State-Space Hierarchical Compression with Gated Attention and Learnable Sampling for Hour-Long Video Understanding in Large Multimodal Models

Geewook Kim[1, 2], **Minjoon Seo**[2]
[1]NAVER Cloud AI
[2]KAIST AI
gwkim.rsrch@gmail.com, minjoon@kaist.ac.kr

Abstract

We propose an efficient framework to compress massive video-frame features before feeding them into large multimodal models, thereby mitigating the severe token explosion arising from hour-long videos. Our design leverages a bidirectional state-space model equipped with a gated skip connection and a learnable weighted-average pooling mechanism applied to periodically inserted learned queries. This structure enables hierarchical downsampling across both spatial and temporal dimensions, preserving performance in a cost-effective manner. Across challenging hour-long video understanding tasks, our approach demonstrates competitive results against state-of-the-art models, while significantly reducing overall token budget. Notably, replacing our state-space model with conventional modules results in substantial performance degradation, highlighting the advantages of the proposed state-space modeling for effectively compressing multi-frame video information. Our framework emphasizes resource-conscious efficiency, making it practical for real-world deployments. We validate its scalability and generality across multiple benchmarks, achieving the dual objectives of efficient resource usage and comprehensive video understanding.

Code — https://github.com/naver-ai/mambamia
Extended version — https://arxiv.org/abs/2506.13564

Introduction

The ability to process and understand long-form video is rapidly becoming central to the next generation of multimodal AI systems. Large language models (LLMs) augmented with visual input—so-called large multimodal models (LMMs)—have recently achieved impressive results on images and short video clips (Liu et al. 2023; Kim and Seo 2024; Li et al. 2025a). However, lifting these models to hour-scale video remains a formidable challenge. Simply representing every frame and patch in a long video leads to a dramatic spike in token sequences—often numbering in the hundreds of thousands—far surpassing the capacity of standard models and hardware. Notably, this phenomenon renders state-of-the-art approaches either impractical or inefficient for scalable video understanding (Zhang et al. 2025c, 2024a).

A wide range of prior work attempts to mitigate this "token explosion." Per-frame spatial pooling and token pruning can reduce redundancy locally (Li et al. 2025a; Zhang et al. 2024a; Cheng et al. 2024; Zhang et al. 2025a), but they fail to address accumulation over long time horizons. Other approaches depend on task- or query-specific selection (Cheng et al. 2025; Shen et al. 2025), trading away flexible context modeling and limiting downstream applicability. Consequently, it remains an open problem to design a general-purpose, learnable system that can efficiently compress both spatial and temporal redundancy, while preserving the wide context needed for robust reasoning about long, real-world videos.

In this paper, we address this limitation with **MambaMia** (**Mamba** for **M**assive **i**nput-frame **a**ggregation), a general and modular framework for hierarchical video token compression. Our approach introduces two key innovations: a gated patch aggregation module that integrates spatial and short-range temporal cues via bi-directional state-space (Mamba) modeling, and a lightweight time-axis aggregator that summarizes global video structure and adaptively filters frames using a cumulative, data-driven importance signal. By combining these stages in a hierarchical pipeline, MambaMia systematically distills massive video input to a compact set of tokens that preserves both fine-grained details and holistic temporal context.

What sets our approach apart is not just efficiency, but also its ability to scale to extremely long videos without sacrificing accuracy. On LVBench, a challenging benchmark requiring reasoning over hour-long videos, MambaMia reduces LLM token usage to just 4.7K—yet achieves a score of 44.6, outperforming contemporary models such as mPLUG-Owl3 (43.5) and Qwen2-VL (42.0). These results demonstrate that our model brings efficient, hour-long video understanding within reach on standard hardware—a key step toward practical, deployable LMMs.

Our main contributions are as follows. First, we introduce a unified architecture, MambaMia, which leverages gated patch aggregation (GPA) and a time-axis aggregator (TAA) to deliver highly efficient and accurate token compression for long videos, as demonstrated through extensive and controlled comparisons with prior compression and sequence modeling approaches. Second, we propose a novel delta-time-based adaptive filtering algorithm, implemented on top

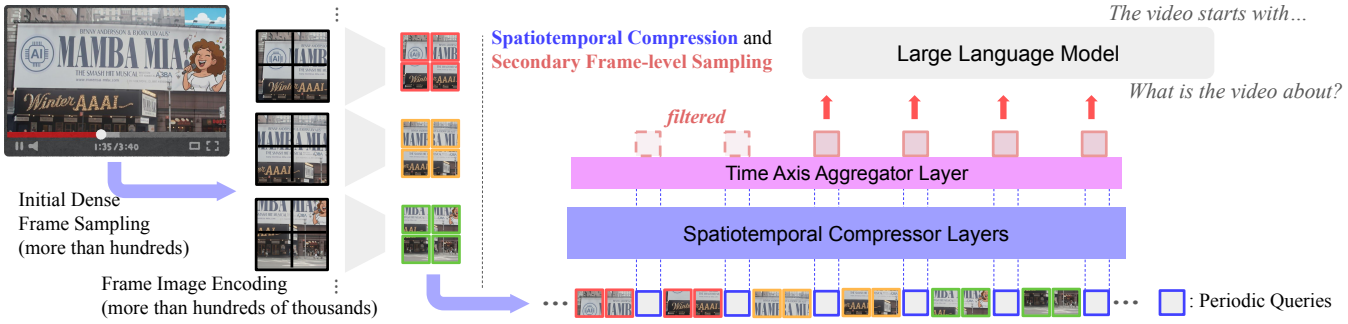

Figure 1: Overview of the MambaMia framework. Given a long video, we densely sample frames and embed patches to form a large sequence of visual tokens. Our framework then applies two-stage compression: (i) a spatiotemporal compression layer with periodic learnable queries aggregates local features, (ii) a time-axis aggregator uses delta-time values for adaptive frame selection. This pipeline efficiently reduces token count while preserving rich video context for LLM processing.

of the TAA, which further reduces temporal redundancy by selecting salient frames in a data-driven way; our ablation studies confirm its effectiveness for long-form video inputs. Finally, we benchmark and release MambaMia alongside the latest large multimodal models, demonstrating that our approach offers state-of-the-art accuracy with drastically fewer LLM tokens and minimal compute, while remaining simple to implement and fast to train from scratch or on top of existing LLM pipelines.

In the following sections, we detail the design, rationale, and empirical benefits of MambaMia, and chart promising future paths for hour-long video understanding research.

Related Work

State-Space Models and Mamba State-space models (SSMs) have gained traction for handling long sequences efficiently, thanks to their linear complexity compared to the quadratic scaling of attention (Gu and Dao 2024; Vaswani et al. 2017; Black et al. 2022). While initially proposed for language modeling, Mamba and related architectures have been adapted for vision tasks as well, demonstrating strong scalability in works such as ViM (Zhu et al. 2024). Recent advances, including bi-directional Mamba variants, have further enhanced video sequence modeling (Li et al. 2025c; Park et al. 2024). Building on these developments, some studies have begun leveraging SSMs for large-scale video feature aggregation and compression in multimodal models. Nonetheless, this area is still emerging, and optimal architectural strategies remain under exploration.

Spatiotemporal Video Token Compression For video understanding in LMMs, it is common practice to treat videos as image sequences, often using per-frame spatial token reduction before temporal concatenation. Various works propose pooling-based spatial compression (Li et al. 2025a; Zhang et al. 2024b, 2025b; Chung et al. 2025; Cha et al. 2024) and lightweight cross-attention modules like Q-Formers (Zhang et al. 2025c; Bai et al. 2023; Wang et al. 2024a; Bai et al. 2025) to reduce token count (Zhang et al. 2025c; Li et al. 2025b). However, these 2D reductions, when applied independently per frame, can still lead to very long token sequences for long or densely-sampled videos, resulting in high computational overhead (Zhang et al. 2024a; Li et al. 2025a).

To further mitigate redundancy, some approaches adopt spatiotemporal compression via 3D pooling (Maaz et al. 2024; Cheng et al. 2024), token pruning (Zhang et al. 2025a), or attention-based resampling of informative tokens either within or across frames (Li, Wang, and Jia 2024; Li et al. 2024). More recently, Mamba-based models have been explored for video token compression, capitalizing on their ability to efficiently aggregate long visual sequences for LLM input. Examples include VAMBA (Ren et al. 2025), integrating cross-attention Mamba into LLMs, and BIMBA (Islam et al. 2025), which uses periodic non-parametric queries with bi-directional Mamba blocks for summarization. Despite rapid progress, optimal SSM-based video compression architectures for LMMs remain an open research problem. Additional discussion of related work can be found in Appendix A.

Method

We address the problem of condensing long-form video into compact, information-rich representations suitable for large language models. As illustrated in Figure 1, our framework comprises two hierarchical stages.

First, a spatiotemporal compression layer aggregates local context across patches and frames into a small set of anchor tokens. Second, a time-axis aggregator models temporal dynamics over these anchors and adaptively selects salient frames. This hierarchical compression enables video understanding over hundreds of frames under a controlled token budget while preserving downstream performance. Below, we describe each component in detail.

Preliminary: State-Space Models and Mamba

A key building block of our approach is the use of SSMs as scalable "sequence compressors" between the vision encoder and LLM. SSMs such as the Mamba family (Gu and Dao 2024; Dao and Gu 2024) enable *linear* computational complexity in sequence length, a critical advantage over the quadratic scaling of Transformers for long-range modeling.

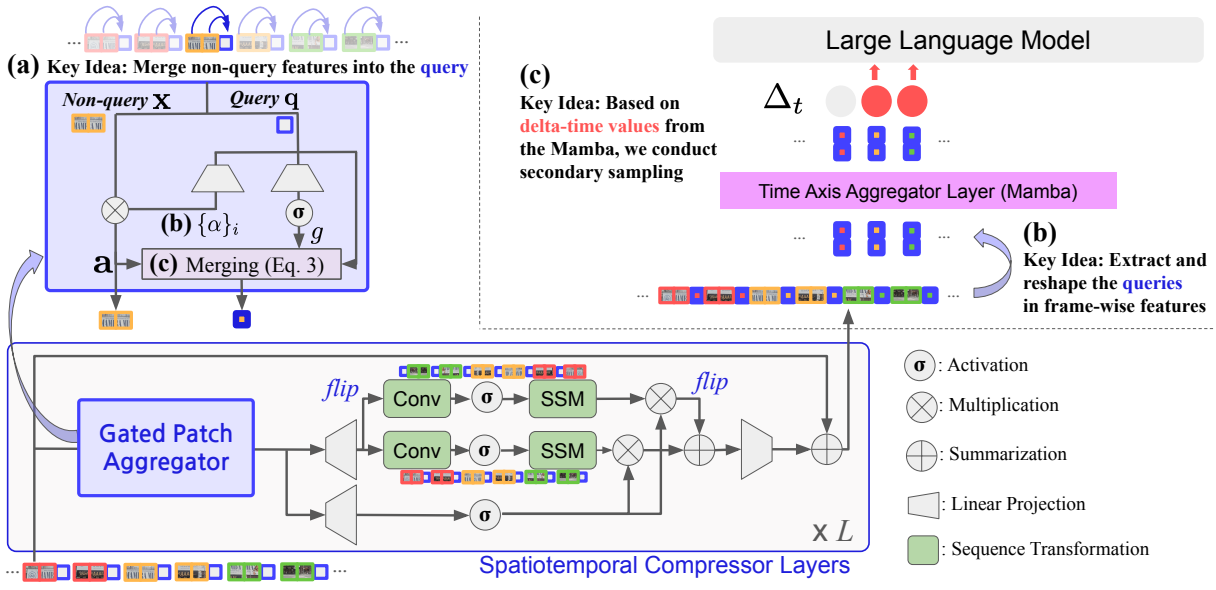

Figure 2: Architecture. (a) Periodic query tokens aggregate local context from nearby tokens using learnable pooling and a gating mechanism. (b) Frame-wise queries are extracted and reorganized into temporal sequences. (c) The time-axis aggregator models temporal dependencies and uses delta-time values for adaptive frame sampling before LLM input.

Formally, a discrete SSM updates hidden state h_t with input x_t as:

$$h_t = \overline{\mathbf{A}} h_{t-1} + \overline{\mathbf{B}} x_t, \quad y_t = \mathbf{C} h_t, \tag{1}$$

where $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C}$ are state-transition and projection matrices (see (Gu and Dao 2024; Dao and Gu 2024) for details).

In particular, the Mamba architecture achieves high expressivity by dynamically updating select parameters—notably, the input/output mixing and the adaptive step-size Δ_t. The **adaptive** Δ_t mechanism allows a flexible trade-off between quickly incorporating new input (large Δ_t) and sustaining long-term memory (small Δ_t), **which is essential for capturing salient events in long sequences**.

We refer readers to (Gu and Dao 2024; Dao and Gu 2024) and our supplementary material for additional background and derivations. In this work, we leverage these selective SSMs to achieve superior scalability and efficiency for long-form video modeling.

Proposed Framework and Architecture

Our architecture instantiates the two-stage design above with two main modules:

- **Spatiotemporal Compression Layer with Gated Patch Aggregation (GPA):** Aggregates spatial–temporal patch tokens into a compact set of representative anchor tokens using a bi-directional state-space block and a lightweight gated pooling around learnable query anchors.
- **Time Axis Aggregator (TAA):** Reorganizes anchor tokens by frames, applies a uni-directional state-space block along the temporal axis, and uses its adaptive step size to derive per-frame importance scores for selective frame sampling.

Together, GPA and TAA perform hierarchical compression: local aggregation at the patch level followed by adaptive temporal selection. This structure matches the multi-scale nature of real-world videos and provides a favorable trade-off between information retention and computational efficiency. A detailed overview of the pipeline is shown in Figure 2; we next describe each module in detail.

Spatiotemporal Compression Layer with GPA Given a video of $M = 384$ frames, each resized to 384×384 resolution and divided into 16×16 image patches, a vision encoder extracts $N = 576$ patch embeddings per frame, arranged on a 24×24 spatial grid. We then insert learnable query anchors *row-wise*, i.e., one anchor per spatial row that aggregates information from the $k = 24$ patches along that row, resulting in a long spatiotemporal token sequence of shape $(230\text{K} \times d_{\text{model}})$. We set $k = 24$ to match the full row width of the 24×24 grid.

This sequence is processed by a bi-directional Mamba block (Li et al. 2025c; Park et al. 2024) to share information across space and time. To promote explicit and adaptive context aggregation, we introduce the GPA module, wherein each anchor aggregates information from its row-wise neighborhood via a learnable gating function (details in the following and Fig. 2(a)). This hybrid module combines the merits of state-space modeling and localized attention, enabling efficient collapse of highly redundant patch-level information into a compressed set of anchor features.

After this layer, we obtain per-frame anchor features, effectively compressing the input sequence while retaining local and global video contexts (e.g., 24 anchors $\times$ 384 frames = 9.2K tokens).

Detail of GPA Figure 2 (a) illustrates the detailed structure of our proposed GPA module. To explicitly guide information aggregation toward the inserted query tokens, we introduce an adaptive gating mechanism. Formally, given a query token $\mathbf{q} \in \mathbb{R}^d$ and its neighboring patch embeddings $\{\mathbf{x}_i\}_{i=1}^k$, we generate aggregation weights $\{\alpha_i\}$ through a small linear layer (parameters $\mathbf{W}_\alpha, \mathbf{b}_\alpha$) followed by a softmax:

$$\boldsymbol{\alpha} = \mathrm{softmax}\left(\mathbf{W}_\alpha \mathbf{q} + \mathbf{b}_\alpha\right), \quad \mathbf{a} = \sum_{i=1}^{k} \alpha_i \mathbf{x}_i. \tag{2}$$

Note that the aggregation weights $\boldsymbol{\alpha}$ are intentionally conditioned only on the query token $\mathbf{q}$. This design implements a lightweight, query-conditioned pooling mechanism, avoiding the computational overhead of full content-aware attention at this fine-grained level.

Next, we compute a scalar gate $g \in [0, 1]$ from the query representation $\mathbf{q}$ using another linear layer (parameters $\mathbf{W}_g, \mathbf{b}_g$) and sigmoid function $\sigma(\cdot)$:

$$g = \sigma(\mathbf{W}_g \mathbf{q} + b_g), \quad \mathbf{f} = (1 - g)\,\mathbf{q} + g\,\mathbf{a}. \tag{3}$$

This learnable scalar gate g adaptively modulates how much neighboring token information replaces the original query representation: $g \approx 0$ preserves previous query contexts, while $g \approx 1$ heavily aggregates local information. Through this adaptive gating, each query token selectively captures key neighboring context, efficiently summarizing both local details and broader spatiotemporal contexts. Collecting these updated query features $\mathbf{f}$ over all anchors and frames yields the per-frame anchor tokens $\{f_t\}$ that serve as the input to the TAA.

Time Axis Aggregator for Adaptive Temporal Summarization After spatiotemporal compression, the aggregated anchor tokens from all frames are concatenated into a 1D temporal sequence with shape $(M,\ 24 \times d_{\mathrm{model}})$ (e.g., $M = 384$). This sequence is processed by the Time Axis Aggregator (TAA), a bi-directional Mamba block along the temporal axis.

As discussed in Section , Mamba (both v1 and v2) implements a state-space model with an *adaptive step size* Δ_t at each position, which controls how far the internal state is advanced between tokens (larger Δ_t induces a larger state update). We reuse this internal adaptive step size for each frame-level feature f_t (Gu and Dao 2024; Dao and Gu 2024), which is computed by a small scalar head:

$$\Delta_t = \mathrm{softplus}(\mathbf{W}_\Delta f_t + \mathbf{b}_\Delta), \tag{4}$$

yielding a non-negative scalar per frame. We interpret these Δ_t values as per-frame importance scores: frames with higher Δ_t are treated as more salient for the downstream task. Since Equation (4) is fully differentiable, $\mathbf{W}_\Delta$ and $\mathbf{b}_\Delta$ are learned end-to-end via the main LLM loss, so the model automatically assigns large Δ_t to informative frames.

To retain only the most informative frames, we apply cumulative delta-based sampling (Algorithm 1): we accumulate Δ_t over time and select a frame whenever the sum exceeds a threshold δ_{thresh}, then reset the accumulator. This adaptively reduces the sequence length (e.g., from 384 to 192 frames) while focusing on frames with large Δ_t. The selected frame features are then reshaped back to a sequence (e.g., $(192 \times 24, d_{\mathrm{model}})$) before being passed to the LLM. We add a residual connection around the TAA so that it functions primarily as an importance-based selector, minimally altering the aggregated features.

Algorithm 1: Cumulative Delta-based Frame Sampling

Require: Frame-level anchor features $\{f_1, \ldots, f_M\}$, importance scores (delta values) $\{\Delta_1, \ldots, \Delta_M\}$, delta threshold δ_{thresh}
Ensure: Sampled set of key frame features

```
Initialize accumulator A ← 0
Initialize selected set S ← ∅
for t = 1 to M do
    A ← A + Δ_t
    if A ≥ δ_thresh then
        Add f_t to S
        Reset A ← 0
    end if
end for
return S
```

Delta-based sampling is especially advantageous for long videos (e.g., 384 frames), while its computational benefits are less pronounced for shorter clips. Accordingly, we apply delta-based secondary sampling only to our final model and large-scale settings; this length-aware policy applies the secondary sampling only to inputs exceeding M_{thresh} frames, effectively bypassing it for shorter clips. Comprehensive ablations analyzing its impact are provided in the following sections and in the supplementary material.

Experiments

Experimental Setup

We summarize the essential elements of our experimental design here. Supplementary materials contain comprehensive information, including all dataset splits, exact hyperparameter settings (such as random seeds), software/hardware environments, and main scripts used in our experiments.

Training We first construct an image understanding model following the LLaVA-style pipeline (Liu et al. 2023), using 1 million instructional images collected according to the Elva recipe (Kim and Seo 2024). Next, we introduce our compression layers and perform modular alignment on the LLaVA-Pretrain dataset (Liu et al. 2023), where only the parameters of the compression layers are updated. Finally, the language model is unfrozen, and we conduct video-level instruction tuning using both the LLaVA-Video-Set (Zhang et al. 2024b) and a subset of the Vista-Set (Ren et al. 2024). For ablation studies, we use a subset of 133K video samples, while the final comparative model is trained on the full 1.4 million samples.

Implementation The default setup employs SigLIP2 (Tschannen et al. 2025) as the vision encoder (producing $N = 576$ tokens from 384×384 pixels),

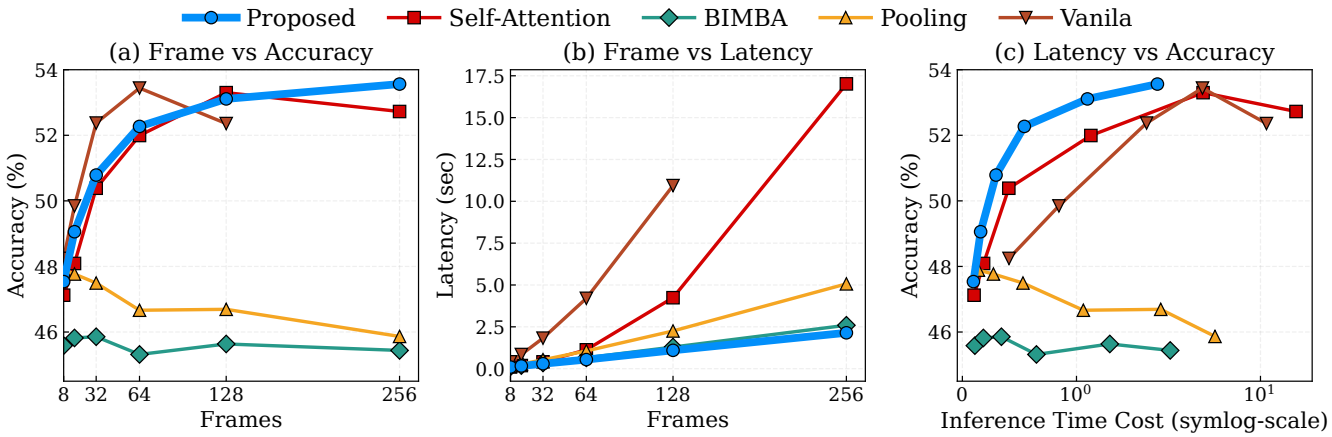

Figure 3: Trade-off analysis between input frames, inference latency, and accuracy. We benchmark five models (Proposed, BIMBA, Self-Attention, Pooling, and Vanilla) over varying input frame counts. (a) Average accuracy across LVBench, MLVU, and VideoMME as a function of frame number; (b) Inference latency (seconds) versus frame number; (c) Average accuracy as a function of inference time cost (symlog scale). The proposed method achieves the best balance, maintaining high accuracy with low test-time compute even as sequence length increases. Notably, (c) shows that our method consistently delivers the highest accuracy under any compute budget, clearly outperforming baselines—especially in high-latency regimes.

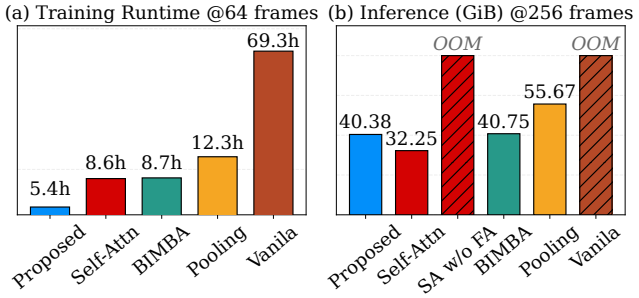

Figure 4: Comparison of training and inference costs for all baselines. (a) Training runtime (8A100-hours) at 64-frame input. (b) Peak per-GPU memory usage during inference at 256 frames. All models leverage FlashAttention-2 (Dao 2023) where possible. Self-Attention without Flash-Attn (red hatched bar) causes OOM errors at 256 frames.

paired with Qwen2-7B (Yang et al. 2024) as the language backbone. The vision encoder remains frozen throughout, consistent with efficient training practices (Liu et al. 2024; Kim and Seo 2024). For some ablation experiments, we also test CLIP-ConvNeXt-Large (Liu et al. 2022), Vicuna-7B (Chiang et al. 2023), Mamba2-2.7B (Dao and Gu 2024), and Pythia-2.8B (Biderman et al. 2023).

Videos are subsampled at up to 2.0 fps, with 64 frames for ablations and 128 frames for final training. At inference, various frame sampling strategies are considered. Module-only alignment (LM frozen) uses an initial learning rate of 1×10^{-4}, reduced to 2×10^{-5} during full multimodal fine-tuning. Our MambaMia block processes queries row-wise ($k = 24$) with $(M_{\text{thresh}}, \delta_{\text{thresh}})$ values of $(64, 0.5)$ for Vicuna-based, $(180, 0.6)$ for Qwen-based models (see Appendix J for the sensitivity analysis). All primary experiments use a fixed random seed (2025) for reproducibility. Furthermore, where computationally practical, we provide additional statistical validation over multiple independent trials in Appendix H.

Evaluation Benchmarks Our main objective is rigorous assessment of token compression architectures for hour-long video understanding. We focus on LVBench (Wang et al. 2024c) and MLVU (Zhou et al. 2024), both requiring reasoning over hour-long videos. To contextualize findings, we test five more benchmarks: HourVideo (Chandrasegaran et al. 2024), VideoMME (w/o subtitles) (Fu et al. 2024), LongVideoBench (Wu et al. 2024), VNBench (Zhao et al. 2024), and NExTQA (Xiao et al. 2021). Further details on the benchmark datasets used are provided in Appendix F.

Comparison Methods and Baselines Our experimental analyses aim to (1) identify the most effective methods for compressing video features, and (2) quantify their performance relative to state-of-the-art models. Experiments are structured accordingly.

We first compare five compression approaches under controlled conditions: (1) self-attention-based compression (Bai et al. 2023); (2) per-frame 2D average pooling (Zhang et al. 2024b; Li et al. 2025a); (3) a vanilla no-compression baseline (upper bound); (4) the BIMBA (Islam et al. 2025) architecture (Mamba-based baseline); and (5) our proposed module. All other factors are held constant, isolating the effect of compression.

We then benchmark against leading LMMs, including LongVA (Zhang et al. 2025b), Qwen2-VL (Wang et al. 2024b), Video-XL (Shu et al. 2025), LongVU (Shen et al. 2025), VAMBA (Ren et al. 2025), Video-LLAVA (Lin et al. 2024), LLAMA-VID (Li, Wang, and Jia 2024), LLAVA-NeXT-Video (Zhang et al. 2024a), BIMBA-LLAVA (Islam et al. 2025), LLaVA-Video (Lin et al. 2024), and LLaVA-

Model Duration	**Year**	**Inst. Data** (Img / Vid)	**LLM**	**# Max Tokens***	**LVBench** 30–140min	**MLVU** 3–120min	**HVid** <95min	**MME** <60min	**LongV** <60min	**VNB** <3min	**NQA** <3min
LongVA	Arxiv24	779K (Img only)	Qwen2-7B	18K+	-	56.3	-	52.6	51.8	41.5	68.3
Qwen2-VL	Arxiv24	n/a	Qwen2-7B	-	42.0	64.2	-	**63.3**	55.6	33.9	-
mPLUG-Owl3	Arxiv24	n/a	Qwen2-7B	-	43.5	-	-	53.5	52.1	-	78.6
VideoLLaMA2.1	Arxiv24	0.8M / 0.5M	Qwen2-7B	1.2K	-	-	-	54.9	-	-	-
LLaVA-Video†	Arxiv24	4.8M / 1.7M	Qwen2-7B	6.3K	40.3	66.3	33.5	62.4	**58.0**	55.9	**81.8**
LLaVA-OneVision†	TMLR25	8.4M / 0.4M	Qwen2-7B	6.3K	40.8	65.2	33.6	58.5	56.6	40.4	79.3
Video-XL	CVPR25	762K / 100K	Qwen2-7B	-	36.8	64.9	-	55.5	49.5	**61.6**	77.5
LongVU	ICML25	3.2M / 553K	Qwen2-7B	8K	37.8	65.4	-	55.3	53.5	-	78.0
VAMBA	ICCV25	4.8M / 1.7M	Qwen2-7B	-	42.1	65.9	-	57.8	55.9	-	78.1
Proposed		1M / 1.4M	Qwen2-7B	4.7K	**44.6**	**68.0**	**39.9**	58.3	57.1	56.5	80.6
Video-LLaVA	EMNLP24	665K / 100K	Vicuna-7B	2K	-	47.3	-	39.9	39.1	-	-
LLaMA-VID	ECCV24	625K / 107K	Vicuna-7B	16K+	-	33.2	-	25.9	-	10.8	-
LLaVA-NeXT-Video†	Blog24	760K / 100K	Vicuna-7B	2.3K	30.3	36.9	27.5	34.1	44.1	17.3	53.6
BIMBA-LLaVA	CVPR25	370K (Vid only)	Vicuna-7B	2.3K	-	47.2	-	45.7	-	-	72.4
Proposed$_{\text{Mini}}$		133K (Vid only)	Vicuna-7B	2.3K	**37.9**	**58.5**	**33.7**	**47.8**	**48.3**	**24.2**	**73.3**

Table 1: Comparison with Modern LMMs. We group models by their backbone LLM and mainly include recent, representative models with reproducible and from-scratch training pipelines. The table summarizes each method's publication venue, scale of training data for both images and videos, backbone LLM, and the maximum number of tokens processed for evaluation (measured on LVBench). Performance across diverse long video benchmarks is reported (higher is better). * Maximum tokens are based on best available information from original papers or released code. † Results marked are reproduced under identical evaluation settings for fair comparison; for details, see supplementary material in the extended version.

OneVision (Li et al. 2025a). Ablation configurations and more experimental specifics are detailed in the supplementary materials.

Controlled Comparisons of Compression Layers

To isolate the effect of different video compression and sequence modeling strategies, we conduct controlled experiments where only the compression layer is varied, with all other factors (e.g., data and LLM) held constant. This ensures a fair comparison of efficiency and scalability.

As shown in Figure 3, our method consistently achieves the best balance of accuracy, inference speed, and token efficiency across varying input lengths. Accuracy remains high even as input frames increase, without the steep latency or memory costs observed in self-attention and pooling baselines. Notably, our approach maintains practical inference time and resource use, while methods such as self-attention and vanilla become increasingly intractable for longer videos.

Figure 4 further highlights our method's lower computational and memory footprint during **both training and inference**, due to effective compression. In contrast, methods lacking a dedicated compression layer face significant inefficiency or out-of-memory errors at scale.

These results demonstrate that the proposed framework enables scalable long-video processing with a modest compute budget, driven by advances in compression and sequence modeling rather than increased compute or data.

Benchmark Comparison to Modern LMMs

We benchmark our proposed models against recent LMMs on a diverse suite of long-video understanding tasks (see Table 1). Our approach, based on Qwen2-7B, achieves competitive or state-of-the-art scores across key benchmarks—most notably, reaching 44.6 on LVBench and 68.0 on MLVU—while maintaining a substantially lower token budget (4.7K) compared to competing methods. Furthermore, our strong performance on VNBench, a *needle-in-a-video-haystack* benchmark, addresses concerns on missing brief yet critical events. This demonstrates that our architecture delivers both accuracy and efficiency, even as other models rely on larger compute and token counts.

To ensure fair comparison with Vicuna-based models, we additionally train a lightweight variant (**Proposed**$_{\text{Mini}}$) under similar data and training conditions. This mini model continues to outperform all Vicuna-based LMMs—even with limited training samples and visual tokens—highlighting the scalability and generalizability of our approach.

We emphasize that all our models are trained and evaluated on academic-scale, publicly available datasets with transparent pipelines, enabling reproducible comparison. We strictly adhere to matched-setting comparisons; results reported under different conditions (e.g., using proprietary backbones or further fine-tuning, like some models in STORM (Jiang et al. 2025) and BIMBA (Islam et al. 2025)) are discussed accordingly in Appendix A. See also Fig. 4 for efficiency analyses. While many recent LMMs pursue brute-force scaling of data and tokens, our results show that efficient architectural design and principled compression are key to scalable, accessible long-video understanding.

Analyses and Discussions

Proposed Module Ablations The proposed MambaMia architecture comprises several key components: GPA and TAA. To understand the individual contributions of these modules, we perform a series of ablation studies. As noted, these ablations are conducted using a lighter training recipe for efficiency. BIMBA, which can be regarded as the base model for MambaMia, utilizes a bi-directional Mamba block and periodically summarizes information into dedicated queries, but it generates these queries through 3D average pooling. We re-implement this baseline and systematically

GPA	TAA	Mamba	LVBench	MLVU	MME	Avg.
		V2	35.31	53.75	47.26	45.44
✓		V2	41.12	62.36	53.19	52.22
✓	✓	V2	41.06	63.96	55.67	53.56
✓	✓	V1	41.51	63.02	54.85	53.13

Table 2: Ablation study results for our compression layer architecture design. GPA and TAA denote each module's usage. The V2 and V1 indicate the Mamba block version.

turn the proposed modules on or off to assess their effects.

Table 2 presents the ablation results. First, replacing the average pooling query with our learnable GPA leads to marked improvements in performance across all benchmarks. While the addition of the TAA does not yield substantial accuracy gains on its own, its value lies in enabling our delta-based frame sampling approach, as discussed in the next section. Lastly, we verify that our proposed framework is compatible with both versions of the Mamba block (V1 and V2), with consistently strong results observed in all cases. We also confirm that the bi-directional spatiotemporal compressor blocks are beneficial; replacing it with unidirectional blocks on our best setting (Table 2, row 3) degrades the average score from 53.56 to 51.86. As the GPA module can be interpreted as a form of attention, we provide comprehensive qualitative visualizations for it in Appendix K.

Delta-based Frame Filtering Ablations The TAA layer outputs per-frame delta-time values, which we leverage for an adaptive secondary frame filtering step during inference. This mechanism, incorporated in our final model, further reduces the computational burden (see Table 1). For example, with 384 input frames, the initial sequence contains up to 221K patch tokens, which is compressed to about 9K tokens after the first stage ($k = 24$ anchors per frame). While this reduction is substantial, directly processing 9K tokens can still be challenging; thus, we apply delta-based frame filtering to select a more compact, information-rich subset. As a result, as shown in Table 1, the token budget can be reduced by roughly half, down to 4.7K.

We compare delta-based filtering (Algorithm 1)—which retains frames with the highest delta values, targeting approximately 50% retention using $\delta_{\text{thresh}} = 0.6$—to simple uniform downsampling at the same rate. As shown in Table 3, delta-based filtering consistently outperforms uniform sampling, demonstrating that the learned delta values are effective for identifying salient frames. To further illustrate this, qualitative analysis in Figure 5 shows that delta spikes often align with scene boundaries or salient events, corresponding to moments of semantic importance. We provide additional visualizations of per-frame delta-time values, including on diverse video samples, in Appendix K.

Ablations with Mamba as LLM One might wonder whether simply adopting a Mamba-based LLM backbone suffices for efficient long-video modeling, given SSMs' strong long-range processing capabilities. To address this, we conduct controlled experiments using CLIP-ConvNeXt-

Method	Sampling Rate	LVBench	MLVU	MME	Avg.
No Sampling	-	**45.58**	67.64	**58.59**	**57.27**
Uniform	50%	43.38	67.52	58.00	56.30
Delta-based	approx. 50%	44.61	**67.99**	58.26	56.95

Table 3: Ablation of secondary frame sampling methods. Results are shown for no sampling (i.e., disabling this secondary step), uniform 50% sampling, and delta-based filtering (using a threshold selected to retain half the frames). Delta-based sampling preserves accuracy while halving the input token count.

Figure 5: Visualization of per-frame delta-time values produced by the TAA layer. Peaks correspond to scene transitions or distinctive events (e.g., *needle-in-a-video-haystack*).

Method	LLM	Throughput	LVBench	MLVU	MME	Avg.
Vanilla	Pythia-2.8B	1.37	31.25	29.34	37.74	32.78
Vanilla	Mamba2-2.7B	1.16	30.28	40.33	31.81	34.14
Proposed	Pythia-2.8B	3.79	33.25	51.03	40.85	41.71
Proposed	Mamba2-2.7B	2.90	32.60	47.70	39.26	39.85

Table 4: Comparison of vanilla and proposed architectures with Pythia-2.8B and Mamba2-2.7B as language models. Compression-augmented models consistently yield higher efficiency and accuracy.

Large as the vision encoder, paired with either Pythia-2.8B or Mamba2-2.7B as the language model. As shown in Table 4, our compression-augmented method significantly outperforms the vanilla Mamba LLM setting (which uses an MLP projector, per LLaVA (Liu et al. 2024)), both in throughput and overall accuracy. Similar trends are observed with Pythia. While Mamba LLMs are promising, these results confirm that dedicated compression is crucial for practical long-video processing.

Conclusion

We introduce MambaMia, a unified framework for practical and efficient long-video understanding that integrates novel modular compression with advanced sequence modeling. Through extensive experiments on challenging hour-long video benchmarks, MambaMia demonstrates state-of-the-art accuracy and substantial efficiency gains while operating under realistic memory and compute constraints. Comprehensive ablations and analyses validate the critical importance of effective compression, and adaptive sequence processing across diverse backbones and benchmarks. We will fully release our codebase, models, and resources to enable reproducibility and further accelerate progress toward scalable, real-world video understanding.

References

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. *arXiv preprint arXiv:2308.12966*.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Biderman, S.; Schoelkopf, H.; Anthony, Q.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E.; Skowron, A.; Sutawika, L.; and van der Wal, O. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. arXiv:2304.01373.

Black, S.; Biderman, S.; Hallahan, E.; Anthony, Q.; Gao, L.; Golding, L.; He, H.; Leahy, C.; McDonell, K.; Phang, J.; Pieler, M.; Prashanth, U. S.; Purohit, S.; Reynolds, L.; Tow, J.; Wang, B.; and Weinbach, S. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*.

Cha, J.; Kang, W.; Mun, J.; and Roh, B. 2024. Honeybee: Locality-enhanced Projector for Multimodal LLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chandrasegaran, K.; Gupta, A.; Hadzic, L. M.; Kota, T.; He, J.; Eyzaguirre, C.; Durante, Z.; Li, M.; Wu, J.; and Li, F.-F. 2024. HourVideo: 1-Hour Video-Language Understanding. In *Advances in Neural Information Processing Systems*.

Cheng, C.; Guan, J.; Wu, W.; and Yan, R. 2025. Scaling Video-Language Models to 10K Frames via Hierarchical Differential Distillation. In *Forty-second International Conference on Machine Learning*.

Cheng, Z.; Leng, S.; Zhang, H.; Xin, Y.; Li, X.; Chen, G.; Zhu, Y.; Zhang, W.; Luo, Z.; Zhao, D.; and Bing, L. 2024. VideoLLAMA 2: Advancing Spatial-Temporal Modeling and Audio Understanding in Video-LLMs. arXiv:2406.07476.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chung, J.; Zhu, T.; Saez-Diez, M. G.; Niebles, J. C.; Zhou, H.; and Russakovsky, O. 2025. Unifying Specialized Visual Encoders for Video Language Models.

Dao, T. 2023. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. arXiv:2307.08691.

Dao, T.; and Gu, A. 2024. Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Fu, C.; Dai, Y.; Luo, Y.; Li, L.; Ren, S.; Zhang, R.; Wang, Z.; Zhou, C.; Shen, Y.; Zhang, M.; et al. 2024. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. *arXiv preprint arXiv:2405.21075*.

Gu, A.; and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling*.

Islam, M. M.; Nagarajan, T.; Wang, H.; Bertasius, G.; and Torresani, L. 2025. BIMBA: Selective-Scan Compression for Long-Range Video Question Answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*.

Jiang, J.; Li, X.; Liu, Z.; Li, M.; Chen, G.; Li, Z.; Huang, D.-A.; Liu, G.; Yu, Z.; Keutzer, K.; Ahn, S.; Kautz, J.; Yin, H.; Lu, Y.; Han, S.; and Byeon, W. 2025. Token-Efficient Long Video Understanding for Multimodal LLMs. arXiv:2503.04130.

Kim, G.; and Seo, M. 2024. On Efficient Language and Vision Assistants for Visually-Situated Natural Language Understanding: What Matters in Reading and Reasoning. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 16978–17000. Miami, Florida, USA: Association for Computational Linguistics.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; and Li, C. 2025a. LLaVA-OneVision: Easy Visual Task Transfer. *Transactions on Machine Learning Research*.

Li, K.; Goyal, S.; Semedo, J. D.; and Kolter, J. Z. 2025b. Inference Optimal VLMs Need Only One Visual Token but Larger Models. In *The Thirteenth International Conference on Learning Representations*.

Li, K.; Li, X.; Wang, Y.; He, Y.; Wang, Y.; Wang, L.; and Qiao, Y. 2025c. VideoMamba: State Space Model for Efficient Video Understanding. In *Computer Vision – ECCV 2024*, 237–255. Cham.

Li, K.; Wang, Y.; He, Y.; Li, Y.; Wang, Y.; Liu, Y.; Wang, Z.; Xu, J.; Chen, G.; Luo, P.; Wang, L.; and Qiao, Y. 2024. MVBench: A Comprehensive Multi-modal Video Understanding Benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22195–22206.

Li, Y.; Wang, C.; and Jia, J. 2024. LLaMA-VID: An Image is Worth 2 Tokens in Large Language Models. In *Computer Vision – ECCV 2024*.

Lin, B.; Ye, Y.; Zhu, B.; Cui, J.; Ning, M.; Jin, P.; and Yuan, L. 2024. Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 5971–5984. Miami, Florida, USA: Association for Computational Linguistics.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 26296–26306.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 34892–34916. Curran Associates, Inc.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. 2024. Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12585–12602. Bangkok, Thailand: Association for Computational Linguistics.

Park, J.; Kim, H.-S.; Ko, K.; Kim, M.; and Kim, C. 2024. VideoMamba: Spatio-Temporal Selective State Space Model. In *ECCV 2024: Proceedings, Part XXV*.

Ren, W.; Ma, W.; Yang, H.; Wei, C.; Zhang, G.; and Chen, W. 2025. Vamba: Understanding Hour-Long Videos with Hybrid Mamba-Transformers. arXiv:2503.11579.

Ren, W.; Yang, H.; Min, J.; Wei, C.; and Chen, W. 2024. VISTA: Enhancing Long-Duration and High-Resolution Video Understanding by Video Spatiotemporal Augmentation. arXiv:2412.00927.

Shen, X.; Xiong, Y.; Zhao, C.; Wu, L.; Chen, J.; Zhu, C.; Liu, Z.; Xiao, F.; Varadarajan, B.; Bordes, F.; Liu, Z.; Xu, H.; Soran, B.; Krishnamoorthi, R.; Elhoseiny, M.; and Chandra, V. 2025. LongVU: Spatiotemporal Adaptive Compression for Long Video-Language Understanding.

Shu, Y.; Liu, Z.; Zhang, P.; Qin, M.; Zhou, J.; Liang, Z.; Huang, T.; and Zhao, B. 2025. Video-XL: Extra-Long Vision Language Model for Hour-Scale Video Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 26160–26169.

Tschannen, M.; Gritsenko, A.; Wang, X.; Naeem, M. F.; Alabdulmohsin, I.; Parthasarathy, N.; Evans, T.; Beyer, L.; Xia, Y.; Mustafa, B.; Hénaff, O.; Harmsen, J.; Steiner, A.; and Zhai, X. 2025. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv preprint arXiv:2502.14786*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Fan, Y.; Dang, K.; Du, M.; Ren, X.; Men, R.; Liu, D.; Zhou, C.; Zhou, J.; and Lin, J. 2024a. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Fan, Y.; Dang, K.; Du, M.; Ren, X.; Men, R.; Liu, D.; Zhou, C.; Zhou, J.; and Lin, J. 2024b. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv:2409.12191.

Wang, W.; He, Z.; Hong, W.; Cheng, Y.; Zhang, X.; Qi, J.; Huang, S.; Xu, B.; Dong, Y.; Ding, M.; and Tang, J. 2024c. LVBench: An Extreme Long Video Understanding Benchmark. arXiv:2406.08035.

Wu, H.; Li, D.; Chen, B.; and Li, J. 2024. LongVideoBench: A Benchmark for Long-context Interleaved Video-Language Understanding. In *Advances in Neural Information Processing Systems*, volume 37, 28828–28857.

Xiao, J.; Shang, X.; Yao, A.; and Chua, T.-S. 2021. NExT-QA: Next Phase of Question-Answering to Explaining Temporal Actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9777–9786.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024. Qwen2 Technical Report. arXiv:2407.10671.

Zhang, B.; Li, K.; Cheng, Z.; Hu, Z.; Yuan, Y.; Chen, G.; Leng, S.; Jiang, Y.; Zhang, H.; Li, X.; Jin, P.; Zhang, W.; Wang, F.; Bing, L.; and Zhao, D. 2025a. VideoLLAMA 3: Frontier Multimodal Foundation Models for Image and Video Understanding. arXiv:2501.13106.

Zhang, P.; Zhang, K.; Li, B.; Zeng, G.; Yang, J.; Zhang, Y.; Wang, Z.; Tan, H.; Li, C.; and Liu, Z. 2025b. Long Context Transfer from Language to Vision.

Zhang, S.; Fang, Q.; Yang, Z.; and Feng, Y. 2025c. LLaVA-Mini: Efficient Image and Video Large Multimodal Models with One Vision Token. In *The Thirteenth International Conference on Learning Representations*.

Zhang, Y.; Li, B.; Liu, h.; Lee, Y. j.; Gui, L.; Fu, D.; Feng, J.; Liu, Z.; and Li, C. 2024a. LLaVA-NeXT: A Strong Zero-shot Video Understanding Model.

Zhang, Y.; Wu, J.; Li, W.; Li, B.; MA, Z.; Liu, Z.; and Li, C. 2024b. Video Instruction Tuning with Synthetic Data.

Zhao, Z.; Lu, H.; Huo, Y.; Du, Y.; Yue, T.; Guo, L.; Wang, B.; Chen, W.; and Liu, J. 2024. Needle In A Video Haystack: A Scalable Synthetic Framework for Benchmarking Video MLLMs. *arXiv preprint*.

Zhou, J.; Shu, Y.; Zhao, B.; Wu, B.; Xiao, S.; Yang, X.; Xiong, Y.; Zhang, B.; Huang, T.; and Liu, Z. 2024. MLVU: A Comprehensive Benchmark for Multi-Task Long Video Understanding. *arXiv preprint arXiv:2406.04264*.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: efficient visual representation learning with bidirectional state space model. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.