LONGVU-TTT: CAUSAL TEST-TIME TRAINING FOR VISUAL RESAMPLING IN LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-video MLLMs must model temporal change before a limited visual-token budget removes most frame evidence. We introduce **LongVU-TTT**, which inserts a convolutional Test-Time Training (TTT) resampler with causal fast-weight updates between the vision encoder and the LLM. Its grouped 2D fast weights adapt to each video and contextualize frame features before compression, while a hybrid uniform-and-change-aware selector retains explicit visual evidence for downstream reasoning. Under controlled conditions, TTT-Conv improves over TTT-MLP by up to +2.12 and bidirectional Mamba2 by up to +3.04 on MLVU, and it is stronger than attention- and fixed-state recurrent resamplers across three benchmarks. Analysis shows that the fast weights behave as a temporal aggregation state rather than a reliable long-horizon episodic memory: their benefit attenuates as evidence becomes more distant, motivating explicit frame retention. LongVU-TTT processes up to 512 frames before reducing them to 128 LLM frames and achieves competitive performance across five video understanding benchmarks. Code and models will be publicly released.

1 INTRODUCTION

Recent advancements in video MLLMs (Lin et al., 2023; Zhang et al., 2023) have improved AI reasoning over temporal content. These models encode individual frames independently via a pretrained vision encoder before feeding tokens into an LLM (Zhu et al., 2023; Li et al., 2024a), forcing the LLM to perform temporal modeling within a constrained context window. As video lengths increase, token counts grow linearly while quadratic attention makes long sequences prohibitively expensive.

Existing efforts reduce spatiotemporal redundancy through spatial pooling, token merging, or differential token dropping (Bolya et al., 2022; Li et al., 2025; Yao et al., 2025; Choudhury et al., 2024), but generally compress frame-level features before explicitly modeling what changed. Recurrent architectures such as State-Space Models (Jiang et al., 2025; Islam et al., 2025) offer temporal propagation before compression, but their flattened sequence operators do not explicitly impose the 2D local-neighborhood interactions that complement the global representations of the ViT encoder.

We introduce **LongVU-TTT**, which inserts a convolutional Test-Time Training (TTT) resampler between the vision encoder and the LLM. It processes frames recurrently through continuously updated grouped-convolutional fast weights, preserving the visual feature grid while adapting to temporal change. The inner-loop gradient signals provide a change-aware importance measure that, combined with uniform sampling, reduces sequences of up to 512 frames to 128 LLM frames without an auxiliary learned scorer. Crucially, we find that the fast state is best understood as a temporal aggregator rather than an episodic store: selected frames retain the explicit evidence required for long-range reasoning. Here, *causal* refers specifically to the fast-weight update rule: each frame is contextualized only by preceding updates. Importance normalization and the final fixed-budget frame selection are performed after the full video has been processed.

Through a controlled comparison with a temporal Transformer, Gated DeltaNet, Mamba2, and TTT-MLP, we study which projector-stage temporal operator is most effective. Our results identify causal update ordering and explicit 2D local processing as important design choices, while retention diag-

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

nostics clarify the boundary between temporal aggregation and long-horizon memorization. Our goal is not to claim a new state of the art across models trained with heterogeneous data and backbones, but to provide a controlled study of TTT at the projector stage of a video MLLM. We therefore emphasize comparisons with models of similar scale that start from the same trained baseline. System optimizations enabling 512-frame training on A100 40GB GPUs are detailed in Section A.3 of the supplementary material.

Our contributions are as follows:

- A convolutional TTT resampler with causal fast-weight updates that adapts grouped 2D fast weights to contextualize video features before the LLM's visual-token bottleneck.
- A saliency-driven hybrid frame selector that combines gradient-direction shift and alignment-loss delta to retain scene transitions and within-scene visual change without an auxiliary learned scorer.
- A controlled study spanning temporal Transformers, gated linear recurrence, Mamba2, and TTT variants, together with retention diagnostics showing that projector fast weights provide temporal aggregation rather than reliable episodic storage.

2 Related Work

2.1 Video Multimodal Large Language Models

The prevailing paradigm in video MLLMs involves treating videos as sequences of independent frame embeddings extracted via pretrained vision encoders such as CLIP (Sun et al., 2023) or SigLIP 2 (Tschannen et al., 2025), and projecting them into the input space of an LLM (Li et al., 2024a; Zhang et al., 2023). While effective for short-range spatial reasoning, these models delegate temporal modeling entirely to the LLM's self-attention mechanism, which becomes computationally prohibitive as video length increases.

To manage the resulting token count, various compression strategies have been proposed. Early methods employ uniform temporal or spatial pooling to reduce the visual footprint (Wang et al., 2024a), while more recent adaptive approaches such as token merging and differential token dropping (Bolya et al., 2022; Li et al., 2025; Yao et al., 2025; Choudhury et al., 2024) perform more selective reduction. LongVU (Shen et al., 2024) takes a more structured approach by leveraging DINOv2 (Oquab et al., 2024) features to identify and remove redundant frames, and further applies text-guided cross-modal queries to selectively reduce frame features based on the input prompt. However, all of these methods operate on raw frame-level features without first propagating temporal context, and therefore compress without a principled understanding of what has changed across frames. In contrast, LongVU-TTT derives its compression signal directly from the TTT inner-loop updates, capturing visual change as a natural byproduct of temporal propagation rather than relying on external similarity metrics or auxiliary models.

2.2 Recurrent Models and Test-Time Training

Linear-complexity recurrent models offer an alternative to the quadratic cost of Transformer self-attention for long sequence modeling (Vaswani et al., 2017; Dao & Gu, 2024). Methods such as VideoMamba (Li et al., 2024b) and STORM (Jiang et al., 2025) apply selective scan operations over video tokens to capture temporal dependencies prior to compression. While effective, these methods apply flattened sequence operators that do not explicitly encode the 2D local-neighborhood interactions useful for fine-grained visual understanding. Gated DeltaNet (Yang et al., 2024) augments efficient linear recurrence with a gated delta-rule update; we include it as a fixed-state recurrent comparison at the same projector location as our method.

Test-Time Training (TTT) (Sun et al., 2024) replaces fixed recurrent hidden states with fast-weight (Schlag et al., 2021) models that are updated as a function of the input sequence, acting as a continuously evolving adaptation state. Recent TTT variants such as Titans (Behrouz et al., 2024) and Atlas (Behrouz et al., 2025) have improved the memory capacity of these fast-weight models, but remain focused on 1D sequence modeling with MLP-based fast weights. LaCT (Zhang et al., 2025b) improves the hardware efficiency of TTT-MLP by maximizing chunk size and head

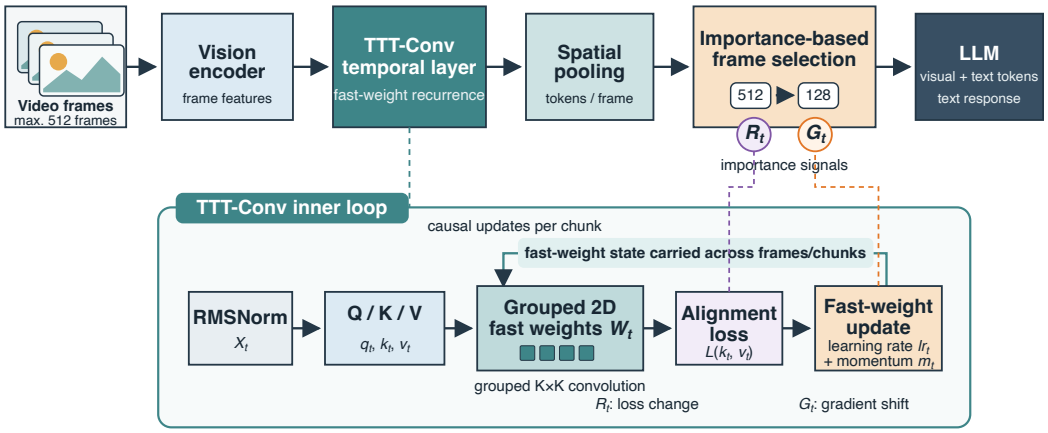

Figure 1: Overview of LongVU-TTT. Video frames are encoded independently by a pretrained vision encoder to produce raw visual tokens. The TTT-Conv layer then processes all frames recurrently via continuously updated grouped convolutional fast weights, propagating temporal context across the sequence in chunks. The temporally enriched representations are passed through spatial pooling and importance-based frame sampling to produce a compact set of visual tokens. The existing MLP projector maps the selected features into the LLM embedding space, where they are concatenated with text embeddings and used to generate the response.

dimension to better utilize GPU tensor cores, and serves as the basis for our TTT-MLP baseline. We depart from this convention by parameterizing the fast weights as grouped convolutions, preserving the 2D spatial structure of frame features during temporal propagation and capturing local spatial patterns that are complementary to the global representations of the ViT encoder. Rather than changing the LLM architecture itself, we place the temporal operator at the projector stage of a pretrained MLLM and fine-tune the complete model end to end. This setting lets us identify the architectural choices—including causal fast-weight updates and preservation of spatial structure—that make projector-stage TTT effective.

2.3 Efficient Training for Long-Video Models

Training on long video sequences creates activation bottlenecks that parameter-sharding methods such as DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) do not remove. Sequence parallelism (Jacobs et al., 2023), ring attention (Liu et al., 2023), and systems such as LongVILA's MM-SP (Chen et al., 2024) distribute long attention sequences, but require specialized communication across devices. A complementary direction is operator tiling: Arctic Long Sequence Training (ALST) (Bekman et al., 2025) partitions LLM MLP computation into tiles and defers ZeRO-3 gradient synchronization until the final tile. We adapt this idea to the independently processed ViT frame batch and pair it with asynchronous activation-memory management; the method and measurements are given in Section 3.5.

3 Methodology

3.1 Architecture Overview

LongVU-TTT introduces a dedicated temporal processing stage between the Vision Transformer (ViT) encoder and the Large Language Model (LLM), as illustrated in Figure 1. Rather than delegating all temporal reasoning to the LLM, this stage processes frames recurrently before they reach the LLM, enriching each frame's representation with context from all preceding frames and enabling principled compression of long video sequences. The layer receives frame-level features $X \in \mathbb{R}^{T\times(H\cdot W)\times C}$ from the ViT, where each frame maintains a spatial resolution of $H \times W = 27 \times 27$ tokens. To stabilize optimization, the TTT-Conv stage is wrapped in a residual block preceded by an RMSNorm layer:

$$X_{out} = X + \text{TTT-Conv}(\text{RMSNorm}(X)) \tag{1}$$

After spatial pooling and frame selection, the existing MLP projector maps the selected features from the vision-encoder width to the LLM embedding width. Thus, TTT-Conv is a projector-stage temporal operator, while the MLP projector remains the modality-alignment mapping used by the base architecture. The following sections describe the parameterization of the TTT-Conv layer, its inner-loop update rule, the frame selection strategy it enables, and the system optimizations required for scalable long-video training.

3.2 TTT-Conv Parameterization and Projections

The key design choice that distinguishes our TTT layer from prior work is its convolutional fast-weight parameterization, which directly leads to consistent improvements of up to $+2.12$ on MLVU over TTT-MLP and up to $+3.04$ over bidirectional Mamba2 under controlled conditions. Rather than flattening frame features into 1D sequences as TTT-MLP and SSM methods do, we parameterize the hidden state as two grouped convolutional kernels that operate directly on the 2D spatial structure of each frame. We use N_g groups to partition the channel dimension, allowing each group to specialize in a subset of feature channels while maintaining parameter efficiency. Let $d = C/N_g$ be the input channels per group and $D_{int} = 4d$ the per-group intermediate width. The two fast-weight tensors are $W_1 \in \mathbb{R}^{(N_g D_{int}) \times d \times K \times K}$ and $W_2 \in \mathbb{R}^{C \times D_{int} \times K \times K}$, which project each group into and out of its intermediate representation. Frame-wise linear heads produce keys and values (k_t, v_t) for the inner update, queries q_t for retrieval, two per-group learning rates $lr_t \in (0,1)^{N_g \times 2}$, and per-group momentum coefficients $m_t \in (0,1)^{N_g}$. Both gates use a sigmoid; zero-initialized learning-rate logits therefore begin at $\mathrm{sigmoid}(0) = 0.5$.

3.3 The TTT-Conv Inner Loop

The TTT hidden state is a function rather than a fixed vector. Specifically, the fast weights $(W_{1,t}, W_{2,t})$ define a convolutional mapping $f_{W_t} : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^{H \times W \times C}$ that is updated as frames arrive. It contextualizes the current frame from the observed video history, but is not assumed to store individual past frames. Frames are processed in chunks of fixed size C_{size}, with the weight state carried over across chunk boundaries to maintain temporal continuity across the full video. The projections and initial fast weights are learned by the outer training objective. At inference, the fast weights and momentum reset to their learned initial state at each video boundary and adapt using only the self-supervised alignment objective, without answer labels.

At each frame t, the fast weights are updated by minimizing a self-supervised alignment objective. Given the key k_t and value v_t projections of the current frame, the inner loop optimizes:

$$\mathcal{L}(k_t, v_t) = -\langle f_W(k_t),\, v_t \rangle \tag{2}$$

which encourages the mapping to produce representations aligned with the current frame's value. Empirically, frames that introduce visual change produce larger or directionally different updates, a property later used for importance-based frame selection.

3.3.1 Parallel Causal Materialization

A purely sequential inner-loop update preserves causal ordering but limits GPU utilization. Conversely, treating each chunk as a single minibatch removes distinct intra-chunk states. We retain causal ordering while parallelizing the computation.

We resolve this with a Parallel Causal Materialization strategy. For each frame t within the chunk, we first compute its raw gradient independently:

$$g_t = -\nabla_W \mathcal{L}(k_t, v_t) \tag{3}$$

Per-group learning rates first produce the scaled gradient $u_t = lr_t \cdot g_t$. Given the carried momentum state p_{init} from the previous chunk, the pre-update history h_t and momentum-adjusted fast-weight increment δ_t are

$$h_t = p_{init} + \sum_{i=1}^{t-1} u_i, \qquad \delta_t = u_t + m_t \cdot h_t. \tag{4}$$

The causal weight state excludes the current frame's own update:

$$W_t = W_{init} + \sum_{i=1}^{t-1} \delta_i. \tag{5}$$

Here g_t, u_t, and δ_t denote the raw gradient, learning-rate-scaled gradient, and momentum-adjusted increment, respectively. The coefficient $m_t \in (0, 1)$ controls the contribution of accumulated update history. At a chunk boundary, the final momentum-adjusted increment is carried as p_{init} for the next chunk. Using `torch.vmap` and prefix sums, all per-frame weight states within a chunk are materialized simultaneously. This trades activation memory for parallel compute; we use asynchronous activation checkpointing to rematerialize these states during the backward pass instead of retaining the full materialized tensor. Implementation details are in Section A.3.

3.3.2 Parallel Apply Operation

Following materialization, the query q_t is transformed through its corresponding weight state $(W_{1,t}, W_{2,t})$ in parallel across all frames in the chunk:

$$z_t = \sigma(\text{Conv2d}(q_t, W_{1,t})), \qquad \hat{x}_t = \text{Conv2d}(z_t, W_{2,t}) \tag{6}$$

where σ denotes the SiLU activation. The output $\hat{x}_t$ is a temporally contextualized representation of frame t influenced by preceding fast-weight updates before being passed to the LLM.

3.4 Saliency-Driven Hybrid Frame Sampling

A key property of our TTT layer is that the gradient signals produced during the inner-loop optimization provide a natural and principled measure of frame importance, requiring no additional learned components. This allows us to compress sequences of up to 512 frames to 128 frames while retaining the most informative content, as validated by the $+6.4$ MLVU gain over a baseline that subsamples without temporal propagation.

3.4.1 Importance Scoring

We extract two complementary scalar signals from the TTT inner loop to identify salient frames, where $t \in \{1, \ldots, T\}$ denotes the frame index.

Gradient-Direction Shift (G_t) captures scene-level changes by measuring the cosine distance between the fast-weight gradients of successive frames, yielding a scalar in $[0, 2]$ that is large when the direction of the fast-weight update shifts abruptly between frames $t-1$ and t:

$$G_t = 1 - \frac{\nabla_W \mathcal{L}_t \cdot \nabla_W \mathcal{L}_{t-1}}{\|\nabla_W \mathcal{L}_t\| \|\nabla_W \mathcal{L}_{t-1}\|} \tag{7}$$

Alignment-Loss Delta (R_t) captures event-level changes within a scene by measuring positive increases in the per-frame inner-loop alignment loss between consecutive frames, yielding a non-negative scalar that reflects how much less well aligned frame t is than frame $t-1$:

$$R_t = \max(0,\, \mathcal{L}_t - \mathcal{L}_{t-1}) \tag{8}$$

We set $G_1 = R_1 = 0$. The final importance score I_t for frame t is a weighted combination of the two signals, each normalized to $[0, 1]$ via min-max normalization across all T frames. A small constant ϵ prevents division by zero when a signal is constant:

$$I_t = \alpha \cdot \frac{G_t - \min_t G_t}{\max_t G_t - \min_t G_t + \epsilon} + (1 - \alpha) \cdot \frac{R_t - \min_t R_t}{\max_t R_t - \min_t R_t + \epsilon} \tag{9}$$

where $\alpha = 0.4$ is set empirically to balance the contribution of the two signals.

3.4.2 Hybrid Downsampling

At inference, the video is downsampled from T frames to exactly N_{sel} frames. We first form a Highlight Set $\mathcal{S}$ containing the top-k frames ranked by I_t. We then select a Uniform Set $\mathcal{U}$ of

$N_{sel} - k$ frames from the remaining indices $\{1, \ldots, T\} \setminus \mathcal{S}$. Because the two sets are disjoint, their sorted union always meets the target budget:

$$\mathcal{F}_{final} = \text{Sort}(\mathcal{S} \cup \mathcal{U}), \qquad |\mathcal{F}_{final}| = N_{sel}. \tag{10}$$

Importance-based selection therefore retains salient changes first, and uniform sampling fills the remaining slots to preserve temporal coverage. The causal designation applies to the fast-weight updates; this fixed-budget selection is performed after scores for the full video are available.

3.5 System Optimizations for Long-Video Training

Training on long video sequences introduces two memory bottlenecks that standard distributed training frameworks do not address: the large activation footprint of processing many frames through the ViT backbone, and the combined activation memory of the TTT recurrent loop and LLM layers for long packed sequences. The optimizations described below directly enable the experimental setting of this paper, extending the maximum trainable sequence length by $4\times$ and reducing training step time by up to $2.23\times$ at 64 GPUs, making 256-frame training on 40GB GPUs practical without sequence parallelism or specialized distributed attention infrastructure.

3.5.1 Batch-Tiled ViT Processing

Building on ALST's MLP tiling (Bekman et al., 2025), we partition the ViT frame batch into independently processed tiles and defer ZeRO-3 gradient reduce-scatter until the final tile. Frame independence makes this adaptation exact, while replacing repeated inter-node collectives with one reduce-scatter per layer. It yields speedups from $1.39\times$ at 8 GPUs to $2.23\times$ at 64 GPUs, where InfiniBand communication is the dominant bottleneck.

3.5.2 Asynchronous CPU Activation Offloading

For long packed sequences, the combined activation memory of the TTT recurrent loop and the LLM layers exceeds 40GB even with selective gradient checkpointing. To quantify this, for a 28-layer LLM with hidden dimension $D = 3{,}584$ and sequence length $L = 25{,}088$ tokens in 16-bit precision, the activation footprint per layer is approximately:

$$\text{Memory}_{\text{layer}} \approx 6 \cdot L \cdot D \cdot 2 \approx 1.08\,\text{GB} \tag{11}$$

where the factor of 6 accounts for the normalization inputs, attention projection inputs, and MLP intermediate activations retained under selective checkpointing, yielding approximately 30.2 GB for the full LLM alone, before accounting for ViT activations or TTT intermediate states. We address this with an asynchronous CPU offloading pipeline using dedicated CUDA streams. Activations are transferred to CPU memory during the forward pass and prefetched back to the GPU asynchronously during the backward pass, overlapping communication with computation. This extends the maximum trainable sequence length by $4\times$ on 40GB hardware. Further implementation details of both optimizations are provided in Section A.3. As neither optimization modifies the attention computation or sequence sharding, they are in principle combinable with sequence parallelism or distributed attention techniques for even longer sequences or larger model scales, given sufficient host memory.

4 Experiments

4.1 Experimental Setup

We evaluate LongVU-TTT on five video understanding benchmarks: MLVU (Zhou et al., 2024), LongVideoBench (Wu et al., 2024), Video-MME (Fu et al., 2024), NExT-QA (Xiao et al., 2021), and LVBench (Wang et al., 2025). Architecture ablations are conducted on MLVU, LongVideoBench, and Video-MME using a fixed 30% subset of the Stage 2 training mixture shared by all variants to isolate architectural contributions from data and scale effects.

4.1.1 Training Setup

We follow the training protocol of LLaVA-Video (Zhang et al., 2024b). Training proceeds in two stages. In the first stage, we train the TTT layer and MLP projector on LLaVA-CC3M-

Table 1: Performance comparison of LongVU-TTT with existing video MLLMs on long and medium video understanding benchmarks. Frame policy reports either the fixed frame count or the temporal sampling policy used by each model. LongVU-TTT processes up to 512 frames through TTT-Conv before selecting 128 frames for the LLM. Results from prior work are shown at their reported precision; our results use two decimals. **Bold** denotes the best result and underline denotes the second best, including ties.

Model	Frame policy	MLVU	LongVideoBench	Video-MME	NExT-QA	LVBench
LongVA (Zhang et al., 2024a)	128	56.3	-	52.6	-	-
InternVL2 (InternVL Team, 2024)	16	48.1	51.8	54.0	-	-
LongVILA (Chen et al., 2024)	256	-	57.1	60.1	80.7	-
VideoChat2 (Li et al., 2024c)	-	54.6	36.0	-	-	-
LLaVA-OV (Li et al., 2024a)	32	64.7	56.3	58.2	79.4	-
Qwen2-VL (Wang et al., 2024a)	256	64.2	55.6	63.3	-	-
Kangaroo (Liu et al., 2024)	64	61.0	54.8	56.0	-	39.4
LLaVA-Video (Zhang et al., 2024b)	64	70.8	58.2	63.3	83.2	42.2
LLaVA-Mini (Zhang et al., 2025a)	-	44.3	19.3	40.3	47.6	-
LongLLaVA (Wang et al., 2024b)	-	53.3	42.1	51.6	72.2	-
LongVU (Shen et al., 2024)	1 fps	65.4	53.5	55.3	78.0	-
Video-XL (Shu et al., 2025)	2048	64.9	50.7	55.5	77.5	-
Vamba (Ren et al., 2025)	1 fps	65.9	55.9	57.8	78.1	-
LongVU-TTT (ours)	$512 \rightarrow 128$	**71.80**	**60.40**	**64.43**	**83.50**	**44.30**

Table 2: Controlled temporal-operator study using the same fixed 30% subset of the Stage 2 training mixture. All variants use the same base model, insertion point, feature width, frame input, LLM token budget, and training setup. (a) compares projector-stage temporal operators; TTT variants use chunks of 8 frames. (b) isolates fast-weight adaptation, cross-chunk state carry, and 2D locality. LVB denotes LongVideoBench; VMME denotes Video-MME.

(a) Temporal operator comparison

Method	MLVU	LVB	VMME
LLaVA-Video baseline	60.35	52.54	58.20
Bidirectional Mamba2	62.46	53.55	60.80
Gated DeltaNet	62.92	53.96	61.31
TTT-MLP	63.38	54.23	61.14
TTT-MLP (causal)	64.38	55.16	61.73
Temporal Transformer	63.94	54.84	61.67
TTT-Conv (non-causal)	61.52	53.46	61.08
TTT-Conv (ours)	**65.50**	**56.17**	**62.11**

(b) TTT-Conv mechanism ablations

Variant	MLVU	LVB	VMME
No temporal layer	60.35	52.54	58.20
Static grouped Conv	62.11	53.48	60.12
Reset every frame	62.84	53.91	60.55
Reset every chunk	64.21	55.08	61.44
TTT-Conv, 1×1	64.36	55.02	61.52
TTT-Conv, shuffled grid	63.77	54.46	60.98
TTT-Conv, 3×3	**65.50**	**56.17**	**62.11**

Pretrain-595K (Sharma et al., 2018) and 200K short-video instruction samples from LLaVA-Video-178K (Zhang et al., 2024b), using a learning rate of 5×10^{-4}. In the second stage, all model components are trained jointly on 1.5M samples from LLaVA-Video-178K (Zhang et al., 2024b) and the LLaVA-OneVision image dataset (Li et al., 2024a), with a minimum of 32 and a maximum of 512 frames per video. We use learning rates of 2×10^{-5} for the LLM, 1×10^{-4} for both the MLP projector and TTT layer, and 2×10^{-6} for the ViT backbone. Stage 2 uses a global batch size of 128. Batch-tiled ViT processing and asynchronous activation offloading are enabled throughout Stage 2.

4.2 Main Results

Table 1 places LongVU-TTT in the broader video-MLLM landscape. This is a contextual comparison rather than a claim of a new state of the art: models differ in training data, backbones, and frame policies. Our primary comparison is LLaVA-Video, which uses the same pipeline, training data, and Qwen2-7B baseline and therefore most directly isolates the contribution of projector-stage TTT. LongVU-TTT improves over this reference on all five benchmarks, with the largest gains on LongVideoBench and LVBench.

Table 3: Temporal contextualization and frame selection. (a) separates the number of frames processed by the temporal module from the visual budget delivered to the LLM. (b) compares selection policies with 512 input frames and a fixed 128-frame LLM budget; all variants use the same TTT-Conv checkpoint. In/out denotes input/selected frame counts; LVB denotes LongVideoBench.

(a) Context and compression

Model	In/out	Selection	MLVU	LVB
LLaVA-Video	64/64	Uniform	70.80	58.20
LLaVA-Video	512/128	Uniform	65.40	55.14
LongVU-TTT	512/64	Uniform	68.42	58.46
LongVU-TTT	512/128	Hybrid	**71.80**	60.40
LongVU-TTT	512/256	Uniform	70.80	59.42
LongVU-TTT	128/128	All	71.50	**61.02**

(b) Frame-selection ablation

Selection	MLVU	LVB
Random	69.66	58.74
Uniform	70.63	59.72
Gradient shift (G_t)	71.19	60.07
Loss delta (R_t)	71.30	60.16
Hybrid (I_t)	**71.80**	**60.40**

4.3 Controlled Temporal-Operator Study

Table 2(a) compares temporal modules at the same projector location. All temporal modules improve over independent frames in most settings, while TTT-Conv is strongest across all three benchmarks. The causal/non-causal Conv comparison isolates update ordering, and causal TTT-MLP versus TTT-Conv isolates the 2D local fast-weight operator. Panel (b) shows that the gain is not explained by an added convolution alone: cross-chunk carry improves all benchmarks, while the 1×1 and shuffled-grid controls reduce the advantage. Detailed variant definitions are in Section A.5.

4.4 Temporal Context and Frame Selection

Table 3(a) shows that 512-frame temporal processing followed by 128-frame hybrid selection improves over the frame-independent 512-to-128 baseline. The 64-frame point shows that contextualization also preserves useful information at a smaller LLM budget, whereas the direct 128-frame result shows that longer pre-aggregation is not uniformly beneficial. Panel (b) isolates the selector: both TTT-derived signals improve over uniform sampling, and their weighted combination I_t is strongest. G_t responds to abrupt changes in update direction, while R_t captures sustained within-scene novelty. Importance-ranked frames are selected first and the remaining budget is filled uniformly. The pre-aggregation length sweep remains in Section A.6, and qualitative score traces are in Section A.8.

4.5 Temporal Aggregation versus Episodic Retention

We identify the temporal evidence window supporting each question in the LongVideoBench temporal subset and remove that window from the explicit frames passed to the LLM. The remaining contextualized frames are retained. Questions are grouped by the normalized gap between the omitted evidence and question timestamp, testing whether evidence can be recovered from propagated representations rather than directly retained frames.

Table 4: Evidence-withholding diagnostic on the LongVideoBench temporal subset. Source evidence is removed from the explicit LLM input for the first three rows; the final row retains it as an explicit-evidence upper bound. Columns denote normalized temporal-gap bins between the evidence and question timestamp. **Bold** denotes the best result among the evidence-withheld methods.

Method	0–25%	25–50%	50–75%	75–100%
No temporal layer	53.18	50.81	48.95	48.31
TTT-Conv, reset every 8 frames	54.37	51.22	49.10	48.42
TTT-Conv, state carried	**56.42**	**53.07**	**49.84**	**48.66**
TTT-Conv + explicit evidence (upper bound)	61.35	58.68	55.29	53.11

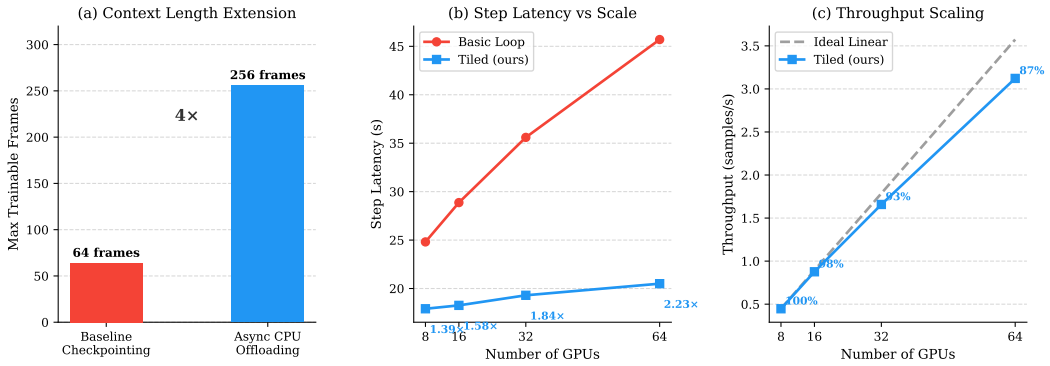

Figure 2: System optimization results measured on A100 40GB GPUs with NVLink intra-node and InfiniBand inter-node connections. (a) Maximum trainable frame count for standard gradient checkpointing versus asynchronous CPU activation offloading on a fixed 8-GPU setup. (b) Training step latency for batch-tiled ViT processing versus the standard per-shard loop from 8 to 64 GPUs at 256 frames with 8 tiles. (c) Throughput scaling efficiency under the same 256-frame, 8-tile setting relative to ideal linear scaling.

Table 4 shows that carrying the fast state improves recovery when omitted evidence is nearby, with the largest gains at short and medium gaps. The benefit attenuates beyond half of the video, whereas retaining explicit evidence helps at every gap. The fast state is therefore a causal temporal aggregator rather than a dependable episodic store.

4.6 System Efficiency

Figure 2 validates the optimizations used for long-video training. Asynchronous CPU activation offloading extends the maximum trainable frame count from 64 to 256 on a fixed 8-GPU setup. Batch-tiled ViT processing defers ZeRO-3 reduce-scatter until the final tile, yielding speedups from $1.39\times$ at 8 GPUs to $2.23\times$ at 64 GPUs while retaining 87% throughput-scaling efficiency. Projector-stage latency is reported separately in Section A.7.

5 Conclusion

We presented LongVU-TTT, a video MLLM that addresses spatiotemporal redundancy in long video understanding through a convolutional Test-Time Training layer inserted between the ViT encoder and the LLM. By parameterizing the fast weights as grouped convolutions rather than the MLP formulation common in prior TTT work, the layer uses causal fast-weight updates with explicit 2D local interactions before the LLM. The gradient signals produced during the inner-loop optimization provide a natural importance measure that, combined with uniform sampling, allows compression from 512 to 128 frames while retaining explicit evidence for the LLM. Controlled operator and retention studies show that causal 2D fast weights are effective temporal aggregators but do not replace long-horizon episodic storage; their value lies in contextualizing features before evidence selection. To enable training at this scale on commodity 40GB hardware, we introduced a batch-tiled ViT processing strategy and asynchronous CPU activation offloading that together extend the maximum trainable sequence length by $4\times$ and reduce training step time by up to $2.23\times$ at 64 GPUs without requiring sequence parallelism. LongVU-TTT achieves competitive performance across five video understanding benchmarks, improving over its closest training baseline on all five benchmarks.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing and organization, including improving clarity and readability, drafting and revising captions and result summaries, checking consistency between methodological descriptions and experimental reporting, and modifying scientific-figure layouts. These tools were not used to generate the underlying research ideas, implement, run experiments, or produce experimental data.

All AI-assisted text, calculations, interpretations, citations, and figures were reviewed and verified by the authors. We take responsibility for the final content of this work, including all claims and artifacts produced with AI assistance.

REFERENCES

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

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Atlas: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025.

Stas Bekman, Samyam Rajbhandari, Michael Wyatt, Jeff Rasley, Tunji Ruwase, Zhewei Yao, Aurick Qiao, and Yuxiong He. Arctic long sequence training: Scalable and efficient training for multi-million token sequences. *arXiv preprint arXiv:2506.13996*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris Kitani, and László Jeni. Don't look twice: Faster video transformers with run-length tokenization. *Advances in Neural Information Processing Systems*, 37:28127–28149, 2024.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024.

InternVL Team. Internvl2. `https://internvl.readthedocs.io/en/latest/internvl2.0/introduction.html`, 2024. Accessed 2024-07-04.

Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Gedas Bertasius, and Lorenzo Torresani. Bimba: Selective-scan compression for long-range video question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29096–29107, 2025.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, Sungjin Ahn, Jan Kautz, Hongxu Yin, Yao Lu, Song Han, and Wonmin Byeon. STORM: Token-efficient long video understanding for multimodal LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 5889–5900, 2025. URL `https://openaccess.thecvf.com/content/ICCV2025W/CLVL/html/Jiang_STORM_Token-Efficient_Long_Video_Understanding_for_Multimodal_LLMs_ICCVW_2025_paper.html`.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024a. URL https://arxiv.org/abs/2408.03326.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European conference on computer vision*, pp. 237–255. Springer, 2024b.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024c.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. Videochat-flash: Hierarchical compression for long-context video modeling, 2025. URL https://arxiv.org/abs/2501.00574.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2023.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context, 2023.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, and Wenhu Chen. Vamba: Understanding hour-long videos with hybrid mamba-transformers, 2025. URL https://arxiv.org/abs/2503.11579.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2556–2565, 2018.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. Longvu: Spatiotemporal adaptive compression for long video-language understanding, 2024. URL https://arxiv.org/abs/2410.17434.

Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26160–26169, 2025.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

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.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024a. URL https://arxiv.org/abs/2409.12191.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22958–22967, 2025.

Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via hybrid architecture, 2024b. URL https://arxiv.org/abs/2409.02889.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=3G1ZDXOI4f.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa:next phase of question-answering to explaining temporal actions, 2021. URL https://arxiv.org/abs/2105.08276.

Songlin Yang and Yu Zhang. FLA: A triton-based library for hardware-efficient implementations of linear attention mechanisms, 2024. URL https://github.com/fla-org/flash-linear-attention.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, et al. Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10807–10816, 2025.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. URL https://arxiv.org/abs/2306.02858.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024a.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*, 2025a.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024b.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding, 2024. URL https://arxiv.org/abs/2406.04264.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.