MeMix: Writing Less, Remembering More for Streaming 3D Reconstruction

Jiacheng Dong[2*] **Huan Li**[1*] **Sicheng Zhou**[2*] **Wenhao Hu**[2] **Weili Xu**[2] **Yan Wang**[1†]

[1]Institute for AI Industry Research, Tsinghua University [2]Zhejiang University

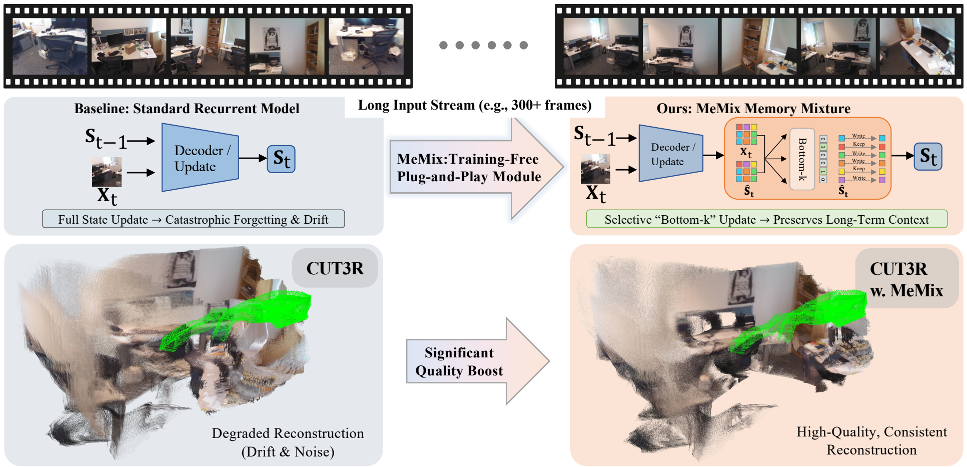

Long Input Stream (e.g., 300+ frames)
Baseline: Standard Recurrent Model
s_{t-1}
x_t
Decoder / Update
s_t
Full State Update → Catastrophic Forgetting & Drift
MeMix:Training-Free Plug-and-Play Module
Ours: MeMix Memory Mixture
s_{t-1}
x_t
Decoder / Update
x_t
$\hat{s}_t$
Bottom-k
0 1 0 0 1 0
Write
Keep
Write
Write
Keep
Write
$\hat{s}_t$
s_t
Selective "Bottom-k" Update → Preserves Long-Term Context
CUT3R
Degraded Reconstruction (Drift & Noise)
Significant Quality Boost
CUT3R w. MeMix
High-Quality, Consistent Reconstruction

Figure 1: **MeMix.** A training-free, plug-and-play state-update module for recurrent streaming 3D reconstruction. MeMix recasts the recurrent state as a mixture of memory patches, updates Bottom-k patches and preserves the rest. This reduces interference and improves long-horizon stability with $O(1)$ inference memory.

Abstract

Reconstruction is a fundamental task in 3D vision and a fundamental capability for spatial intelligence. Particularly, streaming 3D reconstruction is central to real-time spatial perception, yet existing recurrent online models often suffer from progressive degradation on long sequences due to state drift and forgetting, motivating inference-time remedies. We present MeMix, a training-free, plug-and-play module that improves streaming reconstruction by recasting the recurrent state into a **Me**mory **Mix**ture. MeMix partitions the state into multiple independent memory patches and updates only the least-aligned memory patches while exactly preserving others. This selective update mitigates catastrophic forgetting while retaining $O(1)$ inference memory, and requires no fine-tuning or additional learnable parameters, making it directly applicable to existing recurrent reconstruction models. Across standard benchmarks (ScanNet, 7-Scenes, KITTI, etc.), under identical backbones and inference settings, MeMix reduces reconstruction completeness error by **15.3%** on average (up to **40.0%**) across 300–500 frame streams on 7-Scenes. The code is available at https://dongjiacheng06.github.io/MeMix/

[*] Equal contribution.
[†] Corresponding author.

1 Introduction

End-to-end 3D reconstruction aims to directly infer camera poses and scene structure from a set of input RGB images, enabling efficient recovery of 3D structure for downstream tasks. Existing methods broadly fall into two paradigms: offline batch reconstruction [1, 2] and streaming online reconstruction [3–5]. Offline batch methods process complete image sequences with global optimization or global-consistency modeling, achieving high reconstruction quality. However, they cannot process arbitrarily long sequences under bounded resources, and their high latency is incompatible with downstream applications that demand real-time spatial perception, such as autonomous driving and robotic navigation [6, 7]. These constraints motivate streaming online reconstruction, which incrementally consumes a continuously arriving RGB stream and updates geometry and poses in real time.

However, extending online reconstruction to long sequences faces a fundamental tension between exploiting historical context and maintaining constant inference. One family of methods leverages causal-attention KV caches to store the full history [4, 5, 8]. However, this approach incurs memory growth proportional to sequence length, usually leading to out-of-memory errors over long horizons.

An alternative method is fixing latent states to summarize historical context. CUT3R [3] formulates reconstruction as a recurrent model with linear attention [9, 10], achieving $O(1)$ inference memory and computation. TTT3R [11] reinterprets this under test-time training, selectively suppressing low-quality updates with adaptive learning rate. Yet since each frame writes into the same set of state tokens, previously stored memories are updated by new information, leading to catastrophic forgetting [12, 13]. In practice, this manifests as geometric drift, accumulated pose errors, and degraded long-range consistency.

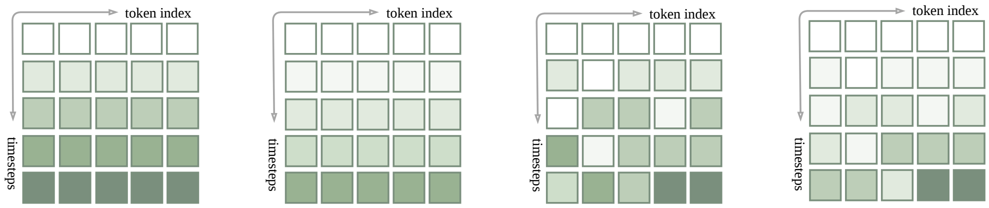
To address this, we revisit the mixture-of-memories (MoM) idea [14] from an engineering perspective. Recent online reconstruction improvements are often presented as standalone methods, and obtaining gains usually requires model-specific redesigns with nontrivial code changes, making reuse across backbones difficult. In contrast, we propose **MeMix** as a *training-free, plug-and-play* state-update module that can be inserted into existing fixed-state recurrent reconstruction pipelines [3, 11, 15]. Concretely, we partition the state into independent memory patches [16, 17] and update `Bottom-k` patches at each timestep while the others are preserved. This design reduces cross-time interference without introducing new learnable parameters or any fine-tuning, and we verify clear improvements on three representative baselines.

Our contribution can be summarized as follows:

1. We introduce MeMix, a training-free, plug-in memory update module that recasts the recurrent state as a mixture of memory patches, substantially improving long-sequence reconstruction quality.
2. We identify a fundamental bottleneck in fixed-state streaming 3D reconstruction: fully rewriting the recurrent state at each step causes cumulative interference and catastrophic forgetting in long-horizon inference.
3. MeMix can integrate seamlessly into mainstream recurrent reconstruction models, consistently improving performance with negligible overhead in GPU memory and inference latency.

2 Related Work

Feedforward Offline Reconstruction. Methods such as DUSt3R [1, 18] encode image features via a ViT [19] encoder and achieve 3D matching with cross attention, but they only support pairwise image inputs, and multi-view processing relies on post-processing. VGGT [2] uses a feedforward Transformer with intra-frame and global self-attention to establish geometric constraints, ensuring global geometric consistency across multiple views. Subsequent methods [20–22] have further optimized VGGT in inference speed and reconstruction accuracy. However, these offline methods require the full image sequence at inference time and cannot handle arbitrarily long streams under bounded resources, while downstream applications demand real-time perception [6, 7]. Thus, there is a growing demand for online reconstruction.



(a) Memory of CUT3R

(b) Memory of TTT3R

(c) Memory of CUT3R (w. MeMix)

(d) Memory of TTT3R (w. MeMix)

Figure 2: **Where to write: Mixture Memory Updates.** (a) CUT3R overwrites all state tokens at every timestep. (b) TTT3R applies a dense per-token gate to modulate how much to write, but still updates every token. (c–d) MeMix enables where-to-write updates via Mixture Memory: only a subset of memory patches/tokens are written while the rest are exactly preserved, and it can be plugged into CUT3R (c) or combined with TTT-style gating (d). Colored token squares 🟩🟩🟩🟩 indicate tokens that are progressively reinforced over time.

Feedforward Online Reconstruction. Online reconstruction methods [3–5, 23, 24] incrementally accept inputs and produce geometry in real time. Existing approaches can be categorized by how they manage historical context. KV-cache based methods [5, 25, 26] store historical features in a causal-attention KV cache, retaining long-range context but incurring memory growth with sequence length. Fixed-state methods such as CUT3R [3] and TTT3R [11] maintain a fixed-length recurrent state, achieving constant-memory inference. CUT3R interacts input tokens with memory states via cross-attention, but errors accumulate over long sequences due to unconditional full-step writes. TTT3R reinterprets state update as test-time learning [27, 28], which eases drift but still suffers from state degradation. Point3R [4] anchors historical tokens to explicit 3D point positions, achieving strong recall but with memory consumption that grows linearly with the number of views. While effective, many improvements in this line are tightly coupled to specific architectures and training/inference pipelines, so transferring the same idea to another recurrent backbone often requires substantial re-implementation and engineering effort. This limits practical reusability in real systems where model stacks evolve quickly.

Memory Mixture. In sequence modeling [29], linear attention and state-space models have explored various gating mechanisms to control information retention: RetNet [30] employs exponential decay for multi-scale retention; Mamba [31] introduces input-dependent selection for selective state propagation; DeltaNet [9] and Gated DeltaNet [32] adopt the delta rule to address key collisions in additive state updates; and Titans [33] introduces a neural long-term memory module that learns to memorize at test time through nested optimization. Subsequent work [34–36] has further deepened the theoretical understanding of how gating controls information flow. However, these approaches all employ continuous gates that never produce exact zeros [37], meaning every state dimension receives a nonzero update at every step. MoM [14] brings sparse routing into linear attention, partitioning the recurrent state into independent memory blocks to reduce cross-time interference. Our work follows this direction and builds a training-free, plug-and-play state-update module for online 3D reconstruction, validating its effectiveness across three or more recurrent baselines.

3 Method

MeMix is a training-free method to modify current online 3D reconstruction models [3, 11, 15]. Its core update is executed during the forward pass without any parameter fine-tuning or additional learnable modules. Inspired by related works in memory mixture [10, 14, 33, 38], our key idea is to design the recurrent state as memory blocks [14] and selectively update, to reduce cross-time interference under a fixed state.

3.1 Reconstruction with Continuous Update

Given a continuous image stream $\{\mathbf{I}_t\}_{t=1}^{T}$, we aim to estimate per-frame camera pose $\mathbf{T}_t$, intrinsic $\mathbf{K}_t$, and pixel-aligned pointmap $\mathbf{P}_t$ in an online fashion. Following CUT3R [3], the process is formulated as a recurrent sequence model operating on a fixed-length state:

$$\mathbf{X}_t = \texttt{Tokenizer}(\mathbf{I}_t) \quad \mathbf{S}_t = \texttt{Update}(\mathbf{S}_{t-1},\, \mathbf{X}_t) \quad \mathbf{Y}_t = \texttt{Read}(\mathbf{S}_t,\, \mathbf{X}_t) \quad \mathcal{M}_t = \texttt{Head}(\mathbf{Y}_t) \quad (1)$$

where $\mathbf{X}_t \in \mathbb{R}^{n\times d}$ are image tokens, $\mathbf{S}_t \in \mathbb{R}^{n\times d}$ is the recurrent state initialized from learnable embeddings, and $\mathbf{Y}_t$ are decoded tokens from which $(\mathbf{T}_t, \mathbf{K}_t, \mathbf{P}_t)$ are regressed.

State Input Interaction.

The `Update` and `Read` are realized jointly by an L-layer dual-stream cross-attention decoder. Focusing on the state stream, each layer performs:

$$\mathbf{S}^{(\ell)} = \mathbf{S}^{(\ell-1)} + \texttt{softmax}\Big(\mathbf{Q}_{\mathbf{S}}^{(\ell)}\, {\mathbf{K}_{\mathbf{X}}^{(\ell)}}^{\top}\Big)\, \mathbf{V}_{\mathbf{X}}^{(\ell)} \qquad (2)$$

where $\mathbf{Q}_{\mathbf{S}}^{(\ell)}$ is projected from $\mathbf{S}^{(\ell-1)}$ and $\mathbf{K}_{\mathbf{X}}^{(\ell)}, \mathbf{V}_{\mathbf{X}}^{(\ell)}$ from $\mathbf{X}^{(\ell-1)}$. A symmetric stream updates $\mathbf{X}^{(\ell)}$ by attending to $\mathbf{S}^{(\ell-1)}$. After L layers, $\mathbf{Y}_t = \mathbf{X}^{(L)}$ is fed to the prediction head Eq. 16.

Continuous State Update.

The continuous update method [3] directly takes the decoder output as the new state. Expanding Eq. (2) across L layers:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \underbrace{\sum_{\ell=1}^{L} \texttt{softmax}\Big(\mathbf{Q}_{\mathbf{S}}^{(\ell)}\, {\mathbf{K}_{\mathbf{X}}^{(\ell)}}^{\top}\Big)\, \mathbf{V}_{\mathbf{X}}^{(\ell)}}_{\Delta \mathbf{S}_t} \qquad (3)$$

where $\Delta\mathbf{S}_t$ is the accumulated cross-attention residual [39–41]. This unconditional full-step write, where information from earlier frames is erased by new features, leads to geometric degradation on long sequences.

Test-Time Learning for State Update.

Another method [11] reinterprets the state update through the lens of test-time training. Viewing $\mathbf{S}$ as model parameters and each incoming frame $\mathbf{X}_t$ as a test sample, $\Delta\mathbf{S}_t$ in Eq. (3) can be seen as a gradient step [28] that minimizes a self-supervised loss on $\mathbf{X}_t$. This method derives $\boldsymbol{\beta}_t$ by aggregating the attention map and scales the entire residual $\Delta\mathbf{S}_t$ before it is added back to $\mathbf{S}_{t-1}$. As shown in Eq. (4)

$$\boldsymbol{\beta}_t = \sigma\Bigg(\frac{1}{LHm}\sum_{\ell,h,j} Q_{S_{t-1}} K_{X_t}^{T}\Bigg) \qquad (4)$$

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{\beta}_t \odot \sum_{\ell=1}^{L} \texttt{softmax}\Big(\mathbf{Q}_{\mathbf{S}}^{(\ell)}\, {\mathbf{K}_{\mathbf{X}}^{(\ell)}}^{\top}\Big)\, \mathbf{V}_{\mathbf{X}}^{(\ell)} \qquad (5)$$

Each state token now retains history in proportion to its relevance to the current observation, alleviating drift. Nevertheless, the gate remains dense, so every token receives a nonzero write at every step, differing only in magnitude.

3.2 Rethinking Sparse Update

As mentioned above, when the state is continuously updated, the state at layer ℓ evolves as Eq. (2):

$$\mathbf{S}^{(\ell)} = \mathbf{S}^{(\ell-1)} + \mathbf{A}^{(\ell)}\, \mathbf{V}^{(\ell)}, \quad \mathbf{A}^{(\ell)} = \texttt{softmax}\Big(\mathbf{Q}_{\mathbf{S}}^{(\ell)}\, {\mathbf{K}_{\mathbf{X}}^{(\ell)}}^{\top}\Big)$$

After L layers, the decoder produces a candidate state $\hat{\mathbf{S}}_t = \mathbf{S}^{(L)}$. If there is no gating, the state is directly overwritten: $\mathbf{S}_t = \hat{\mathbf{S}}_t$ as Eq. (3).

Now introduce a gate matrix $\boldsymbol{G_t} \in [0,1]$ that modulates the attention:

$$\mathbf{S}^{(\ell)} = \mathbf{S}^{(\ell-1)} + (\boldsymbol{G}_t \odot \mathbf{A}^{(\ell)})\, \mathbf{V}^{(\ell)} \qquad (6)$$

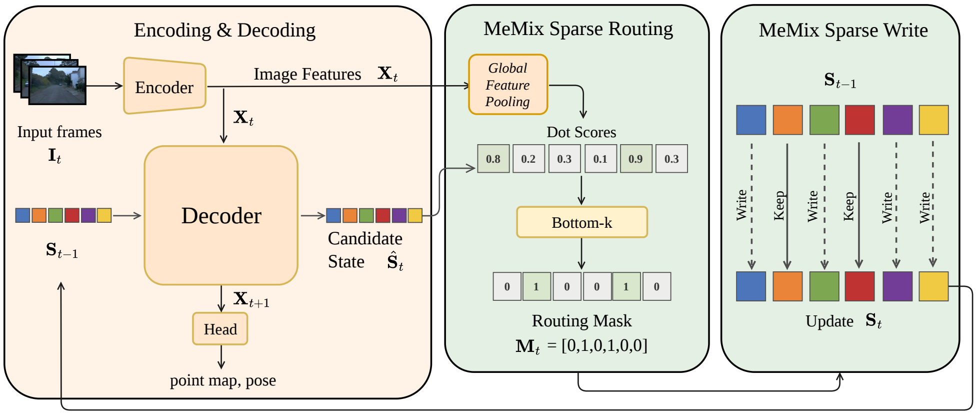


Figure 3: **Overview of MeMix.** A ViT encoder encodes each frame to tokens $\mathbf{X}_t$, which interact with state tokens $\mathbf{S}_{t-1}$ through a dual-stream cross-attention decoder to produce predictions $\mathbf{Y}_t$ and candidate state $\hat{\mathbf{S}}_t$. MeMix computes dot scores between $\hat{\mathbf{S}}_t$ and $\mathbf{X}_t$, selects `Bottom-K` patches to construct a binary mask $\mathbf{M}_t$, updating only `Bottom-K` patches. Decoded image tokens $\mathbf{Y}_t$ are fed to the prediction head for output.

Substituting into the residual connection, the gated candidate state becomes

$$\hat{\mathbf{S}}_t = \mathbf{S}_{t-1} + \sum_{\ell=1}^{L} (\boldsymbol{G}_t \odot \mathbf{A}^{(\ell)}) \mathbf{V}^{(\ell)} \tag{7}$$

The state update can then be written as

$$\mathbf{S}_t = \boldsymbol{G}_t \odot \hat{\mathbf{S}}_t + (1 - \boldsymbol{G}_t) \odot \mathbf{S}_{t-1} \tag{8}$$

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{G}_t \sum_{\ell=1}^{L} \mathbf{A}^{(\ell)} \, \mathbf{V}^{(\ell)} \tag{9}$$

Table 1: **Unified Memory Update Rules.** We write CUT3R, TTT3R, TTSA3R, and their MeMix variants under a shared gate formulation.

Method	**Memory Update Rule**
Unified form	$\boldsymbol{S}_t = \boldsymbol{G}_t \odot \hat{\boldsymbol{S}}_t + (1 - \boldsymbol{G}_t) \odot \boldsymbol{S}_{t-1}$
CUT3R	$\boldsymbol{G}_t = \mathbf{1}$
TTT3R / TTSA3R	$\boldsymbol{G}_t = \boldsymbol{\beta}_t$
CUT3R + MeMix	$\boldsymbol{G}_t = \boldsymbol{M}_t$
TTT3R / TTSA3R + MeMix	$\boldsymbol{G}_t = \boldsymbol{M}_t \odot \boldsymbol{\beta}_t$

when $\boldsymbol{G}_t \in (0, 1)$, Eq. (9) is equivalent to Eq. (5) , when $\boldsymbol{G}_t \in [0, 1]$, we have the sparse update rule.

3.3 MeMix Design

Memory Mixture

Unlike the full update strategy of CUT3R and the dense learning-rate adaptation of TTT3R, MeMix constructs a routing mask $\mathbf{M}_t$ [10] for the state update in Eq. (8). In step t, the decoder produces a candidate state $\hat{\mathbf{S}}_t \in \mathbb{R}^{n \times d}$ and image features $\mathbf{X}_t \in \mathbb{R}^{n \times d}$. The interaction between state token and the observation is measured by dot-product similarity:

$$r_t = \langle \hat{\mathbf{S}}_t, \ \mathbf{X}_t \rangle \tag{10}$$

Then, we select the k patches with the bottom scores:

$$\mathcal{P}_t = \texttt{Bottom-k}\{r_t\} \tag{11}$$

The routing mask is then constructed as M_t, Patches most aligned with the current frame will be constructed as $M_t = 1$, which encode relevant information while the `Bottom-k` patches are not enriched.

State Update

Substituting the routing mask $\mathbf{M}_t$ into Eq. (8):

$$\mathbf{S}_t = \mathbf{M}_t \odot \hat{\mathbf{S}}_t + (1 - \mathbf{M}_t) \odot \mathbf{S}_{t-1} \tag{12}$$

Tokens within the selected patches are fully replaced by the decoder output; all others are exactly preserved. This is the binary-gate instance of Eq. (8) discussed in Sec. 3.2.

State Update with Test-Time Training

Routing mask $\mathbf{M}_t$ determines where to write, yet within the selected patches every token is still fully overwritten. We can further modulate how much to write by combining $\mathbf{M}_t$ with the attention-derived learning rate $\boldsymbol{\beta}_t$ from Eq. (4):

$$\mathbf{S}_t = (\mathbf{M}_t \odot \boldsymbol{\beta}_t) \odot \hat{\mathbf{S}}_t + (1 - \mathbf{M}_t \odot \boldsymbol{\beta}_t) \odot \mathbf{S}_{t-1} \tag{13}$$

Algorithm. MeMix Inference

```
Require: Input sequence $\{I_t\}_{t=1}^{T}$, base model $f$, patch partition $\{P_j\}_{j=1}^{p}$
$S \leftarrow S_0$
$\mathcal{M} \leftarrow \emptyset$
for $t = 1$ to $T$ do
    $X_t \leftarrow \texttt{Tokenize}(I_t)$
    $\hat{S}_t, Y_t \leftarrow \text{cross attn}(S_{t-1}, X_t)$
    $r_t \leftarrow \text{RouteScore}(\hat{S}_t, X_t)$
    $\mathcal{P}_t \leftarrow \texttt{Bottom-K}(r_t)$
    $\mathbf{M_t} \leftarrow \text{Gate}(\mathcal{P}_t)$
    $S_t \leftarrow \mathbf{M_t} \odot \hat{S}_t + (1 - \mathbf{M_t}) \odot S_{t-1}$
    $\mathcal{M}_t \leftarrow \texttt{Head}(Y_t)$
end for
return $\mathcal{M}$
```

For unselected patches $M_{t,i} = 0$, the state is exactly preserved regardless of β_t; for selected patches, β_t provides token-level soft modulation. This combination is still training-free since $\boldsymbol{\beta}_t$ is derived from the existing decoder's cross-attention weights. Note that $\mathbf{M}_t \odot \boldsymbol{\beta}_t \in [0, 1]$, so Eq. (13) remains equivalent to the unified gate framework in Eq. (8).

Readout and Output

Symmetrically to the state stream Eq. (2), the image stream of the dual-stream decoder lets each image token cross-attend to the state at every layer:

$$\mathbf{X}^{(\ell)} = \mathbf{X}^{(\ell-1)} + \texttt{softmax}\Big(\mathbf{Q}_{\mathbf{X}}^{(\ell)} \mathbf{K}_{\mathbf{S}}^{(\ell)^{\top}}\Big) \mathbf{V}_{\mathbf{S}}^{(\ell)} \tag{14}$$

$$\mathbf{X} = \mathbf{X}_{t-1} + \sum_{\ell=1}^{L} \texttt{softmax}\Big(\mathbf{Q}_{\mathbf{X}}^{(\ell)} \mathbf{K}_{\mathbf{S}}^{(\ell)^{\top}}\Big) \mathbf{V}_{\mathbf{S}}^{(\ell)} \tag{15}$$

where $\mathbf{Q}_{\mathbf{X}}^{(\ell)}$ is projected from $\mathbf{X}^{(\ell-1)}$ and $\mathbf{K}_{\mathbf{S}}^{(\ell)}, \mathbf{V}_{\mathbf{S}}^{(\ell)}$ from $\mathbf{S}^{(\ell-1)}$. After L layers, the decoded tokens $\mathbf{Y}_t = \mathbf{X}$ are fed to the DPT head [42, 43]. Finally we have:

$$(\mathbf{T}_t, \mathbf{K}_t, \mathbf{P}_t) = \texttt{Head}(\mathbf{Y}_t) \tag{16}$$

4 Experiments

We evaluate MeMix on multi-view 3D reconstruction, camera pose estimation, and video depth estimation. As a *training-free* and *plug-and-play* module, we keep the backbone weights, input resolution, and inference hyper-parameters identical to the corresponding baseline.

Baselines. Following common practice in recent streaming reconstruction evaluations [3, 11, 15], we compare MeMix with representative offline and online families. We first include strong pairwise 3D reconstruction foundation models, including DUSt3R [1], MASt3R [18], MonST3R [44], and Easi3R [45], which take a pair of views as input and typically require an extra global alignment stage to consolidate pairwise predictions. We also compare with full-attention multiview models such as AETHER [46] and VGGT [2], which can jointly predict pointmaps/cameras but are usually limited to long sequences due to the need to run full attention when new frames arrive.

For online methods, we plug MeMix into recurrent streaming backbones and compare the results with their original versions, including CUT3R [3] and its training-free adaptations TTT3R [11] and TTSA3R [15]. We further include KV-cache / causal streaming Transformers (StreamVGGT [25] and STREAM3R$^{\alpha}$ [5]) to test our overall competitiveness. Finally, we compare with explicit/external memory designs (Spann3R [23], Point3R [4]), which improve long-horizon recall by maintaining additional pointmap memories.

Settings. In all experiments, we run inference at a single NVIDIA A100 40GB PCIe or RTX 4090 (24GB). The `Bottom-k` is set to 708 while the entire state is 768 tokens/frame—this is the most fine-grained design, ensuring that every token can be involved. Following prior training-free methods[11, 15, 47], we apply MeMix to existing pipelines, using the same released checkpoint and evaluation script, without any fine-tuning. Computation costs are shown in Table 4.

4.1 3D Reconstruction

Following common practice in long-horizon streaming reconstruction [3, 11, 25], we evaluate multi-view reconstruction on 7-Scenes [48] and NRGBD [49]. To explicitly probe length generalization under bounded memory, we tested three long sequence lengths (300/400/500 frames) and reported accuracy, completeness, and normal consistency.

As shown in Table 2, offline full-attention models (e.g. VGGT) and KV-cache streaming Transformers (e.g. StreamVGGT) run out of memory at all tested lengths (300/400/500 frames). For constant-memory recurrent baselines, reconstruction quality generally degrades as the input horizon increases. In contrast, inserting MeMix into the same backbone consistently improves 7-Scenes performance in accuracy, completeness, and normal consistency for all lengths tested. In NRGBD, MeMix also provides overall gains across all backbones and input lengths, with especially clear improvements in accuracy and completeness.

Fig. 4 shows comparisons between methods with or without MeMix. Without MeMix, recurrent baselines accumulate errors over time, which typically appear as pose drift and degraded geometry. MeMix leads to more coherent surfaces and better preserved structures. More results are shown in Table 5 of the supplementary material.

Table 2: **3D Reconstruction Results on 7-Scenes [48] and NRGBD [49].** We test MeMix on 7-Scenes and NRGBD, with one frame sampled every two frames (Sparse Sampling, **-S**). Green boxes indicate improved or unchanged performance over the base model (w/o MeMix) under the same input length.

Model	MeMix	Input	7-Scenes-S						NRGBD-S					
			Acc. ↓		Comp. ↓		NC ↑		Acc. ↓		Comp. ↓		NC ↑	
			Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT *(Offline)* [2]	–	*300*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*400*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*500*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
StreamVGGT [25]	–	*300*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*400*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*500*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
CUT3R [3]	×	*300*	0.141	0.096	0.076	0.034	0.543	0.564	0.234	0.139	0.074	0.018	0.575	0.614
	✓		0.106	0.076	0.053	0.019	0.550	0.575	0.186	0.086	0.050	0.009	0.595	0.651
	×	*400*	0.178	0.121	0.115	0.069	0.532	0.546	0.342	0.227	0.127	0.067	0.561	0.591
	✓		0.147	0.100	0.076	0.039	0.540	0.559	0.321	0.180	0.099	0.031	0.565	0.594
	×	*500*	0.190	0.138	0.090	0.033	0.530	0.543	0.359	0.264	0.173	0.081	0.560	0.591
	✓		0.167	0.119	0.077	0.026	0.533	0.547	0.328	0.218	0.161	0.040	0.560	0.590
TTT3R [11]	×	*300*	0.040	0.025	0.024	0.005	0.567	0.602	0.101	0.044	0.025	0.005	0.610	0.678
	✓		0.034	0.020	0.023	0.005	0.567	0.603	0.099	0.037	0.020	**0.004**	0.616	0.692
	×	*400*	0.052	0.031	0.027	0.005	0.558	0.588	0.143	0.065	0.071	0.012	0.600	0.658
	✓		0.043	0.025	0.026	0.005	0.560	0.590	0.146	0.066	0.070	0.018	0.602	0.665
	×	*500*	0.066	0.039	0.031	0.006	0.551	0.577	0.166	0.092	0.087	0.021	0.593	0.647
	✓		0.059	0.032	0.030	0.005	0.553	0.580	0.183	0.094	0.094	0.031	0.595	0.650
TTSA3R [15]	×	*300*	0.036	0.020	0.035	0.006	0.566	0.600	0.090	0.036	0.020	**0.004**	0.620	0.696
	✓		**0.026**	**0.013**	**0.021**	**0.004**	**0.568**	**0.604**	**0.086**	**0.031**	**0.015**	**0.004**	**0.626**	**0.709**
	×	*400*	0.036	0.019	0.024	**0.004**	**0.561**	0.592	0.104	0.045	0.035	0.006	**0.618**	**0.692**
	✓		**0.030**	**0.015**	**0.023**	**0.004**	**0.561**	**0.593**	**0.100**	**0.042**	**0.031**	**0.005**	0.617	**0.692**
	×	*500*	0.042	0.021	0.024	**0.004**	0.556	0.585	0.121	0.054	0.050	**0.006**	0.613	0.684
	✓		**0.033**	**0.016**	**0.023**	**0.004**	**0.558**	**0.587**	**0.114**	**0.050**	**0.040**	0.007	**0.615**	**0.687**

4.2 Camera Pose Estimation

Following previous online pose evaluation protocols [3, 11, 15], we further evaluate long-sequence camera pose estimation on both TUM and ScanNet for three recurrent backbones, reporting the absolute trajectory error (ATE) as the number of input views increases; lower is better. The results

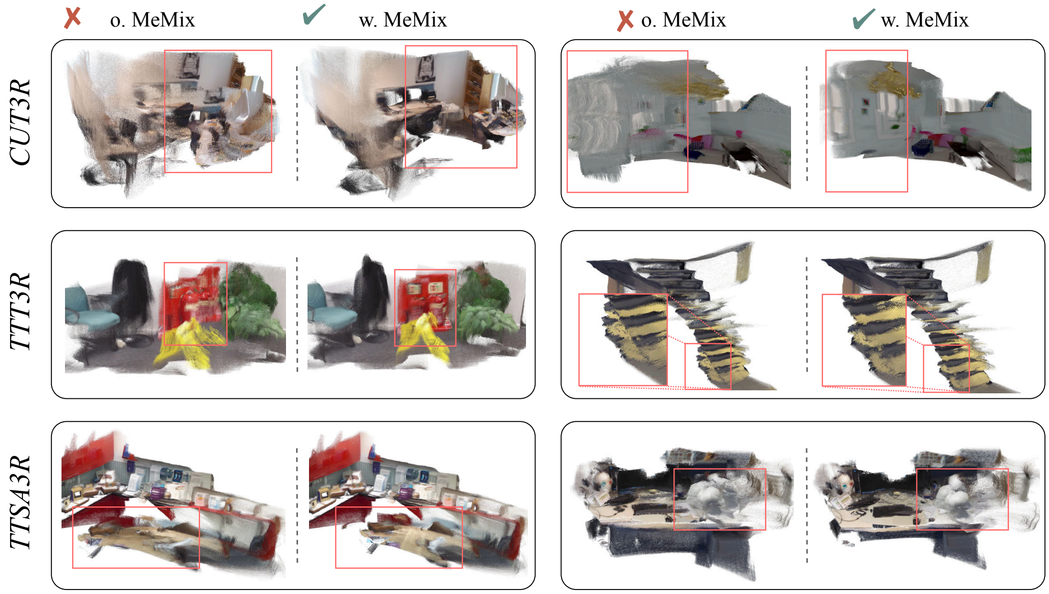


Figure 4: **Qualitative results of 3D reconstruction.** We compare CUT3R, TTT3R, and TTSA3R with their MeMix variants on long input streams. MeMix consistently improves reconstruction quality by reducing drift and recovering more complete, sharper surfaces. Red boxes highlight representative regions where MeMix corrects failures such as surface tearing, missing geometry, and ghosting.

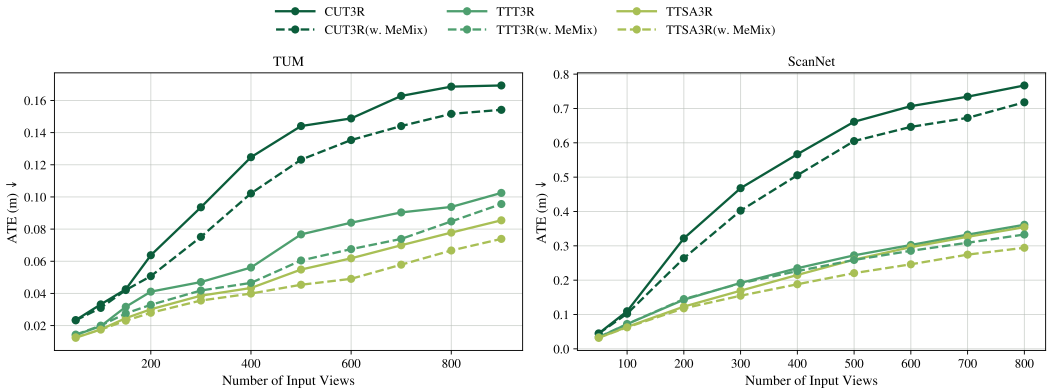


Figure 5: **Long-sequence pose estimation on TUM and ScanNet.** We compare CUT3R, TTT3R, and TTSA3R with their MeMix variants on long input streams, and report the absolute trajectory error (ATE) as the number of input views increases.

are summarized in Fig. 5. Across both datasets, MeMix consistently improves pose estimation over the corresponding baselines.

Pose drift in streaming reconstruction is tightly coupled with state degradation. Once the latent state is updated, errors accumulate during readouts. By preserving `Bottom-k` tokens, MeMix reduces accumulated drift, improving both global trajectory accuracy and frame-to-frame consistency. More results are shown in Table 6 of the supplementary material.

4.3 Depth Estimation

Following common practice in streaming video depth benchmarks [3, 11, 47], we evaluate video depth estimation on KITTI [50], Bonn [51], and Sintel [52]. For fair comparison, we keep the same checkpoints and inference settings as their corresponding baselines, and report both scale-invariant and metric-scale metrics in the main paper and supplementary material.

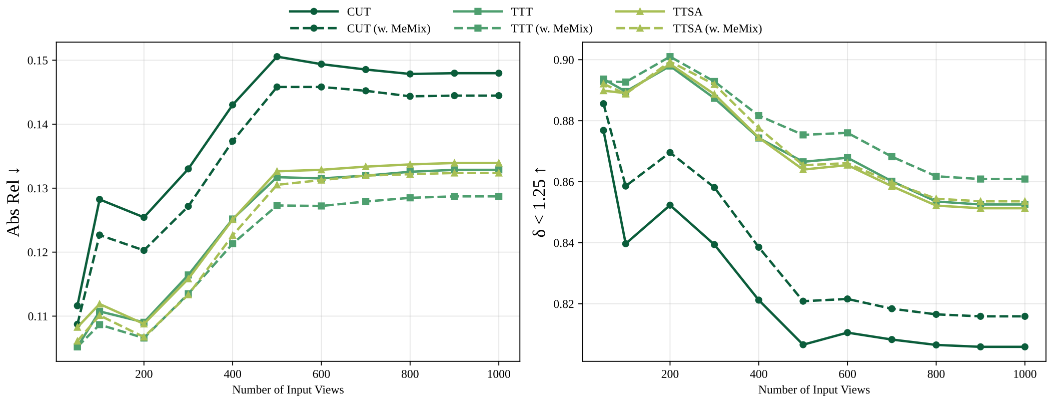


Figure 6: **Evaluation on Video Depth Estimation.** We compare CUT3R, TTT3R and TTSA3R with their w/o MeMix version under long input streams. MeMix generally improves depth estimation quality for input lengths ranging from 50 to 1000 frames. Notably, the outcomes largely depend on the capacity of the original model.

As shown in Fig. 6, MeMix generally improves depth estimation as the input horizon increases. The gains become more evident on longer streams, where recurrent state degradation accumulates over time. At the same time, MeMix also preserves, and in several cases slightly improves, short-horizon performance, suggesting that its benefit is not limited to long-range stability but also comes from more effective state updates. The magnitude of the improvement depends on the strength of the underlying backbone: stronger baselines already mitigate part of the drift, leaving less room for improvement, whereas more fragile baselines benefit more from selective memory writes. More results are shown in Table 7 of the supplementary material.

4.4 Ablation Studies

We conduct ablations to isolate how routing (where to write) and write-back (how to write) affect stability under a fixed-capacity state. Following the analysis protocol used in training-free state interventions [11, 15, 47], we keep the backbone, patch partition, and k fixed unless otherwise specified, and report depth, pose and reconstruction.

Default configuration and k selection. Unless stated otherwise, MeMix uses `Bottom-k` patch routing with the dot-product score $S_t^{\text{dot}} = \langle \hat{\mathbf{S}}_t, \mathbf{Y}_t \rangle$, and applies the single-update write-back after the decoder (optionally gated by $\boldsymbol{\beta}_t$ when enabled by the backbone). We set k = 708 as a single global default since it is consistently competitive across the reported tasks/metrics; Fig. 7 presents a representative sweep of k on KITTI[50] video depth estimation and TUM camera pose estimation. Table 3 reports this configuration as the **Default** row, and each subsequent block varies one factor while keeping the others fixed.

Patch selection strategy.

We compare three routing policies: `Top-k` (update k highest-score patches/tokens), `Bottom-k` (update k lowest-score patches/tokens; **default**) and `Random-k` (uniformly sample k patches/tokens). For patch-level routing, token scores are averaged within each patch prior to selection.

Scoring function.

We evaluate cosine similarity, dot product (**default**), and attention-derived scores as routing signals:

$$S_t^{\cos} = \left\langle \frac{\hat{\mathbf{S}}_t}{\|\hat{\mathbf{S}}_t\|_2}, \frac{\mathbf{Y}_t}{\|\mathbf{Y}_t\|_2} \right\rangle, \quad S_t^{\text{dot}} = \langle \hat{\mathbf{S}}_t, \mathbf{Y}_t \rangle \tag{17}$$

Table 3: **Ablations on routing policy and score design.** The Default row corresponds to `Bottom-k` + `Dot` + score($\hat{\mathbf{S}}_t, \bar{\mathbf{x}}_t$); other rows change one component at a time.

Variant	**Depth@KITTI**		**Camera@TUM**			**3R@NRGBD**		
	AbsRel ↓	$\delta < 1.25$ ↑	ATE ↓	RPE$_{\text{trans}}$ ↓	RPE$_{\text{rot}}$ ↓	Acc ↓	Comp ↓	NC ↑
Default (TTT3R with MeMix)	0.103	**92.1**	**0.028**	**0.013**	0.376	**0.099**	**0.020**	**0.616**
Patch selection								
Top-k	**0.102**	91.8	0.068	0.021	0.595	0.178	0.049	0.587
Random-k	0.108	91.4	**0.028**	**0.013**	0.382	0.102	0.023	**0.616**
Scoring function								
Cosine (s^{cos})	0.105	91.6	**0.028**	**0.013**	**0.375**	**0.099**	0.023	**0.616**
Attn (s^{attn})	0.107	90.7	0.030	0.014	0.418	0.105	0.025	0.611
Update strategy								
Full-update	0.108	91.4	0.035	0.015	0.443	0.127	0.034	0.604
No-update	0.114	89.0	0.162	0.066	1.624	0.461	0.672	0.528
Routing score								
score($\mathbf{S}_{t-1}, \mathbf{X}_t$)	0.107	91.1	0.039	0.016	0.437	0.111	0.027	0.608
score($\mathbf{S}_{t-1}, \mathbf{Z}_t$)	0.124	85.9	0.041	0.016	0.426	0.249	0.055	0.575
score($\hat{\mathbf{S}}_t, \mathbf{Z}_t$)	0.128	84.7	0.046	0.046	0.465	0.308	0.112	0.573

where $\hat{\mathbf{S}}_t$ is the candidate state token for the final decoder and $\mathbf{X}_t$ are the image tokens used for routing. For TTT3R-style attention routing [11], we use aggregated decoder cross-attention as TTT3R[11] did.

Write-back strategy.

We compare single-update (one update after the decoder):

$$\mathbf{S}_t = \mathbf{M}_t \odot \hat{\mathbf{S}}_t + (1 - \mathbf{M}_t) \odot \mathbf{S}_{t-1} \tag{18}$$

and full-update (per-block update inside each decoder layer):

$$\mathbf{S}_t^{(\ell+1)} = \mathbf{M}_t^{(\ell)} \odot \hat{\mathbf{S}}_t^{(\ell+1)} + (1 - \mathbf{M}_t^{(\ell)})\mathbf{S}_t^{(\ell)} \tag{19}$$

together with a no-update (freeze) control.

Feature source for routing.

Let $\mathbf{S}_{t-1}$ be the pre-update state, $\hat{\mathbf{S}}_t$ the decoder-final candidate state, $\mathbf{X}_t$ the raw image tokens, and $\mathbf{Y}_t$ the interaction tokens used for routing. We further compare score($\mathbf{S}_{t-1}, \mathbf{X}_t$), score($\hat{\mathbf{S}}_t, \mathbf{Y}_t$), score($\hat{\mathbf{S}}_t, \mathbf{X}_t$), and score($\mathbf{S}_{t-1}, \mathbf{Y}_t$) to determine which routing score strategy is the best.

Bottom-k Selection.

In the ablation study, we also evaluate our `Bottom-k` strategy. We integrate MeMix into CUT3R and TTT3R and compare the results with the original versions to determine the optimal value of k. By sweeping k in 12 token steps, ranging from 0 to 768, the experiments show that MeMix achieves the best performance when k is set to 708.

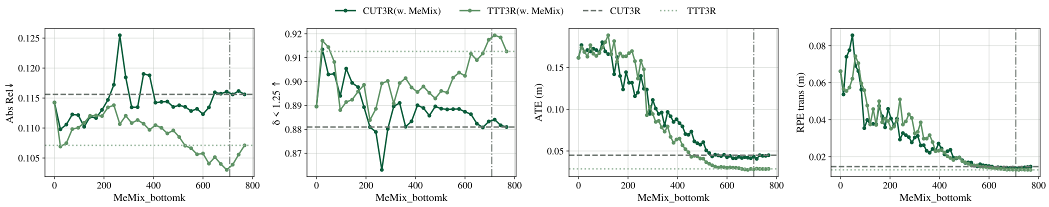


Figure 7: **Sensitivity to k.** We present a representative sweep of k on KITTI video depth estimation and TUM camera pose estimation for CUT3R and TTT3R with/without MeMix.

Inference Efficiency

To enhance the practical applicability of our approach, we conduct an ablation study comparing the inference speed and GPU memory consumption of three methods with and without MeMix. We test on KITTI using whole-frame input and calculate the average results for each scene. As shown in Table 4, introducing MeMix has negligible impact on both the inference FPS and peak GPU memory usage across all methods. These results demonstrate that MeMix does not introduce additional side effects.

Table 4: **Efficiency.** Inference FPS and peak GPU memory with/without MeMix.

Method	**FPS (f/s)**		**GPU (GB)**	
w. MeMix	×	✓	×	✓
`CUT3R`	14.39	14.13	5.31	5.31
`TTT3R`	12.72	12.81	6.96	6.96
`TTSA3R`	12.58	12.78	6.63	6.63

5 Conclusion

Summary. Fully rewriting the recurrent state at each step causes cumulative interference and catastrophic forgetting in long-horizon inference. We identify this fundamental bottleneck in fixed-state streaming 3D reconstruction and propose MeMix: a training-free, plug-in memory update module that recasts the recurrent state as a mixture of memory patches, substantially improving long-sequence reconstruction quality. MeMix seamlessly integrates into mainstream recurrent reconstruction models, consistently improving performance while preserving short-sequence accuracy, with negligible overhead in GPU memory and inference latency.

Limitations. Although MeMix surpasses previous main-stream methods in long-horizon inference, we have not tested what happens when the input contains thousands of frames. Inference on kilometer scale is vital for navigation and perception; some methods [8, 53, 54] have achieved this goal, but it still needs more exploration. Moreover, the `Bottom-k` selection is heuristic, and we have not analyzed the interpretability of this parameter for the state update. In the future, update strategies based on geometric properties or physical scenarios may further enhance the potential of this process.

Acknowledgement

We thank Xingyu Chen [11], Wendi Hu, Chengyi Gao for valuable insights and supports during the project. A small step for your devotion, a huge leap for your successors.

References

[1] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceeding of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[2] 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.

[3] 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.

[4] Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. In *Advances in Neural Information Processing Systems*, 2025. NeurIPS 2025.

[5] Yushi LAN, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. STream3r: Scalable sequential 3d reconstruction with causal transformer. In *The Fourteenth International Conference on Learning Representations*, 2026.

[6] Dharmendra Selvaratnam and Dena Bazazian. 3d reconstruction in robotics: A comprehensive review. *Computers & Graphics*, 130:104256, 2025.

[7] Liewen Liao, Weihao Yan, Wang Xu, Ming Yang, Songan Zhang, and Hongtei Eric Tseng. Learning-based 3d reconstruction in autonomous driving: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[8] Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. Infinitevggt: Visual geometry grounded transformer for endless streams. *arXiv preprint arXiv:2601.02281*, 2026.

[9] Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[10] Jintao Zhang, Haoxu Wang, Kai Jiang, Kaiwen Zheng, Youhe Jiang, Ion Stoica, Jianfei Chen, Jun Zhu, and Joseph E. Gonzalez. Sla2: Sparse-linear attention with learnable routing and qat. *arXiv preprint arXiv:2602.12675*, 2026.

[11] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3r: 3d reconstruction as test-time training. In *The Fourteenth International Conference on Learning Representations*, 2026.

[12] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[14] Jusen Du, Weigao Sun, Disen Lan, Jiaxi Hu, and Yu Cheng. Mom: Linear sequence modeling with mixture-of-memories. *arXiv preprint arXiv:2502.13685*, 2025.

[15] Zhijie Zheng, Xinhao Xiang, and Jiawei Zhang. Ttsa3r: Training-free temporal-spatial adaptive persistent state for streaming 3d reconstruction. *arXiv preprint arXiv:2601.22615*, 2026.

[16] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[17] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022.

[18] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, volume 15130 of *Lecture Notes in Computer Science*, pages 71–91, Cham, 2024. Springer. ISBN 978-3-031-73219-5. doi: 10.1007/978-3-031-73220-1_5.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[20] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

[21] Zipeng Wang and Dan Xu. Flashvggt: Efficient and scalable visual geometry transformers with compressed descriptor attention. *arXiv preprint arXiv:2512.01540*, 2025.

[22] Xianbing Sun, Zhikai Zhu, Zhengyu Lou, Bo Yang, Jinyang Tang, Liqing Zhang, He Wang, and Jianfu Zhang. Avggt: Rethinking global attention for accelerating vggt. *arXiv preprint arXiv:2512.02541*, 2025.

[23] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pages 78–89, 2025. doi: 10.1109/3DV66043.2025.00013.

[24] 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.

[25] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *The Fourteenth International Conference on Learning Representations*, 2026.

[26] Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3r: Training-free token eviction for memory-bounded streaming visual geometry transformers, 2025. URL `https://arxiv.org/abs/2509.17650`.

[27] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

[28] Wenhao Chai and Weili Xu. View transformer layers from online optimization perspective. 2025.

[29] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[30] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

[31] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[32] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations (ICLR)*, 2025.

[33] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

[34] Yingcong Li, Davoud Ataee Tarzanagh, Ankit Singh Rawat, Maryam Fazel, and Samet Oymak. Gating is weighting: Understanding gated linear attention through in-context learning. *arXiv preprint arXiv:2504.04308*, 2025.

[35] Yuan Cao and Dong Wang. Saga: Selective adaptive gating for efficient and expressive linear attention. *arXiv preprint arXiv:2509.12817*, 2025.

[36] Yu Zhang, Songlin Yang, Rui-Jie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, et al. Gated slot attention for efficient linear-time sequence modeling. *Advances in Neural Information Processing Systems*, 37:116870–116898, 2024.

[37] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[38] Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *arXiv preprint arXiv:2401.15969*, 2024.

[39] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[40] J Nagoor Kani and Ahmed H Elsheikh. Dr-rnn: A deep residual recurrent neural network for model reduction. *arXiv preprint arXiv:1709.00939*, 2017.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[42] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[43] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[44] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *The Thirteenth International Conference on Learning Representations*, 2025. ICLR 2025 Spotlight.

[45] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9158–9168, October 2025.

[46] Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8535–8546, October 2025.

[47] Guole Shen, Tianchen Deng, Xingrui Qin, Nailin Wang, Jianyu Wang, Yanbo Wang, Yongtao Chen, Hesheng Wang, and Jingchuan Wang. Mut3r: Motion-aware updating transformer for dynamic 3d reconstruction. 2025.

[48] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[49] Dejan Azinović, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6290–6301, 2022.

[50] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[51] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. 2019.

[52] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012.

[53] Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyuang Guo, and Hao Wang. Longstream: Long-sequence streaming autoregressive visual geometry. *arXiv preprint arXiv:2602.13172*, 2026.

[54] Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. Loger: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026.

[55] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012. doi: 10.1109/IROS.2012.6385773.

[56] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[57] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1611–1621, 2021. doi: 10.1109/CVPR46437.2021.00166.

[58] Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T. Freeman. Structure and motion from casual videos. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13693 of *Lecture Notes in Computer Science*, pages 20–37, Cham, 2022. Springer. ISBN 978-3-031-19827-4. doi: 10.1007/978-3-031-19827-4_2.

Supplementary Material

A1. Comprison between Top-k & Bottom-k

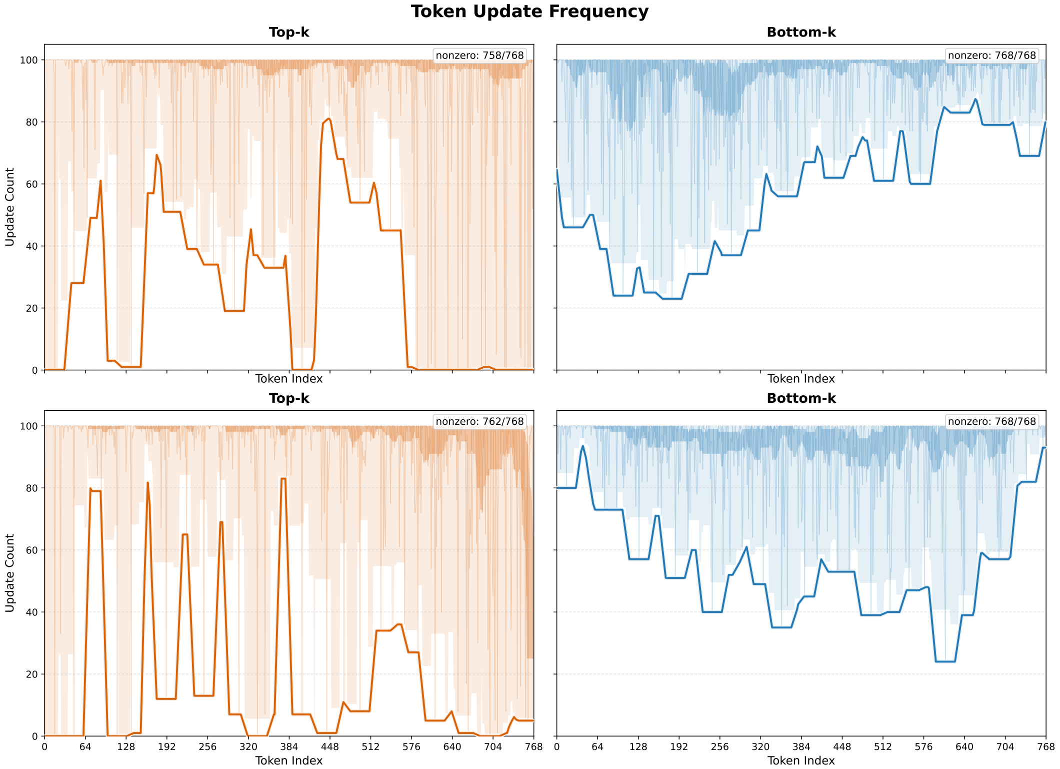


Figure 8: **Visualization of distinct strategies.** We employ `Top-k` and `Bottom-k` strategies separately, and tally the state tokens that are updated in each input frame. Results show some typical results that `Bottom-k` not only achieves a higher update frequency but also yields a more balanced updates across all state tokens.

`Top-k` updates the most-aligned tokens, creating a positive feedback loop that a small set of high-score tokens is repeatedly selected and reinforced, while the rest receive few updates and gradually become stale. This behavior reduces the effectiveness of memory diversity and utilization. In contrast, `Bottom-k` updates the least-aligned tokens, which naturally spreads writes across the state over time and improves overall memory coverage. For CUT3R(w. MeMix) on 7-Scenes, `Top-k` leaves a distinct subset of tokens nearly unupdated, whereas Bottom-k yields far more uniform token updates overall.

A2. 3D Reconstruction

We additionally report dense long-sequence 3D reconstruction results on 7-Scenes and NRGBD, using input streams of 300, 400, and 500 frames. We evaluate reconstruction quality with accuracy, completeness, and normal consistency, where lower accuracy/completeness and higher normal consistency indicate better performance. Green cells indicate metrics that are improved or preserved relative to the corresponding base model under the same input length.

As shown in Table 5, MeMix consistently improves or preserves dense long-sequence 3D reconstruction performance across different recurrent backbones and datasets, with especially clear gains on CUT3R. These results suggest that MeMix serves as a general memory-update improvement rather than a backbone-specific design.

Table 5: **3D Reconstruction Results on 7-Scenes [48] and NRGBD [49].** We test MeMix on 7-Scenes and NRGBD, with every frame sampled (Dense Sampling, **-D**). Green boxes indicate improved or unchanged performance over the base model (w/o MeMix) under the same input length.

Model	MeMix	Input	7-Scenes-D						NRGBD-D					
			Acc. ↓		Comp. ↓		NC ↑		Acc. ↓		Comp. ↓		NC ↑	
			Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT *(Offline)* [2]	–	*300*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*400*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*500*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
StreamVGGT [25]	–	*300*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*400*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	–	*500*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
CUT3R [3]	×	*300*	0.099	0.062	0.048	0.014	0.542	0.562	0.137	0.092	0.066	0.024	0.572	0.609
	✓		0.076	0.045	0.039	0.010	0.549	0.573	0.113	0.081	0.060	0.035	0.578	0.618
	×	*400*	0.150	0.093	0.090	0.037	0.531	0.543	0.225	0.155	0.119	0.076	0.554	0.579
	✓		0.117	0.071	0.056	0.015	0.536	0.552	0.196	0.128	0.098	0.062	0.572	0.609
	×	*500*	0.165	0.114	0.094	0.039	0.522	0.531	0.313	0.203	0.202	0.148	0.554	0.580
	✓		0.146	0.094	0.067	0.022	0.528	0.541	0.273	0.173	0.162	0.110	0.568	0.602
TTT3R [11]	×	*300*	0.030	0.016	0.019	**0.004**	0.558	0.588	0.057	0.035	0.016	**0.003**	0.595	0.650
	✓		0.030	0.016	0.019	**0.004**	**0.559**	0.589	0.052	0.032	0.015	**0.003**	0.599	0.656
	×	*400*	0.044	0.026	0.024	**0.004**	0.551	0.577	0.093	0.053	0.018	**0.003**	0.587	0.635
	✓		0.039	0.023	0.025	**0.004**	0.552	0.578	0.078	0.042	0.016	**0.003**	0.592	0.644
	×	*500*	0.068	0.046	0.033	0.009	0.542	0.562	0.127	0.061	0.033	**0.003**	0.586	0.635
	✓		0.057	0.039	0.030	0.008	0.546	0.568	0.105	0.048	0.026	0.004	0.586	0.633
TTSA3R [15]	×	*300*	0.023	0.011	0.018	**0.004**	0.558	0.588	0.039	**0.022**	0.011	**0.003**	**0.606**	**0.669**
	✓		**0.022**	**0.009**	**0.017**	**0.004**	**0.559**	**0.588**	**0.037**	**0.022**	**0.010**	**0.003**	0.605	0.668
	×	*400*	0.030	0.016	0.022	**0.004**	0.553	0.580	0.060	**0.027**	**0.010**	**0.003**	**0.598**	**0.655**
	✓		**0.025**	**0.012**	**0.021**	**0.004**	**0.554**	**0.581**	**0.059**	**0.027**	**0.010**	**0.003**	0.596	0.651
	×	*500*	0.045	0.029	0.025	**0.004**	0.545	0.567	0.085	0.034	0.020	**0.003**	**0.596**	**0.651**
	✓		**0.035**	**0.021**	**0.023**	**0.004**	**0.548**	**0.571**	**0.081**	**0.032**	**0.014**	**0.003**	0.595	0.649

A3. Pose Estimation

We benchmark camera pose estimation on Sintel [52], TUM-dynamics [55], and ScanNet [56]. We set 50 frames on Sintel and 90 frames on TUM&ScanNet as our setting, reporting standard trajectory metrics in Table 6.

Table 6: **Evaluation on Short-Sequence Pose Estimation**. To illustrate that MeMix will not undermine performance even in short-sequence, we evaluate on three datasets less than 100 frames input. Green boxes indicate improved or unchanged performance over the base model(w/o MeMix).

		TUM-dynamics (90 frames)			ScanNet (90 frames)			Sintel (50 frames)		
Method	**Online**	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓	ATE ↓	RPE trans ↓	RPE rot ↓
Robust-CVD [57]	×	0.153	0.026	3.528	0.227	0.064	7.374	0.360	0.154	3.443
CasualSAM [58]	×	0.071	0.010	1.712	0.158	0.034	1.618	0.141	0.035	0.615
DUSt3R [1]	×	0.083	0.017	3.567	0.081	0.028	0.784	0.417	0.250	5.796
MASt3R [18]	×	0.038	0.012	0.448	0.078	0.020	0.475	0.185	0.060	1.496
MonST3R [44]	×	0.098	0.019	0.935	0.077	0.018	0.529	0.111	0.044	0.869
Easi3R [45]	×	0.105	0.022	1.064	0.061	0.017	0.525	0.110	0.042	0.758
AETHER[46]	×	0.092	0.012	1.106	0.176	0.028	1.204	0.189	0.054	0.694
VGGT [2]	×	**0.012**	**0.010**	**0.310**	**0.035**	**0.015**	**0.377**	**0.172**	**0.062**	**0.471**
Spann3R [23]	✓	0.056	0.021	0.591	0.096	0.023	0.661	0.329	0.110	4.471
Point3R [4]	✓	0.075	0.029	0.642	0.106	0.035	1.946	0.351	0.128	1.822
StreamVGGT [25]	✓	0.061	0.033	3.209	0.161	0.057	3.647	0.251	0.149	1.894
CUT3R [3]	✓	0.045	0.015	0.443	0.096	0.022	0.600	0.210	**0.069**	0.628
CUT3R(w. MeMix)	✓	0.043	0.014	0.424	0.090	0.022	0.604	**0.190**	0.075	**0.627**
TTT3R[11]	✓	0.029	**0.013**	0.380	0.065	**0.021**	0.640	0.208	0.093	0.725
TTT3R(w. MeMix)	✓	0.028	**0.013**	0.376	0.065	**0.021**	0.677	0.210	0.083	0.733
TTSA3R[15]	✓	0.026	**0.013**	**0.372**	0.058	**0.021**	**0.561**	0.210	0.084	0.738
TTSA3R(w. MeMix)	✓	**0.025**	**0.013**	**0.372**	**0.057**	**0.021**	0.569	0.209	0.084	0.763

A4. Video Depth Estimation

Following common practice in streaming video depth benchmarks [3, 11, 47], we evaluate on KITTI [50], Bonn [51], and Sintel [52] using 110-frame input sequences. We keep the same settings of each baseline, and report both scale-invariant and metric-scale metrics. Notably, MeMix can yield better performance even on short horizons, indicating that sparse routing improvements are not limited to long-range stability.

Table 7: **Video Depth Estimation** We evaluate scale-invariant and metric depth accuracy on KITTI [50], Sintel [52], and Bonn [51] datasets. Methods that require global alignment are denoted as "GA" and green boxes indicate metrics that improve compared to corresponding base model (w/o MeMix).

Alignment	Method	Online	KITTI		Sintel		Bonn	
			Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
Per-sequence scale	DUSt3R-GA [1]	×	0.144	81.3	0.656	45.2	0.155	83.3
	MASt3R-GA [18]	×	0.183	74.5	0.641	43.9	0.252	70.1
	MonST3R-GA [44]	×	0.168	74.4	0.378	55.8	0.067	96.3
	Easi3R [45]	×	0.102	91.2	0.377	55.9	0.059	97.0
	VGGT [2]	×	**0.070**	**96.5**	**0.287**	**66.1**	**0.055**	**97.1**
	Spann3R [23]	✓	0.198	73.7	0.622	42.6	0.144	81.3
	Point3R [4]	✓	0.136	84.2	0.452	48.9	0.060	96.0
	STREAM3R$^\alpha$ [5]	✓	0.116	89.6	0.478	51.1	0.075	94.1
	StreamVGGT [25]	✓	0.173	72.1	**0.323**	**65.7**	**0.059**	**97.2**
	CUT3R [3]	✓	0.116	88.1	0.426	47.3	0.079	93.7
	CUT3R(w. MeMix)	✓	0.115	88.6	0.436	46.2	0.078	93.8
	TTT3R [11]	✓	0.107	91.2	0.409	48.9	0.069	95.5
	TTT3R(w. MeMix)	✓	**0.103**	92.1	0.407	49.2	0.070	95.1
	TTSA3R [15]	✓	**0.103**	91.9	0.410	49.6	0.064	96.4
	TTSA3R(w. MeMix)	✓	**0.103**	**92.2**	0.400	50.2	0.065	96.0
Metric scale	MASt3R-GA [18]	×	0.467	15.2	1.022	14.3	0.272	70.6
	Point3R [4]	✓	0.191	73.8	**0.777**	17.1	0.137	94.7
	STREAM3R$^\alpha$ [5]	✓	0.234	57.6	1.041	21.0	0.084	94.4
	CUT3R [3]	✓	0.129	82.8	1.020	23.7	0.103	88.9
	CUT3R(w. MeMix)	✓	0.122	85.0	1.068	24.1	0.104	88.8
	TTT3R [11]	✓	0.107	89.2	0.978	23.3	0.090	94.4
	TTT3R(w. MeMix)	✓	**0.103**	**89.9**	0.984	23.6	0.094	92.9
	TTSA3R [15]	✓	0.110	88.6	0.959	24.5	**0.080**	**96.4**
	TTSA3R(w. MeMix)	✓	0.107	89.1	0.962	**24.9**	0.083	96.1

A5. Visualization

Fig. 9 shows qualitative trajectory visualizations on long sequences. Across different backbones, the MeMix variants generally stay closer to the ground-truth trajectories and exhibit reduced drift. More specifically, the improvement is most visible in challenging segments with longer temporal horizons and larger camera motion, where the baseline trajectories tend to gradually deviate from the ground truth or accumulate local drift. In contrast, the MeMix variants better preserve the overall trajectory shape and remain more consistent with the reference path over time. These qualitative observations are consistent with the quantitative ATE improvements reported in Sec. A3, further supporting that selective memory updates help stabilize long-sequence pose estimation.

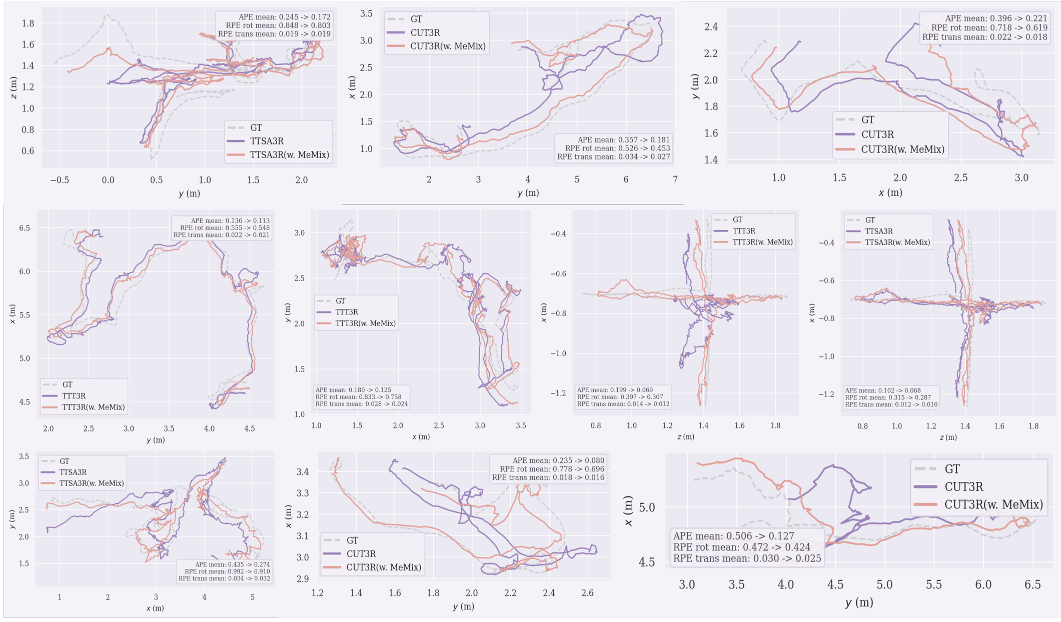


Figure 9: **Visualization of Estimated Camera Trajectories – Long Sequence.** The trajectories are projected onto the two axes with the highest variance to capture the dominant motion. We compare ● GT, ● baseline trajectories, and ● their corresponding MeMix variants across three backbones: CUT3R, TTT3R, and TTSA3R. Across different backbones, the MeMix variants generally stay closer to the ground-truth trajectories and exhibit reduced drift over long sequences.