Em-Garde: A Propose-Match Framework for Proactive Streaming Video Understanding

Yikai Zheng[1,2*], Xin Ding[3*], Yifan Yang[4], Shiqi Jiang[4], Hao Wu[5], Qianxi Zhang[4], Weijun Wang[1], Ting Cao[1†], and Yunxin Liu[1]

[1] Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China
[2] Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, Beijing, China
[3] University of Science and Technology of China, Hefei, China
[4] Microsoft Research Asia
[5] Nanjing University, Nanjing, China
[*] Equal contribution [†] Corresponding author
tingcao@mail.tsinghua.edu.cn

Abstract. Recent advances in Streaming Video Understanding has enabled a new interaction paradigm where models respond proactively to user queries. Current proactive VideoLLMs rely on per-frame triggering decision making, which suffers from an efficiency-accuracy dilemma. We propose **Em-Garde**, a novel framework that decouples semantic understanding from streaming perception. At query time, the Instruction-Guided Proposal Parser transforms user queries into structured, perceptually grounded visual proposals; during streaming, a Lightweight Proposal Matching Module performs efficient embedding-based matching to trigger responses. Experiments on StreamingBench and OVO-Bench demonstrate consistent improvements over prior models in proactive response accuracy and efficiency, validating an effective solution for proactive video understanding under strict computational constraints. Code and model are available at https://air-embodied-brain.github.io/Em-Garde/.

Keywords: Streaming Video Understanding · Proactive VideoLLM · Multimodal LLM

1 Introduction

The recent advances in Multimodal LLMs (MLLMs) [1, 4, 13, 18, 27, 43, 46, 50], have brought about significant success in streaming video understanding. Industrial models like Gemini Live [29] are able to comprehend streaming video content and respond to user queries in real time, bringing an unprecedented interactive experience. Academic efforts push even further towards models that can answer a user query proactively [2, 3, 8, 24, 30, 37]: instead of passively switching on and answering at query time, these models keep monitoring the video stream, and

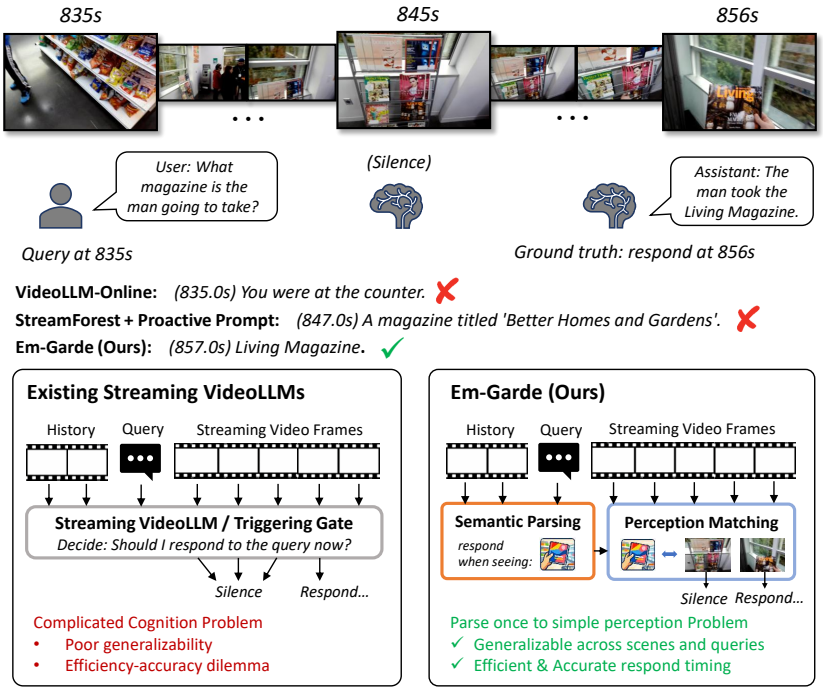

Fig. 1: Demonstration of our model v.s. existing Streaming VideoLLMs on the Proactive Streaming Understanding task. While existing models solve a complicated response/silence decision-making problem at every timestep, we turn the problem into a simple perception problem with query-time semantic parsing, allowing for efficient and accurate proactive response timing.

automatically decide when to respond to the query (*e.g.* remind the user when water boils). This new interactive paradigm opens up vast new applications from sports commentary to household assistance [3,5,15,20,31,36,39], yet also posing a fundamental new challenge to the community.

The unique challenge of proactive streaming video understanding lies in the need to decide whether to respond in real time at every timestep [2,24,35]. Whenever a new frame arrives, the system must perform multiple perceptual and semantic reasoning processes to determine whether a response should be triggered, including visual perception (*e.g.* object and action recognition), event transition detection, and query-aware relevance assessment. However, the processing speed must match the video frame rate (at least 5-10 frames per second (fps)), making the available computational budget severely limited [6,7,25,42]. This creates an inherent tension between the need for *complicated visual-semantic understanding and the requirement for fast per-frame computation.*

Existing approaches mostly formulate the problem as a **per-frame decision making problem**. Specifically, they utilize an MLLM or lightweight gate to output a response/silence decision at every timestep. To keep up with the

streaming speed, they reduce the model size [8, 24, 30] or increase information compression rate [21, 25, 32, 42, 44, 48] to perform fast framewise understanding. However, this naturally limits the granularity of visual perception and the decision-making quality of the model, making the tension between the need for rich visual understanding and the strict real-time constraint fundamentally unsolved.

To solve the efficiency-accuracy dilemma, we try to break away from this paradigm. We note that the complicated decision-making problem can break down into two stages: Upon the query, the model can already decide which kind of events are worth responding to and imagine the visual evidence that points to them. Then, only simple visual perception is needed to match the evidence [12, 40]. Based on this insight, we propose a **per-query parsing paradigm** (Fig. 1) that shifts expensive semantic interpretation entirely outside the streaming loop. Specifically, the system parses the user query into perceptually grounded proposals at query time, enabling subsequent streaming processing to be reduced to efficient frame-level visual matching. For example, if the user requests to be notified when "the water boils", the system does not need to repeatedly interpret this request at every frame. Instead, it parses the instruction at query time into concrete visual proposals, such as "vigorous bubbling" or "sustained steam emission", and then continuously performs lightweight frame-level matching to detect these cues.

This separation introduces two key technical challenges. Firstly, there is no universal parsing scheme that can generalize across diverse and open-ended user queries. The system must handle varied query structures, multiple possible event types, and different levels of abstraction, and produce structured visual proposals that a lightweight perception model can reliably use. Secondly, given multiple fine-grained visual proposals, it is non-trivial for a lightweight model to quickly and reliably match them to actual visual evidence in streaming video.

By addressing these challenges, we realize a novel framework for proactive streaming video understanding, called **Em-Garde**, which separates semantic reasoning from perception and enables real-time proactive response under strict computational constraints. It integrates two key techniques.

For semantic parsing, we introduce the Instruction-Guided Proposal Parser (IGPP), which leverages the reasoning ability of the MLLM to parse high-level instructions into structured visual proposals. Each proposal consists of parallel visual cues describing either distinct noteworthy events or detailed aspects of a single event. IGPP incorporates a short video context to adapt proposals over time, enabling the parsing process to evolve with the ongoing video while still respecting the query instruction.

To train IGPP, we curate the Parse2Prop-1K dataset containing pairs of queries, proposals, and target response times written by humans or GPT-5 [28], covering diverse proposing methods, variable proposal lengths, and different levels of abstraction.

For streaming perception, we propose the Lightweight Proposal Matching Module (LPMM). LPMM encodes recent video context with a sliding window

and embeds both the video segment and the IGPP proposals using a lightweight embedding model [11,19,45,49]. The embeddings are compared in the embedding space to produce a temporal sequence of similarity scores. A surge in similarity indicates a potential matching, triggering the MLLM responder to generate a response and verify its appropriateness. This design decouples expensive reasoning from high-frequency perception, enabling efficient, frame-rate-matched proactive responses.

To our knowledge, Em-Garde is the first proactive streaming framework that explicitly decomposes triggering decision into query-time proposal generation and streaming-time proposal matching, thus reducing the complex streaming understanding problem to lightweight perception problem.

To evaluate the performance of the Em-Garde framework, we perform extensive tests on proactive response tasks and online video understanding tasks. On proactive response tasks, we outperform existing real-time streaming models by more than 3% accuracy on StreamingBench [17] and 10% F1 score on OVO-Bench [22], while achieving state-of-the-art processing speed of 10-15 fps on A100 GPUs on arbitrarily long videos (Fig. 3). On online video understanding tasks, we score an overall 76.7% on StreamingBench real-time perception tasks, 63.0% on OVO-Bench real-time perception tasks, and 52.2% on OVO-Bench backward tracing tasks. Further analysis shows that the quality of the IGPP proposal contributes to the trigger accuracy and that RL helps generate perception-friendly proposals. These results show the efficiency and generalizability of our framework on the proactive streaming understanding task.

2 Related Work

The concept of an always-on, proactive streaming assistant was first articulated in VideoLLM-Online [2], which introduces a conversational setting where users provide a query in advance and the model autonomously responds when relevant events occur. This interaction paradigm has since been widely adopted in subsequent works [8, 9, 16, 24, 30, 38, 47, 51], establishing proactive response as a core problem in streaming video understanding.

A central challenge in proactive streaming is determining when to respond, often referred to as the **triggering decision** problem. VideoLLM-Online [2] trains the model to output a silence or respond token after each frame, continuing generation only if the respond token is produced. StreamMind [8] and Dispider [24] follow this paradigm but introduce a separate gating module to improve efficiency. However, due to the difficulty of collecting high-quality supervision for triggering signals, such models often suffer from limited generalizability and struggle to adapt to diverse or unseen queries.

An alternative strategy, introduced in OVO-Bench [22], prompts a VideoLLM to continuously predict whether a target event has occurred, triggering a response when the prediction changes from "No" to "Yes." Streaming VideoLLMs such as FlashVStream [48] and StreamForest [44] adopt this scheme and focus on improving efficiency through context management. Nevertheless, VideoLLMs are known

to follow such proactive prompts unreliably [17]. Moreover, the triggering decision frequently requires fine-grained visual–semantic reasoning, which conflicts with the aggressive information compression typically employed by streaming models to maintain real-time performance.

More recent approaches, including StreamAgent [40] and MMDuet-2 [33], recognize that determining the appropriate response time requires non-trivial reasoning over evolving visual evidence. Instead of solving it with naive supervised training or prompting, they formulate proactive response as a sequential decision-making problem and address it via reinforcement learning or step-wise planning. Although these methods improve accuracy, their reliance on long contexts and frame-wise reasoning introduces substantial computational overhead, making real-time deployment challenging.

Overall, existing methods attempt to make triggering decisions from scratch at every timestep. This leaves unresolved the fundamental tension between the rich visual–semantic understanding required for accurate triggering and the limited computational budget available in streaming settings. In contrast, we move semantic parsing outside the streaming loop, thereby reducing the computational burden of triggering decisions. The streaming loop can then rely on a lightweight perception model operating independently to detect relevant visual signals, without repeated semantic reasoning. By decoupling query understanding from visual perception, our approach enables accurate, real-time proactive responses with substantially lower computational overhead.

3 Methodology

3.1 Problem Formulation

We adopt the standard formulation of proactive streaming video understanding introduced in prior work [2, 24]. In this setting, a model receives a natural language instruction once at query time and subsequently processes a continuous video stream, deciding when to respond without further user queries.

Formally, an instruction I is provided at query time t_0. A video stream is represented as an unbounded sequence of frames $\{x_t\}_{t\geq 0}$ where frame x_t arrives at time step t. At each time step $t \geq t_0$, the model observes the video frames until t and outputs a binary triggering decision

$$D_t = \pi_{trigger}(I, \{x_\tau\}_{0\leq\tau\leq t}) \in \{0, 1\}, \tag{1}$$

Where $D_t = 1$ indicates that a response should be issued at time t.

Suppose the model decides to respond at time step t_1, i.e. $D_{t_1} = 1$, the model generates a response

$$R_{t_1} \sim \mathcal{G}(I, \{x_\tau\}_{0\leq\tau\leq t_1}), \tag{2}$$

which may take the form of an answer, alert, explanation, or tool invocation, depending on the instruction.

Let

$$T_R = \{t | D_t = 1\} \tag{3}$$

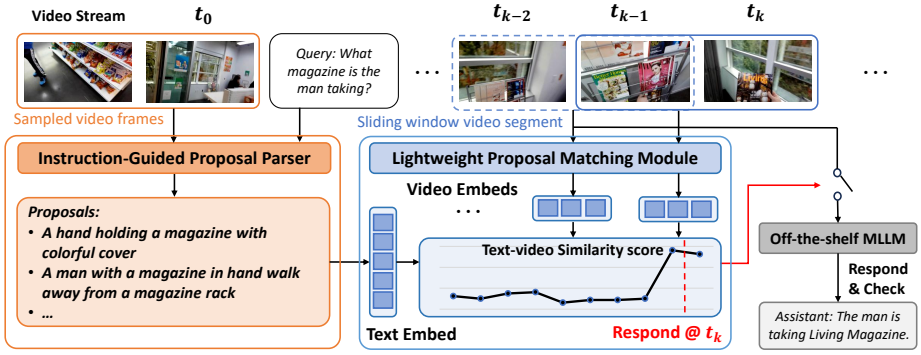

Fig. 2: Overview of the Em-Garde Framework: IGPP (Orange) receives the Instrcution I and the sampled video frames before query time, and parse the instruction into perceptually-grounded visual cues. **LPMM (Blue)** runs in the streaming loop, matching the current sliding-window video segment to the proposal in the embedding space. The similarity scores are utilized as the temporal signal for triggering decision. Together, the two modules separate semantic understanding from the streaming loop, allowing for efficient and accurate triggering decisions.

denote the set of predicted response times. Performance is evaluated by matching predicted triggers to ground-truth event times within a tolerance window, following established benchmark protocols (e.g., OVO-Bench, StreamingBench).

3.2 The Overview of Em-Garde

To improve the accuracy and efficiency of triggering decisions, we decompose the decision process into two stages: semantic parsing and visual perception. Formally, given an instruction I and video frames $\{x_\tau\}_{t_0-h_c\leq\tau\leq t_0}$ of length h_c around the query time, a proposer model generates a proposal set:

$$P = \pi_{propose}(I, \{x_\tau\}_{t_0-h_c\leq\tau\leq t_0}). \tag{4}$$

The proposal describes observable visual evidence about the events that should trigger a response according to the instruction. This transforms the original triggering decision problem into a perception-centric matching problem.

After semantic parsing, a lightweight visual matching module runs continuously in the streaming loop. At each time step t, the module receives a short streaming video segment $\{x_\tau\}_{t-h_s\leq\tau\leq t}$ of length h_s and the proposal set P, and outputs a set of matching score

$$S^t = \pi_{match}(P, \{x_\tau\}_{t-h_s\leq\tau\leq t}). \tag{5}$$

Triggering decisions are then derived via simple temporal processing over the score history:

$$D_t = \pi_{tp}(\{S^\tau\}_{\tau\leq t}). \tag{6}$$

Based on this formulation, we proposed **Em-Garde** (Fig. 2), a proactive streaming understanding framework that explicitly separates semantic parsing from visual perception. Em-Garde consists of two key components:

- **Instruction-Guided Proposal Parser (IGPP)**: IGPP is invoked once at query time. It performs the complex reasoning to translate high-level instructions into concrete perceptually-grounded cues, effectively defining a decision schema for triggering responses.
- **Lightweight Proposal Matching Module (LPMM)**: LPMM runs continuously in the streaming loop. It matches incoming video segments against the proposal set in a multimodal embedding space and produces temporal matching scores without performing full semantic understanding of the video.

This design achieves efficient allocation of computational budget and avoids repeated semantic understanding, preserving task fidelity while enabling real-time performance.

3.3 Instruction-Guided Proposal Parser

At query time t_0, IGPP receives the instruction I along with a segment of video history $\{x_\tau\}_{t_0-h_c\leq\tau\leq t_0}$. Using a large multimodal language model, it generates a proposal set:

$$P = \{p_1, p_2, \ldots, p_k\} \tag{7}$$

Each proposal p_i is a **concise, declarative** visual cue describing a response-worthy event.

Properties of the proposals Proposals are intentionally designed to have the following properties:

- **Temporal Localizability** Proposals should only be matched when a response is required.
- **Perceptual Groundability** Proposals can be matched with a short video segment by simple visual perception without understanding the long context.
- **Redundancy** Multiple proposals are allowed to describe complementary visual details of the same underlying event.

Property 1 and 2 follows naturally from the role of IGPP to turn response/silence decision-making into simple perception problem, while Property 3 reflects the inherent uncertainty of a streaming video. Rather than committing to a single hypothesis, IGPP maintains a set of plausible evidences.

Learning effective proposals Generating useful proposals is non-trivial, involving instruction understanding and task-specific reasoning. Naively trained models might produce proposals that the perception module cannot reliably recognize.

To address this, we train IGPP in two stages. First, **supervised fine-tuning (SFT)** encourage the model to learn the proposal format and pre-defined proposal methods. Then, we introduce a **reinforcement learning (RL)** stage that directly optimizes downstream triggering behavior. In this stage, the model tests out the learned proposal methods, aligns the visual cues with the perception module, and learns to generate proposal sets that are temporally localized and perceptually grounded.

For training data, we curated **Parse2Prop-1K**, a small-scale dataset containing 668 queries and proactive responses on 92 videos across COIN [41], Ego4d [10] and BEHAVIOR [14]. Each query is issued at time t_0 prior to the occurrence of the target event and may require one or multiple responses after t_0. Half of the examples include human-or-GPT-5-authored proposal annotations. The proposals exhibit diverse styles, lengths, and levels of visual detail, with an average of 3.99 visual cues per query and an average cue length of 13.5 words. Details of the dataset can be found in section 3 of the supplementary details.

During reinforcement learning, the proposer is rewarded for correct triggering decisions near the onset of each ground-truth event, receiving one point per correct trigger. A trigger is considered correct if it is within a tolerance window after the start of an event (4 seconds in our experiments), accounting for potential information-gathering latency. Triggers outside these windows are treated as false positives and penalized. A coefficient λ is introduced to balance between event recall and false trigger suppression. The final reward is defined as

$$r = (1 - \lambda r_{fp}) \frac{n_c}{n}$$

Where

$$r_{fp} = 1 - 2^{-n_{fp}/n}$$

and n_c, n_{fp}, n are the number of correct triggers, false triggers, and ground-truth events, respectively. We set $\lambda = 1$, and train the model using GRPO [26].

3.4 Lightweight Proposal Matching Module

LPMM is instantiated as a VLM-based multimodal embedding model [19, 49]. During streaming, at each time step t, a short sliding window of recent frames $\{x_\tau\}_{t-h_s \le \tau \le t}$ is embedded into a video representation $e_v(t)$. Unlike IGPP, LPMM is not required to perform semantic reasoning; instead, it focuses on *detecting immediate visual cues* relevant to the proposal set. Therefore, it uses a short temporal context with a higher frame rate to capture time-sensitive actions and object appearances.

Each proposal $p_i \in P$ is pre-embedded into a proposal representation e_{p_i}. The matching score between the video segment and proposal p_i at time t is

computed using cosine similarity:

$$s_i^t = \cos(e_v^t, e_{p_i}) \tag{8}$$

Triggering decisions are derived from the temporal evolution of similarity scores. Specifically, a trigger occurs when at least one proposal exhibits a sharp increase in similarity exceeding a predefined threshold:

$$D_t = \mathbf{1}\big[\max_i (s_i^t - s_i^{t-1}) > \theta\big]$$

The threshold θ governs the *trade-off between triggering sensitivity and conservativeness*. Increasing θ generally reduces spurious triggers but also increases the risk of missed events, which may be unacceptable in safety-critical applications such as healthcare and surveillance. Conversely, lower θ values favor more frequent triggering, improving recall at the cost of increased downstream verification, which may be undesirable in daily-life scenarios such as household assistance. Em-Garde exposes θ as an explicit control knob, allowing practitioners to select operating points appropriate to application requirements.

Since LPMM does a general perception task which does not involve task-specific instruction followings, we employ an off-the-shelf embedding model, Ops-MM-V1 [23], for the LPMM without any training.

3.5 Computational Efficiency

By shifting semantic understanding out of the streaming loop, Em-Garde runs the streaming loop with a lightweight model on constant length input, inherently removing the problem of linearly growing inference time without applying aggressive information compression.

To further improve the inference efficiency, we introduce a visual encoding cache, which stores the encodings of the overlapped video frames in the sliding window. This allows the visual encoder in the embedding model to encode only one video frame at every timestep, further improving the efficiency by $2\times - 3\times$. With these improvements, the streaming loop can run at a maximum of 10-15 fps on arbitrarily long video sessions on A100 GPUs.

The proposal and response generation, while more computationally expensive, are executed infrequently and can be scheduled asynchronously. Consequently, they do not block or degrade the efficiency of the main streaming loop, yielding a scalable and practical solution for long-horizon proactive streaming understanding.

4 Experiments

4.1 Experiment Setup

Implementation Details Our proposal model is initialized from Qwen2.5VL-7B [1] and trained with our two-stage framework on 8 A100 GPUs. The SFT and

RL stages took around 1 and 5 hours respectively. We use Ops-MM-V1-2B [23] as the embedding model. During inference, the embedding model embeds 2-second, 2fps video segments at a process rate of 2fps. The proposal model receives the original query together with 5 seconds of video history at 1 fps as the context. The triggering threshold is set to 0.04 by default.

Benchmarks for Evaluation With our experiments, we aim to evaluate three key aspects in proactive streaming understanding:

- Whether our model yields superior **proactive response** abilities, i.e. the ability to decide when to respond to a user query.
- Whether our model preserves **online video question-answering** abilities of Streaming VideoLLMs, which evaluate the quality of the responses.
- Whether our model runs streaming inference **efficiently**.

For Proactive Response, we selected three popular benchmarks: The Proactive Response (PO) task of streamingbench [17], The Forward Active Response (FAR) task of OVO-Bench [22], and ProactiveVideoQA [34]. For Online Video QA, we selected the Real-time Understanding task of StreamingBench, and the Real-time Perception and Backward Tracing tasks of OVO-Bench.

Table 1: The Proactive Response Timing comparison (recall, precision and F1 score) on OVO-Bench Future Active Responding tasks.

Model	Size	FPS	CRR			SSR			REC			Avg. F1
			R	P	F1	R	P	F1	R	P	F1	
VideoLLM-Online [2]	8B	2	2.08	2.08	2.08	11.07	6.88	8.49	7.43	16.46	10.24	6.93
FVStream [48]	7B	2	6.25	5.21	5.68	2.46	9.52	3.91	3.46	7.49	4.73	4.77
StreamForest [44]	7B	2	2.08	2.08	2.08	8.02	13.69	10.11	30.49	28.90	29.67	13.95
MMDuet-2 [33]	3B	2	47.92	8.47	14.40	–	–	– [a]	19.33	**42.72**	26.62	20.51
Em-Garde (Ours)	2B [b]	2	**47.92**	**18.22**	**26.40**	**54.88**	**14.87**	**23.40**	**47.33**	39.66	**43.16**	**30.99**

[a] The SSR score is not reported due to slow inference and OOM Error.
[b] The size of the perception module.

4.2 Results on Proactive Response tasks

We first evaluate our model's proactive streaming understanding ability on three benchmarks: StreamingBench, OVO-Bench, and ProactiveVideoQA.

For StreamingBench and ProactiveVideoQA, we follow the original evaluation metrics (accuracy for StreamingBench and PAUC for ProactiveVideoQA). For OVO-Bench, proactive models were scarce when the benchmark was released, and it therefore adopted an indirect offline accuracy metric. However, we observe that this metric is not always aligned with the actual correctness of triggering decisions (see Section 2 of the supplementary details). We therefore return to

Table 2: Comparison of models on the accuracy for StreamingBench PO Task. Our model outperforms existing proactive streaming models by more than 3% accuracy.

Model	Size	FPS	Accuracy
FVStream [48]	7B	-	2.0
VideoLLM-Online [2]	8B	2	4.0
Dispider [24]	7B	1	25.3
StreamAgent [40]	3B	1	28.9
MMDuet-2 [33]	3B	1	34.6
Em-Garde (Ours)	2B	1	37.6
Em-Garde (Ours)	2B	2	**38.0**

Table 3: Comparison of models on the **PAUC (↑) / reply duplicate proportion (↓)** of the ProactiveVideoQA benchmark. We achieve competitive results against MMDuet-2, which *directly* optimizes the PAUC metric and is trained on related datasets.

Model	WEB	EGO	VAD
VideoLLM-Online	25.9 / –	25.0 / –	25.0 / –
MMDuet	38.9 / 81.3	46.0 / 99.4	27.4 / 99.2
MMDuet-2	**53.3** / **4.2**	33.6 / **8.1**	**28.9** / 15.2
Em-Garde(Ours)	44.3 /4.5	**52.3** / 17.4	27.4 / **1.4**

the benchmark's original evaluation goal and adopt an **Online Recall and Precision** metric:

$$Rec = \frac{|T_c|}{|T_{gt}|}, \quad Pre = \frac{|T_c|}{|T_r|} \tag{9}$$

Here, T_{gt} denotes the set of ground-truth response times and T_r the set of model response times. T_c represents the set of correct response times, defined as responses that fall within a tolerance window of a ground-truth response time.

For OVO-Bench, we evaluate existing streaming models under this online protocol using each model's original proactive response mechanism. In particular, for models that require proactive prompting (FVStream and StreamForest), we follow the prompting method provided by OVO-Bench but evaluate triggering correctness using our online metric instead of the original offline accuracy. As Dispider and StreamAgent have not released their proactive inference code, they are not included in our evaluation.

The results are shown in Tab. 1, Tab. 2, and Tab. 3. For each benchmark, we compare with baselines with reported results or available implementation. Our method significantly outperforms all baselines on OVO-Bench and StreamingBench while achieving competitive performance on the newer and less-tested ProactiveVideoQA benchmark, demonstrating the strong proactive response capability of our model.

Table 4: Performance comparison on online video understanding benchmarks. Higher is better for all metrics.

Model	Size (B)	StreamingBench	OVO-Bench	
		Real-time VU ↑	Real-time VP ↑	Backward Tracing ↑
VideoLLM-Online	8B	36.0	20.8	17.7
FVStream	7B	23.2	29.9	25.4
Dispider	7B	67.6	54.6	36.1
StreamAgent	7B	74.3	<u>61.3</u>	46.2
StreamForest	7B	**77.3**	61.2	<u>52.0</u>
Em-Garde (Ours)	7B [a]	<u>76.7</u>	**63.0**	**52.2**

[a] The size of the response model

4.3 Results on Online Video Understanding tasks

To evaluate the online video question-answering abilities of our framework, we evaluated our framework on the real-time perception tasks of StreamingBench, and the backward tracing and real-time understanding tasks of OVO-Bench. The results (Tab. 4) show that our framework achieves results comparable to state-of-the-art (SOTA) streaming MLLMs which explicitly optimized context management and real-time understanding. Overall, these results show that our model maintains the online visual understanding abilities of the responding model.

4.4 Computational efficiency

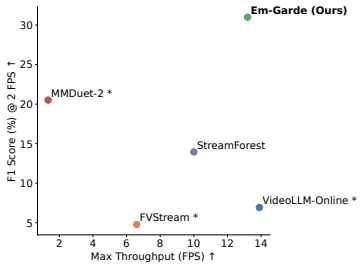

Fig. 3: Throughput-Performance comparison on the Proactive Streaming Understanding task. The performance is measured by the average F1 Score on OVO-Bench. Models with * means that the throughput degrade as the context grows without KV Cache truncation. We show the throughput without degrading.

Fig. 3 shows the efficiency and performance of state-of-the-art Streaming VideoLLMs on the proactive streaming understanding task. Our model achieves

Table 5: Ablation on IGPP & LPMM. Without our propose-match design, the performance significantly degrades.

Method	OVO-Bench (FAR)	StreamingBench (PO)
Em-Garde	**31.0**	**38.0**
w/o. IGPP	24.0	28.8
w/o. LPMM	14.6	23.2
w/o. IGPP & LPMM	18.6	17.2
sliding window	21.0	26.8

Table 6: OVO-Bench average F1 score under different training and testing thresholds.

Train θ \ Test θ	0.03	0.035	0.04
0.03	26.59	26.89	26.67
0.035	**28.74**	**30.12**	29.44
0.04	27.25	28.24	**30.99**

a competitive efficiency of around 13 fps, while outperforming all former models. Furthermore, unlike VideoLLM-Online and MMDuet-2, our model does not suffer from degrading efficiency as the video context grows.

4.5 Ablation Studies

Ablations on IGPP and LPMM To isolate the contribution of our designed modules, we replace IGPP with direct prompt rewriting using Qwen2.5VL-7B and replace LPMM with a direct yes/no decision module using Qwen2VL-2B (VLM backbone of Ops-MM-V1-2B) We additionally evaluate a naive sliding-window baseline that directly predicts whether to answer from the current query and 2-second video segment. Results are shown in Tab. 5. Removing either IGPP or LPMM causes substantial degradation, while removing both components performs even worse. The sliding-window baseline also underperforms despite using the same underlying models. These results validate that the gains of Em-Garde arise from the proposed propose-match decomposition rather than pretrained model capacity alone.

Triggering Threshold Choices We analyze the robustness of under different threshold settings during RL training and inference.Across threshold combinations between 0.03 and 0.04, Em-Garde consistently maintains strong performance on OVO-Bench, with all settings substantially outperforming prior methods. This robustness arises from two factors: (1) triggering is based on changes in similarity rather than absolute similarity values, which is stable across different videos, and (2) RL training adapts proposals to the operating threshold. Therefore, the framework is not overly sensitive to a single manually selected threshold.

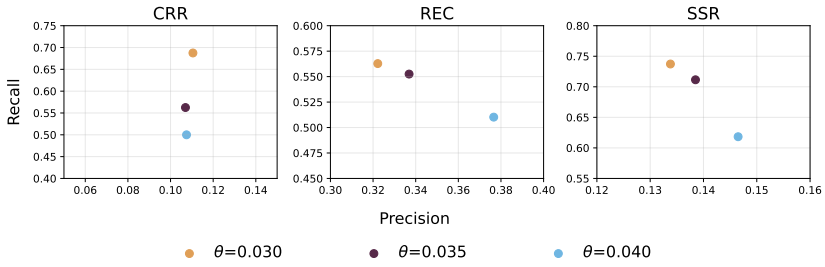

Fig. 4: Recall-Precision plot for varied detection thresholds across OVO-Bench Forward Active Response Tasks.

Table 7: Ablation of RL training and the false-positive penalty coefficient λ. The OVO-Bench score is the overall F1 score for Forward Active Responding tasks.

Method	OVO-Bench (FAR)	StreamingBench (PO)
SFT	23.4	34.8
RL ($\lambda = 0$)	27.7	30.8
RL ($\lambda = 0.5$)	28.2	33.2
RL ($\lambda = 0.75$)	28.6	36.4
RL ($\lambda = 1$)	**31.0**	**38.0**
RL ($\lambda = 1.5$)	19.3	33.6

Furthermore, users are able to tune the threshold to adapt to application needs. As shown in Fig. 4, the threshold choice governs the tradeoff between precision and recall. Our framework explicitly exposes this control knob and allows users to select θ that suits real-world requirements on triggering sensitivity.

RL coefficient λ We additionally study the false-positive penalty coefficient λ used during RL training.As shown in Tab. 7,consistent trends are observed: performance improves when moderate false-positive control is introduced and peaks around $\lambda = 1$, while both $\lambda = 0$ (no FP suppression) and overly large λ values lead to degraded performance. This suggests that moderate control of false triggers is important for learning perception-friendly proposals. Nevertheless, the performance is relatively stable across $\lambda \in [0.5, 1]$, indicating that the performance is not too sensitive to a single λ.

5 Limitations

Despite its effectiveness, Em-Garde has several limitations that point to promising directions for future work.

First, our triggering mechanism relies on thresholding surges in temporal signals, which is sensitive to uncertainty in streaming video. Sudden scene changes

or visually correlated yet semantically irrelevant transitions can lead to unstable triggering behavior. We explored suppressing such errors using negative proposals, but current embedding models struggle to reliably distinguish subtly different textual cues. Developing better triggering criteria is an important direction for future research, and we also expect future multimodal embedding models—an active research area—to provide stronger discriminative capabilities.

Second, our approach does not explicitly address online video understanding challenges such as long-horizon reasoning. Following prior work [24], we treat triggering decision as a problem decoupled from response generation. Em-Garde focuses on *when* to respond, and its modular design allows state-of-the-art responding model to be plugged in downstream. Joint optimization across decision and generation stages remains an open direction for future work.

6 Conclusion

We introduced **Em-Garde**, which separates query parsing from streaming perception to overcome the efficiency–accuracy dilemma in proactive video understanding. By shifting heavy reasoning outside the streaming loop, we enabled a lightweight model to handle triggering decisions accurately and efficiently.

Acknowledgements

This work was supported in part by the Tsinghua University (AIR)–AsiaInfo Technologies (China), Inc. Joint Research Center for 6G Network and Intelligent Computing and the Xiongan AI Institute. This work was also supported in part by the National Natural Science Foundation of China (NSFC) under Grants No. 62432004 and No. 62302207, and by the Fundamental Research Funds for the Central Universities under Grants No. 2026300278 and No. ZZKT2026A31.

The authors express their sincere gratitude to all organizations and individuals whose support and contributions made this research possible.

References

1. 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)
2. 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)
3. Chen, J., Zeng, Z., Lin, Y., Li, W., Ma, Z., Shou, M.Z.: Livecc: Learning video llm with streaming speech transcription at scale. In: CVPR (2025)
4. 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)

5. Deliège, A., Cioppa, A., Giancola, S., Seikavandi, M.J., Dueholm, J.V., Nasrollahi, K., Ghanem, B., Moeslund, T.B., Droogenbroeck, M.V.: Soccernet-v2 : A dataset and benchmarks for holistic understanding of broadcast soccer videos. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2021)
6. Di, S., Yu, Z., Zhang, G., Li, H., Zhong, T., Cheng, H., Li, B., He, W., Shu, F., Jiang, H.: Streaming video question-answering with in-context video kv-cache retrieval. arXiv preprint arXiv:2503.00540 (2025)
7. Ding, X., Wei, J., Jia, F., Mi, L., Ju, R., Wang, X., Zheng, Y., Zhang, Z., Wang, W., Jiang, S., Liu, Y., Cao, T.: Demo: Edgemind-os: A plug-and-play embodied intelligence system for real-time on-device deployment. pp. 1219–1221 (11 2025). https://doi.org/10.1145/3680207.3765596
8. Ding, X., Wu, H., Yang, Y., Jiang, S., Bai, D., Chen, Z., Cao, T.: Streammind: Unlocking full frame rate streaming video dialogue through event-gated cognition. arXiv preprint arXiv:2503.06220 (2025)
9. Fu, S., Yang, Q., Li, Y.M., Peng, Y.X., Lin, K.Y., Wei, X., Hu, J.F., Xie, X., Zheng, W.S.: Vispeak: Visual instruction feedback in streaming videos (2025), https://arxiv.org/abs/2503.12769
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. Jian, W., Zhang, Y., Liang, D., Xie, C., He, Y., Leng, D., Yin, Y.: Rzenembed: Towards comprehensive multimodal retrieval. arXiv preprint arXiv:2510.27350 (2025)
12. Lee, H., Kim, H., Kim, I.J., Choi, Y.: Flashback: Memory-driven zero-shot, real-time video anomaly detection. arXiv preprint arXiv:2505.15205 (2025)
13. 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)
14. Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Ai, W., Martinez, B., Yin, H., Lingelbach, M., Hwang, M., Hiranaka, A., Garlanka, S., Aydin, A., Lee, S., Sun, J., Anvari, M., Sharma, M., Bansal, D., Hunter, S., Kim, K.Y., Lou, A., Matthews, C.R., Villa-Renteria, I., Tang, J.H., Tang, C., Xia, F., Li, Y., Savarese, S., Gweon, H., Liu, C.K., Wu, J., Fei-Fei, L.: Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. arXiv preprint arXiv:2403.09227 (2024)
15. Li, C., Yuan, Z., Zhang, J., Deng, Y., Bi, H., Jia, Z., Duan, X., Luo, P., Zhang, J.: Less redundancy: Boosting practicality of vision language model in walking assistants (2025), https://arxiv.org/abs/2508.16070
16. 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)
17. 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)
18. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/
19. Meng, R., Jiang, Z., Liu, Y., Su, M., Yang, X., Fu, Y., Qin, C., Chen, Z., Xu, R., Xiong, C., et al.: Vlm2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. arXiv preprint arXiv:2507.04590 (2025)

20. Merrill, K., Kim, J., Collins, C.: Ai companions for lonely individuals and the role of social presence. Communication Research Reports **39**, 93 – 103 (2022), https://api.semanticscholar.org/CorpusID:247420609
21. Ning, Z., Liu, G., Jin, Q., Ding, W., Guo, M., Zhao, J.: Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval (2025), https://arxiv.org/abs/2505.15269
22. 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)
23. OpenSearchAI: Ops-mm-embedding-v1-2b. https://huggingface.co/OpenSearch-AI/Ops-MM-embedding-v1-2B (2025), apache License 2.0. Accessed: 2026-03-02
24. 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)
25. 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)
26. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
27. 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)
28. Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., Nathan, A., Luo, A., Helyar, A., Madry, A., et al.: Openai gpt-5 system card (2025), https://arxiv.org/abs/2601.03267
29. Team, G., Georgiev, P., Lei, V.I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al.: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024)
30. 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)
31. Wang, Y., Li, Z., Qian, T., Zheng, H., Wang, Z., Fu, Y., Wang, X.: Streameqa: Towards streaming video understanding for embodied scenarios (2025), https://arxiv.org/abs/2512.04451
32. Wang, Y., Liu, X., Gui, X., Lin, X., Yang, B., Liao, C., Chen, T., Zhang, L.: Accelerating streaming video large language models via hierarchical token compression. arXiv preprint arXiv:2512.00891 (2025)
33. Wang, Y., Liu, S., Wang, D., Xu, N., Wan, G., Zhang, H., Zhao, D.: Mmduet2: Enhancing proactive interaction of video mllms with multi-turn reinforcement learning. arXiv preprint arXiv:2512.06810 (2025)
34. Wang, Y., Meng, X., Wang, Y., Zhang, H., Zhao, D.: Proactivevideoqa: A comprehensive benchmark evaluating proactive interactions in video large language models. arXiv preprint arXiv:2507.09313 (2025)
35. 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)

36. Wen, Z., Wang, Y., Liao, C., Yang, B., Li, J., Liu, W., He, H., Feng, B., Liu, X., Lyu, Y., Zheng, X., Hu, X., Zhang, L.: Ai for service: Proactive assistance with ai glasses (2025), https://arxiv.org/abs/2510.14359
37. Wu, S., Chen, J., Lin, K.Q., Wang, Q., Gao, Y., Xu, Q., Xu, T., Hu, Y., Chen, E., Shou, M.Z.: Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. Advances in Neural Information Processing Systems **37**, 109922–109947 (2024)
38. Xia, J., Chen, P., Zhang, M., Sun, X., Zhou, K.: Streaming video instruction tuning (2025), https://arxiv.org/abs/2512.21334
39. Xu, R., Xiao, G., Chen, Y., He, L., Peng, K., Lu, Y., Han, S.: Streamingvlm: Real-time understanding for infinite video streams (2025), https://arxiv.org/abs/2510.09608
40. 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)
41. Yansong Tang, Dajun Ding, Y.R.Y.Z.D.Z.L.Z.J.L.J.Z.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
42. 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. arXiv preprint arXiv:2504.17343 (2025)
43. 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)
44. Zeng, X., Qiu, K., Zhang, Q., Li, X., Wang, J., Li, J., Yan, Z., Tian, K., Tian, M., Zhao, X., et al.: Streamforest: Efficient online video understanding with persistent event memory. arXiv preprint arXiv:2509.24871 (2025)
45. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
46. 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)
47. Zhang, G., Hannan, T., Kleiner, H., Aydemir, B., Xie, X., Lan, J., Seidl, T., Tresp, V., Gu, J.: Avila: Asynchronous vision-language agent for streaming multimodal data interaction (2025), https://arxiv.org/abs/2506.18472
48. Zhang, H., Wang, Y., Tang, Y., Liu, Y., Feng, J., Dai, J., Jin, X.: Flash-vstream: Memory-based real-time understanding for long video streams. arXiv preprint arXiv:2406.08085 (2024)
49. Zhang, X., Zhang, Y., Xie, W., Li, M., Dai, Z., Long, D., Xie, P., Zhang, M., Li, W., Zhang, M.: Gme: Improving universal multimodal retrieval by multimodal llms. arXiv preprint arXiv:2412.16855 (2024)
50. 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)
51. Zhang, Y., Shi, C., Wang, Y., Yang, S.: Eyes wide open: Ego proactive video-llm for streaming video (2025), https://arxiv.org/abs/2510.14560