STRIDE: When to Speak Meets Sequence Denoising for Streaming Video Understanding

Junho Kim[1*], Hosu Lee[2*], James M. Rehg[1],
Minsu Kim[3†‡], and Yong Man Ro[2†]

[1] University of Illinois Urbana-Champaign
[2] Korea Advanced Institute of Science and Technology
[3] Google DeepMind

Abstract. Recent progress in video large language models (Video-LLMs) has enabled strong offline reasoning over long and complex videos. However, real-world deployments increasingly require streaming perception and proactive interaction, where video frames arrive online and the system must decide not only what to respond, but also when to respond. In this work, we revisit proactive activation in streaming video as a structured sequence modeling problem, motivated by the observation that temporal transitions in streaming video naturally form span-structured activation patterns. To capture this span-level structure, we model activation signals jointly over a sliding temporal window and update them iteratively as new frames arrive. We propose *STRIDE* (Structured Temporal Refinement with Iterative DEnoising), which employs a lightweight masked diffusion module at the activation interface to jointly predict and progressively refine activation signals across the window. Extensive experiments on diverse streaming benchmarks and downstream models demonstrate that *STRIDE* shows more reliable and temporally coherent proactive responses, significantly improving "when-to-speak" decision quality in online streaming scenarios. The *STRIDE* codebase and demonstrations are available on our project webpage.

Keywords: Proactive Video Streaming · Activation Modeling · Masked Diffusion Model

1 Introduction

Along with recent advances in large language models (LLMs) [7, 41, 45, 55, 66], large vision-language models (LVLMs) [12, 16, 25, 33, 34] have also achieved impressive performance across a wide range of image understanding and reasoning tasks. Building upon these advances, various video specialized models (*i.e.,* Video-LLMs) [22, 29, 30, 75, 78] further extend them to the temporal sequences, demonstrating remarkable capabilities in reasoning over video contents. However, existing Video-LLMs mostly operate in an offline manner, processing pre-recorded videos with access to the entire temporal context before generating responses.

* Equal contribution. †Corresponding author. ‡Work done as an advisory role only.

This fundamentally limits their capabilities to real-world streaming deployments such as egocentric assistants [21], autonomous driving [63], or embodied AI agents [62], where the model must continuously perceive an ongoing video stream and decide *when* and *what* to respond in real time.

Recognizing this gap, recent works have delved into streaming video understanding (SVU), where models continuously ingest incoming frames and maintain a temporal understanding on-the-fly [39, 61, 68, 69, 76, 77]. Despite these advances, the approach is still *reactive*, lacking a capability to determine when a response should be triggered. Expanding beyond the streaming scope, several works have explored *proactive* response generation by leveraging special tokens [9, 10, 65] to implicitly learn response timing or an agent-driven interaction approach [64, 67]. More recently, several standalone activation modules [43, 44, 56] have been proposed, especially those that decouple the streaming pipeline into two stages: a lightweight front-end that predicts activation signals at each frame to identify triggering moments, followed by a downstream Video-LLM that, when activated, consumes the accumulated frame cache to generate responses.

Within this decomposed framework, a straightforward way to train the activation module is to treat it as a binary classification problem as in [43, 44, 56], where at each time step a model predicts whether to trigger a response under binary supervision. However, such approach reduces activation to point-wise `0/1` decisions, answering "*should I respond now?*" at each time step, without explicitly modeling how activation states transition across a temporal span. This often results in flickering activations and poorly resolved transition boundaries, causing unstable triggering behavior and fragmented activation spans. In practice, a reliable activation module must not only predict isolated labels, but also model how activation states change over time, capturing consistent `0`→`1` onsets, sustained `1`→`1` persistence, and well-resolved `1`→`0` offsets, so as to form coherent contiguous activation spans. In this sense, streaming and proactive triggering is more analogous to a span-structured decision rather than a point-wise one. To account for this span-level structure, an activation module should jointly model the activation sequence within a temporal neighborhood, so that the downstream Video-LLM can be activated under well-scoped visual context (neither prematurely with insufficient evidence nor too late after the moment has passed).

Motivated by recent advances in masked diffusion models [27, 38, 72] (MDMs), which enable joint prediction over partially masked discrete sequences, we revisit streaming and proactive activation as *structured sequence modeling* over an activation window. Unlike point-wise decision-making, masked diffusion operates on an entire sequence and iteratively refines corrupted states within context, naturally aligning with the span-structure of streaming trigger. Building on this, we propose *STRIDE* (*S*tructured *T*emporal *R*efinement with *I*terative *DE*noising), a proactive streaming framework that models the when-to-speak decision as structured sequence prediction, explicitly capturing span-level structure and activation state transitions. Specifically, during training, we employ boundary-aware span masking strategies that corrupt contiguous regions of the activation sequence, encouraging the model to reason about onset and offset from broader temporal

context rather than relying on isolated binary signals. At inference time, as new frames arrive, *STRIDE* progressively updates the activation window by carrying forward confident states and remasking uncertain positions, enabling temporally coherent span under partial observability while remaining plug-and-play and compatible with off-the-shelf Video-LLMs.

Through extensive experiments and comprehensive analyses on streaming benchmarks and downstream models, we corroborate that *STRIDE* produces more reliable and temporally coherent proactive responses in online settings, significantly improving the *when-to-speak* decisions.

Our contributions can be summarized as follows:

- We revisit proactive streaming activation in Video-LLMs and reformulate the when-to-speak problem as structured sequence modeling over a temporal activation window, establishing span-level activation as the prediction unit.
- We propose *STRIDE* (*S*tructured *T*emporal *R*efinement with *I*terative *DE*noising), a lightweight masked diffusion-based activation model that jointly predicts activation sequences and captures span-level structure.
- We validate *STRIDE* through extensive experiments on diverse streaming benchmarks and downstream backbones, demonstrating more stable proactive triggering and improved temporal consistency in online settings.

2 Related Work

2.1 Large Vision-Language Models

Early works on LVLMs [16, 23, 34] have demonstrated that visual instruction tuning, which pairs a vision encoder with a LLM backbone and trains on instruction-following data, can output strong general-purpose capabilities for back-and-forth multi-modal conversation. Subsequent efforts [12, 59, 60, 83] have focused on scaling model and data, improving visual tokenization, and aligning vision and language representations at scale. Especially, Qwen families [3–5, 58] improve visual processing efficiency and capability with dynamic resolution and stronger multi-modal pretraining, enabling more robust perception and reasoning over complex visual inputs. In addition, Video-LLMs [26, 53, 75, 78] extend its scope to temporal understanding by treating video as a sequence of images, introducing video-specific connector [22, 30, 74] and training pipelines [24, 48, 79] that better capture spatiotemporal dynamics, thereby leading to stronger performance on video QA and captioning tasks. Despite these advances, most LVLMs remain confined to an *offline* setting, where the entire video clip is available prior to inference, limiting their applicability in real-time streaming scenarios.

2.2 Streaming Video Understanding

A growing body of works [28, 44, 80] has explored expanding video understanding into the *streaming* regime, where frames arrive online and frameworks must maintain state over time. One line of research adapts models to streaming interaction

by redesigning training objectives and data formats for continuous inputs [9], incorporating memory-augmented architectures for multi-turn streaming [64, 76], and leveraging real-time commentary pipelines that integrate video speech transcripts with instruction tuning [10, 65]. Another branch emphasizes efficiency for unbounded video streams through memory aggregation for long streams [76], streaming-aligned KV-cache strategies [39, 65, 68], and redundant visual token dropping based on inter-frame similarity [69]. While these approaches have enabled Video-LLMs to process continuous streams, they remain fundamentally *reactive*, generating responses only upon instantaneous user queries.

Addressing this gap, another direction tackles the *proactive response*, which targets deciding *when to respond* as the video unfolds. Several approaches exploit `EOS` token within autoregressive generation to implicitly determine response timing [9,65], conflating the triggering with language generation. Agentic methods explicitly model task-relevant temporal intervals for goal-driven triggering [67], or query-aware visual pruning with proactive response mechanisms [77]. Most relevant to our work, recent modular approaches [43, 44, 56, 61] explicitly decouple the pipeline into a lightweight front-end that predicts per-frame binary activation signals and a downstream Video-LLM that generates responses upon triggering. While such a modular design preserves the downstream Video-LLM's capabilities, reducing activation to point-wise binary supervision undermines the temporal coherence of contiguous activation spans. In this work, we retain the modular design while recasting activation as a *structured sequence prediction* problem, leveraging masked diffusion to jointly model activation sequences over a temporal window and capture span-level temporal coherence.

2.3 Discrete Diffusion Language Models

Recent progress in discrete diffusion language models (dLLMs) [37, 38, 47] revisits diffusion as an alternative to autoregressive decoding for text generation using masked diffusion models mechanism. Instead of generating tokens strictly left-to-right, dLLMs iteratively denoise masked token sequences, enabling bidirectional conditioning and parallel token updates, which naturally supports controllable generation. Subsequent efforts have further scaled dLLMs by converting pretrained autoregressive models into diffusion-based counterparts [17, 71], and improved their alignment and inference efficiency through parallel decoding strategies [13]. Especially, LLaDA series scale masked diffusion to large LLMs [38] and further explores post-training alignment [82] as well as system-level scaling by converting pretrained AR models into diffusion models [6], thereby inheriting knowledge while retaining the non-autoregressive generation benefits. This research scope has also been extended to the multi-modal setting, where vision encoders are coupled with diffusion language backbones for visual instruction following [14, 27, 72, 73], demonstrating that dLLMs can benefit from parallel decoding and bidirectional reasoning in vision-language tasks. Different from these works that primarily replace the autoregressive decoder for textual response generation, our work leverages masked diffusion for proactive streaming activation. We treat the *when-*

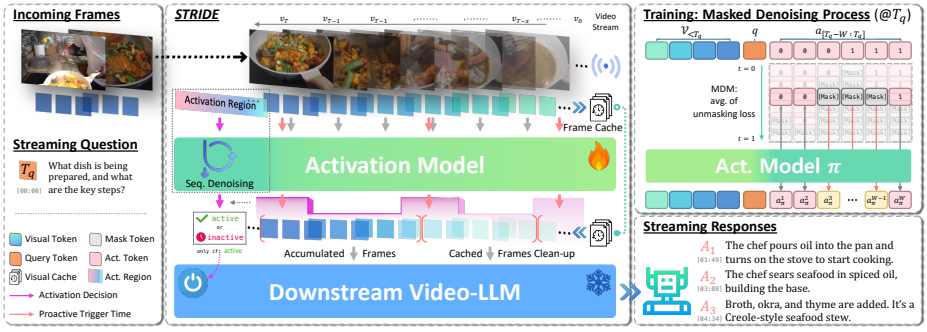


Fig. 1: Overview of ***STRIDE***, which operates in a streaming setting where frames arrive online. A lightweight activation model based on masked diffusion maintains an activation region over a sliding temporal window and iteratively denoises masked activation states to predict a coherent trigger segment. A trigger is issued only if an active span is sustained for a predefined span ratio. When activation is triggered, the accumulated frame context is forwarded to a downstream Video-LLM to generate the response.

to-speak signal as a structured discrete activation sequence over a temporal window, jointly predicting the activation states for the incoming video streams.

3 Proposed Method

Preliminaries: Masked Diffusion Models. Recently, diffusion language models (dLLMs) [27, 38, 72, 82] have shown remarkable progress as an alternative paradigm to autoregressive language modeling, replacing left-to-right token generation with a masked diffusion process that iteratively denoises discrete token sequences. Given a sequence of L tokens $\mathbf{x}_0 = (x_0^1, \ldots, x_0^L)$, the forward process progressively corrupts $\mathbf{x}_0$ by independently replacing each token with a mask token `[M]` with probability $t \in [0, 1]$, generating a partially masked sequence $\mathbf{x}_t$. At $t = 0$ the sequence is fully observed, while at $t = 1$ it is entirely masked.

The core of MDMs is a *mask predictor* $p_\theta(\cdot \mid \mathbf{x}_t)$ with bidirectional attention that takes $\mathbf{x}_t$ as input and predicts all masked tokens simultaneously. The reverse process [2, 47, 50] recovers $\mathbf{x}_0$ from $\mathbf{x}_t$ by iteratively applying this mask predictor, which is trained by minimizing a cross-entropy loss computed only over the masked positions:

$$\mathcal{L}(\theta) = -\mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t}\left[\frac{1}{t}\sum_{i=1}^{L}\mathbb{1}[x_t^i = \mathtt{M}]\log p_\theta(x_0^i \mid \mathbf{x}_t)\right], \tag{1}$$

where $t \sim U[0, 1]$ and $\mathbf{x}_t$ is sampled from the forward process. This serves as an upper bound on the negative log-likelihood of the model distribution [6, 38].

At inference, generation proceeds by initializing a fully masked sequence $\mathbf{x}_1$ and simulating the reverse process through K discrete steps decreasing from

$t = 1 \rightarrow 0$. At each step, the mask predictor predicts all masked positions, and a subset of predictions is accepted while the remaining positions are remasked for subsequent refinement. This iterative predict-and-refine procedure enables MDMs to generate coherent sequences through progressive unmasking with bidirectional context.

3.1 STRIDE: Proactive Streaming Framework

Problem Formulation. The proposed *STRIDE* (shown in Fig. 1) considers the streaming video understanding setting where a model continuously processes video streams $\mathcal{V}=\{v_1, v_2, \ldots, v_T, \ldots\}$ with v_T denoting the incoming visual frame arriving at time step T, interleaved with user queries and model-generated responses over time. Unlike offline Video-LLMs that have access to the holistic video sequences before generating a response, a streaming model must work under partial observability, where only the frames observed so far $\mathcal{V}_{\leq T}=\{v_1, \ldots, v_T\}$ and context priors $\mathcal{C}_T$ (*e.g.,* user query q and prior interaction history) are available. At every time step T, the model faces two sequential decisions: (i) *whether* to respond, and (ii) if so, *what* to respond. *STRIDE* adopts a two-stage streaming framework to decouple these decisions.

Two-Stage Architecture. As illustrated in Fig. 1, *STRIDE* is designed with the two-stage streaming framework. A lightweight *Activation Model* π continuously monitors the incoming stream and determines whether a proactive response should be triggered. Once a response is triggered at time step T, the accumulated visual context since the most recent query time T_q, denoted $\mathcal{V}_{[T_q:T]}$, together with the interaction context $\mathcal{C}_T$, is forwarded to a downstream Video-LLM, which generates the response $\mathcal{R}_T = f(\mathcal{C}_T, \mathcal{V}_{[T_q:T]})$. The generated response $\mathcal{R}_T$ is appended to the interaction context, updating it to $\mathcal{C}_{T'} = \mathcal{C}_T \cup \mathcal{R}_T$, enabling awareness of prior responses and maintaining dialogue coherence across multiple activation events. After each triggered response, the visual accumulation is cleared and restarted from the current time step, ensuring that subsequent activation decisions operate on fresh streaming context. This modular design cleanly separates *when-to-speak* modeling from downstream response generation.

Span-Level Activation Modeling. To formalize the activation decision, we represent activation as a window-level sequence of size W anchored at time step T, and model it as a sequence-level prediction over this temporal window. Specifically, we define an activation region $\boldsymbol{a}_T = [a_{T-W}, \ldots, a_T] \in \{\texttt{0}, \texttt{1}\}^W$, indicating inactive or active states within the window. This windowed formulation enables the activation model to learn contiguous activation spans and their transition dynamics ($\texttt{0}\rightarrow\texttt{1}$ onset, $\texttt{1}\rightarrow\texttt{1}$ persistence, $\texttt{1}\rightarrow\texttt{0}$ offset), aligning the prediction unit with span-level structures rather than isolated point-wise decisions.

As the video stream unfolds, incoming frames are sampled at 1 FPS and encoded into visual tokens by a vision encoder, which are accumulated in a running visual cache. At each time step T, the activation region $\boldsymbol{a}_T$ is appended

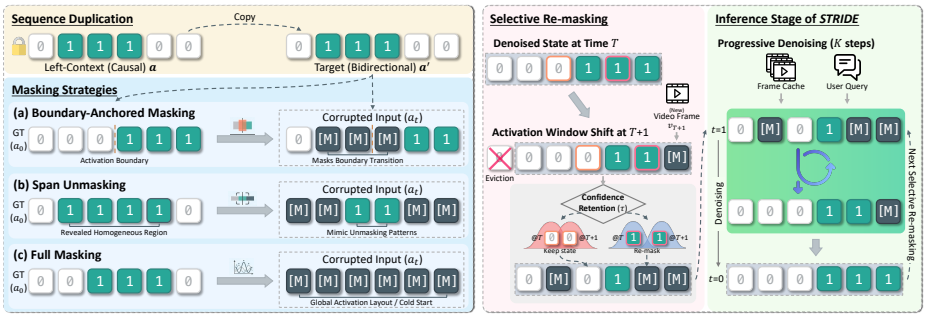


Fig. 2: Activation modeling and inference stage of ***STRIDE***. Training applies sequence duplication and three masking strategies (boundary-anchored masking, span unmasking, full masking). During inference, the activation window slides with incoming frames, retaining confident past decisions while selectively re-masking and progressively denoising uncertain positions.

after the visual cache as the prediction target. Each activation token takes values from the discrete vocabulary $\{0, 1, \texttt{[M]}\}$, where `[M]` denotes masked positions to be denoised. The activation model conditions on the visual cache and jointly infers masked activation states within the temporal window. When the activation state is determined to be active under the span-based criterion, the accumulated visual context is forwarded to the downstream Video-LLM for response generation.

3.2 Training: Activation as Sequence Denoising

Structured Masking Strategies for Activation Denoising. To train the activation model under the structured formulation, we propose a mixture of three corruption strategies instead of the standard MDM [38], which samples mask positions independently. Such masking is inappropriate for our activation learning as the target sequence consists of contiguous active regions; isolated unmasked tokens between active positions make the denoising task trivially solvable through local interpolation, bypassing the need for genuine temporal understanding. The proposed masking mixture shown in Fig. 2 (left) is composed of:

- **Boundary-Anchored Span Masking** masks a contiguous block overlapping with at least one activation boundary, forcing the model to determine where the active region begins and ends from broader temporal context.
- **Span Unmasking** starts from a fully masked sequence and reveals a contiguous block while keeping boundary-adjacent positions masked, mimicking the inference-time pattern where high-confidence tokens are unmasked consecutively in homogeneous regions.
- **Full Masking** initially masks the entire activation sequence (cold-start) to stabilize the reverse step by training the model to estimate the global activation layout from visual context alone.

During training, each sample is randomly corrupted using one of three structured masking strategies, each selected with equal probability. These structured strategies encourage the model to reason over contiguous activation spans and their boundary transitions, rather than relying on isolated token predictions. As a result, the activation module learns span-level consistency that better aligns with the sequential and partial observability of streaming proactive triggering.

Recovering Bidirectional Conditioning with Sequence Duplication. Masked diffusion predicts masked positions using full-sequence context, whereas an AR-pretrained activation model is trained with causal attention that only exposes left context. We therefore introduce an input reparameterization that enables bidirectional conditioning without altering the underlying causal attention layers. Specifically, we employ *sequence duplication*, appending a copy of the activation region to form $[\boldsymbol{a},\ \boldsymbol{a}']$, where the copies carry identical activation tokens but serve distinct roles. The duplicated sequence $\boldsymbol{a}'$ produces diffusion predictions, while $\boldsymbol{a}$ serves as a conditioning prefix under causal attention. Concretely, since $\boldsymbol{a}$ is entirely placed before $\boldsymbol{a}'$, every token in $\boldsymbol{a}'$ can access all positions of $\boldsymbol{a}$ as left-context, providing full-window visibility for denoising without modifying the causal attention mask.

Training Objective. Following the denoising process in Eq. (1), we train the activation module by minimizing the masked cross-entropy loss over $\boldsymbol{a}'$, conditioned on the user query q and the visual cache $\mathcal{V}_{\leq T}$:

$$\mathcal{L}(\theta) = -\mathbb{E}_{t,\,\boldsymbol{a}'_0,\,\boldsymbol{a}'_t}\left[\frac{1}{t}\sum_{j=1}^{W}\mathbb{1}[a'^{j}_t = \mathtt{M}]\log p_\theta(a'^{j}_0 \mid q,\,\mathcal{V}_{\leq t},\,\boldsymbol{a}'_t)\right], \tag{2}$$

where $\boldsymbol{a}'_0$ is the ground-truth activation sequence, $\boldsymbol{a}'_t$ is obtained by applying our aforementioned masking strategies at noise level $t \sim U[0,1]$, and the user query q along with the visual cache $\mathcal{V}_{\leq t}$ serves as a fixed conditioning prefix, analogous to the prompt in the supervised fine-tuning of dLLMs [27, 38].

3.3 Inference: Streaming as Progressive Unmasking

At inference time, *STRIDE* maintains a sliding activation window and performs progressive refinement as illustrated in Fig. 2 (right); confident past decisions are preserved, while uncertain and newly introduced positions are jointly refined with masked diffusion. Concretely, new time step $T{+}1$ proceeds in two stages:
(i) Selective Re-masking: The activation sequence of size W is shifted forward so that the region falling outside the window is evicted and a new frame is appended, causing the activation sequence to advance in time. The fully resolved activation a_T^{j+1} previously assigned to position $j{+}1$ at time T now maps to position j at time $T{+}1$. To determine whether each carried-forward decision remains reasonable given the new visual evidence v_{T+1}, we apply a confidence-based retention: if

$p_\theta(a^j_{T+1} = a^{j+1}_T \mid q, \mathcal{V}_{\leq T+1}, \boldsymbol{a}_{T+1}) > \tau$, position j inherits its previous decision; otherwise, it is re-masked to `[M]` so that uncertain positions re-enter the denoising process alongside the newly appended slots.
(ii) K-Step Progressive Denoising: The masked positions obtained from the previous stage, comprising both newly appended slots and low-confidence re-masked slots, are resolved over K denoising steps by prioritizing high-confidence positions first. At each step, the model computes the activation probability $p^j = p_\theta(a^j = 1 \mid q, \mathcal{V}_{\leq T+1}, \boldsymbol{a}_{T+1})$ for every masked position and derives a confidence score $c^j = \max(p^j, 1 - p^j)$, which measures how strongly the prediction leans toward either triggering or not. The top-k positions ranked by c^j are unmasked, where $k = \lceil N_{\text{init}}/K \rceil$ and N_{init} is the number of masked positions established in stage (i), while the rest remain masked for subsequent refinement. By revealing high-confidence decisions first, this schedule establishes reliable temporal anchors that progressively stabilize the remaining ambiguous boundary regions.

After K steps, the activation window is fully resolved. A trigger at time $T+1$ is issued only if an active span is sustained for at least γ consecutive positions, where γ denotes the required span ratio for triggering.

4 Experiments

4.1 Experimental Setting

Implementation & Training Details. The activation model is initialized from a compact vision-language model using Qwen3-VL-2B [4] to minimize streaming overhead. The downstream Video-LLMs are kept frozen, ensuring full modularity between the two stages. Incoming video frames are sampled at 1 FPS and encoded into the visual cache as the stream progresses. For the denoising process, we adopt the low-confidence remasking strategy [38] with $K=8$ sampling steps during inference. τ is set to 0.75, and γ is set to 1 following the benchmark evaluation protocol. The entire activation model is trained on a single node of 8 NVIDIA H100 GPUs, while evaluation is conducted on a single H100 GPU. Comprehensive hyperparameter settings and additional configurations are provided in the Appendix.

For the training data, we curate a diverse collection of temporally annotated video datasets spanning multiple video understanding tasks, including dense video captioning [8,36], temporal activity detection [51,52], grounded video QA [20,57], sequential step recognition [81], and moment localization [1]. We convert each temporal annotation into a binary activation sequence aligned with the frame sampling rate, where frames within annotated spans are labeled as active (`1`) and the remaining frames as inactive (`0`).

Benchmarks & Baselines. We evaluate *STRIDE* on three complementary benchmarks. OVO-Bench [40] assesses online video understanding across backward tracing, real-time visual perception, and forward active responding, where the model must delay its response until sufficient future evidence is available.

Table 1: Evaluation results on OVO-Bench [40]. Baseline-AR uses autoregressive binary prediction. Offline models follow the original single-turn protocol with segmented clips, whereas streaming methods process frames sequentially.

Method	# of Frames	Real-Time Visual Perception							Backward Tracing				Forward Act. Responding				Overall
		OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	Avg.
Human																	
Human	-	93.96	92.57	94.83	92.70	91.09	94.02	93.20	92.59	93.02	91.37	92.33	95.48	89.67	93.56	92.90	92.81
Proprietary Models (Offline), Single-Turn Evaluation																	
Gemini 1.5 Pro [45]	1 FPS	85.91	66.97	79.31	58.43	63.37	61.96	**69.32**	58.59	76.35	52.64	**62.54**	35.53	74.24	61.67	**57.15**	**63.00**
GPT-4o [42]	64	69.80	64.22	71.55	51.12	70.30	59.78	64.46	57.91	75.68	48.66	60.75	27.58	73.21	59.40	53.40	59.54
Open-Source Models (Offline), Single-Turn Evaluation																	
LLaVA-Video-7B [79]	64	69.13	58.72	68.83	49.44	74.26	59.78	63.52	56.23	57.43	7.53	40.40	34.10	69.95	60.42	54.82	52.91
LLaVA-OV-7B [23]	64	66.44	57.80	73.28	53.37	71.29	61.96	64.02	54.21	55.41	21.51	43.71	25.64	67.09	58.75	50.50	52.74
LLaVA-N-Video-7B [24]	64	69.80	59.60	66.40	50.60	72.30	61.40	63.30	51.20	64.20	9.70	41.70	34.10	67.60	60.80	54.20	53.10
Qwen2-VL-7B [58]	64	60.40	50.46	56.03	47.19	66.34	55.43	55.98	47.81	35.48	56.08	46.46	31.66	65.82	48.75	48.74	50.39
InternVL-V2-8B [11]	64	67.11	60.55	63.79	46.07	68.32	56.52	60.39	48.15	57.43	24.73	43.44	26.50	59.14	54.14	46.60	50.15
LongVU-7B [49]	1 FPS	55.70	49.50	59.50	48.30	68.30	63.00	57.40	43.10	66.20	9.10	39.50	16.60	69.00	60.00	48.50	48.50
Open-Source Models (Streaming)																	
Flash-VStream-7B [76]	1 FPS	24.16	29.36	28.45	33.71	25.74	28.80	28.37	39.06	37.16	5.91	27.38	8.02	67.25	60.00	45.09	33.61
VideoLLM-Online-8B [9]	2 FPS	8.05	23.85	12.07	14.04	45.54	21.20	20.79	22.22	18.80	12.18	17.73	-	-	-	-	-
VideoLLM-EyeWO [80]	1 FPS	24.16	27.52	31.89	32.58	44.55	35.87	32.76	39.06	38.51	6.45	28.00	-	-	-	-	-
Dispider [43]	1 FPS	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.30	36.06	18.05	37.36	48.75	34.72	41.78
TimeChat-Online-7B [69]	1 FPS	69.80	48.60	64.70	44.90	68.30	55.40	58.60	53.90	62.80	9.10	42.00	32.50	36.50	40.00	36.40	45.60
StreamAgent-7B [67]	1 FPS	71.20	53.20	63.60	53.90	67.30	58.70	61.30	54.80	58.10	25.80	41.70	35.90	48.40	52.00	45.40	49.40
QueryStream-7B [77]	1 FPS	74.50	47.70	70.70	46.60	71.30	57.60	61.40	54.20	63.50	8.60	42.10	33.20	43.10	40.80	39.03	47.51
Offline Backbones → Online Inference with STRIDE																	
Qwen3-VL-8B [4]	1 FPS	69.80	59.60	73.30	57.30	71.30	58.70	65.00	55.60	63.50	12.90	44.00	37.70	60.80	40.40	46.30	51.77
+ Baseline-AR [56]	1 FPS	73.80	65.10	73.30	62.40	70.30	71.20	69.35	54.90	66.90	17.20	46.33	29.70	56.00	42.50	42.73	52.81
Gemma3-4B [54]	1 FPS	65.80	48.60	56.00	36.00	66.30	50.00	53.78	44.40	41.90	3.20	29.83	14.40	61.40	52.50	42.77	42.13
+ ***STRIDE***	1 FPS	73.20	60.60	64.70	39.30	71.30	56.50	60.93	47.80	52.00	4.80	34.87	42.60	64.60	60.00	55.73	50.51
InternVL3-8B [60]	1 FPS	65.80	52.30	68.10	51.10	71.30	62.00	61.77	58.90	66.90	9.70	45.17	36.60	64.10	43.30	48.00	51.64
+ ***STRIDE***	1 FPS	75.80	54.10	80.20	56.70	74.30	65.20	67.72	58.90	65.50	11.30	45.23	40.10	67.70	66.20	58.00	56.98
Qwen3-VL-8B [4]	1 FPS	69.80	59.60	73.30	57.30	71.30	58.70	65.00	55.60	63.50	12.90	44.00	37.70	60.80	40.40	46.30	51.77
+ ***STRIDE***	1 FPS	76.50	64.20	79.30	61.20	73.30	63.60	**69.68**	57.20	72.30	14.00	**47.83**	46.40	63.10	69.60	**59.70**	**59.07**

StreamingBench [32] evaluates streaming comprehension through 18 tasks spanning real-time visual understanding, omni-source understanding, and contextual understanding including proactive output and sequential question answering. In addition, we evaluate on subsets of ET-Bench [36], including Temporal Video Grounding (TVG), Episodic Memory (EPM), Temporal Action Localization (TAL), Dense Video Captioning (DVC), and Step Localization and Captioning (SLC). This setup evaluates activation timing precision by measuring how accurately the model identifies event boundaries. For baselines, we compare against various offline Video-LLMs, online streaming proactive models [9, 43, 56], and proprietary models [42,45]. In addition, we include **Baseline-AR**, which serves as the primary counterpart to *STRIDE*. Baseline-AR follows the same architecture and training setup as our method but replaces the masked diffusion activation module with an autoregressive binary prediction head trained with BCE loss, following the activation formulation described in prior work [56]. This setup isolates the activation modeling strategy, enabling a direct comparison between masked denoising and autoregressive binary prediction.

4.2 Qualitative Results on Streaming Video Understanding

Tabs. 1 and 2 present results on OVO-Bench and StreamingBench, respectively. The proposed *STRIDE* outperforms the autoregressive baseline (*i.e.*, Baseline-AR) [56] by introducing the proposed masked denoising process. Furthermore, across all three downstream models [4, 54, 60] on OVO-Bench, *STRIDE* achieves significant gains in Forward Active Responding, which directly evaluates proactive when-to-speak control. This setting benefits from our span-structured predic-

Table 2: Evaluation results on StreamingBench [32]. Baseline-AR uses autoregressive binary prediction. Offline models follow the original single-turn protocol with segmented clips, whereas streaming methods process frames sequentially.

Method	# of Frames	Real-Time Visual Understanding											Omni-Source Understanding					Contextual Understanding					Overall
		OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Avg.	ER	SCU	SD	MA	Avg.	ACU	MCU	SQA	PO	Avg.	Avg.
Human																							
Human	-	89.47	92.00	93.60	91.47	95.65	92.52	88.00	88.75	89.74	91.30	91.46	88.00	88.24	93.60	90.27	90.26	88.80	90.40	95.00	100	93.55	91.66
Proprietary Models (Offline)																							
Gemini 1.5 pro [45]	1 FPS	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	**75.69**	46.80	39.60	74.90	80.00	**60.22**	51.41	40.73	54.80	45.10	**48.73**	**67.07**
GPT-4o [42]	64	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28	41.20	37.20	43.60	56.00	44.50	41.20	38.40	32.80	56.86	38.70	60.15
Open-Source Models (Offline)																							
LLaVA-OV-7B [23]	32	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12	40.80	37.20	33.60	44.80	38.40	35.60	36.00	27.27	29.55	32.74	56.36
Qwen2-VL-7B [58]	1 FPS	75.20	82.81	73.19	77.45	68.32	71.03	72.22	61.19	61.47	46.11	69.04	41.20	22.00	32.80	43.60	34.90	31.20	26.00	39.60	22.73	31.66	54.14
MiniCPM-V 2.6 8B [70]	32	71.93	71.09	77.92	75.82	64.60	65.73	70.37	56.10	62.32	53.37	67.44	40.80	24.00	34.00	41.20	35.00	34.00	31.60	41.92	22.22	34.97	53.85
InternVL-V2-8B [11]	16	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72	37.60	26.40	37.20	42.00	35.80	32.00	31.20	32.32	40.91	32.42	51.40
Kangaroo-7B [35]	64	71.12	84.38	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60	37.60	31.20	28.80	39.20	34.20	32.80	26.40	33.84	16.00	30.06	51.10
LongVA-7B [78]	128	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96	39.60	32.40	28.00	41.60	35.40	32.80	29.60	30.30	15.91	29.95	48.66
VILA-1.5-8B [31]	14	53.68	49.22	70.98	56.86	53.42	53.89	54.63	48.78	50.14	17.62	52.32	41.60	26.40	28.40	36.00	33.10	26.80	34.00	23.23	17.65	27.35	43.20
Video-LLaMA2-7B [15]	32	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52	30.40	32.40	30.40	36.00	32.40	24.80	26.80	18.67	0.00	21.93	40.40
Open-Source Models (Streaming)																							
Flash-VStream-7B [76]	1 FPS	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23	25.91	24.90	25.60	28.40	26.00	24.80	25.20	26.80	1.96	24.12	24.04
VideoLLM-Online-8B [9]	2 FPS	39.07	40.06	34.49	31.05	45.54	32.40	31.48	34.16	42.49	27.89	35.99	31.20	26.51	24.10	32.00	28.45	24.19	29.20	30.80	3.92	26.55	32.48
Dispider [43]	1 FPS	74.92	75.53	74.10	73.08	74.44	59.92	76.14	62.91	62.16	45.80	67.63	35.46	25.26	38.57	43.34	35.66	39.62	27.65	34.80	25.34	33.61	53.12
StreamAgent [67]	1 FPS	79.63	78.31	79.28	75.87	74.74	76.92	82.94	66.31	73.69	55.40	74.31	35.86	26.26	38.87	44.04	36.26	39.72	30.25	39.60	28.90	34.62	57.02
TimeChat-Online-7B [69]	1 FPS	80.80	79.70	80.80	83.30	74.80	78.80	78.70	64.20	68.80	58.00	**75.28**	-	-	-	-	-	-	-	-	-	-	-
QueryStream-7B [77]	1 FPS	82.11	83.59	78.23	82.69	75.47	80.06	79.63	63.01	67.90	42.55	74.04	-	-	-	-	-	-	-	-	-	-	-
Offline Backbones → Online Inference with STRIDE																							
Qwen3-VL-8B [4]	1 FPS	62.70	68.00	69.70	53.30	67.50	65.10	67.60	48.00	68.00	40.90	60.88	36.80	18.00	32.80	34.00	30.40	25.60	23.60	31.20	32.40	28.20	46.84
+ Baseline-AR [56]	1 FPS	79.00	72.70	85.50	70.80	73.30	76.60	81.50	63.40	80.70	44.60	73.79	45.20	29.20	35.20	35.20	36.20	35.60	37.20	48.40	24.30	36.38	57.12
Gemma3-4B [54]	1 FPS	63.80	69.50	68.80	54.40	60.30	65.10	59.30	40.70	62.70	21.80	57.49	28.80	31.20	36.40	46.40	34.20	34.40	31.20	38.80	12.40	29.20	46.03
+ ***STRIDE***	1 FPS	66.80	71.90	66.60	57.20	66.50	70.70	60.20	43.10	65.00	23.80	60.00	35.60	31.60	36.00	44.00	36.80	33.60	35.20	44.80	41.60	38.80	50.14
InternVL3-8B [60]	1 FPS	74.90	82.00	75.70	61.20	72.00	67.60	74.10	66.70	78.10	34.20	68.71	40.40	27.60	38.80	45.60	38.10	38.00	26.00	36.80	31.20	33.00	53.97
+ ***STRIDE***	1 FPS	74.90	78.90	76.70	68.60	77.00	77.30	77.80	71.50	83.00	33.20	72.45	39.60	22.40	44.00	50.80	39.20	34.00	35.20	43.20	42.80	38.80	57.58
Qwen3-VL-8B [4]	1 FPS	62.70	68.00	69.70	53.30	67.50	65.10	67.60	48.00	68.00	40.90	60.88	36.80	18.00	32.80	34.00	30.40	25.60	23.60	31.20	32.40	28.20	46.84
+ ***STRIDE***	1 FPS	77.10	75.00	77.30	72.80	76.40	77.90	76.90	69.90	84.30	46.10	74.24	42.80	24.00	45.20	53.20	**41.30**	32.00	38.40	46.40	42.80	**39.90**	**59.29**

tion, which models response timing over temporal activation region rather than through independent per-frame decisions. *STRIDE* also consistently improves Real-Time Visual Perception, indicating that stable triggering allows the downstream Video-LLM to ingest well-scoped visual context at the appropriate moment. On StreamingBench, this advantage extends broadly across all three evaluation dimensions: Real-time Visual Understanding, Omni-Source Understanding, and Contextual Understanding, with the most notable improvements in Proactive Output (PO) subtask that requires the model to determine response timing without explicit timing cue. Together, these results suggest that the proposed framework reliably enhances both the precision of when to respond and the relevance of responses across diverse streaming conditions.

4.3 Activation Evaluation via Temporal Grounding

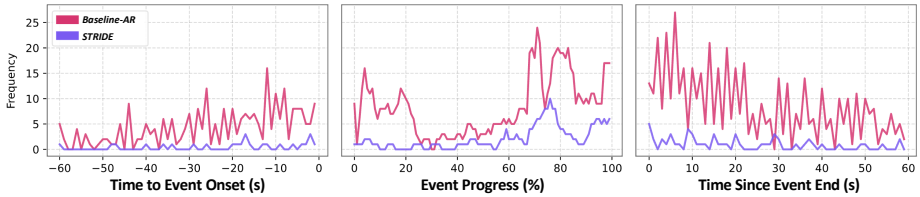
Accurate temporal grounding is central to proactive activation. While the previous benchmarks [32, 40] evaluate the end-to-end behavior of streaming pipelines, they do not directly measure the quality of the activation model itself. To isolate this component, we evaluate the activation model independently on ET-Bench [36], which focuses on fine-grained event-level temporal understanding. As shown in Tab. 3, the gain from replacing binary classification with masked diffusion is substantial: *STRIDE* outperforms Baseline-AR by 27.1 on TVG and 8.3 on average, demonstrating that structured sequence denoising provides considerably sharper boundary resolution than per-frame supervision. Notably, *STRIDE* achieves these results with only 2B parameters, outperforming both streaming baselines and temporal-localization specialized MLLMs of standard size (7–13B parameters) on the overall average.

Table 3: Online activation accuracy on ET-Bench [36]. The Baseline-AR uses autoregressive prediction, while other setups are the same as *STRIDE*.

	Frames	Params	TVG_{F1}	EPM_{F1}	TAL_{F1}	DVC_{F1}	SLC_{F1}	Avg
Temporal-Localization Specialized MLLMs								
VTimeLLM [19]	100	7B	7.6	1.9	18.2	12.4	8.7	9.8
VTG-LLM [18]	96	7B	15.9	3.7	14.4	40.2	20.8	19.0
TimeChat [46]	96	7B	26.2	3.9	10.1	16.6	5.6	12.5
LITA [20]	100	13B	22.2	4.6	18.0	39.7	21.0	21.1
ETChat [36]	1 FPS	5B	38.6	10.2	30.8	38.4	24.4	28.5
Streaming Baselines								
VideoLLM-Online [9]	2 FPS	8B	13.2	3.8	9.1	24.0	9.9	12.0
Dispider [43]	1 FPS	9B	36.1	**15.5**	**27.3**	33.8	18.8	26.3
StreamBridge [56]	1 FPS	8B	34.3	–	24.3	38.3	22.6	–
Baseline-AR [56]	1 FPS	2B	35.7	2.5	21.2	**39.6**	22.6	24.3
STRIDE	1 FPS	2B	**62.8**	10.7	24.6	36.5	**28.5**	**32.6**

Table 4: Ablation studies on ET-Bench evaluating (a) masking strategy design, (b) sequence duplication, and (c) selective re-masking.

	TVG_{F1}	EPM_{F1}	TAL_{F1}	DVC_{F1}	SLC_{F1}	Avg
(a) Masking Strategy						
Indep. only	8.5	3.3	6.1	8.8	9.2	7.2
Span only	30.6	6.1	22.9	25.4	20.6	21.1
Span + Full	36.8	7.0	**26.0**	24.0	21.3	23.0
Span + Full + $\overline{\text{Span}}$	**62.8**	**10.7**	24.6	**36.5**	**28.5**	**32.6**
(b) Sequence Duplication						
w/o Seq. Duplication	49.6	6.0	23.6	19.9	15.2	22.9
w/ Seq. Duplication	**62.8**	**10.7**	**24.6**	**36.5**	**28.5**	**32.6**
(c) Selective Re-masking						
w/o Re-masking (last-only)	39.5	2.5	19.1	30.7	21.2	22.6
w/ Re-masking (selective)	**62.8**	**10.7**	**24.6**	**36.5**	**28.5**	**32.6**



(a) Pre-Event (b) During Event (c) Post-Event

Fig. 3: Activation transition frequency results around event boundaries on ET-Bench TVG. Baseline-AR model shows frequent oscillations near boundaries, whereas *STRIDE* produces more robust activation spans.

4.4 Ablation Studies on STRIDE Components

Effects of Masking Strategies. Tab. 4(a) evaluates how different masking strategies affect the learning of span-structured activation sequences. The standard MDM protocol independent masking performs the worst across all metrics, indicating that activation prediction cannot be treated as independent pointwise denoising since it fails to capture the temporal structure of activation transitions. To better reflect the span-level nature of activation, we adopt three complementary masking patterns described in Sec. 3.2: boundary-anchored span masking (Span), full masking (Full), and span unmasking ($\overline{\text{Span}}$). Span masks contiguous regions near activation boundaries, Full masks the entire sequence to simulate the cold-start condition, and $\overline{\text{Span}}$ exposes boundary refinement patterns encountered during denoising. Combining these strategies significantly improves performance, suggesting that diverse span corruption patterns help the model learn coherent activation spans for better boundary prediction.

Effect of Sequence Duplication. Masked diffusion relies on contextual reasoning over the activation window, whereas the pretrained backbone used in *STRIDE* follows causal attention and therefore only exposes left context during prediction. This mismatch limits the model's ability to jointly infer activation states within the window. To mitigate this, we apply *sequence duplication*, which

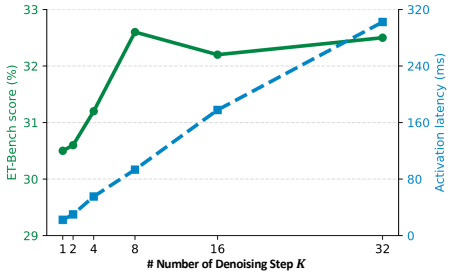


Fig. 4: Trade-off between ET-Bench performance (mean F1) and inference latency for denoising step K.

Table 5: Latency and VRAM usage of the downstream Video-LLM and *STRIDE* activation modules (with AR variation) during streaming inference.

Procedure	Latency (ms)	VRAM
Downstream Video-LLM		
Response Gen. (TTFT)	1276	17.8 GB
Response Gen. (TTLT)	1511	+ 13 MB
STRIDE		
Activation Sate (Base)		5.2 GB
+ 1 Denoising Step	12	+ 10 MB
+ Append Frame	20	+ 30 MB
Baseline-AR		
Activation Sate (Base)		5.2 GB
+ Append Frame	26	+ 30 MB

provides full-window context to the prediction tokens while preserving the causal backbone. As shown in Tab. 4(b), removing sequence duplication leads to a consistent performance drop across all tasks, reducing the average score from 32.6 to 22.9. The degradation is particularly notable in temporally sensitive tasks such as TVG and DVC, indicating that accurate boundary reasoning benefits from access to the full activation window. These results demonstrate that sequence duplication effectively recovers bidirectional context for diffusion-based refinement, enabling full-window conditioning through a simple input reparameterization without modifying the causal architecture.

Effect of Selective Re-masking. In the streaming setting, activation predictions are carried forward as the window advances. If these states are preserved without revision, early mistakes can propagate and gradually corrupt the activation sequence. To examine this effect, we compare our selective re-masking strategy with a simplified variant that masks only the newly appended position (*last-only*), leaving previous decisions fixed. As shown in Tab. 4(c), restricting re-masking to the last position leads to a substantial performance drop, reducing the average score from 32.6 to 22.6. As only predicting last token falls into autoregressive prediction, the resulting performance is also similar to *Baseline-AR* in Tab. 3. In contrast, selectively re-masking low-confidence positions allows the model to revise uncertain decisions as new frames arrive, enabling refinement of the activation sequence by using the updated context information.

4.5 Behavioral Analysis of STRIDE Properties

Flickering Analysis around Event Boundaries. While ET-Bench quantifies activation accuracy, it is also limited to capture the temporal stability of activation decisions. In particular, per-frame activation models may suffer from flickering behavior due to their inherently isolated predictions, where predictions rapidly oscillate between active and inactive states ($0 \leftrightarrow 1$), resulting in unstable triggering and poorly resolved transition boundaries. To analyze this phenomenon, we

measure the frequency of activation transitions relative to event boundaries. Specifically, we align predictions around ground-truth events and accumulate transition counts within three regions (Fig. 3): (a) pre-event (-60s to onset), (b) during-event (normalized by event progress %), and (c) post-event (offset to +60s), using the TVG task of ET-Bench where each instance corresponds to a single event.

As in the figure, across all regions, *Baseline-AR* exhibits substantially higher transition frequency, indicating unstable activation sequences with frequent on/off oscillations. This instability becomes particularly striking near event boundaries, where transition frequency sharply increases, suggesting difficulty in resolving the precise onset and offset of events. In contrast, *STRIDE* produces significantly smoother activation patterns with far fewer transitions. The reduced flickering indicates that modeling activation as structured sequence denoising encourages temporally coherent predictions, allowing the model to maintain consistent activation spans and more reliably capture event boundaries.

Latency–Accuracy Trade-off for Denoising Steps. We analyze the effect of the denoising step K on both activation model accuracy and inference latency as illustrated in Fig. 4. This trade-off is particularly important in the streaming setting, where the activation model operates online and directly determines the model's response latency. Increasing K allows the model to perform more refinement steps, improving activation accuracy but also increasing inference time. In practice, we observe that performance saturates quickly: around $K = 8$ steps already achieves near-maximum mean F1 across ET-Bench subtasks. This behavior likely stems from the small output space of the activation sequence, where each position only takes binary states (0 or 1), making the denoising process relatively easier to converge than large vocabulary space. At $K = 8$, the inference latency is approximately $\sim$100 ms, which is practical enough to support real-time operation for streaming frame rates of downstream models.

Streaming Efficiency and Memory Footprint. Extending the latency-accuracy trade-off analysis in Fig. 4, we decompose the computational overhead introduced by the activation model under a streaming setup. The measurement is conducted on a single H100 GPU with a 128-frame context budget. As shown in Tab. 5, when a subsequent response is required, the 113 ms (new frame and $K=8$ denoising steps) added by *STRIDE* incurs only a 7% additional latency compared to the 1511 ms required by the base model Qwen3VL-8B [4] without the triggering module. When a trigger is unnecessary, *STRIDE* saves approximately 91% of the total processing time (113 ms vs. 1276 ms). Furthermore, compared to the per-frame decision baseline (Baseline-AR, 26 ms), the extra latency from the diffusion process (113 ms) is limited to 87 ms. In terms of memory, *STRIDE* maintains a lightweight footprint of 5.2 GB. Executing denoising process requires an additional 10 MB, and each new frame introduces 30 MB of incremental memory usage. These highlight the advantage of the two-stage design: a lightweight activation model gates the expensive downstream model. Even with the masked diffusion

module employed in STRIDE, trigger modeling introduces only minimal latency and memory overhead, maintaining efficient streaming inference.

5 Conclusion

We present *STRIDE*, a framework for proactive streaming video understanding that models activation as a structured temporal sequence rather than independent per-frame decisions. By leveraging a lightweight masked diffusion module to jointly refine activation signals over a sliding window, *STRIDE* captures span-level temporal structure and produces more stable and coherent triggering behavior in streaming settings. Extensive experiments and analyses show that jointly modeling activation over a temporal window significantly improves event boundary localization and reduces unstable triggering, while introducing only minimal overhead to the overall streaming pipeline.

6 Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2022-NR070162).

References

1. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2017)
2. Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R.: Structured denoising diffusion models in discrete state-spaces. Advances in neural information processing systems **34**, 17981–17993 (2021)
3. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 **1**(2), 3 (2023)
4. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
5. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
6. Bie, T., Cao, M., Chen, K., Du, L., Gong, M., Gong, Z., Gu, Y., Hu, J., Huang, Z., Lan, Z., et al.: Llada2. 0: Scaling up diffusion language models to 100b. arXiv preprint arXiv:2512.15745 (2025)

7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
8. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 961–970 (2015)
9. Chen, J., Lv, Z., Wu, S., Lin, K.Q., Song, C., Gao, D., Liu, J.W., Gao, Z., Mao, D., Shou, M.Z.: Videollm-online: Online video large language model for streaming video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18407–18418 (2024)
10. Chen, J., Zeng, Z., Lin, Y., Li, W., Ma, Z., Shou, M.Z.: Livecc: Learning video llm with streaming speech transcription at scale. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29083–29095 (2025)
11. Chen, Z., Wang, W., Tian, H., Ye, S., Gao, Z., Cui, E., Tong, W., Hu, K., Luo, J., Ma, Z., et al.: How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. Science China Information Sciences **67**(12), 220101 (2024)
12. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Muyan, Z., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)
13. Chen, Z., Fang, G., Ma, X., Yu, R., Wang, X.: dparallel: Learnable parallel decoding for dllms. arXiv preprint arXiv:2509.26488 (2025)
14. Cheng, S., Jiang, Y., Zhou, Z., Liu, D., Tao, W., Zhang, L., Qi, B., Zhou, B.: Sdar-vl: Stable and efficient block-wise diffusion for vision-language understanding. arXiv preprint arXiv:2512.14068 (2025)
15. Cheng, Z., Leng, S., Zhang, H., Xin, Y., Li, X., Chen, G., Zhu, Y., Zhang, W., Luo, Z., Zhao, D., et al.: Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. arXiv preprint arXiv:2406.07476 (2024)
16. Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In: Advances in Neural Information Processing Systems (2023)
17. Gong, S., Agarwal, S., Zhang, Y., Ye, J., Zheng, L., Li, M., An, C., Zhao, P., Bi, W., Han, J., et al.: Scaling diffusion language models via adaptation from autoregressive models. arXiv preprint arXiv:2410.17891 (2024)
18. Guo, Y., Liu, J., Li, M., Cheng, D., Tang, X., Sui, D., Liu, Q., Chen, X., Zhao, K.: Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 3302–3310 (2025)
19. Huang, B., Wang, X., Chen, H., Song, Z., Zhu, W.: Vtimellm: Empower llm to grasp video moments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14271–14280 (2024)
20. Huang, D.A., Liao, S., Radhakrishnan, S., Yin, H., Molchanov, P., Yu, Z., Kautz, J.: Lita: Language instructed temporal-localization assistant. In: European Conference on Computer Vision. pp. 202–218. Springer (2024)
21. Huang, Y., Xu, J., Pei, B., He, Y., Chen, G., Yang, L., Chen, X., Wang, Y., Nie, Z., Liu, J., et al.: Vinci: A real-time embodied smart assistant based on egocentric vision-language model. arXiv preprint arXiv:2412.21080 (2024)
22. Kim, J., Kim, H., Lee, H., Ro, Y.M.: Salova: Segment-augmented long video assistant for targeted retrieval and routing in long-form video analysis. arXiv preprint arXiv:2411.16173 (2024)

23. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Li, Y., Liu, Z., Li, C.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
24. 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)
25. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International Conference on Machine Learning. PMLR (2023)
26. 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)
27. Li, S., Kallidromitis, K., Bansal, H., Gokul, A., Kato, Y., Kozuka, K., Kuen, J., Lin, Z., Chang, K.W., Grover, A.: Lavida: A large diffusion language model for multimodal understanding. arXiv preprint arXiv:2505.16839 (2025)
28. Li, W., Hu, B., Shao, R., Shen, L., Nie, L.: Lion-fs: Fast & slow video-language thinker as online video assistant. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 3240–3251 (2025)
29. Li, Y., Wang, C., Jia, J.: Llama-vid: An image is worth 2 tokens in large language models. In: European Conference on Computer Vision. pp. 323–340. Springer (2025)
30. Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122 (2023)
31. Lin, J., Yin, H., Ping, W., Molchanov, P., Shoeybi, M., Han, S.: Vila: On pre-training for visual language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26689–26699 (2024)
32. Lin, J., Fang, Z., Chen, C., Wan, Z., Luo, F., Li, P., Liu, Y., Sun, M.: Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. arXiv preprint arXiv:2411.03628 (2024)
33. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023)
34. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Advances in Neural Information Processing Systems (2023)
35. Liu, J., Wang, Y., Ma, H., Wu, X., Ma, X., Wei, X., Jiao, J., Wu, E., Hu, J.: Kangaroo: A powerful video-language model supporting long-context video input. arXiv preprint arXiv:2408.15542 (2024)
36. Liu, Y., Ma, Z., Qi, Z., Wu, Y., Shan, Y., Chen, C.W.: Et bench: Towards open-ended event-level video-language understanding. Advances in Neural Information Processing Systems **37**, 32076–32110 (2024)
37. Lou, A., Meng, C., Ermon, S.: Discrete diffusion modeling by estimating the ratios of the data distribution. arXiv preprint arXiv:2310.16834 (2023)
38. Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.R., Li, C.: Large language diffusion models. arXiv preprint arXiv:2502.09992 (2025)
39. Ning, Z., Liu, G., Jin, Q., Ding, W., Guo, M., Zhao, J.: Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. arXiv preprint arXiv:2505.15269 (2025)
40. Niu, J., Li, Y., Miao, Z., Ge, C., Zhou, Y., He, Q., Dong, X., Duan, H., Ding, S., Qian, R., et al.: Ovo-bench: How far is your video-llms from real-world online video understanding? In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 18902–18913 (2025)

41. OpenAI: ChatGPT. https://openai.com/blog/chatgpt/ (2022)
42. OpenAI: Hello gpt-4o (2024), https://openai.com/index/hello-gpt-4o/
43. Qian, R., Ding, S., Dong, X., Zhang, P., Zang, Y., Cao, Y., Lin, D., Wang, J.: Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24045–24055 (2025)
44. Qian, R., Dong, X., Zhang, P., Zang, Y., Ding, S., Lin, D., Wang, J.: Streaming long video understanding with large language models. Advances in Neural Information Processing Systems **37**, 119336–119360 (2024)
45. Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
46. Ren, S., Yao, L., Li, S., Sun, X., Hou, L.: Timechat: A time-sensitive multimodal large language model for long video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14313–14323 (2024)
47. Sahoo, S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J., Rush, A., Kuleshov, V.: Simple and effective masked diffusion language models. Advances in Neural Information Processing Systems **37**, 130136–130184 (2024)
48. Share: Sharegemini: Scaling up video caption data for multimodal large language models (June 2024), https://github.com/Share14/ShareGemini
49. Shen, X., Xiong, Y., Zhao, C., Wu, L., Chen, J., Zhu, C., Liu, Z., Xiao, F., Varadarajan, B., Bordes, F., et al.: Longvu: Spatiotemporal adaptive compression for long video-language understanding. arXiv preprint arXiv:2410.17434 (2024)
50. Shi, J., Han, K., Wang, Z., Doucet, A., Titsias, M.: Simplified and generalized masked diffusion for discrete data. Advances in neural information processing systems **37**, 103131–103167 (2024)
51. Sigurdsson, G.A., Gupta, A., Schmid, C., Farhadi, A., Alahari, K.: Charades-ego: A large-scale dataset of paired third and first person videos. arXiv preprint arXiv:1804.09626 (2018)
52. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: European conference on computer vision. pp. 510–526. Springer (2016)
53. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y., et al.: Moviechat: From dense token to sparse memory for long video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18221–18232 (2024)
54. Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al.: Gemma 3 technical report. arXiv preprint arXiv:2503.19786 (2025)
55. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
56. Wang, H., Feng, B., Lai, Z., Xu, M., Li, S., Ge, W., Dehghan, A., Cao, M., Huang, P.: Streambridge: Turning your offline video large language model into a proactive streaming assistant. arXiv preprint arXiv:2505.05467 (2025)
57. Wang, H., Xu, Z., Cheng, Y., Diao, S., Zhou, Y., Cao, Y., Wang, Q., Ge, W., Huang, L.: Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models. arXiv preprint arXiv:2410.03290 (2024)

58. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
59. Wang, W., Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Zhu, J., Zhu, X., Lu, L., Qiao, Y., et al.: Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. arXiv preprint arXiv:2411.10442 (2024)
60. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
61. Wang, Y., Meng, X., Wang, Y., Liang, J., Wei, J., Zhang, H., Zhao, D.: Videollm knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. arXiv preprint arXiv:2411.17991 (2024)
62. Wei, M., Wan, C., Yu, X., Wang, T., Yang, Y., Mao, X., Zhu, C., Cai, W., Wang, H., Chen, Y., et al.: Streamvln: Streaming vision-and-language navigation via slowfast context modeling. arXiv preprint arXiv:2507.05240 (2025)
63. Xie, B., Liu, Y., Wang, T., Cao, J., Zhang, X.: Glad: A streaming scene generator for autonomous driving. arXiv preprint arXiv:2503.00045 (2025)
64. Xiong, H., Yang, Z., Yu, J., Zhuge, Y., Zhang, L., Zhu, J., Lu, H.: Streaming video understanding and multi-round interaction with memory-enhanced knowledge. arXiv preprint arXiv:2501.13468 (2025)
65. Xu, R., Xiao, G., Chen, Y., He, L., Peng, K., Lu, Y., Han, S.: Streamingvlm: Real-time understanding for infinite video streams. arXiv preprint arXiv:2510.09608 (2025)
66. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
67. Yang, H., Tang, F., Zhao, L., An, X., Hu, M., Li, H., Zhuang, X., Lu, Y., Zhang, X., Swikir, A., et al.: Streamagent: Towards anticipatory agents for streaming video understanding. arXiv preprint arXiv:2508.01875 (2025)
68. Yang, Y., Zhao, Z., Shukla, S.N., Singh, A., Mishra, S.K., Zhang, L., Ren, M.: Streammem: Query-agnostic kv cache memory for streaming video understanding. arXiv preprint arXiv:2508.15717 (2025)
69. Yao, L., Li, Y., Wei, Y., Li, L., Ren, S., Liu, Y., Ouyang, K., Wang, L., Li, S., Li, S., et al.: Timechat-online: 80% visual tokens are naturally redundant in streaming videos. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 10807–10816 (2025)
70. Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., et al.: Minicpm-v: A gpt-4v level mllm on your phone. arXiv preprint arXiv:2408.01800 (2024)
71. Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., Kong, L.: Dream 7b: Diffusion large language models. arXiv preprint arXiv:2508.15487 (2025)
72. You, Z., Nie, S., Zhang, X., Hu, J., Zhou, J., Lu, Z., Wen, J.R., Li, C.: Llada-v: Large language diffusion models with visual instruction tuning. arXiv preprint arXiv:2505.16933 (2025)
73. Yu, R., Ma, X., Wang, X.: Dimple: Discrete diffusion multimodal large language model with parallel decoding. arXiv preprint arXiv:2505.16990 (2025)
74. Zhang, B., Li, K., Cheng, Z., Hu, Z., Yuan, Y., Chen, G., Leng, S., Jiang, Y., Zhang, H., Li, X., et al.: Videollama 3: Frontier multimodal foundation models for image and video understanding. arXiv preprint arXiv:2501.13106 (2025)
75. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023)

76. Zhang, H., Wang, Y., Tang, Y., Liu, Y., Feng, J., Jin, X.: Flash-vstream: Efficient real-time understanding for long video streams. arXiv preprint arXiv:2506.23825 (2025)
77. Zhang, K., Yang, Z., Wang, B., Qian, S., Xu, C.: Querystream: Advancing streaming video understanding with query-aware pruning and proactive response. In: The Fourteenth International Conference on Learning Representations
78. 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)
79. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Video instruction tuning with synthetic data. arXiv preprint arXiv:2410.02713 (2024)
80. Zhang, Y., Shi, C., Wang, Y., Yang, S.: Eyes wide open: Ego proactive video-llm for streaming video. arXiv preprint arXiv:2510.14560 (2025)
81. Zhou, L., Xu, C., Corso, J.: Towards automatic learning of procedures from web instructional videos. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
82. Zhu, F., Wang, R., Nie, S., Zhang, X., Wu, C., Hu, J., Zhou, J., Chen, J., Lin, Y., Wen, J.R., et al.: Llada 1.5: Variance-reduced preference optimization for large language diffusion models. arXiv preprint arXiv:2505.19223 (2025)
83. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Duan, Y., Tian, H., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)