CTEPM: Continuous-Time Event Process Memory for Long-Video Language Models

Yangyang Liu[1]

Unaffiliated, Beijing, China
liuyy159@gmail.com

Abstract. Long-video language models (Video-LMMs) face a fundamental bottleneck: temporal reasoning is often performed implicitly over discretized frames/clips and limited-context tokens. As a result, queries that require *computable* temporal structure—such as counting repeated events, comparing rates across time ranges, estimating intervals, or reasoning about order—are handled by heuristic sampling, retrieval, or free-form summarization. We propose **CTEPM**, a continuous-time event process memory that represents a video as a marked temporal point process with learnable intensities and semantic marks. CTEPM first converts dense video features into a sparse set of latent events with continuous timestamps and mark embeddings, then fits an interaction-aware marked point process that models global time trends and cross-event triggering/inhibition. Given a question, CTEPM executes a small library of temporal operators on the learned process to produce compact structured evidence, which conditions the final answer generation. This design makes temporal reasoning explicit, lightweight, and inspectable without performing video generation. Experiments on long-video understanding benchmarks demonstrate consistent improvements on temporally compositional queries under tight context budgets, while retaining competitive accuracy on standard video QA.

Keywords: Continuous-time event modeling · Marked temporal point processes · Long-video language models · Temporal operator execution

1 Introduction

Video–language models have rapidly progressed from short-clip recognition toward open-ended understanding of long videos.[19,40,42,17,1,25,10] However, long-video reasoning remains constrained by a mismatch between the *continuous* nature of time in videos and the *discrete* representations used in contemporary systems. Most Video-LMM pipelines reduce a long video to a limited set of frames, clips, or visual tokens[3,2,21,27], and then rely on a language model to implicitly infer temporal structure from these discretized observations.[41,12,18] This approach is effective for coarse semantic questions, yet it struggles whenever the question requires *computable* temporal quantities.

Consider queries such as: "How many times does the person drink?" "Does the action happen more frequently in the second half?" "Which happens first:

opening the door or sitting down?" "How long does the phone call last?" and "How long after event A does event B occur?". These questions are not merely about recognizing what appears in the video; rather, they require integrating evidence across time, aggregating repeated occurrences, and manipulating temporal relations. In current systems, such reasoning is typically approximated by heuristic clip retrieval[27,25], increased sampling budgets, or free-form summarization.[12,41,29] Yet summarization does not naturally yield answers to count, rate, interval, and order queries, and increasing the number of sampled clips quickly becomes prohibitive for long videos.[36,9,43,23,20]

A key observation is that many long-video questions can be expressed as *operators* over an underlying temporal structure. If we had access to a compact representation of *when* salient events occur and *how* they interact over time, then answering temporal queries could become a matter of executing a small set of well-defined computations. This motivates a shift from *token-level* temporal reasoning to *process-level* temporal reasoning: instead of treating time as an index attached to tokens, we model the video as a continuous-time stochastic process whose dynamics can be queried directly.[11,7,24,44]

In this paper we propose **CTEPM** (Continuous-Time Event Process Memory), a continuous-time temporal memory for long-video language models. CTEPM represents each video as a *marked temporal point process*: a sparse set of latent events with continuous timestamps and semantic mark embeddings, governed by type-specific conditional intensities. To build this representation, CTEPM first performs *event proposal* on dense video features to produce a small set of candidate events. It then learns an interaction-aware marked point process that captures (i) global temporal trends and (ii) cross-event effects, including both triggering and inhibition, enabling the model to represent temporal dependencies beyond local neighborhoods. Given a question, CTEPM parses it into a small set of temporal operators—such as expected counts, average rates, order probabilities, or expected intervals—and executes these operators on the learned process to produce compact structured evidence. A language model then conditions on this evidence to generate the final answer. Importantly, CTEPM does not perform video generation; the continuous-time process acts as a queryable memory that makes temporal reasoning explicit and lightweight.

CTEPM differs from existing long-video strategies in both representation and computation. Token compression and efficient inference methods reduce redundant visual tokens or attention cost [4,6,29,39], while sampling and retrieval strategies aim to fit long videos into limited context windows. However, these methods still leave temporal reasoning, such as counting, ordering, and interval estimation, largely to be implicitly handled by the language model. Recent event-centric memory methods organize videos into episodic or hierarchical event records for retrieval and downstream reasoning [34,35]. In contrast, CTEPM is not merely an event memory or retrieval structure: it is a queryable continuous-time process whose learnable intensities support computable temporal operators. This allows temporal reasoning to be expressed as explicit computations over intensities and sampled events, so that count, rate, order, interval, and duration

queries can be answered with compact, inspectable evidence rather than long retrieved contexts.

We summarize our main contributions as follows.

- We introduce a process-level temporal memory for long-video language models, representing videos as continuous-time marked temporal point processes with learnable dynamics.
- We design an interaction-aware process model that captures global temporal trends as well as triggering and inhibition across event types.
- We develop a temporal operator library and a query execution pipeline that compute structured evidence from the learned process for explicit temporal reasoning.
- We show consistent gains on temporally compositional long-video queries under tight context budgets, while retaining competitive accuracy on standard video QA.

2 Related Work

2.1 Video-LMMs for Long-Video Understanding and Benchmarks

Recent Video-LMMs extend LLMs to video by coupling a video encoder with a learnable interface and instruction tuning. Early systems such as VideoChat [19] and Video-LLaMA [40] show that LLMs can handle open-ended video dialogue and multimodal reasoning under limited frames. Subsequent lines scale data and unify image/video training, including LLaVA-Video [42], LLaVA-OneVision [17], and LLaVA-NeXT-Interleave [18]. In parallel, stronger foundation video encoders (e.g., InternVideo2 [32] and InternVideo2.5 [33]) improve spatiotemporal representations for downstream Video-LMMs.

Long-video understanding further stresses context limits and sparse, compositional events over minutes to hours. LongVideoBench [36] and Video-MME [9] evaluate hour-level video understanding, while MLVU [43] benchmarks multi-task long-form reasoning. EgoSchema [23] provides a diagnostic setting derived from long egocentric videos, and MVBench [20] systematically probes temporal skills across diverse task types. Together, these benchmarks show that gains on short-clip QA do not necessarily transfer to robust long-horizon temporal reasoning, motivating representations that make time *computable* rather than merely an index of sampled clips.

2.2 Memory, Retrieval, and Efficient Inference for Long Videos

A common strategy for long videos is to augment Video-LMMs with external memory or retrieval. MA-LMM [12] performs online processing and writes historical information into a memory bank, enabling long-term reasoning without exceeding the LLM context. Another route is to expand effective context capacity: LongVA [41] transfers long-context capability from language to vision, allowing dense frame ingestion and strong long-video performance.

Orthogonally, token-efficient inference reduces the cost of redundant visual tokens. FastVID [29] prunes spatiotemporally redundant tokens dynamically, while FitPrune [39] derives a training-free pruning rule by matching attention statistics. HoliTom [28] and FlashVID [8] further study training-free spatiotemporal token merging/compression for video LLMs, and PruneVid [14] tailors training-free pruning to video understanding. These lines (memory/retrieval, long-context transfer, and token compression) are closely related to our baselines: they increase capacity or efficiency, yet temporal reasoning is still largely implicit over discretized clips/tokens. In contrast, we construct a continuous-time, queryable temporal memory that supports explicit operators (count/order/interval/duration) under a fixed budget.

For **event-centric video memory**, recent works have also explored event-centric memory for long-video understanding. Video-EM represents long videos as temporally ordered episodic events and retrieves a compact set of informative memories for question answering [34], while EventMemAgent builds a hierarchical event memory for online video understanding, combining short-term event buffering, long-term event archives, and adaptive tool use [35]. These methods are closely related to our motivation of moving beyond frame-level token accumulation. However, their memories are primarily organized as episodic or hierarchical event records that are later retrieved or reasoned over by an LLM/agent. In contrast, CTEPM models sparse events as a continuous-time marked point process, where event occurrence, temporal interaction, triggering, and inhibition are parameterized by learnable intensities. This enables temporal questions such as count, rate, order, interval, and duration to be answered through explicit operators over the process rather than through retrieval alone.

2.3 Temporal Reasoning, Event-Centric Modeling, and Temporal Point Processes

Temporal reasoning in video is often approached via temporal grounding and event-centric representations, where the goal is to localize *when* an event occurs (typically as an interval) to support downstream reasoning. A complementary, principled view is continuous-time event modeling, which represents irregular occurrences and their dependencies with temporal point processes. Classical Hawkes processes [11] model self-/mutual-excitation via kernels, while neural variants such as RMTPP [7] and the Neural Hawkes Process [24] learn flexible history-dependent intensities; Transformer Hawkes Process [44] further uses self-attention to capture long-range dependencies in continuous time. Such formulations directly support temporal quantities that are hard to recover from discretized clips, including expected counts, event rates, order probabilities, and waiting-time distributions. Motivated by this, we represent a video as a marked temporal point process with learnable triggering and inhibition, and execute a small library of temporal operators to produce compact structured evidence for answering.

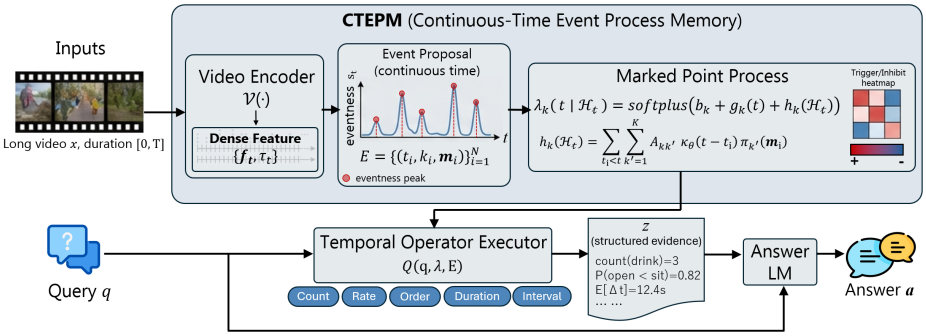

Fig. 1. Given a long video x and a query q, we first encode the video into dense timestamped features $\{f_t, \tau_t\}$ and propose a sparse set of continuous-time events $E = \{(t_i, k_i, m_i)\}_{i=1}^{N}$ from eventness peaks. CTEPM then builds a marked point process memory with type-specific intensities $\lambda_k(t \mid \mathcal{H}_t)$, capturing global temporal trends and cross-event triggering/inhibition (via the interaction matrix $A_{kk'}$). Finally, a temporal operator executor $\mathcal{Q}(q, \lambda, E)$ computes compact structured evidence z (e.g., count/order/interval/duration), which conditions an answer language model to produce the final response a under a fixed input budget.

3 Method

3.1 Problem Setup

We study long-video question answering and temporal reasoning with a video–language model under a strict non-generative setting: the input is a video $\boldsymbol{x}$ and a natural-language query q, and the output is a textual answer a. Let $\boldsymbol{x}$ span a continuous time horizon $[0, T]$. A video encoder produces a dense temporal feature sequence $\boldsymbol{F} = \{\boldsymbol{f}_t\}_{t=1}^{L}$, where $\boldsymbol{f}_t \in \mathbb{R}^D$ is associated with a timestamp $\tau_t \in [0, T]$. Most existing long-video pipelines compress $\boldsymbol{F}$ into a small set of tokens or retrieved clips, but still treat time as a discrete index.[12,41,29] In contrast, we aim to construct a *continuous-time, queryable memory* that supports explicit temporal operators (e.g., count, rate, order, interval, duration) while remaining lightweight enough for long videos.

Formally, we represent a video by a set of latent events

$$\mathcal{E} = \{(t_i, k_i, \boldsymbol{m}_i)\}_{i=1}^{N}, \qquad t_i \in [0, T], \tag{1}$$

where t_i is a *continuous* occurrence time, $k_i \in \{1, \ldots, K\}$ is an event type, and $\boldsymbol{m}_i \in \mathbb{R}^{d_m}$ is a continuous mark (semantic embedding). Given $(\boldsymbol{x}, q)$, our goal is to (i) infer $\mathcal{E}$ from $\boldsymbol{x}$, (ii) model the temporal dynamics of $\mathcal{E}$ as a marked point process, and (iii) answer q by executing a small set of temporal operators on this process and feeding the resulting structured evidence to a language model.

3.2 Overview

Our framework, **CTEPM** (Continuous-Time Event Process Memory), contains three components. Figure 1 provides an overview of CTEPM, illustrating how we

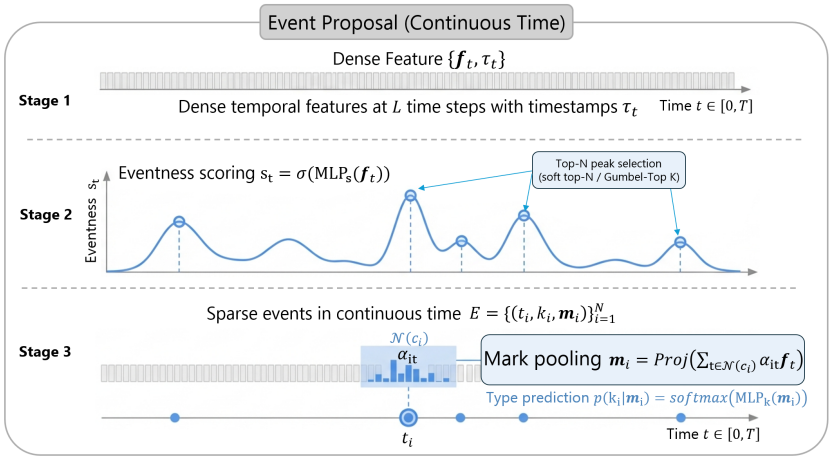

Fig. 2. Given dense timestamped video features $\{\boldsymbol{f}_t, \tau_t\}_{t=1}^{L}$, we predict an eventness score $s_t = \sigma(\mathrm{MLP}_s(\boldsymbol{f}_t))$ along the timeline and select the top-N peaks (e.g., via a soft top-N / Gumbel-Top-N relaxation). Each selected center c_i is mapped to a continuous timestamp $t_i = \tau_{c_i}$, forming a sparse event set $E = \{(t_i, k_i, \boldsymbol{m}_i)\}_{i=1}^{N}$. For each event, we pool local features within a temporal neighborhood $\mathcal{N}(c_i)$ using attention weights α_{it} to obtain the mark embedding $\boldsymbol{m}_i$, and optionally predict an event type distribution $p(k_i \mid \boldsymbol{m}_i)$.

convert dense video features into continuous-time events, build a marked point process memory, and answer queries by executing explicit temporal operators to produce structured evidence. First, an *event proposal* module converts dense video features into a sparse event set $\mathcal{E}$ with continuous times and marks. Second, an *event process memory* module learns a continuous-time marked point process that captures temporal trends and cross-event interactions (triggering and inhibition). Third, a *query executor* maps a question q into a differentiable temporal operator whose output is a compact structured evidence vector $\boldsymbol{z}$, which conditions the final answer model $p(a \mid q, \boldsymbol{z})$. This design avoids video generation and does not rely on active evidence selection: the heavy temporal reasoning is executed on a continuous-time process.

3.3 Video Encoder

We extract dense temporal features with a generic video encoder $\mathcal{V}(\cdot)$:

$$\boldsymbol{F} = \mathcal{V}(\boldsymbol{x}) = \{\boldsymbol{f}_t\}_{t=1}^{L}, \qquad \boldsymbol{f}_t \in \mathbb{R}^D, \quad \tau_t \in [0, T]. \tag{2}$$

The encoder can be instantiated by any modern video backbone or the visual tower of a Video-LMM[3,2,21]. CTEPM is agnostic to this choice; our contribution is the continuous-time memory and operator-based reasoning built on top of $\boldsymbol{F}$.

3.4 Event Proposal with Continuous Times

The event proposal module produces a sparse set of N events from $\boldsymbol{F}$. We predict an *eventness* score for each timestamped feature:

$$s_t = \sigma(\mathrm{MLP}_s(\boldsymbol{f}_t)) \in (0,1), \qquad t = 1, \ldots, L, \tag{3}$$

and select N peaks with a differentiable top-N operator. Concretely, we construct a soft selection distribution

$$\pi_t = \frac{\exp(s_t/\rho)}{\sum_{u=1}^{L} \exp(s_u/\rho)}, \tag{4}$$

where ρ is a temperature. We then obtain N event centers $\{c_i\}_{i=1}^{N}$ using either (i) repeated sampling without replacement via a Gumbel-Top-N relaxation, or (ii) a deterministic soft top-N approximation; both enable gradient-based training.[15,37] Each center index c_i is mapped to a continuous time by timestamp lookup:

$$t_i = \tau_{c_i} \in [0, T]. \tag{5}$$

Figure 2 illustrates our continuous-time event proposal module, which converts dense timestamped features into a sparse set of events by top-N peak selection and local mark pooling.

Event marks. To attach semantic content, we compute a mark embedding by local attention pooling[31] around each center:

$$\boldsymbol{m}_i = \mathrm{Proj}\left(\sum_{t \in \mathcal{N}(c_i)} \alpha_{it} \boldsymbol{f}_t \right), \qquad \alpha_{it} = \frac{\exp(\boldsymbol{w}^\top \boldsymbol{f}_t)}{\sum_{u \in \mathcal{N}(c_i)} \exp(\boldsymbol{w}^\top \boldsymbol{f}_u)}. \tag{6}$$

Here $\mathcal{N}(c_i)$ is a temporal neighborhood (e.g., a window of $\pm w$ clips). We also predict a discrete type distribution

$$p(k_i = k \mid \boldsymbol{m}_i) = \mathrm{softmax}(\mathrm{MLP}_k(\boldsymbol{m}_i))_k, \tag{7}$$

and use either the argmax type at inference or soft type assignments during training.

3.5 Event Process Memory: Marked Temporal Point Process

We model $\mathcal{E}$ as a continuous-time marked point process[5] with event-type-specific conditional intensities.[7,44] Let $\mathcal{H}_t = \{(t_j, k_j, \boldsymbol{m}_j) : t_j < t\}$ denote the event history before time t. For each type k, we define

$$\lambda_k(t \mid \mathcal{H}_t) = \mathrm{softplus}\Big(b_k + g_k(t) + h_k(\mathcal{H}_t)\Big), \tag{8}$$

where b_k is a bias, $g_k(t)$ captures global time trends, and $h_k(\mathcal{H}_t)$ models cross-event interactions. We set $g_k(t) = \mathrm{MLP}_g([\gamma(t)])$ using a time embedding $\gamma(t)$ (e.g., Fourier features).[30]

Triggering and inhibition via learnable kernels. To capture causal-like temporal dependencies, we use a Hawkes-style interaction term[11] that allows both excitation and inhibition:

$$h_k(\mathcal{H}_t) = \sum_{t_j < t} \sum_{k'=1}^{K} A_{kk'} \, \kappa_\theta(t - t_j) \, \pi_{k'}(\boldsymbol{m}_j), \tag{9}$$

where $A_{kk'} \in \mathbb{R}$ can be positive (triggering) or negative (inhibition), $\pi_{k'}(\boldsymbol{m}_j)$ is the soft assignment of event j to type k' (from Eq. 7), and $\kappa_\theta(\Delta t)$ is a nonnegative, learnable temporal kernel. We implement κ_θ as a mixture of exponentials,

$$\kappa_\theta(\Delta t) = \sum_{r=1}^{R} \omega_r \exp(-\beta_r \Delta t), \qquad \omega_r \geq 0, \ \beta_r > 0, \ \Delta t > 0, \tag{10}$$

which is expressive, stable, and admits efficient incremental evaluation.

Mark distribution. We further model the mark conditioned on history:

$$p(\boldsymbol{m}_i \mid \mathcal{H}_{t_i}) = \mathcal{N}\big(\boldsymbol{\mu}_\psi(\mathcal{H}_{t_i}), \mathrm{diag}(\boldsymbol{\sigma}^2_\psi(\mathcal{H}_{t_i}))\big), \tag{11}$$

where $(\boldsymbol{\mu}_\psi, \boldsymbol{\sigma}_\psi)$ are produced by a history encoder $\mathcal{G}_\psi$ that summarizes $\mathcal{H}_{t_i}$ (e.g., a small causal transformer over past events). In practice, Eq. 11 serves as a regularizer that encourages temporally consistent marks; when strong supervision for types/marks is available, we optionally replace it with a discriminative loss.

3.6 Temporal Query Executor

Given a query q, we produce a compact structured evidence vector $\boldsymbol{z} = \mathcal{Q}(q, \lambda, \mathcal{E})$ by selecting an operator family and executing it on the learned process.[7,24,44] We use a lightweight query parser $\mathcal{P}$ (a small LM head or classifier) to map q to an operator $\omega \in \Omega$ and its arguments (event types, time ranges, etc.). The executor then computes operator outputs using intensities and/or sampled events from the process.

Count and rate operators. For questions of the form "how many times does type k occur?", we use the expected count:

$$z_{\mathrm{count}}(k; [t_a, t_b]) = \mathbb{E}[N_k([t_a, t_b])] = \int_{t_a}^{t_b} \lambda_k(t \mid \mathcal{H}_t) \, dt, \tag{12}$$

and the average rate $z_{\mathrm{rate}}(k; [t_a, t_b]) = z_{\mathrm{count}}(k; [t_a, t_b])/(t_b - t_a)$. In practice, we approximate the integral by Monte Carlo quadrature with J time samples:

$$\int_{t_a}^{t_b} \lambda_k(t) \, dt \approx \frac{t_b - t_a}{J} \sum_{j=1}^{J} \lambda_k(\tilde{t}_j), \qquad \tilde{t}_j \sim \mathrm{Unif}(t_a, t_b). \tag{13}$$

Order operator. For questions such as "did k_1 happen before k_2?", we estimate an order probability via process sampling[26]. Let $t_k^{(s)}$ be the first-arrival time of type k in sample s. We compute

$$z_{\text{order}}(k_1, k_2) = \mathbb{P}(t_{k_1} < t_{k_2}) \approx \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}\Big[t_{k_1}^{(s)} < t_{k_2}^{(s)}\Big]. \tag{14}$$

Interval and duration operators. For intervals, we estimate the expected waiting time from k_1 to k_2 by sampling paired arrivals (or conditioning on observed proposals):

$$z_{\text{interval}}(k_1, k_2) = \mathbb{E}[t_{k_2} - t_{k_1} \mid t_{k_2} > t_{k_1}], \tag{15}$$

computed using samples that satisfy the condition. For duration queries, we treat a state as an entry type k_{in} and an exit type k_{out} and measure

$$z_{\text{dur}}(k_{\text{in}}, k_{\text{out}}) = \mathbb{E}[t_{\text{out}} - t_{\text{in}}], \tag{16}$$

again estimated by sampling matched entry/exit events or by pairing nearest feasible proposals.

Structured evidence for answering. The resulting evidence $\boldsymbol{z}$ typically consists of a few scalars or small vectors (counts, probabilities, time estimates). We serialize $\boldsymbol{z}$ into a compact textual schema (or a fixed embedding) and condition a language model to produce the final answer:

$$p(a \mid \boldsymbol{x}, q) = p_{\text{LM}}(a \mid q, \text{Serialize}(\boldsymbol{z})). \tag{17}$$

This keeps the context footprint small and makes temporal reasoning explicit and inspectable.

3.7 Learning Objective

CTEPM is trained end-to-end with a combination of task supervision and point-process likelihood[11,7]. Let $\lambda(t) = \sum_{k=1}^{K} \lambda_k(t \mid \mathcal{H}_t)$. The marked point process log-likelihood is

$$\log p(\mathcal{E}) = \sum_{i=1}^{N} \log \lambda_{k_i}(t_i \mid \mathcal{H}_{t_i}) - \int_0^T \lambda(t \mid \mathcal{H}_t)\, dt + \sum_{i=1}^{N} \log p(\boldsymbol{m}_i \mid \mathcal{H}_{t_i}), \tag{18}$$

yielding the negative log-likelihood loss $\mathcal{L}_{pp} = -\log p(\mathcal{E})$. For question answering, we use standard token-level cross-entropy $\mathcal{L}_{qa} = -\log p_{\text{LM}}(a^\star \mid q, \boldsymbol{z})$. We also include mild proposal regularization to prevent degenerate dense events, e.g., an ℓ_1 sparsity penalty on $\{s_t\}$ and a soft repulsion term that discourages extremely close event times. The final objective is

$$\mathcal{L} = \mathcal{L}_{qa} + \alpha \mathcal{L}_{pp} + \beta \mathcal{L}_{prop}. \tag{19}$$

Discussion. Unlike token pruning or clip retrieval, CTEPM constructs a continuous-time probabilistic memory that supports explicit temporal operators. Unlike event-graph pipelines that discretize time and then rely on an LLM to infer temporal relations implicitly, our operator-based execution computes counts, rates, order probabilities, and temporal intervals directly from the learned intensities, enabling robust long-video reasoning under tight context budgets.

4 Experiments

4.1 Experimental Setup

We evaluate CTEPM as a plug-in module on a strong open-source Video-LMM backbone (**LLaVA-NeXT-Video-7B** [42]) and additionally verify robustness on **VideoChat2** [19]. We benchmark long-video understanding on **LongVideoBench**, **Video-MME** (w/ subtitles) [9], and **MLVU** [43], following the official splits and evaluation scripts. Baselines include **Uniform Sampling**, **Segment Retrieval** (top-k segments under the same frame cap), **FastVID** (training-free token pruning) [29], and an **MA-LMM-style** [12] streaming memory bank, all under matched input budgets. For LongVideoBench and Video-MME we report Top-1 accuracy; for MLVU we report the official overall score (and key sub-task accuracy where applicable).

Budgets and hyperparameters. Unless noted, all methods use a fixed *frame budget* of F=64 sampled frames as the maximum input to the base Video-LMM. CTEPM additionally maintains a separate *event budget* and proposes N=32 events per video to form the continuous-time memory (i.e., event-level reasoning cost is controlled by N and is decoupled from F). We use a mixture-of-exponentials kernel with R=8 components, Monte Carlo quadrature with J=64, and S=32 samples for order/interval estimates. We ablate key hyperparameters (including N and R) in the supplementary material (Sec. C).

We train only lightweight adapters (LoRA) [13] with AdamW [22] while keeping the backbone frozen.

Complete experimental settings are provided in the supplementary material Sec. A.

4.2 Main Results

LongVideoBench. Table 1 compares methods under the same frame budget [36]. CTEPM improves over all baselines, with the largest gains on very long videos (900s–3600s), while simply using more frames provides only marginal benefits.

Video-MME. As shown in Table 2, CTEPM consistently boosts the overall score and performs better on long videos (30–60 min) under the same input cap.

MLVU. Table 3 shows that CTEPM improves M-Avg, with particularly clear gains on temporally compositional subsets (Action Order/Count), whereas retrieval alone yields modest improvements.

Table 1. Main results on LongVideoBench (Top-1 Acc., %). All methods use the same Video-LMM backbone and the same maximum input budget (64 frames unless noted).

Method	Frames	Total	8–15s	15–60s	900–3600s
Backbone: LLaVA-NeXT-Video-7B					
Uniform Sampling (ours)	64	46.0	54.0	56.5	41.0
More-Frames (ours)	128	47.1	54.6	57.2	42.0
Segment Retrieval (ours)	64	46.8	54.3	56.9	41.8
FastVID (ours)	64	46.5	54.2	56.7	41.3
CTEPM (ours)	64	**49.2**	**56.4**	**59.0**	**45.6**
Reference models (reported) for context					
LLaVA-Next-Video-M7B	32	43.5	50.9	53.1	38.9
VideoChat2 (Mistral-7B)	16	41.2	49.3	49.3	37.5

Table 2. Main results on Video-MME (Top-1 Acc., %). "w/ subs" follows the official setting that includes subtitles.

Method	Frames	Overall (w/ subs)	Short	Long
Backbone: LLaVA-NeXT-Video-7B				
Uniform Sampling (ours)	64	52.5	62.8	46.5
Segment Retrieval (ours)	64	52.9	63.0	47.1
FastVID (ours)	64	52.7	62.9	46.8
CTEPM (ours)	64	**55.0**	**64.2**	**49.9**
Reference models (reported) for context				
Long-LLaVA	64	57.1	66.2	50.3
VideoChat2-Mistral	16	43.8	52.8	39.2
Video-LLaVA	8	41.6	46.1	38.1

Analysis. Across three long-video benchmarks, CTEPM improves the backbone baselines under the same input budget. The improvements are largest on the longest-duration subsets (LongVideoBench 900s–3600s; Video-MME long videos) and on temporally compositional tasks (MLVU AO/AC), which aligns with our goal of making time *computable* via a continuous-time event process memory and operator execution. We further analyze the gains by operator type and event budget in supplementary material Sec. B.

4.3 Accuracy–Budget Trade-off

Protocol. We vary the maximum sampled frames $F \in \{16, 32, 64, 128\}$ while keeping the backbone, prompts, and decoding fixed (temperature = 0). All methods are constrained to the same frame cap F on LongVideoBench, Video-MME (w/ subtitles), and MLVU using LLaVA-NeXT-Video-7B.

Table 3. Main results on MLVU. We report M-Avg (higher is better) and two temporally compositional sub-tasks: Action Order (AO) and Action Count (AC).

Method	Input	M-Avg	AO	AC
Backbone: LLaVA-NeXT-Video-7B				
Uniform Sampling (ours)	64 frm	34.0	16.0	21.0
Segment Retrieval (ours)	64 frm	34.6	16.6	22.0
FastVID (ours)	64 frm	34.3	16.2	21.6
CTEPM (ours)	64 frm	**37.8**	**21.5**	**27.8**
Reference models (reported) for context				
MA-LMM	1000 frm	22.4	13.3	18.6
VideoChat2-HD	16 frm	37.4	21.4	23.3
VideoLLaMA2	16 frm	48.4	42.9	16.7

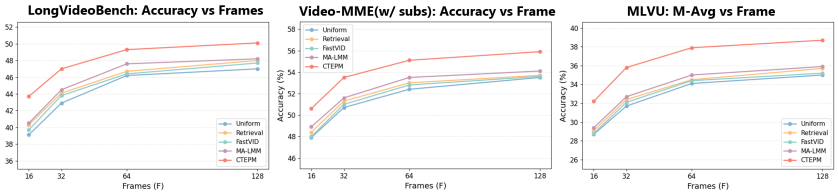

Fig. 3. Accuracy–budget trade-off by varying the maximum sampled frames F on LongVideoBench, Video-MME (w/ subtitles), and MLVU, comparing uniform sampling, retrieval, FastVID, MA-LMM-style memory, and CTEPM under matched frame caps.

Baselines. *Uniform* uses uniform sampling. *Retrieval* retrieves top-k segments such that the total frames match F. *FastVID* prunes visual tokens given the same F frames. *MA-LMM-style* maintains a streaming memory bank under the same F. **CTEPM** builds a continuous-time event process memory from the same F frames and answers via temporal operators.

Analysis. Figure 3 shows consistent improvements from CTEPM across all budgets, with the largest margins at low-to-medium budgets. On LongVideoBench, CTEPM improves over Uniform by +4.6 (at F=16) and +3.1 points (at F=64). On Video-MME and MLVU, CTEPM similarly stays above retrieval, token-pruning, and memory-bank baselines under identical frame caps. Overall, the curves indicate that CTEPM is more sample-efficient and that its gains come from process-level temporal computation rather than increased input length.

4.4 Temporal-Operator Subset Evaluation

MVBench operator-aligned tasks. We evaluate operator-aligned temporal reasoning on MVBench by grouping tasks into **Count** (AC, MC), **Order** (AS, AA,

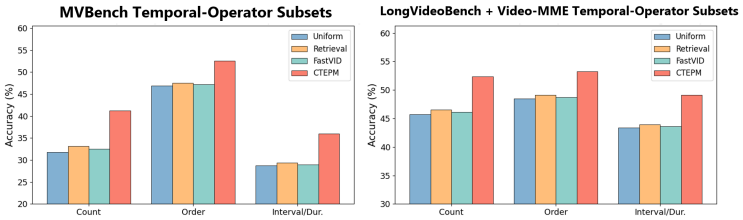

Fig. 4. Operator-aligned subset results on MVBench and on LongVideoBench+Video-MME, comparing uniform sampling, retrieval, FastVID, and CTEPM under a matched input budget.

CO), and **Interval/Duration** (AL, localization-style) [16,38]. All methods use the same backbone, official prompts, and a matched 16-frame budget.

Operator-labeled subsets on LongVideoBench and Video-MME. Since long-video benchmarks mix question types, we further build operator-labeled subsets on LongVideoBench and Video-MME [36,9]. We assign each question to **Count/Order/Interval/Duration** via a high-precision keyword parser and validate with a 200-question spot-check per benchmark, then report per-category accuracy under the same backbone and frame budget.

Analysis. Figure 4 visualizes the operator-aligned results. Across both MVBench and the long-video operator subsets, CTEPM delivers consistent gains over uniform sampling, retrieval, and token-pruning baselines, with the largest improvements on **Count** and **Interval/Duration**. On MVBench, counting and localization-style queries are particularly challenging for generic Video-LMMs, and CTEPM yields a clear margin (about 9–10 points on Count and ∼7 points on Interval), matching our design of explicitly computing expected counts and time windows from a continuous-time event process. On LongVideoBench and Video-MME, the same trend persists under more realistic mixed question distributions: CTEPM improves Count and Interval/Duration without increasing the frame budget, suggesting that the gains stem from *computable temporal structure* rather than heavier sampling or retrieval. Improvements on **Order** are smaller but stable, indicating that many order questions can be partially resolved from local evidence, while CTEPM provides additional benefit when ordering requires aggregating sparse events over long horizons.

4.5 Comparison with MA-LMM

Setup. We compare CTEPM against MA-LMM[12], a representative memory-augmented Video-LMM designed for hour-level videos via online processing and a memory bank. To ensure a fair comparison on long-video QA, we implement an *MA-LMM-style memory baseline* on top of the same backbone used in our main experiments (identical prompting, identical decoding, and identical maximum

Table 4. Comparison with MA-LMM-style memory on LongVideoBench (Top-1 Acc., %). All methods use the same backbone and prompts with a matched 64-frame cap. The > 30 min subset corresponds to videos longer than 1800s.

Method (matched budget)	Frames	Overall	> 30 min
Uniform Sampling (ours)	64	46.0	40.8
Segment Retrieval (ours)	64	46.8	41.6
MA-LMM-style Memory (ours)	64	47.5	42.4
CTEPM (ours)	64	**49.2**	**45.9**

Table 5. Comparison with MA-LMM on MLVU. "Reported" denotes the published MA-LMM result on MLVU test set under a larger input budget. Our runs use the same 64-frame cap as in the main experiments.

Method	Input	M-Avg	Action Order (AO)	Action Count (AC)	G-Avg
MA-LMM	1000 frm	22.4	13.3	18.6	3.83
Uniform Sampling (ours)	64 frm	34.0	16.0	21.0	3.90
MA-LMM-style Memory (ours)	64 frm	34.9	17.1	22.0	3.92
CTEPM (ours)	64 frm	**37.8**	**21.5**	**27.8**	**4.05**

input budget whenever applicable). Since MA-LMM is evaluated in the literature with substantially larger effective frame budgets, we report two settings: (i) **Matched-budget** evaluation where both methods are constrained to the same maximum number of sampled frames/clips; (ii) **Reported** MA-LMM results from the MLVU benchmark for context.

LongVideoBench (matched budget). Table 4 shows results on LongVideoBench under a matched 64-frame budget. CTEPM consistently improves over the MA-LMM-style memory baseline on the overall score and yields larger gains on the *very-long* subset (> 30 minutes), where memory-bank retrieval alone is often insufficient for queries requiring explicit aggregation (e.g., repeated events) and temporal operators. Compared with MA-LMM, which stores and retrieves past information, CTEPM additionally constructs a *continuous-time* event process and executes count/order/interval operators, leading to more robust performance on hour-long referring reasoning.

MLVU (reported vs matched-budget re-evaluation). MLVU includes MA-LMM as a representative long-video model and reports its performance under a large input budget. Table 5 summarizes: (i) the *reported* MA-LMM result on MLVU test set (1000-frame setting), and (ii) our matched-budget evaluations (64 frames) on the same benchmark. Despite using far fewer frames, CTEPM achieves substantially higher multiple-choice average (M-Avg) and improves on temporally compositional tasks, highlighting that explicit process-level temporal computation can be more effective than scaling input length alone.

Analysis. Overall, MA-LMM-style memory improves over uniform sampling by enabling long-term information retention under a fixed context, but its gains saturate on very long videos where many questions require explicit temporal computations rather than recalling a single relevant segment. CTEPM complements memory with a continuous-time event process and an operator interface, yielding larger improvements on > 30 min videos and on temporally compositional subsets (e.g., order and count). These results suggest that, for long-video QA, *process-level temporal modeling* can be more sample-efficient than simply increasing the number of input frames or relying solely on memory retrieval.

5 Conclusion

In this work, we introduced **CTEPM**, a continuous-time event process memory for long-video language models that makes temporal reasoning *computable* rather than implicit. CTEPM represents a video as a marked temporal point process with learnable intensities and semantic marks, capturing both global temporal trends and cross-event triggering/inhibition. By executing a small library of temporal operators (count/rate/order/interval/duration) on the learned process, CTEPM produces compact structured evidence that conditions answer generation without performing video generation or relying on heavy long-context prompting. Empirically, CTEPM improves performance on long-video understanding benchmarks, with particularly strong gains on temporally compositional queries under tight input budgets, while remaining competitive on standard video QA.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. vol. 35, pp. 23716–23736 (2022)
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6836–6846 (2021)
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Icml. vol. 2, p. 4 (2021)
4. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster (2022)
5. Daley, D.J., Vere-Jones, D.: An introduction to the theory of point processes: volume I: elementary theory and methods. Springer (2003)
6. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. vol. 35, pp. 16344–16359 (2022)
7. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1555–1564 (2016)

8. Fan, Z., Chen, K., Xing, R., Li, Y., Jiang, L., Tian, Z.: Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging. arXiv preprint arXiv:2602.08024 (2026)
9. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis pp. 24108–24118 (2025)
10. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18995–19012 (2022)
11. Hawkes, A.G.: Spectra of some self-exciting and mutually exciting point processes. Biometrika **58**(1), 83–90 (1971)
12. He, B., Li, H., Jang, Y.K., Jia, M., Cao, X., Shah, A., Shrivastava, A., Lim, S.N.: Ma-lmm: Memory-augmented large multimodal model for long-term video understanding pp. 13504–13514 (2024)
13. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022)
14. Huang, X., Zhou, H., Han, K.: Prunevid: Visual token pruning for efficient video large language models pp. 19959–19973 (2025)
15. Kool, W., Van Hoof, H., Welling, M.: Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In: International conference on machine learning. pp. 3499–3508. PMLR (2019)
16. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: Proceedings of the IEEE international conference on computer vision. pp. 706–715 (2017)
17. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
18. Li, F., Zhang, R., Zhang, H., Zhang, Y., Li, B., Li, W., Ma, Z., Li, C.: Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. arXiv preprint arXiv:2407.07895 (2024)
19. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: Chat-centric video understanding. Science China Information Sciences **68**(10), 200102 (2025)
20. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: Mvbench: A comprehensive multi-modal video understanding benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22195–22206 (2024)
21. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3202–3211 (2022)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
23. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. Advances in Neural Information Processing Systems **36**, 46212–46244 (2023)
24. Mei, H., Eisner, J.M.: The neural hawkes process: A neurally self-modulating multivariate point process. vol. 30 (2017)
25. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million nar-

rated video clips. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2630–2640 (2019)
26. Ogata, Y.: On lewis' simulation method for point processes. IEEE transactions on information theory **27**(1), 23–31 (1981)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
28. Shao, K., Tao, K., Qin, C., You, H., Sui, Y., Wang, H.: Holitom: Holistic token merging for fast video large language models. arXiv preprint arXiv:2505.21334 (2025)
29. Shen, L., Gong, G., He, T., Zhang, Y., Liu, P., Zhao, S., Ding, G.: Fastvid: Dynamic density pruning for fast video large language models. arXiv preprint arXiv:2503.11187 (2025)
30. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. vol. 33, pp. 7537–7547 (2020)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. vol. 30 (2017)
32. Wang, Y., Li, K., Li, X., Yu, J., He, Y., Chen, G., Pei, B., Zheng, R., Wang, Z., Shi, Y., et al.: Internvideo2: Scaling foundation models for multimodal video understanding pp. 396–416 (2024)
33. Wang, Y., Li, X., Yan, Z., He, Y., Yu, J., Zeng, X., Wang, C., Ma, C., Huang, H., Gao, J., et al.: Internvideo2. 5: Empowering video mllms with long and rich context modeling. arXiv preprint arXiv:2501.12386 (2025)
34. Wang, Y., Zhang, L., Liu, J., Yan, J., Zhang, Z., Zheng, J., Yang, X., Wu, D., Chen, X., Li, X.: Episodic memory representation for long-form video understanding. arXiv preprint arXiv:2508.09486 (2025)
35. Wen, S., Wang, Z., Zhang, X., Huang, L., Wu, W.: Eventmemagent: Hierarchical event-centric memory for online video understanding with adaptive tool use. arXiv preprint arXiv:2602.15329 (2026)
36. Wu, H., Li, D., Chen, B., Li, J.: Longvideobench: A benchmark for long-context interleaved video-language understanding. Advances in Neural Information Processing Systems **37**, 28828–28857 (2024)
37. Xie, S.M., Ermon, S.: Reparameterizable subset sampling via continuous relaxations (2019)
38. Yang, A., Nagrani, A., Seo, P.H., Miech, A., Pont-Tuset, J., Laptev, I., Sivic, J., Schmid, C.: Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10714–10726 (2023)
39. Ye, W., Wu, Q., Lin, W., Zhou, Y.: Fit and prune: Fast and training-free visual token pruning for multi-modal large language models **39**(21), 22128–22136 (2025)
40. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding pp. 543–553 (2023)
41. Zhang, P., Zhang, K., Li, B., Zeng, G., Yang, J., Zhang, Y., Wang, Z., Tan, H., Li, C., Liu, Z.: Long context transfer from language to vision. arXiv preprint arXiv:2406.16852 (2024)
42. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Llava-video: Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713 (2024)

43. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., et al.: Mlvu: Benchmarking multi-task long video understanding pp. 13691–13701 (2025)
44. Zuo, S., Jiang, H., Li, Z., Zhao, T., Zha, H.: Transformer hawkes process. In: International conference on machine learning. pp. 11692–11702. PMLR (2020)