methods achieve strong reconstruction performance by learning powerful priors over structure and appearance from large-scale multi-view data. Despite these advances, these models remain constrained by the amount of activation memory available for a single forward pass, leaving long-context modeling largely unresolved. This is particularly the case in 4D domain, where videos that are temporally extended yet spatially sparsely observed, and their reconstruction quality degrades sharply beyond the training context length, indicating limited temporal scalability [31]. While several 3D reconstruction works have explored hybrid sequence models that combine linear-time state-based mixers with full attention [73, 74], the central question for practical 4D modeling remains open: *How can we design a simple, scalable, and efficient spatial memory architecture that learns scene-level spatiotemporal representations from long sequences?*

Test-Time Training (TTT) [37,40] has shown promise in addressing the long-context issue in geometric reconstruction and view synthesis [7,45,65]. Especially, Large Chunk Test-Time Training (LaCT) [69] enables in-forward, chunk-wise fast-weight adaptation that lets a transformer recalibrate its internal representations during inference, efficiently updating small parameters from key-value statistics without backpropagation to achieve self-refining, test-time adaptation. Yet, these techniques do not directly generalize to the 4D regime, where scene dynamics evolve across space and time during inference, since the fully plastic nature of continuous LaCT updates leads to uncontrolled fast-weight drift, leading to overfitting in training and unstable updates at test time. This is analogous to catastrophic forgetting at inference time. To address this issue, we introduce Elastic Test-Time Training that executes an additional *consolidate* operation after LaCT *update*, inspired by the Elastic Weight Consolidation (EWC) [21]

in continual learning. Each fast-weight module keeps a reference set of anchor parameters (the values before adaptation) and continuously estimates their importance through an online Fisher-style statistic. During inference, important parameters are softly pulled back toward their anchors, while less critical ones remain free to adjust. This elastic behavior acts as an adaptive spring: it constrains unstable drift without sacrificing responsiveness to new lighting, pose, or scene conditions, transforming the base transformer into a fast, self-refining yet elastic 4D learner, one that keeps adapting to the stream while remembering where it came from. We refer to this new architecture as Large Chunk Elastic Test-Time Training (LaCET).

We scale LaCET up to pretrain a Fast Spatial Memory (FSM) on a curated set of 3D/4D datasets with posed images captured over time and from different cameras. We primarily evaluated FSM on the novel view synthesis (NVS) and demonstrated its competitive performance on a variety of benchmarks and the scalability of LaCET. The model scales effectively with more data and larger model size and generalizes well to novel scenes. With careful ablation studies, we show that LaCET can effectively mitigate the overfitting and undesirable inference time behaviors of LaCT, e.g., camera interpolation. To our knowledge, FSM is the first large-scale 4D reconstruction model design that supports input from long sequences of views and arbitrary timestamps and renders arbitrary novel view-time combinations. Overall, we hope to advance LaCT beyond the bounded single-chunk setting toward robust multi-chunk adaptation, a necessary step for generalization to genuinely longer sequences, while substantially alleviating the activation-memory bottleneck.

2 Algorithmic Preliminaries

2.1 Fast Weights and Test-Time Training

Test-Time Training (TTT) [40] introduces fast weights [37] with rapidly adaptable parameters, which get updated at both training and inference time. This is in sharp contrast to slow weights (conventional model parameters) that remain fixed at inference time. In the context of attention, we consider a sequence of N tokens $\mathbf{x} = [x_1, x_2, \ldots, x_N]$, where each token x_i is projected into key k_i, query q_i, and value v_i vectors. Formally, TTT defines a function $f_{\boldsymbol{\theta}}(\cdot)$ parameterized by the fast weights $\boldsymbol{\theta}$, and it involves an *update* and an *apply* operation. The (per-token) *update* operation defines:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}\big(f_{\boldsymbol{\theta}}(k_i), v_i\big), \tag{1}$$

where η represents the learning rate and $\mathcal{L}(\cdot, \cdot)$ denotes a loss between the transformed key $f_{\boldsymbol{\theta}}(k_i)$ and its corresponding value v_i, encouraging the network to learn key-value associations. Intuitively, this objective trains the model to compress the ever-growing KV cache (whose memory cost scales linearly with context length) into a fixed-size neural memory, preserving critical key-value associations within a bounded memory budget. The *apply* operation defines:

$$z_i = f_{\boldsymbol{\theta}'}(q_i), \tag{2}$$

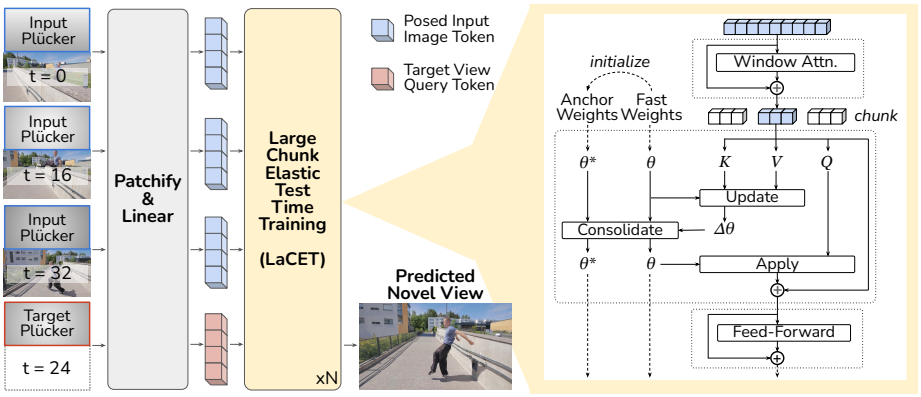


Fig. 2: (Left) Overview of FSM. The model takes a sequence of posed images captured at different times and learns to infer novel view-time combinations. Camera information is converted into Plücker ray maps as geometric augmentation for visual tokens. The model directly predict the target view with decoders. **(Right)** The LaCET Block. It maintains two sets of parameters, *anchor weights* and *fast weights*. During adaptation, the fast weights are updated using information from the current chunk (queries, keys, and values), while the anchor weights act as a stable reference. The model tracks parameter importance online and softly restores critical weights toward their anchors to prevent drift. This stabilizes rapid updates while preserving the adaptability of TTT, addressing the plasticity issue.

where the updated fast weights $\boldsymbol{\theta}'$ are used to compute the output vector z_i given the query q_i. The per-token TTT layer iteratively performs the *update* and *apply* operations on each token x_i in sequence.

2.2 Test-Time Training Done Right

Naïve TTT methods often struggle to scale to long contexts, largely due to the low hardware efficiency of their TTT layers, which operate on extremely small mini-batches. To address this, [69] proposed Large-Chunk Test-Time Training (LaCT), a chunk-wise formulation that improves scalability and throughput. The *apply* operation $o_i = f_{\boldsymbol{\theta}}(q_i)$ follows Eq. (2), where all query vectors q_i within a chunk share the same fast weight. Unlike the per-token update in Eq. (1), LaCT aggregates the loss over all keys k_i and values v_i in a chunk and computes a single surrogate *update* for chunk c:

$$\boldsymbol{\theta}_{c+1} = \boldsymbol{\theta}_c \underbrace{- \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{b} \eta_i(x_i)\, \mathcal{L}\big(f_{\boldsymbol{\theta}}(k_i), v_i\big)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_c}}_{\text{per-chunk surrogate pseudo-gradient}}. \tag{3}$$

Here, b denotes the chunk size and η_i is the (learnable) per-token learning rate. Intuitively, this objective strengthens the association between each key and its corresponding value by updating the fast weights so that $f_{\boldsymbol{\theta}}(k_i)$ becomes more

consistent with v_i under the training loss. In practice, LaCT regularizes the updated fast weights using L2 weight normalization [36] along the input dimension and optionally applies the Muon-style Newton-Schulz iteration [17, 29], without weight decay. Because each chunk aggregates thousands of tokens, updates occur infrequently, amortizing computational cost.

2.3 Test-Time Training Done Better

While LaCT significantly improves the scalability of TTT by amortizing adaptation across large chunks, its updates remain fully plastic, as the fast weights in each chunk drift freely in parameter space at inference time. In the novel view synthesis task, LaCT works the best with one single chunk. In long and dynamic 4D scenes, where illumination, pose, or motion continuously evolve during inference, such unconstrained plasticity can cause cumulative instability, leading to temporal ghosting artifacts. To address this, we propose Elastic Test-Time Training, which enhances the LaCT update operator with an Elastic Weight Consolidation (EWC) [21] regularizer, introducing a soft stability prior over fast-weight dynamics. We refer to our algorithm as Large-Chunk Elastic Test-Time Training (LaCET, to distinguish from LaCT), combining its scalability, efficiency, and elastic stability for robust long sequence modeling.

Elastic Weight Consolidation. [21] introduces a quadratic penalty that discourages important parameters from drifting too far from a reference set of anchor weights, originally designed for a classic continual learning setting where a model learn a new task $\mathcal{T}_B$ without forgetting a previously learned task $\mathcal{T}_A$. All knowledge about $\mathcal{T}_A$ is captured in the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D}_A)$. Since this posterior is intractable for large neural networks, EWC approximates it using a Gaussian centered at the previously optimized parameters $\boldsymbol{\theta}^\star_A$ with a diagonal precision given by the Fisher Information Matrix F, i.e., $p(\boldsymbol{\theta} \mid \mathcal{D}_A) \approx \mathcal{N}\big(\boldsymbol{\theta}^\star_A, F^{-1}\big)$. The Fisher Information has three desirable properties: (i) it corresponds to the local curvature of the loss near $\boldsymbol{\theta}^\star_A$, (ii) it can be estimated from first-order gradients alone, and (iii) it is guaranteed to be positive semi-definite. The overall objective when learning $\mathcal{T}_B$ becomes a combination of the new-task loss and a quadratic penalty at $\boldsymbol{\theta}^\star_A$:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_B(\boldsymbol{\theta}) + \sum_i \frac{\lambda}{2} F_i \left(\boldsymbol{\theta}_i - \boldsymbol{\theta}^\star_{A,i}\right)^2, \tag{4}$$

where $\mathcal{L}_B(\boldsymbol{\theta})$ is the loss for the new task $\mathcal{T}_B$, λ controls the relative importance of retaining old knowledge, and i indexes each model parameter. Intuitively, parameters with high Fisher values F_i are crucial for $\mathcal{T}_A$ and are therefore strongly constrained to remain near $\boldsymbol{\theta}^\star_A$, whereas parameters with small F_i can adapt freely to $\mathcal{T}_B$.

Elastic Test-Time Training. In our formulation, we reinterpret this idea at the time of the test: each incoming chunk of data acts as a new task $\mathcal{T}_B$, and the fast-weight state of the previous chunk plays the role of $\boldsymbol{\theta}^\star_A$. The Fisher-weighted

penalty in Eq. (4) thus serves as a continuously updated elastic prior, stabilizing the model's adaptation over time (e.g., foreground dynamics) while preserving useful past information (e.g., static background). The EWC penalty defines an elastic prior after the LaCT *update* in Eq. (3), which we refer to as the *consolidate* operator. Formally, let $\boldsymbol{\theta}'_c$ denote the intermediate fast weights after the *update* but before elastic consolidation in chunk c, and $\boldsymbol{\theta}^\star_c$ their corresponding *anchor* parameters (the reference state before adaptation or at the last re-anchor).

$$\boldsymbol{\theta}_{c+1} \;=\; \boldsymbol{\theta}'_c \underbrace{-\,\lambda\, F_c \odot \big(\boldsymbol{\theta}'_c - \boldsymbol{\theta}^\star_c\big)}_{\text{elastic consolidation}}, \tag{5}$$

where F_c is a per-parameter Fisher-style importance estimate, $\odot$ denotes the Hadamard (elementwise) product, and λ is a constant controlling the strength of the elastic prior.

Importance Estimates. We maintain the importance matrix F_c as an EMA with decay $\alpha \in [0, 1)$ over chunk index c:

$$F_{c+1} \;=\; \alpha\, F_c \;+\; (1-\alpha)\, \varphi\big(\mathbf{S}_c\big), \tag{6}$$

where $\alpha \in [0, 1)$ is the decay factor. The statistic $\mathbf{S}_c$ depends on the chosen estimator. Besides EWC [21], we also consider two related alternatives motivated by memory-aware synapses (MAS) [1] and synaptic intelligence (SI) [63]. Concretely,

$$\mathbf{S}_c = \begin{cases} \boldsymbol{\theta}'_c - \boldsymbol{\theta}_c, & (\textit{MAS / EWC}) \\ (\boldsymbol{\theta}'_c - \boldsymbol{\theta}_c) \odot (\boldsymbol{\theta}'_c - \boldsymbol{\theta}^\star_c), & (\textit{SI}) \end{cases}$$

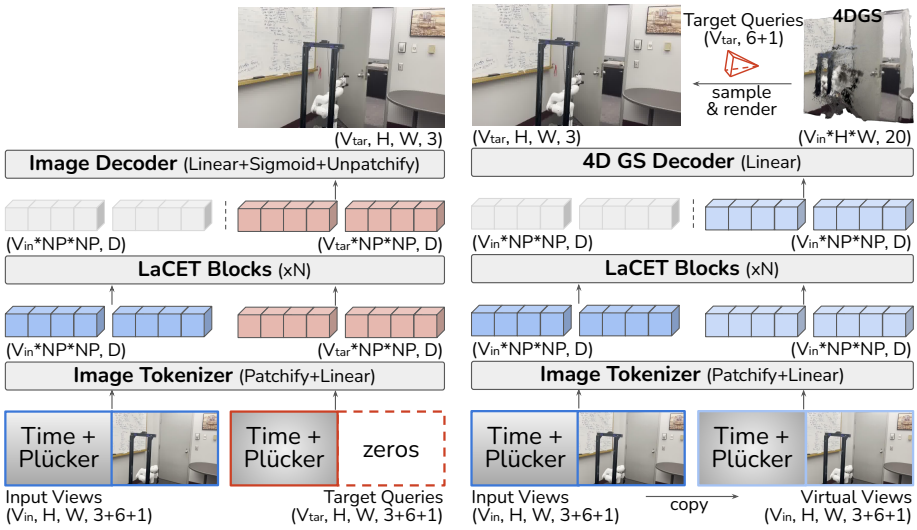
$$\varphi(\mathbf{S}_c) = \begin{cases} |\mathbf{S}_c|, & (\textit{MAS / SI}) \\ \mathbf{S}_c^2, & (\textit{EWC}) \end{cases}$$

with all operations applied elementwise. When $\mathbf{S}_c$ has a leading batch dimension, we average over that dimension before applying Eq. (6). Intuitively, the MAS-like variant tracks the magnitude of the chunkwise update, the EWC-like variant emphasizes parameters that consistently receive large squared updates, and the SI-like variant additionally weights the update by its drift from the current anchor. In our setting, since the anchor-relative displacement is itself induced by the chunkwise update, the SI-like statistic tends to behave similarly to a rescaled squared-update estimator.

Anchor Update Policies. We consider different anchoring policies that control how $\boldsymbol{\theta}^\star$ is maintained:

- **Global:** anchors remain fixed to initialization.
- **Streaming:** anchors update at each chunk boundary, ensuring local temporal continuity.
- **Streaming-EMA:** anchors update via an exponential moving average [43], $\boldsymbol{\theta}^\star \leftarrow \beta\boldsymbol{\theta}^\star + (1-\beta)\boldsymbol{\theta}$, forming a low-pass filter over the fast-weight trajectory.

We will show later that Streaming-EMA is the best practice for genuinely elastic memory behaviors.



(a) FSM-LVSM Overview. **(b)** FSM-LRM Overview.

Fig. 3: FSM-LVSM and FSM-LRM architectural designs. (a) LVSM-style rendering predicts target image patches directly from query tokens and does not build an explicit scene representation. (b) LRM-style rendering first predicts an explicit 4D scene representation with Gaussian primitives and then renders target views from that representation.

3 Fast Spatial Memory (FSM)

FSM adopts an end-to-end feedforward network to learn scene representations, trained using only photometric supervision. Input images are patchified and augmented with temporal and camera information to form visual tokens, which are then processed by the sequence model. We consider two decoding variants: (i) direct RGB patch prediction with a lightweight linear head, in the spirit of LVSMs [15,20]; and (ii) prediction of pixel-aligned Gaussian Splatting primitives followed by rasterization into target views, in the spirit of GS-LRMs [31,45,66].

3.1 Model Architecture

Image Tokenization. As shown in Figure 3, the input consists of V posed images from arbitrary view-time combinations, denoted as $\{\mathbf{I}_j \in \mathbb{R}^{H\times W\times 3}\}_{j=1}^{V}$, together with their camera intrinsics and extrinsics. Here, H and W denote the image height and width, respectively. We convert the provided camera parameters into canonical Plücker ray maps [33], represented as $[\mathbf{r}_d, \mathbf{r}_o \times \mathbf{r}_d]$, where $\mathbf{r}_d$ and $\mathbf{r}_o$ denote the ray direction and origin, respectively. Following 4D-LRM [31], temporal conditioning is encoded using a timestamp map $\{\mathbf{T}_j \in \mathbb{R}^{H\times W\times 1}\}_{j=1}^{V}$, which records the normalized time of each frame. For input j, We concatenate the timestamp $\mathbf{T}_j$, RGB image $\mathbf{I}_j$, and Plücker ray map $\mathbf{P}_j$ along the channel dimension to form a per-view feature map $\widetilde{\mathbf{I}}_j = \mathrm{Concat}(\mathbf{I}_j, \mathbf{P}_j, \mathbf{T}_j) \in \mathbb{R}^{H\times W\times 10}$,

which provides per-pixel spatial and temporal embeddings to distinguish both frame time and camera view. Each $\widetilde{\mathbf{I}}_j$ is partitioned into non-overlapping patches of size $p \times p$. Every patch is flattened into a vector of length $10p^2$ and linearly projected to a D-dimensional token embedding.

LaCET Backbone. We adopt SwiGLU-MLP [39] without bias terms as the fast weight network in Eq. (3), consisting of three parameter matrices $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$. The network and its loss is:

$$\begin{aligned} \mathcal{L}\big(f_{\boldsymbol{\theta}}(k_i), v_i\big) &= -\, f_{\boldsymbol{\theta}}(k_i)^\top v_i \\ &= -\, [\boldsymbol{\theta}_2(\mathrm{SiLU}(\boldsymbol{\theta}_1 k_i) \circ (\boldsymbol{\theta}_3 k_i))]^\top v_i, \end{aligned} \tag{7}$$

where $\circ$ denotes elementwise multiplication. We emphasize that only the input-view tokens are passed through the KV projections to generate gradients for the *update* operation. This design ensures that the target-view tokens do not interact with one another, allowing each novel view to be synthesized independently and efficiently. In contrast, allowing target tokens to interact across views would correspond to a form of dynamic evaluation [22] or few-shot in-context learning [44], which introduces additional information leakage and renders the comparison unfair.

LVSM-Style Rendering. In the LVSM-style variant (Figure 3a), the model does not rely on an explicit scene representation. For each target view-time query, we construct an empty image-token map whose appearance channels are set to zero, while its camera and temporal channels are populated with the target metadata. These query tokens are concatenated with the input tokens and processed jointly by the model. We then use a lightweight image-token decoder to reconstruct RGB patches from the output token embeddings. Concretely, each token is first passed through layer normalization, then projected linearly from the token dimension to $3p^2$. The resulting vector is interpreted as the flattened RGB values of the reconstructed patch. The resulting vector is interpreted as the flattened RGB values of the reconstructed patch, followed by a sigmoid activation to bound predictions to $[0, 1]$ in normalized pixel space.

(Alternatively) LRM-Style Rendering. Following an LRM-style rendering (Figure 3b), we adopt an explicit 4D representation, e.g., 4DGS [60] similar to 4D-LRM [31]. To adapt the sequence model for explicit GS modeling, we follow tttLRM [45] to query the fast weights for a set of virtual view planes for 4DGS and use the input views as virtual views. We adopt pixel-aligned Gaussian rendering, leading to $V \times H \times W$ Gaussian primitives, each parameterized by $\mathbf{g} \in \mathbb{R}^{20}$. We split it into ($\mathbf{g}_{\text{xyz}} \in \mathbb{R}^3, \mathbf{g}_{\text{t}} \in \mathbb{R}, \mathbf{g}_{\text{rgb}} \in \mathbb{R}^3, \mathbf{g}_{\text{scale,xyz}} \in \mathbb{R}^3, \mathbf{g}_{\text{scale,t}} \in \mathbb{R}, \mathbf{g}_{\text{rotation,left}} \in \mathbb{R}^4, \mathbf{g}_{\text{rotation,right}} \in \mathbb{R}^4, \mathbf{g}_{\text{opacity}} \in \mathbb{R}$). We mostly followed the parameterization of 4D-LRM except we set the permissible depth interval $\delta_{\text{near}} = 0.01$ and $\delta_{\text{far}} = 100$ for scene-level reconstruction. We adopt tile-based rasterization with deferred backpropagation during rendering to reduce GPU memory consumption [67].

Fig. 4: Qualitative illustration of the ablation studies, obtained after the same training steps (32K) with the same training and inference random seed on the same Stereo4D test set example.

EWC	Train #Chunk	Test #Chunk	Test Batch Size	Anchor Update	Fisher Estimate	Train ℓ_2 Loss ($\times 10^3$)$^\downarrow$	Test PSNR$^\uparrow$	Test LPIPS$^\downarrow$	Test SSIM$^\uparrow$
✗	1	1	1	-	-	1.80	26.021	0.1179	0.792
✗	4	4	1	-	-	2.04	26.908	0.0988	0.814
✓	4	4	1	streaming-ema	SI	2.36	**29.989**	**0.0517**	**0.903**
✓	4	4	1	streaming-ema	EWC	2.36	29.781	0.0537	0.897
✓	4	4	1	streaming-ema	MAS	2.28	29.922	0.0519	0.899
✓	4	4	1	streaming	MAS	1.71	26.960	0.0966	0.817
✓	4	4	1	global	MAS	3.00	28.347	0.0653	0.863
✓	1	1	1	global*	MAS	1.73	26.965	0.0960	0.817
✓	1	4	1	streaming-ema	MAS	1.73	21.993	0.3429	0.650
✓	4	4	16	streaming-ema	MAS	2.28	29.928	0.0519	0.898

* *The choice of anchor update policy makes no difference when chunk size is set to full sequence.*

Table 1: Ablation Studies. The training ℓ_2 loss is reported from the exponential moving average (EMA) model ($\alpha = 0.1$) to ensure robustness against noise. When the number of chunks is 1, it corresponds to the original full-sequence setup in LaCT. With 4 chunks, each chunk contains 2048 input tokens. We find that EWC effectively mitigates the overfitting issue observed in LaCT due to full plasticity. The *streaming-ema* anchor update policy proves critical for achieving stable performance.

3.2 Training Objectives

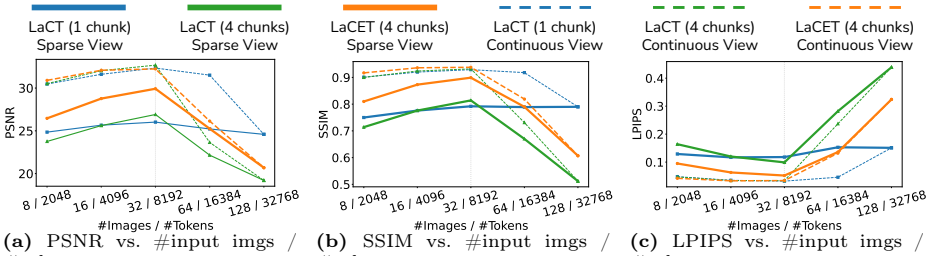
To train the model, we render U target views for supervision and minimize the image reconstruction loss. Let $\{\mathbf{I}^*_{i'} \mid i' = 1, 2, \ldots, U\}$ denote the ground truth views and $\{\widehat{\mathbf{I}}^*_{i'}\}$ the corresponding rendered images. The photometric training loss combines ℓ_2 (MSE) loss and LPIPS (w/ VGGNet) loss [68]:

$$\mathcal{L} = \frac{1}{U} \sum_{i'=1}^{U} \left(\ell_2(\widehat{\mathbf{I}}^*_{i'}, \mathbf{I}^*_{i'}) + \mu \cdot \mathrm{LPIPS}(\widehat{\mathbf{I}}^*_{i'}, \mathbf{I}^*_{i'}) \right), \tag{8}$$

where μ controls the weight of the LPIPS loss and is set to 0.5 empirically.

3.3 Pretraining Dataset

We pretrain FSM on a mixture of real and synthetic 3D/4D datasets, including static multi-view data and dynamic video data. Due to the limited availability of 4D data, we retain several static datasets and assign timestamps according to the natural camera trajectory, while synthetic dynamic datasets use randomly assigned frame timestamps. All datasets are rescaled to maintain a consistent metric scale across sources. Full dataset statistics and data pre-processing details are provided in Appendix **??** and Appendix **??**.



(a) PSNR vs. #input imgs / #tokens. **(b)** SSIM vs. #input imgs / #tokens. **(c)** LPIPS vs. #input imgs / #tokens.

Fig. 5: Test-time scaling curves. Shown are PSNR/SSIM/LPIPS of LaCT (1/4 chunks) and LaCET (4 chunks; streaming-ema), trained with 32 images (vertical line) and evaluated with varying numbers of input images. Each point uses a 136-frame Stereo4D clip. **For sparse views**, input and target frames are randomly sampled across the long full span. For **continuous views**, we select a contiguous sub-sequence (e.g., 40 frames for 32-in/8-out) and randomly mask the target frames inside it for the model to predict, reducing to frame interpolation.

4 Ablation: When and Why Elasticity Helps

Before scaling up the full pretraining pipeline, we perform controlled ablation studies with FSM-LVSM at a moderate scale. These experiments investigate the key algorithmic components added on top of the vanilla LaCT block, including the effects of chunking, anchor update policies, and Fisher estimation. For this purpose, we start by training the model exclusively on internet stereo videos from Stereo4D [16], trimmed to a maximum temporal window of 136 frames. All ablation models use a 12-layer LaCET backbone, trained with a per-GPU batch size of 16 on 8 H100 GPUs, using 32 input and 32 target views, a maximum temporal span of 128 frames, and an image resolution of 128×128 for 32K steps ($\approx$32B tokens). We deliberately use these smaller networks so that its long-context performance saturates with a reasonably small number of tokens. We evaluate on the Stereo4D test set using PSNR [6], SSIM [52], and LPIPS [68], using 32 randomly sampled views along the trajectory as inputs and averaged over 8 randomly sampled target views per scene. The results over different settings are aggregated in Table 1. More details are available in Appendix **??**.

4.1 Anchor Update Policies

We analyze how elastic consolidation behaves under different chunking and anchoring configurations.

Full-sequence setup (single chunk). When the chunk size equals the full sequence length, the model performs exactly one forward pass and one fast-weight update per scene. All anchor update policies become equivalent. The consolidation term scales with both the update magnitude and the anchor-relative drift, and in the single-chunk regime reduces to a second-order correction $\mathcal{O}(\lambda(\Delta\theta)^2)$ in the update size, which is negligible for small λ.

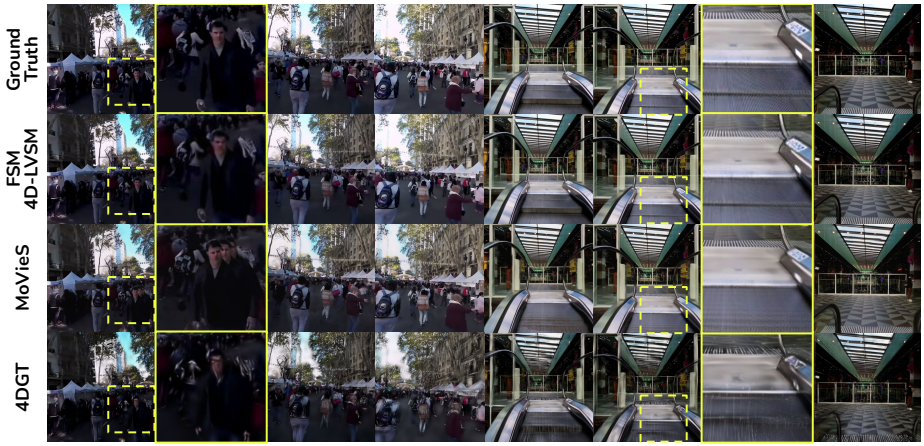


Fig. 6: Qualitative comparison on Steoro4D test set. Note that for MoVieS we use a higher default resolution (504×504).

Global anchoring. If the anchor weights remain fixed globally, consolidation degenerates into an importance-weighted ℓ_2 regularizer. This stabilizes inference-time adaptation, but does not encode temporal continuity beyond the fixed prior, similar to weight decay.

Streaming anchoring. Under streaming (w/o EMA) update, the anchor is reset to the current fast weights at the beginning of each chunk. The consolidation term then only regularizes within-chunk drift, applying adaptive shrinkage to the accumulated fast-weight change. This configuration lacks memory consolidation across chunks, making it more prone to overfitting.

Streaming-EMA anchoring. The non-trivial, genuinely elastic behavior emerges when streaming anchors are combined with EMA updates. The consolidation term acts as a low-pass, importance-weighted constraint on the fast-weight trajectory, penalizing cumulative drift relative to an dynamically evolving consolidated anchor weight rather than the instantaneous update.

4.2 Elasticity Improves Generalization

As shown in Table 1, we observe a clear gap between training and test PSNR, i.e., training vs. test ℓ_2, which points to substantial overfitting. This generalization gap is reduced by consolidation, suggesting that consolidation improves information transfer across chunks while also suppressing fast-weight drift caused by repeated fully plastic inference-time updates. We hypothesize that LaCT-LVSM tends to exploit local pattern shortcuts, effectively memorizing localized cues within its limited fast-weight memory instead of maintaining a more distributed spatiotemporal representation, consistent with similar findings in other efficient architectures [62]. We next provide a deeper analysis of what LaCT-LVSM overfits to in practice.

Model	Stereo4D [16]				NVIDIA [61]			
	Resolution	PSNR↑	LPIPS↓	SSIM↑	Resolution	PSNR↑	LPIPS↓	SSIM↑
Optimization-based								
SoM [49]	—— OOT* ——				379 × 672	15.30	0.509	0.317
MoSca [23]	—— OOT* ——				379 × 672	21.45	0.265	0.712
Rendering-based								
L4GM [35]	—— OOT† ——				256 × 256	10.07	0.587	0.235
4DGT [56]	504 × 504	24.62	0.102	0.785	504 × 504	14.13	0.640	0.131
MoVieS [26]	504 × 504	27.19	0.114	0.888	379 × 672	19.16	0.315	0.514
FSM-LRM	256 × 256	27.29	0.147	0.876	256 × 256	20.17	0.337	0.567
FSM-LVSM	256 × 256	**32.16**	**0.043**	**0.931**	256 × 256	**23.90**	**0.105**	**0.747**

* *SoM takes around 10min per scene and MoSca takes around 45min per scene.*
† *L4GM requires multi-view diffusion as prior.*

Table 2: 4D NVS Results. Metrics are resolution-dependent (e.g., higher resolutions typically produce higher PSNR). We adopt the lowest resolution for meaningful comparison with baselines. Steoro4D test set contains 7109 scenes, which is out of time (OOT) for some methods.

Setups. Figure 5 examines how LaCT and LaCET behave under different test-time input densities. Both models are trained with 32 input images, and we vary the number of input frames at inference on 136-frame Stereo4D clips. In the discrete-view setting, input and target frames are uniformly sampled across the full span. In the continuous-view setting, we crop a contiguous sub-sequence (e.g., 40 frames for the 32-in/8-out case) and mask the target frames within that window, reducing the problem to frame interpolation. Two settings converge when the full 136-frame span is used.

LaCET consistently dominates LaCT under sparse inputs. When input views are sparse in time and space, the advantages of LaCET are large and systematic across all PSNR/SSIM/LPIPS metrics. Both LaCET (4 chunks) and LaCT (4 chunks) degrade sharply as sparsity increases, while LaCT (1 chunk) collapses gracefully, as more activation memory is used to process the full sequence (which is not sustainable for longer sequences). Nevertheless, smaller chunks remain appealing due to their reduced activation memory footprint, since backpropagation spans fewer samples, making them more suitable for scaling and for real streaming applications.

LaCET mitigates camera-pose interpolation shortcuts. In the continuous-view regime, LaCET (4 chunks) begins to outperform both LaCT (1 chunk) but still outperforms LaCT (4 chunks). This behavior reveals that LaCT learns to exploit short-range temporal redundancy rather than learning a true view-conditioned spatial representation. When input frames are continuous, the task effectively degenerates into a frame interpolation problem. The model can simply latch onto neighboring frames in the context window and does not need to perform genuine NVS for 4D representation, i.e., no camera-pose extrapolation or long-range temporal modeling. [32] made similar observations. LaCET still improves with more continuous inputs, but the gap between discrete-view and continuous-view performance is substantially smaller. This indicates that LaCET is less prone to collapsing into an interpolation-only solution and instead preserves the ability to model long-range 4D dynamics.

Model	DL3DV [27]			
	Resolution	PSNR↑	LPIPS↓	SSIM↑
Static Models				
DepthSplat [55]	512 × 448	17.81	0.356	0.596
GS-LRM [66]	256 × 256	23.02	0.266	0.705
LVSM [15]	256 × 256	23.10	0.257	0.703
RayZer† [14]	256 × 256	23.72	0.222	0.733
LongLRM [74]	540 × 960	24.10	0.254	0.783
tttLRM [45]	540 × 960	25.07	0.215	0.822
tttLVSM [69]	540 × 960	**26.90**	0.185	0.837
FSM-LRM	256 × 256	23.59	0.206	0.766
FSM-LVSM	256 × 256	26.69	**0.091**	**0.846**
Dynamic Models				
FSM-LRM	256 × 256	21.89	0.314	0.692
FSM-LVSM	256 × 256	24.61	0.118	0.787

† *RayZer ignores input poses and uses target reference images instead, placing it somewhere between pose-conditioned and fully pose-free approaches.*

Table 3: 3D NVS Results. Metrics are resolution-dependent (e.g., higher resolutions typically produce higher PSNR). We adopt the lowest resolution for meaningful comparison with baselines.

Fig. 7: Qualitative comparison on DL3DV benchmark.

5 Scaling LaCET for Fast Spatial Memory

5.1 Pretraining Curriculum

Based on the controlled studies described above, we default LaCET blocks to (i) the streaming-EMA anchor update policy and (ii) the SI-style importance estimate for empirically better training stability. We train both the FSM-LVSM and FSM-LRM variants. Given compute limitations, we bootstrap the LVSM variant from a DL3DV-pretrained LaCT backbone with a resolution of 128, introduce additional temporal encodings, and continue pretraining it for pose-conditioned 4D reconstruction. For data scheduling, we employ a long-context curriculum that gradually increases the input resolution (128 → 256), the temporal span

(128 → 256), and dynamic number of input views as training progresses. Complete implementation details are available in Appendix **??**.

5.2 Novel View Synthesis Performance

For fair comparisons, we report the highest score among (i) our reproduced results, (ii) reported by the authors, and (iii) those reported by the community. Note that metrics like PSNR are resolution-dependent (e.g., higher resolutions typically produce higher PSNR). We adopt the lowest resolution (256×256) for meaningful comparison with baselines.

4D Novel View Synthesis. Unlike 3D NVS, there is currently no well-established benchmark for feedforward 4D evaluation. Existing datasets were originally designed for optimization-based pipelines, and the community has not yet converged on a standard evaluation protocol. We use the NVIDIA [61] benchmarks (with the same evaluation setup in [26]) and Steoro4D [16] benchmarks for fair comparison within this regime. In Table 2, we show our method outperforms existing approaches evaluated at similar resolutions. In particular, on Stereo4D our model achieves clear improvements over prior rendering-based methods across all metrics. On the NVIDIA benchmark, our method achieves the best performance among feed-forward approaches at 256×256 resolution, and approaches the performance of the strongest optimization-based methods, which require per-scene test-time optimization. These results suggest that the proposed LaCET effectively benefits dynamic scene modeling, where maintaining consistent spatial information across time becomes critical.

3D Novel View Synthesis. We use the DL3DV-140 benchmark [27] for evaluation. Since evaluation metrics scale with resolution, we adopt the minimal 256 resolution to ensure fair comparison across both categories. In Table 3, we show our method delivers performance comparable to existing approaches evaluated at similar resolutions, demonstrating that the proposed LaCET blocks preserve strong capability on static scenes where spatial memory is less critical.

6 Related Work

Fast Weights and Test-Time Training (TTT). Recent sequence models have increasingly been viewed as inference-time learning or regression systems, where recurrent state updates implement online learning from context [2, 10, 28, 44]. This connects them to fast weights [38]: in-context parameters that act as associative memories [5, 34] and balance retention with adaptation, as in DeltaNet [37, 59]. Test-Time Training (TTT) extends this idea to neural modules updated online with self-supervised signals [40, 47], while recent work explores specialized optimizers and objectives [3, 4, 18] for video generation, 3D reconstruction, and beyond [7, 9, 69]. However, naïve TTT remains limited by poor hardware utilization, limited state capacity, and unstable long-horizon dynamics [41]. Large-Chunk Test-Time Training (LaCT) improves efficiency with

in-forward fast-weight updates over larger contexts [30, 69], but its fully plastic dynamics can still overfit or forget over long sequences. We address this issue with Elastic TTT, which stabilizes adaptation by adding elasticity across chunks.

Large Rendering-Based Reconstruction Models. Large Reconstruction Models (LRMs) provide a unified framework for view-consistent 3D reconstruction from few posed views, using triplane-based NeRFs [12, 13, 24, 48] or Gaussian Splatting [45, 53, 66, 73, 74] to encode priors over shape and appearance. In 4D, existing LRMs still often rely on posed inputs, explicit Gaussian primitives, and geometric supervision for rendering consistency [25, 26, 31, 35, 56, 58]. Large View Synthesis Models (LVSMs) relax these constraints by synthesizing views without explicit geometric representations [15, 20, 69], sometimes with self-supervised autoencoding reconstruction [8, 14, 32]. Our work follows this rendering-based direction by learning scene-level spatiotemporal representations and instantiating them both with and without minimal geometric priors. In parallel, feed-forward geometry-centric models [42, 46, 51, 57, 65] and their 4D counterparts [7, 11, 50, 64, 71, 72] estimate dynamic geometry or camera poses, but do not directly target novel view-time synthesis. We instead treat novel view-time synthesis as the core objective for 4D representation learning, following prior work [20, 30, 69] on architecture training, evaluation, and scaling.

7 Conclusion and Limitations

Scaling to Longer Sequences. LaCET enables fast inference-time adaptation from, in principle, arbitrarily long sequences in a single forward pass, making activation memory less of a bottleneck. Due to limited licensable training data, benchmarks, and compute, we focus on the architectural advance rather than fully scaling the model.

Geometrically Faithful 4D Reconstruction. While NVS is central to spatial intelligence, it does not guarantee geometric faithfulness or temporally consistent motion. Accurate 4D geometry may require constraints beyond view synthesis, and the role of explicit geometric supervision remains open. LaCET reduces interpolation of nearby context frames, but this behavior does not fully disappear under rendering-only supervision. Depth, correspondence, multi-view consistency, or optical flow may further mitigate this issue.

Pose Estimation in Dynamic Scenes. Recent works have explored 3D reconstruction from unposed images [14, 32, 48], but jointly estimating intrinsics and poses in dynamic scenes remains challenging. We assume posed input images and do not target unposed reconstruction in this work.

Acknowledgment. The authors would like to thank Zefan Cai, Xuweiyi Chen, Yinpei Dai, Yilun Du, Chenguo Lin, Freda Shi, Hao Tan, Zeyuan Yang, and Tianyuan Zhang for their insightful discussions.

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: European conference on computer vision (ECCV). pp. 139–154 (2018)
2. Behrouz, A., Li, Z., Kacham, P., Daliri, M., Deng, Y., Zhong, P., Razaviyayn, M., Mirrokni, V.: Atlas: Learning to optimally memorize the context at test time. arXiv preprint arXiv:2505.23735 (2025)
3. Behrouz, A., Razaviyayn, M., Zhong, P., Mirrokni, V.: It's all connected: A journey through test-time memorization, attentional bias, retention, and online optimization. arXiv preprint arXiv:2504.13173 (2025)
4. Behrouz, A., Zhong, P., Mirrokni, V.: Titans: Learning to memorize at test time. In: Conference on Neural Information Processing Systems (2025)
5. Bietti, A., Cabannes, V., Bouchacourt, D., Jegou, H., Bottou, L.: Birth of a transformer: A memory viewpoint. In: Conference on Neural Information Processing Systems. pp. 1560–1588 (2023)
6. Chan, L.C., Whiteman, P.: Hardware-constrained hybrid coding of video imagery. IEEE Transactions on Aerospace and Electronic Systems (1), 71–84 (1983)
7. Chen, X., Chen, Y., Xiu, Y., Geiger, A., Chen, A.: Ttt3r: 3d reconstruction as test-time training. In: International Conference on Learning Representations (2026)
8. Chen, X., Zhou, W., Cheng, Z.: Wildrayzer: Self-supervised large view synthesis in dynamic environments. In: Conference on Computer Vision and Pattern Recognition (2026)
9. Dalal, K., Koceja, D., Xu, J., Zhao, Y., Han, S., Cheung, K.C., Kautz, J., Choi, Y., Sun, Y., Wang, X.: One-minute video generation with test-time training. In: Conference on Computer Vision and Pattern Recognition. pp. 17702–17711 (2025)
10. Dherin, B., Munn, M., Mazzawi, H., Wunder, M., Gonzalvo, J.: Learning without training: The implicit dynamics of in-context learning. arXiv preprint arXiv:2507.16003 (2025)
11. Feng, H., Zhang, J., Wang, Q., Ye, Y., Yu, P., Black, M.J., Darrell, T., Kanazawa, A.: St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In: International Conference on Computer Vision. pp. 8503–8513 (2025)
12. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: International Conference on Learning Representations (2024)
13. Jiang, H., Huang, Q., Pavlakos, G.: Real3d: Scaling up large reconstruction models with real-world images. In: International Conference on Computer Vision. pp. 5821–5833 (2025)
14. Jiang, H., Tan, H., Wang, P., Jin, H., Zhao, Y., Bi, S., Zhang, K., Luan, F., Sunkavalli, K., Huang, Q., et al.: Rayzer: A self-supervised large view synthesis model. In: International Conference on Computer Vision (2025)
15. Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: LVSM: A large view synthesis model with minimal 3d inductive bias. In: International Conference on Learning Representations (2025)
16. Jin, L., Tucker, R., Li, Z., Fouhey, D., Snavely, N., Holynski, A.: Stereo4d: Learning how things move in 3d from internet stereo videos. In: Conference on Computer Vision and Pattern Recognition. pp. 10497–10509 (2025)
17. Jordan, K., Jin, Y., Boza, V., Jiacheng, Y., Cecista, F., Newhouse, L., Bernstein, J.: Muon: An optimizer for hidden layers in neural networks. `https://kellerjordan.github.io/posts/muon` (2024)

18. Karami, M., Mirrokni, V.: Lattice: Learning to efficiently compress the memory. arXiv preprint arXiv:2504.05646 (2025)
19. Kerr, J., Kim, C.M., Wu, M., Yi, B., Wang, Q., Goldberg, K., Kanazawa, A.: Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In: Conference on Robot Learning (2024)
20. Kim, E., Ryu, H., Mitchel, T.W., Sitzmann, V.: Scaling view synthesis transformers. arXiv preprint arXiv:2602.21341 (2026)
21. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)
22. Krause, B., Kahembwe, E., Murray, I., Renals, S.: Dynamic evaluation of neural sequence models. In: International Conference on Machine Learning. pp. 2766–2775 (2018)
23. Lei, J., Weng, Y., Harley, A.W., Guibas, L., Daniilidis, K.: Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In: Conference on Computer Vision and Pattern Recognition. pp. 6165–6177 (2025)
24. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In: International Conference on Learning Representations (2024)
25. Liang, H., Ren, J., Mirzaei, A., Torralba, A., Liu, Z., Gilitschenski, I., Fidler, S., Oztireli, C., Ling, H., Gojcic, Z., Huang, J.: Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. In: Conference on Neural Information Processing Systems (2025)
26. Lin, C., Lin, Y., Pan, P., Yu, Y., Hu, T., Yan, H., Fragkiadaki, K., Mu, Y.: Movies: Motion-aware 4d dynamic view synthesis in one second. In: Conference on Computer Vision and Pattern Recognition (2026)
27. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
28. Liu, B., Wang, R., Wu, L., Feng, Y., Stone, P., qiang liu: Longhorn: State space models are amortized online learners. In: International Conference on Learning Representations (2025)
29. Liu, J., Su, J., Yao, X., Jiang, Z., Lai, G., Du, Y., Qin, Y., Xu, W., Lu, E., Yan, J., et al.: Muon is scalable for llm training. arXiv preprint arXiv:2502.16982 (2025)
30. Liu, J., Elflein, S., Litany, O., Gojcic, Z., Li, R.: Test-time training with kv binding is secretly linear attention. arXiv preprint arXiv:2602.21204 (2026)
31. Ma, Z., Chen, X., Yu, S., Bi, S., Zhang, K., Ziwen, C., Xu, S., Yang, J., Xu, Z., Sunkavalli, K., et al.: 4d-lrm: Large space-time reconstruction model from and to any view at any time. In: Conference on Neural Information Processing Systems (2025)
32. Mitchel, T., Ryu, H., Sitzmann, V.: True self-supervised novel view synthesis is transferable. In: International Conference on Learning Representations (2026)
33. Plücker, J.: Xvii. on a new geometry of space. Philosophical Transactions of the Royal Society of London (155), 725–791 (1865)
34. Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Gruber, L., Holzleitner, M., Adler, T., Kreil, D., Kopp, M.K., Klambauer, G., Brandstetter, J., Hochreiter, S.: Hopfield networks is all you need. In: International Conference on Learning Representations (2021)

35. Ren, J., Xie, C., Mirzaei, A., Kreis, K., Liu, Z., Torralba, A., Fidler, S., Kim, S.W., Ling, H., et al.: L4gm: Large 4d gaussian reconstruction model. In: Conference on Neural Information Processing Systems. pp. 56828–56858 (2024)
36. Salimans, T., Kingma, D.P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: Conference on Neural Information Processing Systems (2016)
37. Schlag, I., Irie, K., Schmidhuber, J.: Linear transformers are secretly fast weight programmers. In: International conference on machine learning. pp. 9355–9366 (2021)
38. Schmidhuber, J.: Learning to control fast-weight memories: An alternative to dynamic recurrent networks. Neural Computation **4**(1), 131–139 (1992)
39. Shazeer, N.: Glu variants improve transformer. arXiv preprint arXiv:2002.05202 (2020)
40. Sun, Y., Li, X., Dalal, K., Xu, J., Vikram, A., Zhang, G., Dubois, Y., Chen, X., Wang, X., Koyejo, S., et al.: Learning to (learn at test time): Rnns with expressive hidden states. In: International Conference on Machine Learning. pp. 57503–57522 (2025)
41. Tandon, A., Dalal, K., Li, X., Koceja, D., Rød, M., Buchanan, S., Wang, X., Leskovec, J., Koyejo, S., Hashimoto, T., et al.: End-to-end test-time training for long context. arXiv preprint arXiv:2512.23675 (2025)
42. Tang, Z., Fan, Y., Wang, D., Xu, H., Ranjan, R., Schwing, A., Yan, Z.: Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In: Conference on Computer Vision and Pattern Recognition (2024)
43. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Conference on Neural Information Processing Systems (2017)
44. Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., Vladymyrov, M.: Transformers learn in-context by gradient descent. In: International Conference on Machine Learning. pp. 35151–35174 (2023)
45. Wang, C., Tan, H., Yifan, W., Chen, Z., Liu, Y., Sunkavalli, K., Bi, S., Liu, L., Hu, Y.: tttlrm: Test-time training for long context and autoregressive 3d reconstruction. In: Conference on Computer Vision and Pattern Recognition (2026)
46. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Conference on Computer Vision and Pattern Recognition. pp. 5294–5306 (2025)
47. Wang, K.A., Shi, J., Fox, E.B.: Test-time regression: a unifying framework for designing sequence models with associative memory. arXiv preprint arXiv:2501.12352 (2025)
48. Wang, P., Tan, H., Bi, S., Xu, Y., Luan, F., Sunkavalli, K., Wang, W., Xu, Z., Zhang, K.: Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. In: International Conference on Learning Representations (2024)
49. Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., Kanazawa, A.: Shape of motion: 4d reconstruction from a single video. In: International Conference on Computer Vision. pp. 9660–9672 (2025)
50. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: Conference on Computer Vision and Pattern Recognition. pp. 10510–10522 (2025)
51. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024)

52. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
53. Xie, D., Bi, S., Shu, Z., Zhang, K., Xu, Z., Zhou, Y., Pirk, S., Kaufman, A., Sun, X., Tan, H.: Lrm-zero: Training large reconstruction models with synthesized data. In: Conference on Neural Information Processing Systems (2024)
54. Xie, Y., Yao, C.H., Voleti, V., Jiang, H., Jampani, V.: SV4d: Dynamic 3d content generation with multi-frame and multi-view consistency. In: International Conference on Learning Representations (2025)
55. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. In: Conference on Computer Vision and Pattern Recognition. pp. 16453–16463 (2025)
56. Xu, Z., Li, Z., Dong, Z., Zhou, X., Newcombe, R., Lv, Z.: 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. In: Conference on Neural Information Processing Systems (2025)
57. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: Conference on Computer Vision and Pattern Recognition (2025)
58. Yang, J., Huang, J., Chen, Y., Wang, Y., Li, B., You, Y., Sharma, A., Igl, M., Karkus, P., Xu, D., et al.: Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. In: International Conference on Learning Representations (2025)
59. Yang, S., Wang, B., Zhang, Y., Shen, Y., Kim, Y.: Parallelizing linear transformers with the delta rule over sequence length. In: Conference on Neural Information Processing Systems. pp. 115491–115522 (2024)
60. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In: International Conference on Learning Representations (2024)
61. Yoon, J.S., Kim, K., Gallo, O., Park, H.S., Kautz, J.: Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In: Conference on Computer Vision and Pattern Recognition. pp. 5336–5345 (2020)
62. You, W., Tang, Z., Li, J., Yao, L., Zhang, M.: Revealing and mitigating the local pattern shortcuts of mamba. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 12156–12178 (2025)
63. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International conference on machine learning. pp. 3987–3995 (2017)
64. Zhang, J., Herrmann, C., Hur, J., Jampani, V., Darrell, T., Cole, F., Sun, D., Yang, M.H.: Monst3r: A simple approach for estimating geometry in the presence of motion. In: International Conference on Learning Representations (2025)
65. Zhang, J., Herrmann, C., Hur, J., Sun, C., Yang, M.H., Cole, F., Darrell, T., Sun, D.: Loger: Long-context geometric reconstruction with hybrid memory. arXiv preprint arXiv:2603.03269 (2026)
66. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: European Conference on Computer Vision. pp. 1–19 (2024)
67. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields. In: European Conference on Computer Vision. pp. 717–733 (2022)
68. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: 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)

69. Zhang, T., Bi, S., Hong, Y., Zhang, K., Luan, F., Yang, S., Sunkavalli, K., Freeman, W.T., Tan, H.: Test-time training done right. In: International Conference on Learning Representations (2026)
70. Zhen, H., Sun, Q., Zhang, H., Li, J., Zhou, S., Du, Y., Gan, C.: Learning 4d embodied world models. In: International Conference on Computer Vision. pp. 5337–5347 (2025)
71. Zhou, K., Wang, Y., Chen, G., Beaudouin, G., Zhan, F., Liang, P.P., Wang, M.: Page-4d: Disentangled pose and geometry estimation for 4d perception. In: International Conference on Learning Representations (2026)
72. Zhuo, D., Zheng, W., Guo, J., Wu, Y., Zhou, J., Lu, J.: Streaming 4d visual geometry transformer. arXiv preprint arXiv:2507.11539 (2025)
73. Ziwen, C., Tan, H., Wang, P., Xu, Z., Fuxin, L.: Long-lrm++: Preserving fine details in feed-forward wide-coverage reconstruction. arXiv preprint arXiv:2512.10267 (2025)
74. Ziwen, C., Tan, H., Zhang, K., Bi, S., Luan, F., Hong, Y., Fuxin, L., Xu, Z.: Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In: International Conference on Computer Vision. pp. 4349–4359 (2025)