UNIVIA: A UNIFIED VIDEO INTELLIGENCE ARCHITECTURE FOR OFFLINE UNDERSTANDING AND DUPLEX STREAMING INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming video interaction requires a model to keep perceiving an evolving video while deciding when to respond and generating answers. Existing chunk-based formulations couple response timing with answer generation: fine chunks split each answer across observation updates, departing from the complete-answer supervision of offline video-language training, while coarse chunks sacrifice the temporal granularity of response decisions. We present UniVIA, a unified video intelligence architecture that serves offline video understanding and full-duplex streaming interaction with a single shared vision-language model. At inference, an asynchronous fork-and-commit mechanism lets a persistent parent stream keep perceiving the video while each response decision forks a child stream that generates the complete answer from a fixed context snapshot and commits it back as memory. For training, we split each streaming interaction into response-ended samples that reproduce the parent and child states of inference, and propose a Segment-Balanced Loss that supervises the silence and response decisions of each query with balanced class weights. UniVIA achieves the best results among open-source models on seven of ten tasks across StreamingBench, OVO-Bench, and OmniMMI, with the largest gains on proactive tasks, keeps the perception delay added by answering at 63–94 ms at 1 FPS regardless of answer length, and achieves the best results on nine of eleven offline video metrics among all evaluated models; with the Qwen3-VL-8B backbone, it improves over the base model on eight of these metrics.

1 INTRODUCTION

Vision-language models (VLMs) have demonstrated strong capabilities in understanding visual content and answering questions about videos. Most existing models, however, operate in an offline setting, where a complete video is provided before the model produces a response. In contrast, real-world applications often require a model to continuously perceive an evolving video stream and interact with users as relevant visual evidence emerges. This motivates a more general form of video intelligence that unifies offline video understanding with online streaming interaction. Importantly, these two settings share the same underlying goal of understanding visual content, while differing in how visual evidence is organized and supervised. Offline video understanding typically provides complete visual context and supervises the model with complete responses, whereas online interaction requires the model to identify fine-grained temporal evidence from a continuously arriving stream and determine whether the current evidence is sufficient to trigger a response.

Existing approaches differ in how they handle observations that arrive while an answer is being generated (Figure 1, left). Blocking finishes the current answer before processing subsequent observations (Chen et al., 2024a), so perception and response decisions pause until the answer ends. Interleaving alternates observation updates with answer chunks (Cui et al., 2026); perception continues, but each answer is split across observation steps, deviating from the complete-answer supervision commonly used in offline vision-language training. Because both generate the answer inside the perception stream, they trade off continuous perception against complete-answer supervision: the answer either pauses the stream or is split by it. Multi-model execution processes observations and generates responses concurrently using separate models (Qian et al., 2025), which avoids this trade-off but no longer shares parameters between perception and response, so supervision for one

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

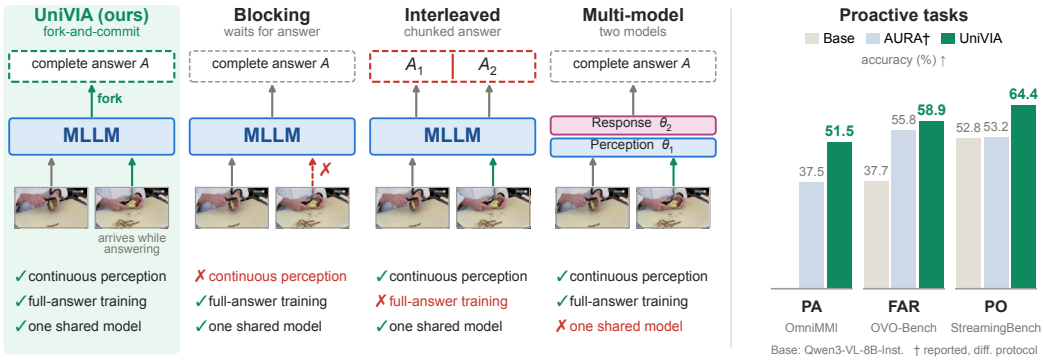

Figure 1: **Streaming execution strategies and proactive results. Left:** Blocking pauses new frames during generation. Interleaved must split the answer A into chunks. Multi-model separates perception (θ_1) and response (θ_2) parameters. UniVIA forks a branch that generates the complete answer while perception continues. **Right:** proactive accuracy (%). Base: Qwen3-VL-8B-Instruct. †Reported in the AURA paper under a different protocol.

role does not train the other. These differences motivate a shared VLM that continues perception while generating each answer as a complete sequence.

Offline vision-language data provides rich supervision for learning the semantic content of videos, whereas online supervision further guides the model to distinguish whether the visual evidence accumulated at each moment supports a response or should be ignored, represented by $\langle\text{response}\rangle$ and $\langle\text{silence}\rangle$ decisions. This complementary relationship allows offline and online data to be jointly exploited to strengthen a shared video understanding capability, without sacrificing the complete-answer supervision of offline training. Based on this insight, we propose **UniVIA**, a unified video intelligence architecture that jointly trains a shared vision-language model with offline and online supervision. Beyond training, UniVIA is designed to support full-duplex interaction at inference time: when the model detects sufficient evidence for a response, it preserves the current causal state and generates a complete response from that state, while continuing to perceive newly arriving video content. We achieve this through an asynchronous fork-and-commit mechanism, which allows response generation and ongoing perception to proceed concurrently within a unified model.

We evaluate UniVIA across both offline video understanding and online streaming interaction benchmarks. The results show that jointly training with offline and online supervision preserves the model's ability to understand visual content while improving its ability to identify fine-grained temporal evidence and make appropriate response decisions. Furthermore, the asynchronous execution enables the model to continuously perceive incoming video while generating responses, providing full-duplex interaction within a single vision-language model. These results demonstrate that offline video understanding and online streaming interaction can be unified within a common video intelligence architecture, rather than requiring separate models or independently optimized training paradigms.

Our contributions are summarized as follows:

- We propose **UniVIA**, a unified video intelligence architecture that integrates offline video understanding and online streaming interaction within a shared vision-language model, enabling both capabilities to benefit from a common visual understanding foundation.
- We identify the fundamental trade-off in existing chunk-based streaming training between preserving complete-answer supervision and achieving fine-grained full-duplex interaction, and introduce a training formulation that retains complete offline responses while learning temporal $\langle\text{silence}\rangle/\langle\text{response}\rangle$ decisions from online supervision.
- We develop an asynchronous fork-and-commit execution mechanism that enables continuous visual perception during response generation, providing duplex streaming interaction without blocking the ongoing perception process.
- We demonstrate the effectiveness of UniVIA through comprehensive experiments on offline video understanding and online streaming interaction benchmarks, showing the benefits of unified training and asynchronous duplex inference.

2 Related Work

Offline video understanding. General-purpose VLMs, including InternVL, LLaVA-OneVision, LLaVA-OneVision-1.5, Qwen3-VL, and Qwen3.5, provide pretrained foundations for video understanding (Chen et al., 2024b; Li et al., 2024; An et al., 2025; Bai et al., 2025; Qwen Team, 2026). LLaVA-OneVision-2, Mage-VL, and VideoChat3 improve encoding efficiency through codec-stream tokenization, codec-guided sparse patch encoding, and spatiotemporal token compression, respectively (An et al., 2026; Yang et al., 2026; Li et al., 2026). Offline supervision also supports streaming adaptation: MMDuet2 and Streamo mix offline QA into supervised training (Wang et al., 2025b; Xia et al., 2025), whereas VideoLLM-online converts timestamped annotations into streaming dialogues (Chen et al., 2024a). Streamo's ET-Instruct ablation reports improved real-time perception but reduced forward active responding on OVO-Bench after adding offline QA (Xia et al., 2025). UniVIA treats offline QA as a special case of streaming interaction within a shared training formulation.

Online streaming interaction. Work on response timing spans supervision for responses and silence (Chen et al., 2024a; Xia et al., 2025; Lu et al., 2026; Li et al., 2026), explicit decision heads or event triggers (Wang et al., 2024; Tian et al., 2026; Ding et al., 2025; Yang et al., 2026; Zheng et al., 2026), and multi-turn reinforcement learning (Wang et al., 2025b). StreamingVLM aligns overlapping-chunk training with bounded-cache inference (Xu et al., 2025). Complementary work manages history through event memory, visual-token or KV-cache compression, and recent-frame windows (Zeng et al., 2025; Yao et al., 2025; Zhang et al., 2026a; Shen et al., 2026). StreamingBench, OVO-Bench, and OmniMMI evaluate online understanding and interaction (Lin et al., 2024; Niu et al., 2025; Wang et al., 2025c). Methods also differ in input handling during generation. VideoLLM-online buffers incoming visual features until decoding for the preceding frame completes (Chen et al., 2024a). Interleaved approaches organize inputs and outputs at the word, segment, or time-slot level (Chen et al., 2025; Cui et al., 2026; Tian et al., 2026; Zhang et al., 2026b; Thinking Machines Lab, 2026); Speak While Watching additionally supports parallel visual prefill and text decoding (Lin et al., 2026). System-level approaches assign perception, response activation, or answering to separate components (Zhang et al., 2024a;b; Wang et al., 2025a; Qian et al., 2025; Yao et al., 2026). Other within-model approaches admit new visual input during generation (Fu et al., 2025; Wang et al., 2026) or multiplex incoming queries with ongoing decoding (Wang et al., 2025c). UniVIA focuses on concurrent perception and complete-answer generation with shared parameters and fixed observation boundaries.

3 Method

UniVIA unifies offline video understanding and full-duplex streaming interaction within a single shared VLM. Figure 2 summarizes the training and inference paths.

3.1 Preliminaries

Let H_t denote the context containing observations and interactions available at step t. A streaming VLM predicts whether to remain silent or respond:

$$d_t \sim p_\theta(d \mid H_t), \qquad d_t \in \{\text{silence}, \text{response}\}. \tag{1}$$

A response decision initiates autoregressive generation of a complete answer $A_t = (a_{t,1}, \ldots, a_{t,L_t})$.

Streaming inference must accommodate new observations while an answer is being generated. Unlike the strategies contrasted in Section 1, we aim to continue perception within a single shared VLM while generating each answer as a complete sequence.

3.2 Asynchronous Fork-and-Commit

We achieve this by separating the inference *states* of perception and response generation while sharing the model parameters θ. A single persistent *parent* stream perceives the video and makes control decisions; each response decision spawns a transient *child* stream that writes the answer and then returns it to the parent (Figure 2, right). Algorithm 1 in Appendix A.1 summarizes the procedure.

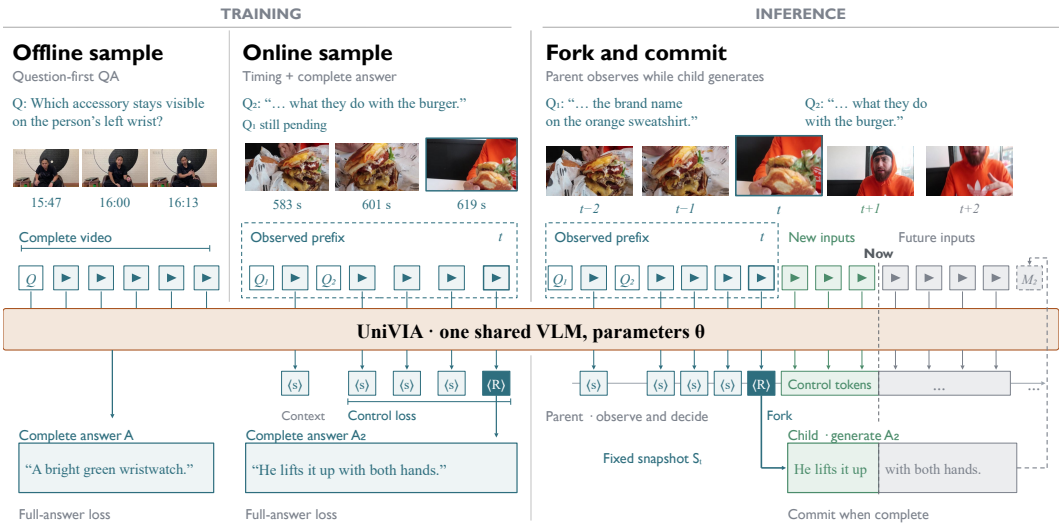

Figure 2: **Overview of UniVIA.** Offline training, online training, and streaming inference share one VLM with parameters θ. **Offline sample.** The complete video conditions the answer, which is supervised in full without timing decisions. **Online sample.** The response-ended prefix stops at the first response to Q_2, triggered at 619 s, while Q_1 remains pending. Control decisions from Q_2's arrival through the response trigger and the complete answer A_2 are supervised; earlier steps serve as context. **Fork and commit.** A response decision $\langle R \rangle$ forks a child from the fixed snapshot S_t. The child generates A_2 while the parent continues receiving observations and emitting control tokens. In the right panel, blue-green marks the context at fork, green marks progress since fork up to *Now*, and gray marks future input or output. The dashed return path commits A_2 as memory M_2 at a subsequent parent step, after the answer is complete. Horizontal progress is schematic. Frames, queries, and answers are drawn from training examples.

Parent stream. At each observation step t, the parent appends the new input to its context—the frame tokens of the step, any newly arrived query, and any answers committed since the previous step—and decodes a single control token d_t following Equation 1. The parent never decodes answer tokens. Its per-step work is therefore a prefill of the new input followed by one decoding step, regardless of how many answers are being generated at the same time.

Fork. A response decision at step t triggers **fork**: the parent submits a child request whose context is the snapshot $S_t = H_t \oplus \langle \text{response} \rangle$ and immediately resumes perception. The two streams then evolve independently under the same parameters:

$$\begin{aligned} &\text{Parent:} \quad d_{t'} \sim p_\theta(d \mid H_{t'}), \qquad t' > t, \\ &\text{Child:} \quad a_{t,j} \sim p_\theta(a \mid S_t, d_t = \text{response}, a_{t,<j}). \end{aligned} \tag{2}$$

Here, t' indexes subsequent parent observation steps while the child is active, and j indexes child decoding steps. The snapshot is fixed: the child never observes frames that arrive after t, so every answer is conditioned on a well-defined observation boundary, exactly as its training target is (Section 3.3).

Forking does not re-encode the video history. Because attention is causal, the keys and values of H_t depend only on H_t and are not modified by tokens that the parent appends later. The child therefore directly reuses the parent's KV cache of S_t, shared by reference as in paged KV-cache serving (Kwon et al., 2023), and decodes its answer on top of it. A fork does not re-prefill the history and allocates new memory only for the child's own answer tokens, independently of the length of the observed history. Several children can be active at once, for example when queries are pending simultaneously; the serving scheduler advances the parent and all children as separate logical requests.

Commit. When a child finishes, its complete answer enters a commit queue. At the next parent step, all answers completed since the previous step are inserted, in completion order, as textual memory

M into the parent's user input ahead of the new frame. Committed answers inform later control decisions and are inherited by children forked afterwards through their snapshots. The parent never sees a partial answer: an answer is either absent from H_t or present in full. Because children run independently, answers may complete in a different order than their queries were asked. Figure 2 illustrates parent control decisions continuing while a child generates A_2.

Perception latency. Let t_{dec} denote the time to decode one token and L the answer length. Under blocking execution, an observation that arrives during an answer waits for the answer to finish, adding up to $L\,t_{\mathrm{dec}}$ to its processing delay. Interleaving with chunks of c tokens bounds this delay by $c\,t_{\mathrm{dec}}$ but stretches the answer over $\lceil L/c \rceil$ observation steps, and training must align each chunk with the stream. In fork-and-commit, the parent step does not wait for any child; its only overhead is sharing the accelerator with concurrently decoding children, which does not grow with L. Figure 3 confirms this behavior: the added perception latency of our implementation stays within 63–94 ms at 1 FPS across answer lengths, whereas blocking decoding grows linearly and is $286\times$ higher at 1024 tokens.

Offline inference. Offline video understanding is the degenerate case of the same procedure: the full video and the question form H, the parent emits a single $\langle\mathrm{response}\rangle$, and the child decodes the complete answer. The same parameters thus serve offline and streaming use without any change to the model.

3.3 Training Objectives

We train the shared parameters θ using offline understanding and online interaction data; their construction is detailed in Appendix A.2.

Question-first serialization. An online query is posed before the evidence it refers to: a request such as "tell me when the chef starts peeling the potato" is issued first, and the frames that answer it arrive afterwards. In terms of input order, an online query is therefore a question-first sample (Q, V, A), which we call *QVA ordering*. Offline video QA is instead conventionally serialized video-first, (V, Q, A) (*VQA ordering*), so that the question is read with the entire video already in context. We convert offline QA to QVA ordering, keeping the question, video, and complete answer of each example unchanged. The conversion requires no additional annotation and aligns offline supervision with the order in which queries are posed online; Section 4.4 evaluates its effect on streaming and offline benchmarks.

Response-ended training samples. Consider a streaming interaction with observation steps $t = 1, \ldots, T$ and responses indexed by $k = 1, \ldots, K$ in temporal order. Response k is emitted at step τ_k, carries the complete answer $A_k = (a_{k,1}, \ldots, a_{k,L_k})$, and answers a query that arrives at step t_k^q; a query may receive several responses. We split the interaction at its responses and construct one training sample X_k per response, which mirrors the parent and child states of Fork-and-Commit at τ_k:

- **Context.** X_k contains all observations and queries up to τ_k and nothing after it.
- **Earlier responses.** For every j with $\tau_j < \tau_k$, X_k keeps the parent's control token $\langle\mathrm{response}\rangle$ at τ_j but not the answer tokens of A_j. Instead, A_j is committed as memory M_j into the first user input at or after $\tau_j + \delta_j$, where δ_j is a simulated generation delay; if $\tau_j + \delta_j > \tau_k$, A_j is still being generated and is absent from X_k. The delay grows with the answer length and is randomly perturbed, $\delta_j = L_j/r + \epsilon_j$, where r is a nominal decoding rate and ϵ_j is a random offset that emulates the decoding speed of different hardware.
- **Current response.** X_k ends with $\langle\mathrm{response}\rangle$ at τ_k followed by the complete answer A_k.

The prefix of X_k preceding A_k is exactly the snapshot S_{τ_k} that the parent would fork at inference, and A_k is the output of the corresponding child. A single sample therefore trains both streams of Equation 2: control decisions under the parent's history, in which earlier answers appear only after their simulated completion, and complete-answer generation from a fixed snapshot. A QVA offline example is the degenerate case: the query arrives at $t = 1$, the single response follows the full video, and no control decision is supervised.

Train–inference consistency. The construction of X_k is dictated by what each stream can observe at inference. A naive serialization of the interaction as a multi-turn dialogue places each answer A_j immediately after its $\langle\text{response}\rangle$ token, so that every later decision is trained on a context in which answers appear instantaneously. Under fork-and-commit, this context never occurs: while the child for response j is decoding, the parent keeps perceiving without access to A_j, and A_j becomes visible only once it is committed. Training on the naive serialization would therefore condition control decisions on text the parent cannot yet see, and would condition answers on the full interaction instead of the snapshot from which they are generated. Response-ended samples remove both mismatches: the context of each control decision in X_k contains exactly the frames, queries, earlier $\langle\text{response}\rangle$ tokens, and committed memories available to the parent at that step, and the answer A_k is conditioned only on S_{τ_k}. The simulated delay δ_j determines where committed memories appear. Its length-dependent term reflects that longer answers return later, and its random offset exposes the model to a range of commit times, so that control decisions do not depend on memories arriving after a fixed delay and remain valid when decoding speed varies across deployments.

Segment-Balanced Loss (SBL). A sample X_k contains control decisions made on behalf of several queries, but only those between the query of response k and τ_k are informative about when A_k should be triggered. We therefore define the *segment* of response k as

$$\mathcal{W}_k = \{\, t \mid s_k \le t \le \tau_k \,\}, \qquad s_k = \begin{cases} t_k^q, & \text{if } k \text{ is the first response to its query,} \\ \tau_{k'} + 1, & \text{otherwise,} \end{cases} \tag{3}$$

where k' is the preceding response to the same query. Within X_k, we supervise control tokens only at steps in $\mathcal{W}_k$; earlier decisions serve as context and are supervised in their own samples. Segments of different queries may overlap when queries are pending simultaneously; a step in the overlap is supervised in each sample whose segment contains it, with its ground-truth target, including $\langle\text{response}\rangle$ targets of other queries.

Responses are sparse within a segment: a typical $\mathcal{W}_k$ contains many $\langle\text{silence}\rangle$ targets and a single $\langle\text{response}\rangle$ target. SBL balances the two classes of control decisions inside the segment and keeps standard cross-entropy on the answer:

$$\mathcal{L}_{\text{SBL}}(X_k) = -\frac{1}{|\mathcal{W}_k| + L_k} \left[\sum_{t \in \mathcal{W}_k} \alpha_{d_t} \log p_t + \sum_{j=1}^{L_k} \log p_{k,j} \right], \tag{4}$$

where $d_t \in \{\text{silence}, \text{response}\}$ is the target at step t, $p_t = p_\theta(d_t \mid H_t)$ follows Equation 1, and $p_{k,j} = p_\theta(a_{k,j} \mid S_{\tau_k}, \langle\text{response}\rangle, a_{k,<j})$ is the child's answer likelihood in Equation 2. The class weight $\alpha_c = |\mathcal{W}_k|/(2n_c)$, where n_c counts the targets of class c in $\mathcal{W}_k$, gives both classes the same total weight $|\mathcal{W}_k|/2$; when $\mathcal{W}_k$ contains only one class, we set $\alpha_c = 1$. Since $\sum_{t \in \mathcal{W}_k} \alpha_{d_t} = |\mathcal{W}_k|$, class balancing redistributes weight between silence and response without changing the total control weight, so the normalizer remains the number of supervised tokens. Every silence decision in the segment thus remains supervised, while the single response decision is no longer outweighed by them. Answer tokens are unweighted: the reweighting targets only the balance between silence and response for each query.

4 Experiments

4.1 Implementation Details

Training. We construct our supervised fine-tuning (SFT) data from three sources: the JoyAI event-grounding dataset, the TimeLens2 dataset, and our in-house data. The JoyAI data are processed through the pipeline described in Appendix A.2, yielding approximately 270K samples. Although these samples account for only a small portion of our training data, they provide valuable coverage of everyday, real-world scenarios. The remaining SFT data are constructed from TimeLens2 and our in-house data.

As discussed in Appendix A.2, the base model exhibits substantial performance differences between the VQA and QVA orderings (Section 3.3). We therefore include offline QA converted to QVA ordering, which accounts for approximately 29% of the training input tokens. We further hypothesize

Table 1: **Streaming benchmark results** (accuracy, %). Higher scores are better. Tasks are grouped by protocol: *real-time* tasks are answerable from content available at query time, *backward* tasks need earlier content or interaction history, and *proactive* tasks require waiting for future evidence. Sources: SB = StreamingBench, OVO = OVO-Bench, OMMI = OmniMMI; scores are not averaged across benchmarks. Proprietary models (gray) are for reference; **bold** and underline mark the best and second best among the remaining models. – denotes unavailable results. Per-task results are in Appendix A.3.

Method	Real-time					Backward		Proactive		
	RTVU SB	OSU SB	MCU SB	ACU SB	RTVP OVO	SQA SB	BT OVO	PA OMMI	FAR OVO	PO SB
Proprietary models										
GPT-4o	73.3	44.5	38.4	41.2	64.5	32.8	60.8	–	53.4	56.9
Gemini-1.5-Pro	75.7	60.2	40.7	51.4	69.3	54.8	62.5	–	57.2	45.1
Open-source models										
StreamAgent	74.3	36.3	30.3	39.7	61.3	39.6	41.7	–	45.4	28.9
Streamo-7B	–	–	–	–	67.4	–	49.2	–	57.0	–
ViSpeak	70.4	61.6	36.4	49.2	66.3	39.2	57.5	–	54.3	50.8
MiniCPM-o-4.5	78.2	42.1	35.2	44.0	67.3	51.2	55.9	–	55.9	47.6
Qwen3-VL-8B-Inst.	74.1	41.9	30.0	38.4	61.6	38.0	41.0	–	37.7	52.8
JoyAI-VL-Int.	83.2	53.7	37.2	51.6	76.4	53.6	53.3	13.5	54.6	39.6
AURA	83.2	62.0	54.8	**70.8**	79.8	57.2	60.4	37.5	55.8	53.2
UniVIA (Qwen3-VL-8B)	82.9	62.1	51.2	68.4	**80.8**	**60.0**	60.4	51.5	**58.9**	64.4
UniVIA (Qwen3.5-9B)	**83.4**	**67.3**	**63.6**	68.8	79.3	**60.0**	**62.3**	**58.0**	56.3	**70.0**

that our architecture and data format can also support offline video understanding. We therefore include offline data, which account for approximately 36% of the training input tokens, and evaluate the models on the corresponding benchmarks. Section 4.3 evaluates the resulting offline video understanding. We train UniVIA (Qwen3-VL-8B) on 8×8 NVIDIA A800 GPUs for approximately 58 hours, and UniVIA (Qwen3.5-9B) on 12×8 NVIDIA A800 GPUs for approximately 49 hours.

Evaluation settings. For our models, PA uses a 100k-pixel budget and full history; PO uses the same pixel budget and the five seconds immediately preceding the query. Both use greedy decoding. AURA values are taken from its paper, whose evaluation protocol is not fully identical to ours. JoyAI-VL-Interaction (Yao et al., 2026) evaluation settings are given in Appendix A.4.

4.2 Streaming Interaction

Table 1 compares UniVIA with proprietary and open-source streaming models on real-time, backward, and proactive tasks from StreamingBench, OVO-Bench, and OmniMMI; per-task results for StreamingBench and OVO-Bench are reported in Tables 5 and 6 (Appendix A.3). UniVIA (Qwen3-VL-8B) outperforms all open-source baselines on six of the ten tasks and ties AURA on BT. The largest margins appear on proactive tasks: on PA and PO, it reaches 51.5 and 64.4, compared with 37.5 and 53.2 for AURA, which fine-tunes the same Qwen3-VL-8B backbone. AURA scores are taken from its paper under a different protocol. Qualitative examples are shown in Appendix A.6.

Compared with JoyAI-VL-Interaction, UniVIA (Qwen3-VL-8B) improves PA and PO by 38.0 and 24.8 percentage points (51.5 vs. 13.5 and 64.4 vs. 39.6), and the backward tasks SQA and BT by 6.4 and 7.1 points. Replacing the backbone with Qwen3.5-9B improves seven of the ten tasks over the Qwen3-VL-8B variant, including PA and PO (58.0 and 70.0), indicating that the recipe transfers across backbones.

4.3 Offline Video Understanding

Table 2 evaluates offline video understanding, with all models run under the same frame and pixel settings; the full breakdown is given in Appendix A.5. To measure the effect of our training on offline

Table 2: **Offline video understanding** (%). All models are evaluated by us under the same frame and pixel settings (Appendix A.5). VideoMME is averaged over the short, medium, and long splits, without (w/o) and with (w/) subtitles; VideoMME-v2 is reported without subtitles as score and accuracy; VideoMMMU reports the perception (Perc.), comprehension (Comp.), and adaptation (Adapt.) tracks; TimeLens reports mIoU on Charades (Cha.), ActivityNet (ANet), and QVHighlights (QVH). LVB.: LVBench. **Bold** and underline mark the best and second best. †At most 64 frames. ‡At most 448 frames.

	VideoMME		LVB.	VideoMME-v2		VideoMMMU			TimeLens		
Method	w/o	w/	Acc.	Score	Acc.	Perc.	Comp.	Adapt.	Cha.	ANet	QVH
Base models											
Qwen2.5-VL-7B-Inst.	65.9	71.2	44.1	10.1	24.2	57.0	46.7	28.0	34.1	26.9	28.9
Qwen3-VL-8B-Inst.	70.3	76.5	53.6	13.6	29.3	70.3	**62.0**	34.7	43.0	41.3	56.0
Streaming models											
MiniCPM-o-4.5†	68.4	71.4	42.7	12.0	28.0	47.3	27.0	25.3	7.4	9.3	8.1
ViSpeak‡	62.0	67.5	42.0	8.1	21.6	46.3	39.3	34.0	8.7	3.8	3.4
JoyAI-VL-Int.	70.1	76.5	49.9	12.3	28.2	61.7	51.7	37.0	43.7	44.4	59.8
AURA	68.5	74.9	51.5	11.9	27.8	**78.7**	51.0	35.7	36.4	27.9	38.3
UniVIA (Qwen3-VL-8B)	71.5	77.1	52.7	17.0	34.1	78.3	55.3	38.3	41.2	45.6	62.1
UniVIA (Qwen3.5-9B)	**73.0**	**78.3**	**56.6**	**18.3**	**36.3**	75.3	51.7	**42.7**	**44.9**	**49.3**	**65.3**

Table 3: **Training-data ablation** (%; Qwen3-VL-8B, 1K steps). Each variant changes one data component of (a). Streaming: PA/PO. Offline: VideoMME without/with subtitles and LVBench. **Bold**: best.

	Variant	Streaming		Offline		
		PA OMMI	PO SB	VideoMME w/o	VideoMME w/	LVB.
(a)	Full recipe	**42.5**	**21.2**	70.3	75.7	52.1
(b)	w/ JoyAI data	21.0	10.8	69.1	73.2	50.1
(c)	w/ VQA order	37.0	18.8	70.1	77.0	53.1
(d)	w/o offline QA	-	-	71.1	76.0	52.8

Table 4: **Loss ablation** (%; Qwen3-VL-8B, 1K steps). $1/N$: AURA silence weighting (Lu et al., 2026) on the same segments.

	Proactive	
Loss setting	PA	PO
$1/N$ silence weighting	15.5	8.4
w/o last-segment loss	31.7	13.6
SBL (Ours)	**42.5**	**21.2**

ability, we compare UniVIA (Qwen3-VL-8B) with its base model, Qwen3-VL-8B-Instruct, evaluated under the same protocol. UniVIA improves on eight of the eleven metrics, including VideoMME without and with subtitles ($+1.2$ and $+0.6$), VideoMME-v2 accuracy ($+4.8$), VideoMMMU perception ($+8.0$), and TimeLens QVHighlights ($+6.1$ mIoU). It is lower on LVBench (-0.9), TimeLens Charades (-1.8), and VideoMMMU comprehension (-6.7). Compared with AURA, which fine-tunes the same backbone, UniVIA (Qwen3-VL-8B) is higher on ten of the eleven metrics and trails only on VideoMMMU perception (78.3 vs. 78.7); the largest gaps appear on temporal grounding, where it reaches 45.6 and 62.1 mIoU on TimeLens ActivityNet and QVHighlights, compared with 27.9 and 38.3 for AURA. With Qwen3.5-9B, UniVIA achieves the best result among all evaluated models on nine of the eleven metrics.

4.4 Ablation Studies

All ablations fine-tune Qwen3-VL-8B for 1K steps, fewer than the models in Table 1, under the same evaluation protocol.

Training data. To assess the role of each data component, we consider two substitutions to the training recipe in Section 4.1 (Table 3). Starting from the full recipe (a), variant (b) replaces our streaming data with the original JoyAI streaming data while retaining QVA; (c) serializes the same offline QA examples in VQA instead of QVA ordering while retaining our streaming data; and (d) removes the offline QA examples, training on streaming data alone. Replacing our streaming data

with the original JoyAI data (b) lowers PA and PO by 21.5 and 10.4 points. Serializing offline QA in VQA ordering (c) lowers them by 5.5 and 2.4 points, indicating that question-first offline data contributes to proactive responding.

Loss. We ablate two components of SBL: control-token weighting and segment selection. The w/o last-segment loss variant extends control-token supervision from $\mathcal{W}_k$ to the entire sample prefix, while retaining class balancing and the answer-generation objective. We compare SBL with AURA's $1/N$ silence weighting (Lu et al., 2026) on the same segments $\mathcal{W}_k$. Both losses give silence and response equal total weight; they differ in how much the control decisions weigh against L_k answer tokens (2 vs. $|\mathcal{W}_k|$). SBL improves PA and PO by 27.0 and 12.8 points (Table 4). Removing last-segment selection lowers PA from 42.5 to 31.7 and PO from 21.2 to 13.6, corresponding to drops of 10.8 and 7.6 percentage points, respectively. These results support restricting control-token supervision to the segment associated with the current response.

Execution strategy and perception latency. To isolate concurrent execution, we compare blocking decode, KV-cache merging, and fork-and-commit under the same answer lengths and hardware (Figure 3).

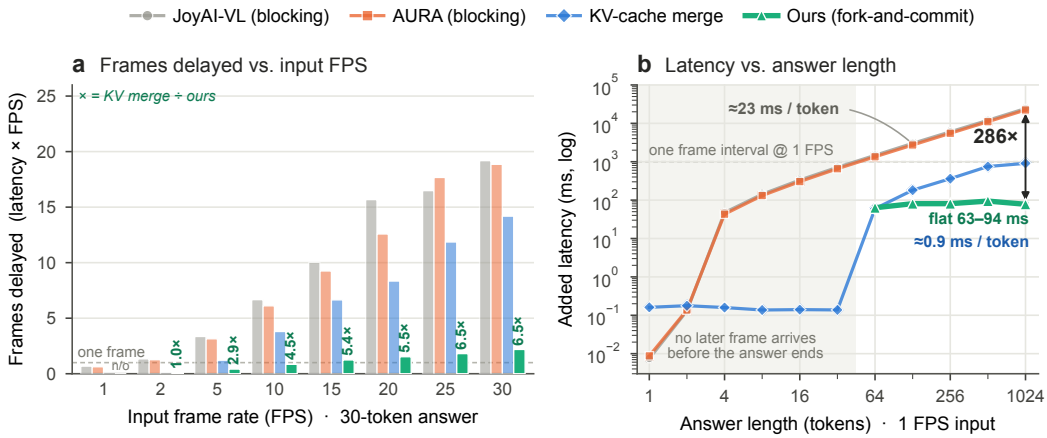

Figure 3: **Perception delay while answering** (one A800 GPU). **(a)** Frames delayed at 30 tokens. **(b)** Added latency at 1 FPS; ours stays at 63–94 ms.

5 Conclusion

We presented UniVIA, a unified architecture for offline video understanding and full-duplex streaming interaction within a shared vision-language model. UniVIA jointly learns complete-answer generation and fine-grained response decisions from offline and online supervision, preserving complete-response training while learning when to respond to emerging visual evidence. Question-first serialization further aligns offline supervision with the order in which queries and visual evidence arrive online. At inference time, asynchronous fork-and-commit separates perception and response-generation states while sharing model parameters, allowing continuous perception during complete-answer generation. Each answer is generated from a fixed causal snapshot and committed back as memory upon completion, informing subsequent response decisions and interactions. Experiments demonstrate strong streaming performance, particularly on proactive tasks, while maintaining competitive offline video understanding. Execution analysis further shows that fork-and-commit substantially reduces perception delay during response generation. Together, these results show that offline understanding and fine-grained online interaction can be jointly supported without fragmenting answer supervision or blocking ongoing perception.

AI USE STATEMENT

In this work, large language models were used as components of the automated data-construction, attribute-extraction, and benchmark-evaluation pipelines described in the paper. AI-assisted tools were also used for proofreading, grammatical correction, language refinement, and code implementation. The authors designed the research questions, methods, experiments, and scientific claims, reviewed all AI-assisted outputs, and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The main paper specifies the model architecture, training objective, datasets, evaluation protocols, and principal implementation settings. Additional training details, ablations, benchmark construction procedures, prompts, and qualitative results are provided in the appendix. Code, model checkpoints, and data resources will be released to support reproduction.

REFERENCES

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. LLaVA-OneVision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025. URL https://arxiv.org/abs/2509.23661.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. LLaVA-OneVision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026. URL https://arxiv.org/abs/2605.25979.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. 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, 2024a. URL https://arxiv.org/abs/2406.11816.

Joya Chen, Ziyun Zeng, Yiqi Lin, Wei Li, Zejun Ma, and Mike Zheng Shou. LiveCC: Learning video LLM with streaming speech transcription at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29083–29095, 2025. URL https://arxiv.org/abs/2504.16030.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024b. doi: 10.1109/CVPR52733.2024.02283. URL https://arxiv.org/abs/2312.14238.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. MiniCPM-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026. URL https://arxiv.org/abs/2604.27393.

Xin Ding, Hao Wu, Yifan Yang, Shiqi Jiang, Donglin Bai, Zhibo Chen, and Ting Cao. StreamMind: Unlocking full frame rate streaming video dialogue through event-gated cognition. *arXiv preprint arXiv:2503.06220*, 2025. URL https://arxiv.org/abs/2503.06220.

Shenghao Fu, Qize Yang, Yuan-Ming Li, Yi-Xing Peng, Kun-Yu Lin, Xihan Wei, Jian-Fang Hu, Xiaohua Xie, and Wei-Shi Zheng. ViSpeak: Visual instruction feedback in streaming videos. *arXiv preprint arXiv:2503.12769*, 2025. URL https://arxiv.org/abs/2503.12769.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626. ACM, 2023. doi: 10.1145/3600006.3613165. URL https://arxiv.org/abs/2309.06180.

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

Xinhao Li, Yuhan Zhu, Xiangyu Zeng, Yuhao Dong, Haoning Wu, Zhiqiu Zhang, Yuandong Yang, Changlian Ma, Qingyu Zhang, Yansong Shi, Xinyu Chen, Haoran Chen, Zizheng Huang, Jun Zhang, Kun Ouyang, Lin Sui, Ziang Yan, Yicheng Xu, Chenting Wang, Yinan He, Hongjie Zhang, Yi Wang, Yu Qiao, Yali Wang, Ziwei Liu, Kai Chen, and Limin Wang. VideoChat3: Fully open video MLLM for efficient and generalist video understanding. *arXiv preprint arXiv:2607.14935*, 2026. URL https://arxiv.org/abs/2607.14935.

Junming Lin, Zheng Fang, Chi Chen, Haoxuan Cheng, Zihao Wan, Fuwen Luo, Ziyue Wang, Peng Li, Yang Liu, and Maosong Sun. StreamingBench: Assessing the gap for MLLMs to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*, 2024. URL https://arxiv.org/abs/2411.03628.

Junyan Lin, Junlong Tong, Hao Wu, Jialiang Zhang, Jinming Liu, Xin Jin, and Xiaoyu Shen. Speak while watching: Unleashing TRUE real-time video understanding capability of multimodal large language models. *arXiv preprint arXiv:2601.06843*, 2026. URL https://arxiv.org/abs/2601.06843.

Xudong Lu, Yang Bo, Jinpeng Chen, Shuhan Li, Xintong Guo, Huankang Guan, Fang Liu, Dunyuan Xu, Peiwen Sun, Heyang Sun, Rui Liu, and Hongsheng Li. AURA: Always-on understanding and real-time assistance via video streams. *arXiv preprint arXiv:2604.04184*, 2026. URL https://arxiv.org/abs/2604.04184.

Junbo Niu, Yifei Li, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, et al. OVO-Bench: How far is your video-LLM from real-world online video understanding? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18902–18913, 2025. URL https://arxiv.org/abs/2501.05510.

Rui Qian, Shuangrui Ding, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Dispider: Enabling video LLMs with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24045–24055, 2025. URL https://arxiv.org/abs/2501.03218.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding. *arXiv preprint arXiv:2604.02317*, 2026. URL https://arxiv.org/abs/2604.02317.

Thinking Machines Lab. Interaction models: A scalable approach to human-AI collaboration. *Thinking Machines Lab: Connectionism*, May 2026. doi: 10.64434/tml.20260511. URL https://thinkingmachines.ai/blog/interaction-models/.

Xueyun Tian, Wei Li, Bingbing Xu, Heng Dong, Yuanzhuo Wang, and Huawei Shen. ROMA: Real-time omni-multimodal assistant with interactive streaming understanding. *arXiv preprint arXiv:2601.10323*, 2026. URL https://arxiv.org/abs/2601.10323.

Haibo Wang, Bo Feng, Zhengfeng Lai, Mingze Xu, Shiyu Li, Weifeng Ge, Afshin Dehghan, Meng Cao, and Ping Huang. StreamBridge: Turning your offline video large language model into a proactive streaming assistant. *arXiv preprint arXiv:2505.05467*, 2025a. URL https://arxiv.org/abs/2505.05467.

Pengyu Wang, Chenkun Tan, Shaojun Zhou, Wei Huang, Qirui Zhou, Zhan Huang, Zhen Ye, Jijun Cheng, Xiaomeng Qian, Yanxin Chen, Xingyang He, Huazheng Zeng, Chenghao Wang, Pengfei Wang, Hongkai Wang, Shanqing Gao, Yixian Tian, Chenghao Liu, Xinghao Wang, Botian Jiang, and Xipeng Qiu. MOSS-Video-Preview: Toward real-time video understanding via cross-attention. *arXiv preprint arXiv:2606.07639*, 2026. URL https://arxiv.org/abs/2606.07639.

Yueqian Wang, Xiaojun Meng, Yuxuan Wang, Jianxin Liang, Jiansheng Wei, Huishuai Zhang, and Dongyan Zhao. VideoLLM knows when to speak: Enhancing time-sensitive video comprehension with video-text duet interaction format. *arXiv preprint arXiv:2411.17991*, 2024. URL https://arxiv.org/abs/2411.17991.

Yueqian Wang, Songxiang Liu, Disong Wang, Nuo Xu, Guanglu Wan, Huishuai Zhang, and Dongyan Zhao. MMDuet2: Enhancing proactive interaction of video MLLMs with multi-turn reinforcement learning. *arXiv preprint arXiv:2512.06810*, 2025b. URL https://arxiv.org/abs/2512.06810.

Yuxuan Wang, Yueqian Wang, Bo Chen, Tong Wu, Dongyan Zhao, and Zilong Zheng. OmniMMI: A comprehensive multi-modal interaction benchmark in streaming video contexts. *arXiv preprint arXiv:2503.22952*, 2025c. URL https://arxiv.org/abs/2503.22952.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. *arXiv preprint arXiv:2512.21334*, 2025. URL https://arxiv.org/abs/2512.21334.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025. URL https://arxiv.org/abs/2510.09608.

Senqiao Yang, Kaichen Zhang, Zhaoyang Jia, Jinghao Guo, Yifei Shen, Xinjie Zhang, Xiaoyi Zhang, Haoqing Wang, Xiao Li, Peng Zhang, Xiang An, Yin Xie, Zhening Liu, Xun Guo, Jiahao Li, Shicheng Zheng, Jinglu Wang, Zongyu Guo, Wenxuan Xie, Zihan Zheng, Yuxuan Luo, Bin Li, and Yan Lu. Mage-VL: An efficient codec-native streaming multimodal foundation model. *arXiv preprint arXiv:2607.24904*, 2026. URL https://arxiv.org/abs/2607.24904.

Dingyu Yao, Junhao Zhou, Chenxu Yang, Chuanyu Qin, Haowen Hou, Zheming Liang, Congcong Wang, Yuhang Cao, Shenglong Ye, Shuai Xie, Shuhuan Gu, Haoyang Huang, Qingyi Si, Nan Duan, and Jiaqi Wang. JoyAI-VL-Interaction: Real-time vision-language interaction intelligence. *arXiv preprint arXiv:2606.14777*, 2026. URL https://arxiv.org/abs/2606.14777.

Linli Yao, Yicheng Li, Yuancheng Wei, Lei Li, Shuhuai Ren, Yuanxin Liu, Kun Ouyang, Lean Wang, Shicheng Li, Sida Li, Lingpeng Kong, Qi Liu, Yuanxing Zhang, and Xu Sun. TimeChat-Online: 80% visual tokens are naturally redundant in streaming videos. *arXiv preprint arXiv:2504.17343*, 2025. URL https://arxiv.org/abs/2504.17343.

Xiangyu Zeng, Kefan Qiu, Qingyu Zhang, Xinhao Li, Jing Wang, Jiaxin Li, Ziang Yan, Kun Tian, Meng Tian, Xinhai Zhao, Yi Wang, and Limin Wang. StreamForest: Efficient online video understanding with persistent event memory. *arXiv preprint arXiv:2509.24871*, 2025. URL https://arxiv.org/abs/2509.24871.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-VStream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024a. URL https://arxiv.org/abs/2406.08085.

Haowei Zhang, Shudong Yang, Jinlan Fu, See-Kiong Ng, and Xipeng Qiu. HERMES: KV cache as hierarchical memory for efficient streaming video understanding. *arXiv preprint arXiv:2601.14724*, 2026a. URL https://arxiv.org/abs/2601.14724.

Kaichen Zhang, Wei Huang, Keming Wu, Bo Li, and Xiaojuan Qi. Aero Realtime: Fully aligned input-output streams for low-latency streaming multimodal generation. *arXiv preprint arXiv:2608.08469*, 2026b. URL https://arxiv.org/abs/2608.08469.

Pan Zhang, Xiaoyi Dong, Yuhang Cao, Yuhang Zang, Rui Qian, Xilin Wei, Lin Chen, Yifei Li, Junbo Niu, Shuangrui Ding, Qipeng Guo, Haodong Duan, Xin Chen, Han Lv, Zheng Nie, Min Zhang, Bin Wang, Wenwei Zhang, Xinyue Zhang, Jiaye Ge, Wei Li, Jingwen Li, Zhongying Tu, Conghui He, Xingcheng Zhang, Kai Chen, Yu Qiao, Dahua Lin, and Jiaqi Wang. InternLM-XComposer2.5-OmniLive: A comprehensive multimodal system for long-term streaming video and audio interactions. *arXiv preprint arXiv:2412.09596*, 2024b. URL https://arxiv.org/abs/2412.09596.

Yikai Zheng, Xin Ding, Yifan Yang, Shiqi Jiang, Hao Wu, Qianxi Zhang, Weijun Wang, Ting Cao, and Yunxin Liu. Em-Garde: A propose-match framework for proactive streaming video understanding. *arXiv preprint arXiv:2603.19054*, 2026. URL https://arxiv.org/abs/2603.19054.

A APPENDIX

This appendix is organized as follows:

- In Appendix A.1, we give the pseudocode of asynchronous fork-and-commit inference (Section 3.2).
- In Appendix A.2, we describe the training data construction, including question-first (QVA) serialization of offline QA data and the timestamped streaming data pipeline.
- In Appendix A.3, we complement the category-level summary in Table 1 with per-task results on StreamingBench and OVO-Bench.
- In Appendix A.4, we detail our evaluation settings for JoyAI-VL-Interaction (Yao et al., 2026), covering recent-window QA, proactive tasks, and metrics.
- In Appendix A.5, we list the frame and pixel settings for offline benchmarks and report the full offline results.
- Finally, in Appendix A.6, we present qualitative examples of UniVIA in live captioning, memory-based question answering, and proactive question answering.

A.1 FORK-AND-COMMIT INFERENCE ALGORITHM

Algorithm 1 summarizes the inference procedure described in Section 3.2.

Algorithm 1 Asynchronous fork-and-commit inference

Input: parameters θ; observations $o_1, o_2, \ldots$ with queries
$H \leftarrow \varnothing$; $\mathcal{Q} \leftarrow$ empty commit queue
for $t = 1, 2, \ldots$ **do**
  $M \leftarrow$ all answers in $\mathcal{Q}$, in completion order; clear $\mathcal{Q}$
  $H \leftarrow H \oplus (M, o_t)$ ▷ prefill new input only
  $d_t \sim p_\theta(d \mid H)$; $H \leftarrow H \oplus d_t$
  if $d_t =$ response **then**
    $S_t \leftarrow H$ ▷ reuse the parent's KV cache
    launch CHILD(S_t) asynchronously; continue without waiting
procedure CHILD(S)
  decode $A = (a_1, \ldots, a_L)$ with $a_j \sim p_\theta(a \mid S, a_{<j})$ until end of answer
  push A to $\mathcal{Q}$ ▷ commit at the next parent step

A.2 TRAINING DATA CONSTRUCTION

Timestamped streaming interactions supervise both response timing and answers grounded in the observed prefix, but require fine-grained temporal localization and causal verification. We complement them with offline video QA to expand video-understanding and answer-generation supervision without additional temporal annotation.

Question-first QA data. Section 3.3 motivates serializing offline QA in QVA ordering. We apply this conversion to offline QA examples from [sources], retaining each example's original question Q, video V, and complete answer A. In a pilot study, [models] perform substantially worse on [benchmarks] under QVA than under VQA ordering [Fig./Table], indicating that base models are rarely trained with the question preceding the video.

Timestamped Streaming Data. Figure 4 illustrates the construction pipeline. We first construct seed video-QA examples with coarse temporal intervals and use a vision model to discard unsupported or uncertain examples. For a query Q inserted at time t_q with seed interval $[s, e]$, the model scans $[t_q, e + \Delta]$ at a low frame rate to propose candidate response windows $\{W_k\}_{k=1}^{K}$, where Δ is a fixed search margin beyond the seed interval.

Each candidate window is then examined at a higher frame rate. Let $\mathcal{T}_k$ denote the sampled timestamps within W_k, and let $\hat{c}_Q(t) \in \{0, 1\}$ denote the vision model's judgment of whether the query's response

condition is satisfied at time t. The response timestamp is the earliest verified sampled time:

$$\tau_k = \min \{t \in \mathcal{T}_k \mid \hat{c}_Q(t) = 1\}. \tag{5}$$

Windows without a verified response time are discarded.

Using the retained timestamps and their visual evidence, a vision model regenerates the complete answers, yielding timestamped responses $\{(\tau_k, A_k)\}$. The model is instructed to include only information available at each response time τ_k. A final model-based verification step rejects interactions with incorrect response conditions, timing, response counts, or answer content, including future-information leakage.

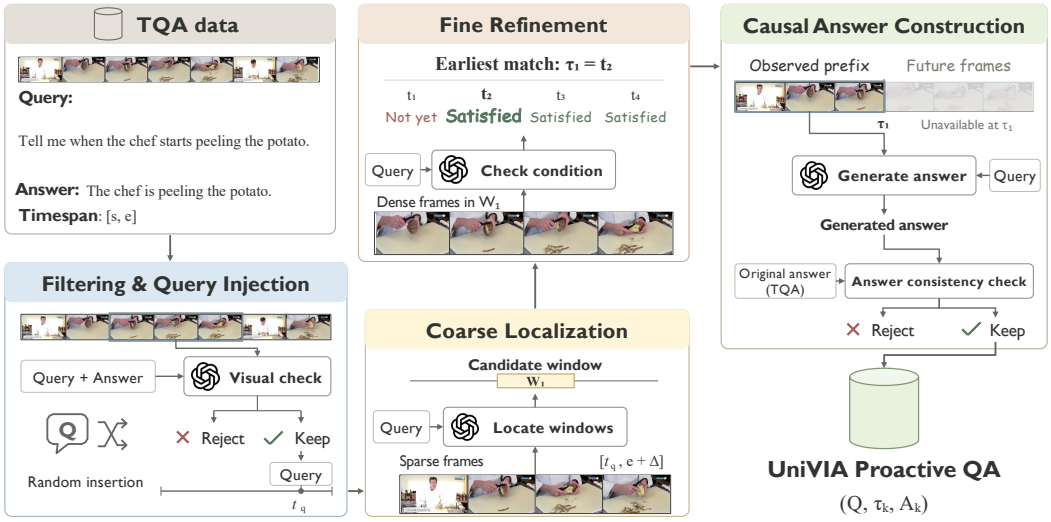

Figure 4: **Timestamped streaming data construction.** Seed video-QA examples are filtered for visual consistency, and queries are inserted at random times t_q. Coarse localization proposes candidate response windows; fine-grained refinement identifies the earliest sampled time τ_k satisfying the query within each window. Answers are generated from the query and the video prefix observed up to τ_k, then retained when consistent with the original seed answer.

Figure 5 shows a resulting training interaction.

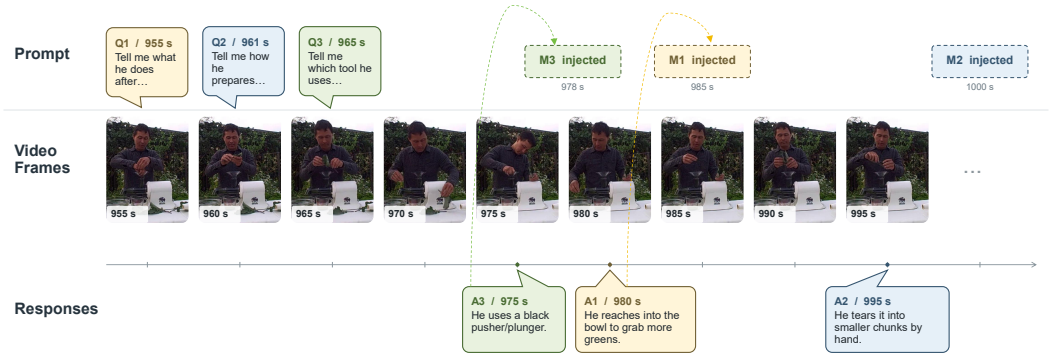

Figure 5: **Training data examples.** Streaming question-answering data: questions may arrive at any point during the video, requiring the model to continuously monitor the video and prepare to answer them.

A.3 Full Streaming Benchmark Results

Tables 5 and 6 report per-task results on StreamingBench and OVO-Bench. Table 1 in the main text summarizes their category-level scores.

Table 5: Quantitative comparison on StreamingBench in terms of accuracy (%). We report performance on each fine-grained task, along with the average results for Real-Time Visual Understanding (RTVU), Omni-Source Understanding (OSU), Contextual Understanding (CU), and the overall average. Higher scores are better. Bold and underline mark the highest and second-highest reported scores among open-source models and ours; protocols differ. – denotes unavailable results. For the two Qwen-based UniVIA models, All is weighted over 4,495 questions, excluding five questions from one corrupted video.

Method	RTVU											OSU					CU					All
	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	Avg	ER	SCU	SD	MA	Avg	MCU	ACU	SQA	PO	Avg	
Proprietary Models																						
GPT-4o	77.1	80.5	83.9	76.5	70.2	83.8	66.7	62.2	69.1	49.2	73.3	41.2	37.2	43.6	56.0	44.5	38.4	41.2	32.8	56.9	38.7	60.2
Gemini-1.5-Pro	79.0	80.5	83.5	79.7	80.0	84.7	77.8	64.2	72.0	48.7	75.7	46.8	39.6	74.9	80.0	60.2	40.7	51.4	54.8	45.1	48.7	67.1
Open-Source Models																						
StreamAgent	79.6	78.3	79.3	75.9	74.7	76.9	82.9	66.3	73.7	55.4	74.3	35.9	26.3	38.9	44.0	36.3	30.3	39.7	39.6	28.9	34.6	57.0
ViSpeak	79.8	71.1	81.4	78.8	74.5	70.1	63.9	64.2	71.4	28.0	70.4	47.2	56.4	61.6	81.2	61.6	36.4	49.2	39.2	50.8	43.9	62.6
Qwen3-VL-8B-Inst.	80.1	76.6	81.1	81.1	77.0	73.2	80.6	63.4	68.8	**57.0**	74.1	44.4	27.2	46.4	49.6	41.9	30.0	38.4	38.0	52.8	39.8	59.3
MiniCPM-o-4.5	80.9	**88.3**	82.0	85.0	78.9	85.4	78.7	67.1	75.6	56.0	78.2	44.4	28.0	41.6	54.4	42.1	35.2	44.0	51.2	47.6	44.5	62.7
JoyAI-VL-Int.	–	–	–	–	–	–	–	–	–	–	83.2	–	–	–	–	53.7	37.2	51.6	53.6	39.6	–	–
AURA	87.5	84.4	93.4	89.5	83.2	87.9	**86.1**	76.4	82.4	47.7	83.2	56.4	48.4	62.8	80.4	62.0	54.8	**70.8**	57.2	53.2	59.0	73.1
Ours																						
UniVIA (OV2-8B)	84.3	80.5	88.3	87.5	76.1	86.6	83.3	66.7	67.0	53.7	78.3	39.2	31.6	46.4	54.0	42.8	24.0	47.2	41.2	57.6	42.5	62.4
UniVIA (Qwen3-VL-8B)	87.5	87.5	**93.7**	91.6	86.0	92.8	84.3	77.2	**83.5**	23.9	82.9	57.2	52.8	60.0	78.4	62.1	51.2	68.4	**60.0**	64.4	61.0	73.4
UniVIA (Qwen3.5-9B)	**90.2**	83.6	91.5	**91.9**	**89.2**	**94.4**	83.3	**80.1**	81.0	27.7	**83.4**	**61.6**	**60.0**	**64.4**	**83.2**	**67.3**	**63.6**	68.8	**60.0**	**70.0**	**65.6**	**75.9**

Table 6: Quantitative comparison on OVO-Bench in terms of accuracy (%). We report performance on each fine-grained task, along with the average results for Real-Time Visual Perception (RTVP), Backward Tracing (BT), Forward Active Responding (FAR), and the overall average. Higher scores are better. Bold and underline mark the highest and second-highest reported scores among open-source models and ours; protocols differ. – denotes unavailable results. For the two Qwen-based UniVIA models, All is the macro-average of RTVP, BT, and FAR.

Method	RTVP							BT				FAR				All
	OCR	ACR	ATR	STU	FPD	OJR	Avg	EPM	ASI	HLD	Avg	REC	SSR	CRR	Avg	
Proprietary Models																
GPT-4o	69.8	64.2	71.6	51.1	70.3	59.8	64.5	57.9	75.7	48.7	60.8	73.2	27.6	59.4	53.4	59.5
Gemini-1.5-Pro	85.9	67.0	79.3	58.4	63.4	62.0	69.3	58.6	76.4	52.6	62.5	35.5	74.2	61.7	57.2	63.0
Open-Source Models																
StreamAgent	71.2	53.2	63.6	53.9	67.3	58.7	61.3	54.8	58.1	25.8	41.7	35.9	48.4	52.0	45.4	49.4
Streamo-7B	77.2	66.1	76.7	45.5	66.3	72.8	67.4	55.6	58.1	33.9	49.2	30.8	57.6	**82.5**	57.0	57.9
ViSpeak	75.2	58.7	71.6	51.1	74.3	66.9	66.3	59.9	48.7	**64.0**	57.5	33.8	68.5	60.4	54.3	61.1
MiniCPM-o-4.5	73.2	65.1	75.0	59.0	69.3	62.5	67.3	55.2	66.9	45.7	55.9	**43.0**	66.0	58.8	55.9	59.7
Qwen3-VL-8B-Inst.	54.5	58.7	68.5	**70.3**	68.1	49.5	61.6	58.8	41.1	23.1	41.0	21.2	51.3	40.4	37.7	46.8
JoyAI-VL-Int.	–	–	–	–	–	–	76.4	–	–	–	53.3	34.1	72.3	57.5	54.6	–
AURA	89.9	79.8	80.2	70.2	77.2	**81.5**	79.8	54.9	67.6	58.6	60.4	30.4	75.8	61.3	55.8	65.3
Ours																
UniVIA (OV2-8B)	83.9	67.0	78.4	59.0	72.3	63.0	70.6	59.6	63.5	1.6	41.6	41.5	70.6	67.9	**60.0**	57.4
UniVIA (Qwen3-VL-8B)	**91.9**	**82.6**	**87.9**	68.0	75.2	79.3	**80.8**	59.6	**68.2**	53.2	60.4	40.8	**76.0**	60.0	58.9	**66.7**
UniVIA (Qwen3.5-9B)	**91.9**	79.8	81.9	66.9	**79.2**	76.1	79.3	**62.3**	**68.2**	56.5	**62.3**	38.1	73.3	57.5	56.3	66.0

A.4 JoyAI Evaluation Settings

Recent-window QA. Each query uses up to 30 of the most recent frames within its allowed temporal interval, sampled at 1 FPS with timestamps. Images use a minimum of 4,096 pixels and a maximum of 200,000 pixels. We use the original task prompts, zero-shot greedy decoding, and a maximum of 256 generated tokens. Qwen3.5-9B scores the generated answers using the original yes/no and 0–5 prompts, with temperature zero and thinking disabled.

Proactive tasks. For PA and PO, we report the higher score from the two evaluated configurations for each task. PA uses a 65,536-pixel limit and measures the accuracy of the first response time (27/200, 13.5%). PO runs with the summary model enabled and the background solver disabled, and measures joint timing and answer accuracy (99/250, 39.6%).

Metrics. SG and MD Avg measure the fraction of fully correct dialogues; 1st–3rd report cumulative accuracy through each turn. FAR is the macro-average of REC, SSR, and CRR, yielding 54.6% at 200K pixels. Questions on which inference failed (83 of 9,175) are counted as incorrect.

A.5 Offline Evaluation Settings and Full Results

Offline benchmarks are evaluated with lmms-eval using uniform frame sampling and a fixed pixel budget per frame, following the LLaVA-OneVision-2 evaluation settings (Table 7). Every model is evaluated under the same settings, except that MiniCPM-o-4.5 and ViSpeak are capped at 64 and 448 frames, respectively. Table 8 reports the full results.

Table 7: Offline evaluation settings: number of uniformly sampled frames and pixel budget per frame. S / M / L: short / medium / long.

Benchmark	Split	Frames	Pixels / frame
VideoMME, w/o subtitles	S / M / L	128 / 256 / 640	321,489 / 136,900 / 102,400
VideoMME, w/ subtitles	S / M / L	128 / 256 / 640	233,289 / 128,164 / 84,100
LVBench	–	768	84,100
VideoMME-v2, w/ subtitles	–	64	330,000
VideoMME-v2, w/o subtitles	–	64	330,000
TimeLens	Charades / ANet / QVH	128	153,664

Table 8: Full offline results (%). Overall VideoMME scores average the three equally sized duration splits. **Bold** and underline mark the best and second best in each row. Qwen2.5-VL and Qwen3-VL denote the official instruction-tuned models. †At most 64 frames. ‡At most 448 frames.

Benchmark	Qwen2.5-VL (7B)	Qwen3-VL (8B)	MiniCPM-o-4.5†	ViSpeak‡	JoyAI-VL-Int.	AURA	UniVIA (Qwen3-VL-8B)	UniVIA (Qwen3.5-9B)
VideoMME, w/o subtitles								
Short	76.4	79.2	78.8	72.0	78.4	77.8	78.6	**80.7**
Medium	66.6	69.8	68.8	59.4	70.1	69.9	**72.8**	72.7
Long	54.7	61.9	57.6	54.7	61.8	57.9	63.2	**65.6**
Overall	65.9	70.3	68.4	62.0	70.1	68.5	71.5	**73.0**
VideoMME, w/ subtitles								
Short	78.3	**81.8**	79.3	74.3	80.2	80.6	80.0	81.0
Medium	72.4	76.9	75.2	65.0	76.7	75.9	78.6	**81.0**
Long	62.9	70.7	59.6	63.3	72.6	68.3	72.6	**73.0**
Overall	71.2	76.5	71.4	67.5	76.5	74.9	77.1	**78.3**
Long video								
LVBench	44.1	53.6	42.7	42.0	49.9	51.5	52.7	**56.6**
VideoMME-v2								
Score, w/o subtitles	10.1	13.6	12.0	8.1	12.3	11.9	17.0	**18.3**
Acc., w/o subtitles	24.2	29.3	28.0	21.6	28.2	27.8	34.1	**36.3**
Score, w/ subtitles	14.9	20.3	17.9	11.4	18.5	18.7	22.3	**24.5**
Acc., w/ subtitles	31.4	37.2	35.0	26.8	35.9	36.2	40.6	**43.1**
VideoMMMU								
Perception	57.0	70.3	47.3	46.3	61.7	**78.7**	78.3	75.3
Comprehension	46.7	**62.0**	27.0	39.3	51.7	51.0	55.3	51.7
Adaptation	28.0	34.7	25.3	34.0	37.0	35.7	38.3	**42.7**
TimeLens (mIoU)								
Charades	34.1	43.0	7.4	8.7	43.7	36.4	41.2	**44.9**
ActivityNet	26.9	41.3	9.3	3.8	44.4	27.9	45.6	**49.3**
QVHighlights	28.9	56.0	8.1	3.4	59.8	38.3	62.1	**65.3**

A.6 Qualitative Examples

Figure 6 shows inference examples of UniVIA.

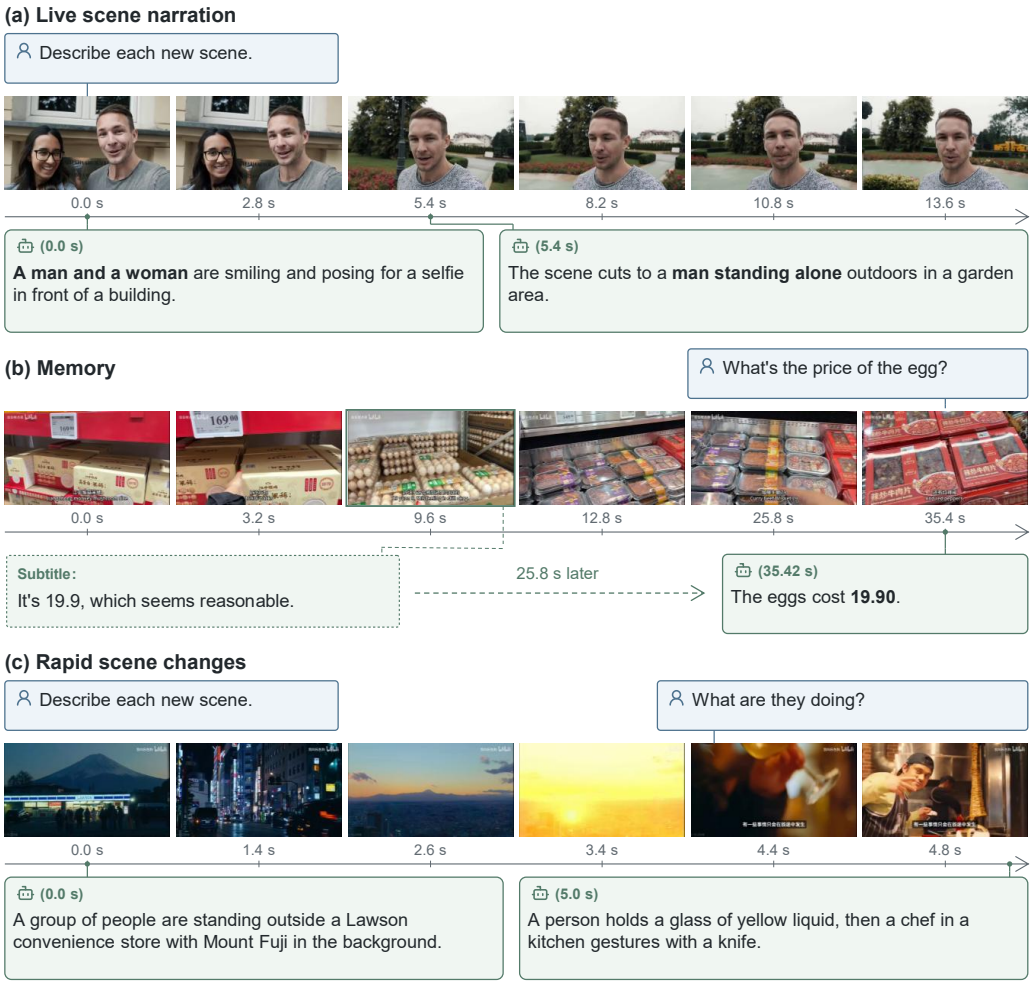

Figure 6: **Qualitative examples** of UniVIA in three usage modes: live captioning, memory-based question answering, and proactive question answering.