AERO REALTIME: DUPLEX VIDEO INTERACTION WITHIN A UNIFIED CAUSAL STREAM

Anonymous authors
Paper under double-blind review

ABSTRACT

The world continues to change while a video-language model generates a response. Continuous interaction requires new observations to enter during generation, yet turn-formatted execution defers them to a later input turn, while separate perception and response paths introduce additional coordination requirements. Can a video-language model support continuous input and output while preserving the training and incremental execution primitives of causal language models? We introduce Aero Realtime, a 4B model that places video, audio, silence, and lexical output on one causal timeline. At each approximately 80-ms audio slot, the model incorporates newly available visual states and predicts one lexical token or silence. Our contributions are threefold. **1)** An append-only causal-stream architecture removes the native turn boundary, enabling continuous observation and generation within a single causal stream while preserving causal-language-model training and incremental execution. **2)** We assemble 1.5M QA samples over 713K unique videos and address dense silence supervision with silence-label masking to adapt a pretrained VLM to the interface. **3)** We provide parallel training and incremental serving as a distributed training and deployment implementation. Aero Realtime achieves 48.31 on OVOBench and records 84-ms median and 173-ms P95 processing lag over the first 20 minutes of a continuous stream on four NVIDIA A6000 GPUs. These results establish the learning and execution feasibility of duplex video interaction within a unified causal stream, with longer-run latency and autonomous response timing remaining limitations.

1 INTRODUCTION

The world does not pause while an interactive video-language model generates a response. An object may move, an action may change, or a user may supply new information before the response is complete. Continuous interaction therefore requires more than processing successive video clips: new observations must remain available to the model throughout generation, the system must represent periods without textual output, and it must keep pace with the incoming stream during both silence and generation. The conventional turn-formatted interface instead separates these activities, as illustrated in Figure 1(a): a multimodal input prefix is encoded before an output suffix is decoded, deferring observations that arrive during generation to a subsequent turn. Causal Transformers do not inherently require this separation, but removing it requires a representation and training objective in which observations and outputs can continue to enter the same sequence.

Existing approaches make different choices about interaction and execution, yet share a boundary that our formulation removes. Repeated invocation of a turn-formatted VLM reuses established pipelines, but observations arriving during an output suffix remain unavailable to that response. Streaming methods arrange incremental observations and temporal context (Chen et al., 2025; Xu et al., 2026), while others coordinate perception, activation, and response: Dispider separates perception and decision from asynchronous reaction, and StreamBridge uses an activation model to trigger a VideoLLM (Qian et al., 2025; Wang et al., 2025). Separate perception and response modules require additional coordination—when to activate generation and how to transfer perceptual state—which increases training and deployment complexity. Systems that target concurrent observation and generation—MOSS-Video-Preview, AURA, and Interaction Models—handle streaming input through separate perception channels, periodic updates, or micro-turns, absorbing new observations at these boundaries rather than within a single active causal stream (Wang et al., 2026;

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.

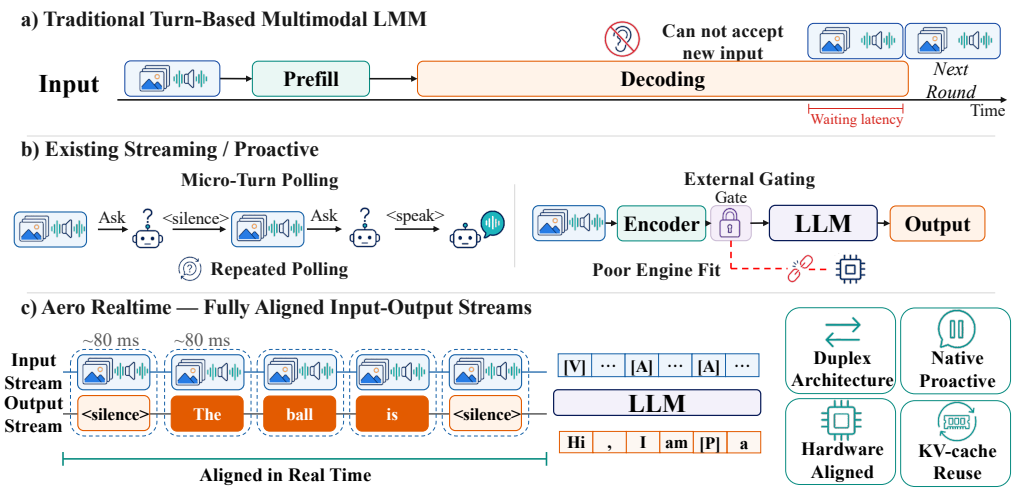


Figure 1: Realtime multimodal architectures: (a) non-duplex turn-based generation, (b) segmented streaming updates and separate response control, and (c) Aero Realtime's aligned input-output stream.

Lu et al., 2026; Thinking Machines Lab, 2026). Across these designs, new observations condition later outputs only after a channel, update, or trigger boundary. This motivates a complementary objective: supporting continuous observation and generation while preserving the core training and incremental execution mechanisms of a causal language model.

Achieving both requires more than interleaving inputs at inference time, and raises three coupled problems: removing the native turn boundary so that input and output advance at the same rate; learning when to emit a response versus remain silent on a dense temporal axis; and keeping long-stream training and incremental inference efficient. Our central question is: *can a pretrained video-language model support continuous observation and generation while retaining causal language-model training and incremental execution primitives?*

We introduce **Aero Realtime**, a 4B streaming multimodal model that advances perception and textual output on a shared temporal grid, as shown in Figure 1(c). Following the audio-text recurrence of Voxtral Realtime (Mistral AI et al., 2026), we combine the preceding output-token embedding with the current audio state. We extend this interface to video-conditioned response generation by inserting timestamped visual tokens into the causal stream and training the language-model head to predict one lexical token or silence at each approximately 80-ms audio slot. New observations can therefore condition the remaining response while it unfolds. Silence and language share one autoregressive objective, without a separate response gate.

The same construction connects the interaction interface to incremental execution. Each step appends the new audio-text state and newly available visual tokens, then predicts the next output. We characterize the conditions under which these extensions leave earlier hidden states and KV entries unchanged (Section 3.5). This permits reuse of causal attention and KV-cache primitives without repeatedly encoding the accumulated history. Integration still requires a streaming input builder and modality-specific encoding; attention to the growing history also retains a context-dependent cost. Here *duplex* denotes admission of observations during textual generation; an external streaming text-to-speech model can consume the resulting text, though spoken interruption handling remains out of scope.

Learning this interface is a separate challenge from defining its recurrence. We assemble 1.5M QA samples over 713K unique videos, including 103k realtime QA samples constructed using only the available video prefix and prior turns. Additional video datasets enter the same format through temporal metadata or terminal answering after the observed video. These sources provide defined output times with different levels of temporal supervision. Dense silence targets can nevertheless cause the adapted model to remain almost entirely silent. We address this failure with silence-label masking; in our Stage 1 ablation, it raises the OVOBench average from 6.40 without masking to 42.36. To make long-stream training practical, we distribute frames according to visual workload

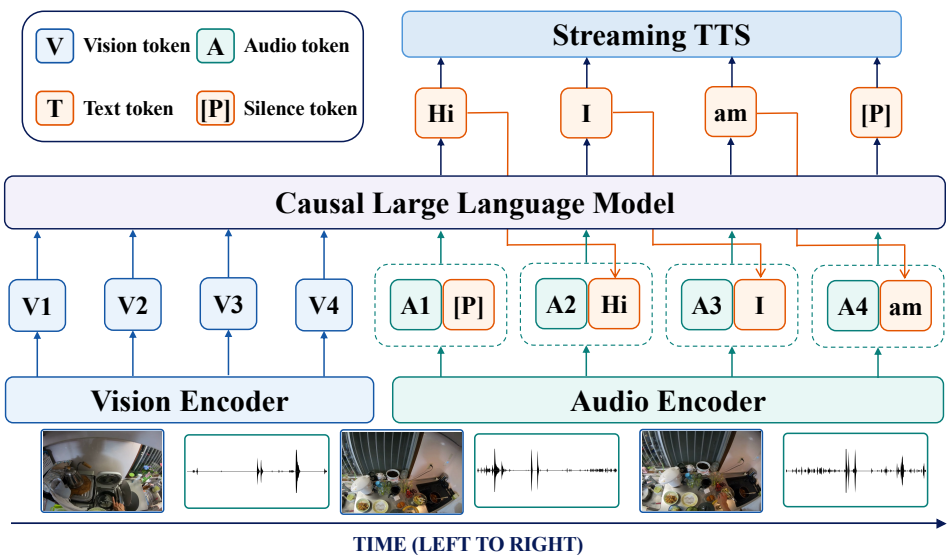


Figure 2: Aero Realtime architecture. Timestamped visual tokens enter a causal stream of audio-text states on an approximately 80-ms slot grid. Each audio state is fused with the preceding output embedding and predicts either a lexical token or the silence token [P]. An external streaming text-to-speech engine can consume the resulting textual stream incrementally.

and partition audio and language sequences across GPUs. These designs adapt both the learning signal and the computation to the shared causal interface.

Our experiments examine training feasibility, processing lag, and retained video understanding. The training stack exceeds 1,600 tokens per GPU per second on A100-40G GPUs. On four NVIDIA A6000 GPUs, incremental inference records 84-ms median and 173-ms P95 processing lag over the first 20 minutes of a continuous stream. On OVOBench, Aero Realtime achieves an average score of 48.31, trailing the strongest baselines while retaining useful video-understanding capability (Table 1). These experiments support the feasibility of concurrent observation and textual generation; they do not establish the quality of autonomous response timing. Our contributions are:

- We introduce an architecture that removes the native turn boundary, letting each output condition on observations that arrive while generation continues, and characterize the conditions under which incremental execution preserves previously computed states.
- We provide a data and training recipe: 1.5M QA samples over 713K unique videos with slot-aligned supervision and silence-label masking, enabling a pretrained VLM to learn the streaming interface while retaining video-understanding capability.
- We implement modality-aware distributed training and incremental serving using existing causal-model primitives, and evaluate training efficiency and the operating limits of continuous inference.

2 Related Work

Streaming video understanding. Prior streaming video-language models process long-running inputs incrementally, e.g., by interleaving timestamped transcripts (Chen et al., 2025), maintaining bounded caches (Xu et al., 2026), or using recent-frame windows, codec-derived representations, and general streaming backbones (Shen et al., 2026; Yang et al., 2026b; Li et al., 2026). MOSS-Video-Preview separates perception from generation through a cross-attention side channel (Wang et al., 2026), while AURA and JoyAI-VL-Interaction add proactive response or agent delegation (Lu et al., 2026; Yao et al., 2026). In all cases, new observations enter at separate-channel, update, or trigger boundaries rather than within an active output stream.

Duplex multimodal models. Duplex modeling has developed largely for speech: Moshi and MoshiVis model time-aligned speech or image streams (Défossez et al., 2024; Royer et al., 2025); Voxtral Realtime applies an audio-plus-previous-token recurrence to ASR (Mistral AI et al., 2026),

which our recurrence extends with timestamped visual states for proactive response; and Interaction Models realize concurrency through time-aligned micro-turns (Thinking Machines Lab, 2026). Aero Realtime instead uses one 80-ms slot grid where each audio slot predicts a lexical token or silence.

3 METHOD

3.1 PRELIMINARIES: MULTIMODAL CAUSAL TRANSFORMERS

Let $\mathcal{V}$, $\mathcal{A}$, and $\mathcal{T}$ denote visual observations, audio, and text, respectively. A conventional multimodal language model maps each supported modality into the hidden dimension of a causal Transformer,

$$\mathbf{V} = P_v(f_v(\mathcal{V})), \qquad \mathbf{A} = P_a(f_a(\mathcal{A})), \qquad \mathbf{T} = E(\mathcal{T}), \tag{1}$$

where f_v and f_a are modality encoders, P_v and P_a are projectors, and E is the token embedding table. Existing multimodal LLMs serialize these states with text into a causal sequence (Qwen Team, 2025; An et al., 2026). In the turn-formatted setting, an input prefix $\mathbf{M} = \mathrm{Serialize}(\mathbf{V}, \mathbf{A}, \mathbf{T})$ precedes response tokens $t_{1:L}$, and training minimizes

$$\mathcal{L}_{\text{turn}} = -\sum_{i=1}^{L} \log p_\theta(t_i \mid \mathbf{M}, t_{<i}). \tag{2}$$

This prefix–suffix formulation cannot accommodate observations that arrive after decoding begins, which motivates the append-only stream developed below.

3.2 MODEL ARCHITECTURE

Figure 2 presents the overall architecture of Aero Realtime. We preserve modality-specific vision and audio towers but replace turn-level conditioning with a dense text stream aligned to the audio timeline. At each audio slot, the current audio representation is combined with the preceding text-stream token embedding before entering the causal language model, which predicts the next aligned lexical token or the silence token [P].

Aero Realtime divides continuous time into N audio slots. The audio encoder and projector temporally downsample the waveform such that each projected audio hidden state represents approximately 80 ms of input. Let a_t denote the audio segment at slot t, v_j the j-th sampled video frame with timestamp τ_j, and y_t the output token aligned with slot t. The modality-specific representations are

$$z_j^v = P_v(f_v(v_j)), \qquad z_t^a = P_a(f_a(a_t)), \tag{3}$$

where $z_j^v \in \mathbb{R}^{n_j \times d}$ is the sequence of n_j visual tokens from frame j, and $z_t^a \in \mathbb{R}^d$ is the audio state at slot t. We denote the timestamp-ordered visual-token sequence available at audio slot t as

$$\mathcal{Z}_{\le t}^v = \mathrm{Concat}_{j:\,\tau_j \le \tau_t}\, z_j^v, \tag{4}$$

where frames are concatenated in timestamp order and τ_t is the wall-clock timestamp of the audio slot.

Each audio slot has exactly one output token,

$$y_t \in \mathcal{V}_{\text{text}} \cup \{[\mathrm{P}]\}, \tag{5}$$

where [P] is a special silence token with a learned embedding $E([\mathrm{P}]) \in \mathbb{R}^d$. A lexical token indicates that the response advances at the current slot, whereas [P] indicates that the model remains silent. Our causal audio-text recurrence follows the realtime formulation of Voxtral Realtime (Mistral AI et al., 2026); Aero Realtime extends it to multimodal generation by incorporating timestamp-aligned visual states into the same causal stream. To carry the output state along the audio timeline, we fuse the current audio hidden state with the preceding output-token embedding,

$$e_t^a = z_t^a + E(y_{t-1}), \qquad y_0 = [\mathrm{P}]. \tag{6}$$

Thus, listening context and response history are represented at every slot. An L-token response occupies at least L 80-ms slots while perception continues, yielding a maximum lexical output rate of 12.5 tokens/s as the input timeline advances.

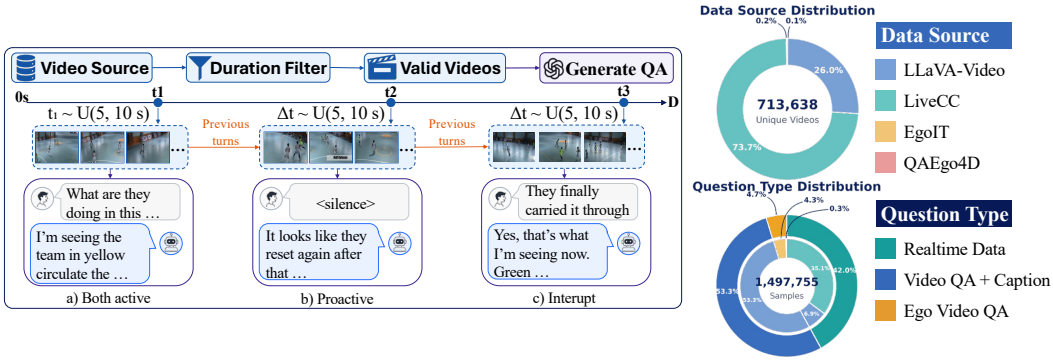


Figure 3: Realtime QA construction and training-corpus composition. The final corpus contains 1.5M QA samples over 713K unique videos with slot-level time-aware supervision, including 103k native realtime QA samples generated using only past visual context and turns.

Visual states remain independent tokens, whereas each audio state shares a position with the preceding text state. Following Qwen2.5-Omni and Qwen3-Omni (Xu et al., 2025a;b), the visual tokens from each sampled frame are ordered by timestamp and inserted before the first fused audio-text slot at or after that timestamp.

Let h_t denote the Transformer hidden state associated with the fused token e^a_t. It is conditioned on all preceding fused audio-text tokens and all visual tokens available by time t,

$$h_t = \text{Transformer}\big(\text{Interleave}(\mathcal{Z}^v_{\le t}, e^a_{\le t})\big)\,. \tag{7}$$

The model predicts the aligned output using the original language-model head,

$$p_\theta(y_t \mid \mathcal{Z}^v_{\le t}, a_{\le t}, y_{<t}) = \text{softmax}(W_{\text{lm}} h_t). \tag{8}$$

Predicting either a lexical token or [P] represents output timing and content within one distribution.

Because lexical tokens are emitted one per slot, the non-silence outputs define a textual stream that advances on the same clock as the audio input,

$$\mathcal{S} = \{y_t \in \mathcal{V}_{\text{text}} \mid t = 1, \ldots, N\}, \qquad \Delta t_{\text{speech}} = \Delta t_{\text{audio}} = 80 \text{ ms}. \tag{9}$$

An external streaming text-to-speech model consumes $\mathcal{S}$ incrementally as tokens arrive; it is not part of the model, and synthesized speech has its own duration.

3.3 Data Curation

Training Aero Realtime requires supervision that specifies both answer content and its placement along a continuous stream. We construct realtime interactions using only the available visual prefix and convert heterogeneous video datasets into a common slot-aligned format. The resulting corpus contains 1.5M QA samples over 713K unique videos, each with a defined response time on the streaming timeline. It includes 103k native realtime QA samples generated by our pipeline and additional converted datasets. A defined response time supplies alignment supervision, but does not by itself identify when an unsolicited response would be useful.

Causal Realtime Data Construction. We construct realtime interactions while strictly preserving causal availability of visual information: at each sampled timestamp, a generator observes *only* the video prefix available up to that time, together with previously generated QA turns as textual memory, and produces a temporally grounded QA pair. Repeating this process yields a time-ordered interaction trajectory without exposing future frames, so each response is grounded only in information available when it is produced. This pipeline generates 103k realtime QA samples; sampling details are provided in Appendix C.

To broaden the supervision beyond synthetically constructed realtime interactions, we further incorporate heterogeneous existing video corpora, including video QA and caption data from LLaVA-Video (Zhang et al., 2025b) and LiveCC (Chen et al., 2025), and egocentric video QA from EgoIT (Yang et al., 2026a) and QAEgo4D (Patel et al., 2025). Data with temporal annotations are converted directly into realtime supervision, while conventional question-answer and caption data are adapted through the unified aligned-stream formulation described below.

Unified Aligned-Stream Conversion. A key feature of our pipeline is that realtime and conventional instruction data can supervise the *same* duplex interface. Temporally annotated responses are aligned to the first audio slot at or after their timestamp, with the remaining positions supervised by the silence token [P]; sources with event or clip metadata are placed at the corresponding point on the timeline, and conventional video QA is treated as a terminal case answered after the observed video. Every converted example therefore has a defined response time, allowing native realtime, metadata-aligned, and terminal QA data to train one slot-level streaming objective. Conversion details are provided in Appendix C.

3.4 Model Training

3.4.1 Training Objective

The architecture retains the standard next-token prediction objective with cross-entropy loss, but replaces the turn-level prefix conditioning in Equation 2 with the aligned multimodal history:

$$\mathcal{L}_{\text{stream}} = -\sum_{t=1}^{N} m_t \log p_\theta(y_t \mid x_{\le t}, y_{<t}), \tag{10}$$

where $x_{\le t} = (\mathcal{Z}^v_{\le t}, a_{\le t})$ denotes all observations causally available up to slot t, and m_t masks positions excluded from supervision. Silence targets dominate the aligned stream, so we independently exclude a fraction $r = 0.95$ of these targets from the loss while retaining lexical targets. This masking changes loss participation without removing positions from the input stream. For packed training, samples are shifted independently to prevent loss terms from crossing sequence boundaries.

3.4.2 Three-Level Parallel Training Infrastructure

Training long aligned multimodal streams introduces distinct computational bottlenecks in the vision tower, audio tower, and language model. We therefore develop a *modality-aware three-level parallel training strategy*, where the three levels refer to these three model components. As illustrated in Figure 4a, each component is partitioned along its dominant workload dimension.

Workload-Balanced Frame Parallelism. Conventional text-token packing does not balance computation in the vision tower: packed samples may contain widely varying numbers and resolutions of video frames, while LM-oriented sequence parallelism typically replicates visual encoding across ranks. To distribute visual computation explicitly, we use the number of visual patches in frame f as a workload proxy, $w_f = H_f W_f$ and assign frames to S sequence-parallel ranks by approximately minimizing

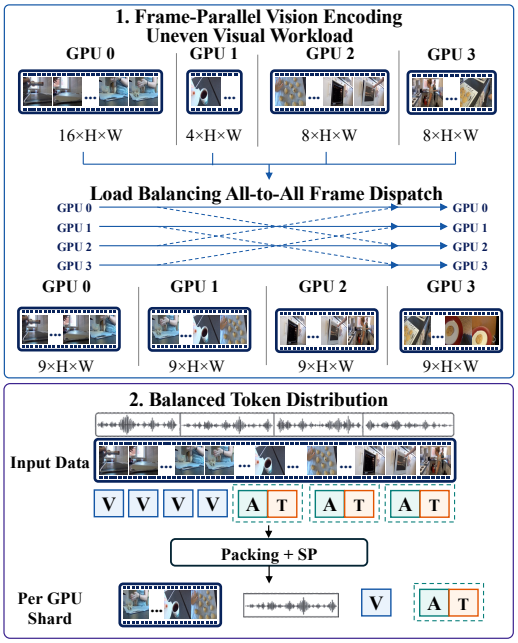
$$\max_{r \in \{1,\dots,S\}} \sum_{f:\, \pi(f)=r} w_f, \tag{11}$$

where $\pi(f)$ denotes the rank assigned to frame f. A deterministic redistribution moves frames from overloaded to underloaded ranks before the vision forward pass and gathers the resulting features afterward, balancing visual computation and lowering peak memory.

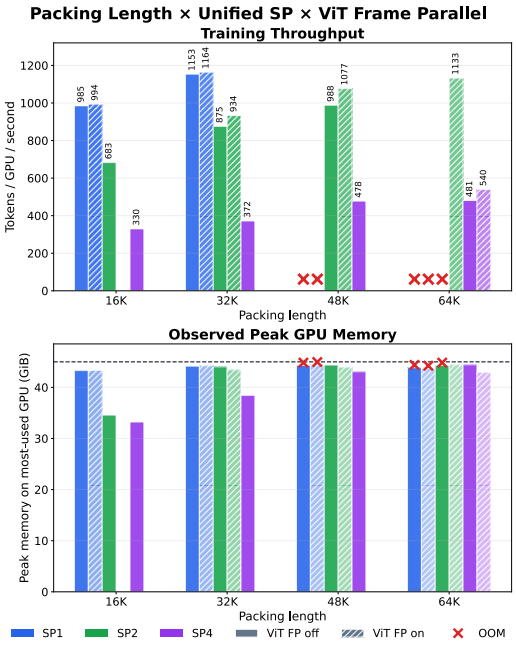
Unified Packed-Sequence Parallelism. After on-the-fly packing, we remove padding and partition both the audio sequence and the final multimodal language sequence across the same sequence-parallel group. The audio tower and language model both use DeepSpeed Ulysses (Jacobs et al., 2023), which exchanges sequence and attention-head dimensions through all-to-all communication while preserving individual sample boundaries. Combining frame parallelism for vision with sequence parallelism for audio and language distributes all three major sources of computation and memory; applying sequence parallelism only to the language model would instead leave visual and audio encoding replicated.

3.5 Cache-Preserving Incremental Inference

Training and inference follow the same slot-wise autoregressive recurrence, so incremental decoding can reuse the causal attention and KV-cache primitives of existing serving engines. Let Δ_t denote the positions newly added at slot t: the fused audio-text state e^a_t, any visual tokens whose timestamps fall



(a) Three-level training parallelism: workload-balanced frame parallelism for vision and Ulysses sequence parallelism for audio and language.



(b) Throughput and peak memory across packing lengths and parallel settings on four A6000 GPUs. Hatching enables ViT frame parallelism; crosses denote OOM.

Figure 4: Three-level parallel training infrastructure and its ablation.

between the preceding and current slots, and the required modality-boundary tokens. We implement the recurrence with vLLM's native streaming-input interface (Kwon et al., 2023), which appends Δ_t to the same streaming request, produces exactly one output token y_t, and feeds that token back into the next fused state of Equation 6,

$$e_{t+1} = P_a(f_a(a_{t+1})) + E(y_t). \tag{12}$$

Because earlier positions are never revised, each update appends only Δ_{t+1} rather than prefilling the accumulated history,

$$(y_t, \mathrm{KV}_t) = F_\theta(\Delta_t, \mathrm{KV}_{t-1}), \tag{13}$$

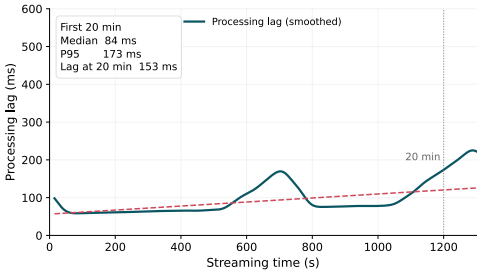
where KV_t is the retained cache after processing Δ_t. This uses the existing streaming-request, attention, and KV-cache primitives without modifying the scheduler, attention kernels, or cache layout.

Cache-Preserving Causal Extension. Let $X_t = \mathrm{Concat}(\Delta_1, \ldots, \Delta_t)$ be the realized input sequence with a fixed history of emitted tokens. If (i) earlier embeddings and positional assignments remain unchanged as positions are appended, (ii) attention is causal and all other Transformer operations are position-wise with fixed parameters, and (iii) inference is deterministic, then appending Δ_{t+1} leaves every earlier hidden state and KV entry unchanged. Incremental execution with the retained cache therefore yields the same new-position states as full causal evaluation of X_{t+1} in exact arithmetic (proof in Appendix E).

The property concerns cache correctness for a fixed realized history; respecting source-time causality additionally requires each modality representation to depend only on observations available when appended, accounting for encoder buffering or lookahead. It does not guarantee constant per-step cost or bounded latency, since attention still accesses an expanding history.

4 Experiments

Experimental setup. Aero Realtime is initialized from Qwen3-VL-4B-Instruct (Bai et al., 2025a), retaining its vision tower and language model, while the audio tower is initialized from Qwen3-Omni (Xu et al., 2025b) and follows the temporal downsampling scheme of Qwen2-Audio (Chu



Streaming text generation						
Streaming time	30 s	1 min	5 min	10 min	15 min	20 min
Processing lag (ms)	37	37	39	95	56	153
Running median (ms)	78	57	60	78	81	84
Running P95 (ms)	126	125	85	99	174	173
Streaming text-to-speech						
TTS time	≤10 s	≤20 s	≤30 s	≤40 s	≤60 s	≤90 s
Cumulative RTF	1.01	1.00	1.01	1.07	1.00	1.00
Output lag p50 (ms)	57	61	69	87	176	77

Figure 5: Long-horizon processing lag under continuous audio-video streaming (left) and lag at selected checkpoints for Aero Realtime (top right) and for the external streaming TTS front end (bottom right). The first 10 seconds are excluded as warmup from the running median and P95.

et al., 2024), producing one projected audio token per 80-ms slot to match the temporal grid in Section 3.2. We train with LMMs-Engine (LMMs-Lab, 2025) in two stages: Stage 1 adapts the model to the aligned input-output stream on the full training mixture, and Stage 2 refines response quality and video understanding on generated realtime data together with subsets of LLaVA-Video and QAEgo4D. All benchmark evaluations are conducted with LMMs-Eval (Zhang et al., 2024; Li* et al., 2024). Complete optimization, data-mixture, and hardware details are provided in Appendix A.

4.1 Realtime Latency and Response Time

We evaluate processing lag under continuous audio-video input. Each 80-ms slot admits the next audio state, any newly available visual tokens, and the previous slot's output token into the same active autoregressive sequence, so the model continues ingesting observations during textual generation. Figure 5 reports the first 20 minutes of a longer continuous video run, streamed at its native wall-clock rate on four NVIDIA A6000 GPUs. Over this window, Aero Realtime records a median processing lag of 84 ms, a P95 of 173 ms, and 153 ms at the 20-minute boundary. These measurements show that the implementation remains close to the source timeline over the reported window. The same run accumulates backlog beyond 20 minutes, reaching approximately 51.6 seconds near its end at 30 minutes, because attention to a growing history still adds cost even when cached states are reused. This backlog reflects the throughput of the tested hardware rather than an intrinsic limit: on newer accelerators such as B200, a single card is expected to sustain this latency beyond 30 minutes.

We also evaluate an external streaming text-to-speech front end under paced textual input. The bottom panel of Figure 5 reports its cumulative RTF, which ranges from 1.00 to 1.07 at the listed checkpoints. This experiment evaluates the speech component only and does not measure end-to-end conversational latency, interruption handling, or the quality of response timing. Measurement details are given in Appendix F.

4.2 Video Understanding Performance

We evaluate streaming video understanding on OVOBench (Li et al., 2025). Table 1 reports the average score for each track together with architectural interaction properties; these properties are not measured by the accuracy scores. Using a 4B language model, Aero Realtime achieves 61.49 on Realtime, 44.07 on Backward, and 39.36 on Forward. The strongest online baselines lead on raw accuracy. These results provide evidence of video-understanding capability under the aligned formulation, but do not isolate the effect of the interface change from training data and model size, or evaluate autonomous response timing.

4.3 Ablation Studies

We ablate three components required by the aligned-stream formulation: silence-aware supervision, staged training, and modality-aware parallelism.

Effect of Silence-Label Masking. Silence labels dominate the aligned stream and can overwhelm sparse lexical targets. We define r as the fraction of [P] targets independently excluded from the loss; lexical targets are never masked. Without masking, the model remains almost entirely silent,

Table 1: OVOBench track averages and interaction properties. Duplex I/O admits observations during generation; Native Proactive models silence and text in one autoregressive objective. Properties follow public sources; column maxima are bold.

Model	LM	Input Mode	Duplex I/O	Native Proactive	Realtime	Backward	Forward	Avg.
Qwen2.5-VL (Bai et al., 2025b)	7B	Offline	✗	✗	59.90	44.70	–	–
LLaVA-OneVision (Li et al., 2024)	7B	Offline	✗	✗	64.00	43.70	50.50	52.73
InternVL2 (Chen et al., 2024b)	8B	Offline	✗	✗	60.40	43.40	46.60	50.13
LLaVA-Video (Zhang et al., 2025b)	7B	Offline	✗	✗	63.50	40.40	**54.82**	52.91
Qwen2-VL (Wang et al., 2024)	7B	Offline	✗	✗	60.70	48.60	48.74	52.68
LongVU (Shen et al., 2025)	7B	Offline	✗	✗	57.40	39.50	47.50	48.13
VideoLLM-online (Chen et al., 2024a)	8B	Online	✗	✗	20.80	17.70	–	–
Flash-VStream (Zhang et al., 2025a)	7B	Online	✗	✗	28.40	27.40	45.09	33.63
Dispider (Qian et al., 2025)	7B	Online	✗	✗	54.60	36.10	34.72	41.81
TimeChat-Online (Yao et al., 2025)	7B	Online	✗	✗	61.90	41.70	36.70	46.77
StreamForest (Zeng et al., 2025)	7B	Online	✗	✗	61.20	**52.00**	53.49	55.56
Streamo (Xia et al., 2025)	7B	Online	✗	✓	66.00	46.10	54.77	**55.62**
HERMES (Zhang et al., 2026)	7B	Online	✗	✗	**69.00**	49.40	–	–
Aero Realtime	**4B**	**Online**	✓	✓	61.49	44.07	39.36	48.31

Table 2: Effects of silence-label masking and staged training on OVOBench.

	OVOBench			
Setting	Realtime	Backward	Forward	Avg.
No masking ($r = 0.00$)	0.36	0.96	17.87	6.40
+ Masking ($r = 0.70$)	9.03	12.01	**47.90**	22.98
+ Masking ($r = 0.95$, Stage 1)	58.03	33.94	35.11	42.36
+ Stage 2	**61.49**	**44.07**	39.36	**48.31**

yielding only 6.40 average on OVOBench. Although $r = 0.70$ recovers Forward performance, $r = 0.95$ produces more balanced results, raising the average to 42.36 while retaining 5% of silence labels for Stage 1.

Training-Stage Ablations. Stage 1 adapts the turn-based initialization to the slot-aligned interface, including silence prediction and continuous lexical output. Stage 2 continues training with realtime data and selected LLaVA-Video and QAEgo4D examples. It raises the overall average from 42.36 to 48.31, with the largest gain on the Backward track. This comparison measures the combined effect of additional training and the changed data mixture; it does not isolate their individual contributions.

Training Infrastructure Ablation. Figure 4b studies packing length, sequence-parallel degree, and ViT frame parallelism on four A6000 GPUs. Sequence parallelism makes longer packing memory-feasible, while frame parallelism improves throughput by balancing visual work and avoiding replicated frame encoding. Guided by this trade-off, our full $\mathrm{SP} = 4 \times \mathrm{DP} = 8$ configuration sustains over 1,600 tokens per GPU per second, allowing both stages to finish within one day.

5 Conclusion

We presented Aero Realtime, a 4B video-language model that supports continuous observation and textual generation within one causal stream while reusing causal language-model training and incremental execution primitives. We characterize the conditions for cache-preserving extension and provide a learning and execution recipe through aligned supervision, silence-label masking, modality-aware distributed training, and incremental serving. These components establish the feasibility of adapting a pretrained VLM to the interface: the model retains video-understanding capability on OVOBench and records low processing lag over the first 20 minutes of a measured continuous run. Its limitations define the scope of these results: the 80-ms grid caps output at 12.5 tokens per second, processing lag accumulates in longer runs on the tested hardware, and benchmark accuracy trails the strongest baselines. The streaming runs do not establish autonomous response timing or interruption handling, and external TTS measurements do not establish end-to-end conversational latency. Path-level data deduplication also does not guarantee content-level independence from evaluation videos. Evaluating interaction behavior, managing growing context, and auditing data overlap remain necessary steps toward broader deployment.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets. We have not used generative AI tools for other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, draft parts of the research paper, and edit the research paper to improve readability. We have reviewed all AI-assisted work and verified and tested AI-assisted code for correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

It is important that the work published in ICLR is reproducible. We provide a complete training and serving recipe in the main text and appendix, including the aligned-stream data construction, the slot-aligned supervision scheme, the modality-aware three-level parallel training configuration, and the cache-valid delta inference procedure. Full hyperparameters, training-stage data mixtures, computing infrastructure, and random seeds are reported in Appendix A.

REFERENCES

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, 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, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

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 *CVPR*, 2024a.

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 *CVPR*, 2025.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024b.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen2-audio technical report, 2024. URL https://arxiv.org/abs/2407.10759.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: A speech-text foundation model for real-time dialogue, 2024. URL https://arxiv.org/abs/2410.00037.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, Xinyu Zhang, Pei Zhang, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-tts technical report, 2026. URL https://arxiv.org/abs/2601.15621.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models, 2023. URL https://arxiv.org/abs/2309.14509.

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 ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Bo Li*, Peiyuan Zhang*, Kaichen Zhang*, Fanyi Pu*, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Accelerating the development of large multimoal models, March 2024. URL https://github.com/EvolvingLMMs-Lab/lmms-eval.

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, 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, 2026. URL https://arxiv.org/abs/2607.14935.

Yifei Li, Junbo Niu, Ziyang Miao, Chunjiang Ge, Yuanhang Zhou, Qihao He, Xiaoyi Dong, Haodong Duan, Shuangrui Ding, Rui Qian, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, and Jiaqi Wang. Ovo-bench: How far is your video-llms from real-world online video understanding? In *CVPR*, 2025.

LMMs-Lab. Lmms engine: A simple, unified multimodal framework for pretraining and finetuning., 2025. URL https://github.com/LMMs-Lab/lmms-engine.

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, 2026. URL https://arxiv.org/abs/2604.04184.

Mistral AI, Alexander H. Liu, Andy Ehrenberg, Andy Lo, Chen-Yo Sun, Guillaume Lample, Jean-Malo Delignon, Khyathi Raghavi Chandu, Patrick von Platen, Pavankumar Reddy Muddireddy, Rohin Arora, Sanchit Gandhi, Sandeep Subramanian, Soham Ghosh, Srijan Mishra, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andrew Bai, Angele Lenglemetz, Anmol Agarwal, Anton Eliseev, Antonia Calvi, Arjun Majumdar, Avi Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Benjamin Tibi, Charlotte Cronjäger, Clémence Lanfranchi, Connor Chen, Corentin Barreau, Corentin Sautier, Cyprien Courtot, Darius Dabert, Diego de las Casas, Elizaveta Demyanenko, Elliot Chane-Sane, Enguerrand Paquin, Etienne Goffinet, Fabien Niel, Faruk Ahmed, Federico Baldassarre, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet,

Genevieve Hayes, Georgii Novikov, Giada Pistilli, Guillaume Kunsch, Guillaume Martin, Guillaume Raille, Gunjan Dhanuka, Gunshi Gupta, Han Zhou, Harshil Shah, Hope McGovern, Hugo Thimonier, Indraneel Mukherjee, Irene Zhang, Jaeyoung Kim, Jan Ludziejewski, Jason Rute, Joachim Studnia, John Harvill, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Julien Tauran, Karmesh Yadav, Kartik Khandelwal, Kilian Tep, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Manan Sharma, Margaret Jennings, Marie Pellat, Mark Prins, Martin Alexandre, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mert Unsal, Mia Chiquier, Minh-Quang Pham, Nathan Grinsztajn, Neha Gupta, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Philippe Pinel, Philomène Chagniot, Pierre Stock, Piotr Miłoś, Prateek Gupta, Pravesh Agrawal, Quentin Torroba, Ram Ramrakhya, Rishi Shah, Romain Sauvestre, Roman Soletskyi, Rosalie Millner, Rupert Menneer, Sagar Vaze, Samuel Barry, Samuel Humeau, Sean Cha, Shashwat Verma, Siddhant Waghjale, Siddharth Gandhi, Simon Lepage, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Edwards, Tyler Wang, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vedant Nanda, Victor Jouault, Vincent Maladière, Virgile Richard, Vladislav Bataev, Wassim Bouaziz, Wen-Ding Li, William Havard, William Marshall, Xinghui Li, Xingran Guo, Xinyu Yang, Yannic Neuhaus, Yassine El Ouahidi, Yassir Bendou, Yihan Wang, Yimu Pan, Zaccharie Ramzi, and Zhenlin Xu. Voxtral realtime, 2026. URL https://arxiv.org/abs/2602.11298.

Alkesh Patel, Vibhav Chitalia, and Yinfei Yang. Advancing egocentric video question answering with multimodal large language models, 2025. URL https://arxiv.org/abs/2504.04550.

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 *CVPR*, 2025.

Qwen Team. Qwen3-vl: Sharper vision, deeper thought, broader action. https://qwen.ai/blog?id=99f0335c4ad9ff6153e517418d48535ab6d8afef&from=research.latest-advancements-list, 2025. Accessed: 2025-11-14.

Amélie Royer, Moritz Böhle, Gabriel de Marmiesse, Laurent Mazaré, Neil Zeghidour, Alexandre Défossez, and Patrick Pérez. Vision-speech models: Teaching speech models to converse about images, 2025. URL https://arxiv.org/abs/2503.15633.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Borber, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. In *ICML*, 2025.

Yujiao Shen, Shulin Tian, Jingkang Yang, and Ziwei Liu. A simple baseline for streaming video understanding, 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. https://thinkingmachines.ai/blog/interaction-models/.

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. In *NeurIPS*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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, 2026. URL https://arxiv.org/abs/2606.07639.

Jiaer Xia, Peixian Chen, Mengdan Zhang, Xing Sun, and Kaiyang Zhou. Streaming video instruction tuning. *arXiv preprint arXiv:2512.21334*, 2025. Introduces Streamo.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025a. URL https://arxiv.org/abs/2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, Baosong Yang, Bin Zhang, Ziyang Ma, Xipin Wei, Shuai Bai, Keqin Chen, Xuejing Liu, Peng Wang, Mingkun Yang, Dayiheng Liu, Xingzhang Ren, Bo Zheng, Rui Men, Fan Zhou, Bowen Yu, Jianxin Yang, Le Yu, Jingren Zhou, and Junyang Lin. Qwen3-omni technical report, 2025b. URL https://arxiv.org/abs/2509.17765.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. Streamingvlm: Real-time understanding for infinite video streams. In *ICLR*, 2026.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, Bei Ouyang, Zhengyu Lin, Marco Cominelli, Zhongang Cai, Yuanhan Zhang, Peiyuan Zhang, Fangzhou Hong, Joerg Widmer, Francesco Gringoli, Lei Yang, Bo Li, and Ziwei Liu. Egolife: Towards egocentric life assistant, 2026a. URL https://arxiv.org/abs/2503.03803.

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, 2026b. 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, 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. In *ACM Multimedia*, 2025.

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.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *ICCV*, 2025a.

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*, 2026.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. URL https://arxiv.org/abs/2407.12772.

Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025b.

Table 3: Final hyperparameters for both training stages.

Hyperparameter	Value
Optimizer	AdamW (fused)
Learning rate	5×10^{-5}
Learning-rate schedule	Constant
Warmup ratio	0.05
Precision	bf16 (tf32 matmul)
Gradient checkpointing	Enabled
Per-device batch size	1
Gradient accumulation	1
Stage 1 steps	4000
Stage 2 steps	2000
Packing length	65536 tokens
Packing strategy	Balanced (10 buckets)
Padding removal	Enabled
Sharding	FSDP2
Sequence-parallel degree	4
Attention implementation	FlashAttention-2
Frame rate	1 fps
Max frames per video	512
Max visual pixels	360448
Min visual pixels	28800
Silence-label masking ratio	0.95

A Training Configuration

Table 3 lists the final hyperparameters used for both training stages. The two stages share all optimization and parallelism settings and differ only in the number of training steps and in the data mixture, as described in the main paper.

We did not perform a systematic hyperparameter search. The learning rate and the learning-rate schedule were the only settings varied during development, and the reported configuration was selected from a small number of preliminary runs. All remaining values were fixed a priori.

B Training-Stage Data Mixtures

Training length is defined by consumed tokens because samples are packed on the fly. Stage 1 uses the complete mixture to adapt the turn-based initialization to aligned-stream generation. Stage 2 starts from the Stage 1 checkpoint and uses a focused mixture to strengthen response quality and video understanding.

Table 4: Data mixtures and token budgets for the two training stages.

Stage	Data mixture	Token budget
Stage 1	Generated realtime QA; LLaVA-Video QA/caption data; LiveCC; EgoIT; QAEgo4D	2B
Stage 2	Generated realtime QA; selected LLaVA-Video data; QAEgo4D	1B

C Data Construction Details

Causal realtime construction. We retain videos between 30 and 600 seconds. For each video, we sample an initial timestamp uniformly from 5–10 seconds and advance through the video using intervals drawn from the same range. At each timestamp, GPT-5.4 observes *only* the video prefix available up to that time, together with previously generated QA turns as textual memory, and

produces a temporally grounded QA pair. Repeating this process yields a time-ordered interaction trajectory without exposing future frames. This pipeline generates 103k realtime QA samples.

Unified aligned-stream conversion. For temporally annotated examples, we align each response to the first audio slot at or after its timestamp, place its tokenized response in subsequent slots, and supervise all remaining positions with the silence token [P]. When a source provides event or clip metadata rather than a response timestamp, we use that metadata to place the QA pair at the corresponding point on the video timeline. For conventional video QA, we treat answering after the observed video as a terminal special case: the question remains in context, silent audio slots extend the timeline beyond the final frame, and the answer occupies the following output slots. Samples without video retain standard conversation-template supervision.

Deduplication. During dataset assembly, we preserve source identifiers and split metadata and remove duplicate source paths. This path-level deduplication does not guarantee content-level independence from every evaluation video; because our training mixture and OVOBench contain related egocentric-video sources, potential source overlap remains a limitation.

D Computing Infrastructure

All training runs use 4 nodes with 8 NVIDIA A100-40G GPUs each, for a total of 32 GPUs. The software stack is CUDA 13.2 and PyTorch 2.11, with LMMs-Engine for training and LMMs-Eval for evaluation. The parallel-training ablation and long-horizon latency experiment use 4 NVIDIA A6000-45G GPUs. CPU model, host memory, and operating system version are not recorded.

E Cache-Preservation Proof

Earlier input states are unchanged by assumption. At each Transformer layer, an earlier position attends only to its unchanged causal prefix, so its attention output and subsequent position-wise transformations are unchanged. Induction over layers establishes invariance of earlier hidden states and their projected keys and values. New positions therefore attend to the same keys and values in cached and full evaluation, establishing equivalence. This is an application of causal Transformer computation to our stream construction, rather than a new attention mechanism.

F Latency Measurement Details

In these experiments, processing lag is the wall-clock completion time of each one-second interval minus its source end timestamp, and is not event-to-response latency. The curve in Figure 5 applies a 120-second Gaussian smoothing window, and the reported median and P95 summarize interval measurements after excluding the first 10 seconds as warmup.

The external streaming text-to-speech front end (Hu et al., 2026) is evaluated under token delivery at one token per 80-ms slot on a single standard inference GPU. The bottom panel of Figure 5 reports its output lag and cumulative RTF over the measured 90-second interval; cumulative RTF ranges from 1.00 to 1.07 at the listed checkpoints.

G Random Seeds

All training runs use a fixed random seed of 42, which controls data shuffling, packing order, and model initialization of newly added parameters. We report results from a single run per configuration and therefore do not report variance across seeds.

H Qualitative Proactive Examples

The following examples are taken directly from the saved realtime runs. The image shows the current stream state, while the text is emitted by Aero Realtime as the stream continues; no separate answer-generation turn is opened.

I REALTIME BENCHMARK DATA SAMPLES

The benchmark samples pair a continuous video with time-aligned user turns and assistant responses. Each row below shows the video state at the annotated interaction time and the corresponding supervision.

Proactive response at 00:32



00:30 I can see a person seated at a table with a bright yellow bag on the left side of the frame. The person is



00:32 gesturing with their hands and appears to be talking. The background includes a bookshelf and a floor lamp. The person is



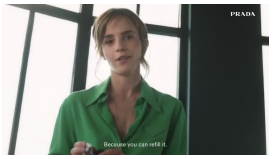
00:34 still seated at the table, continuing to gesture and talk. The person is still seated at the table, continuing to gesture and



00:36 talk. The bright yellow bag remains on the left side of the frame, with the bookshelf and floor lamp in the background.

Standing instruction: Describe what is happening now.

Proactive response at 00:12



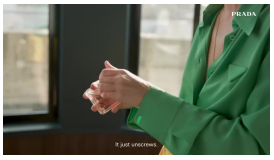
00:10 A woman in a green top and yellow belt is standing by a window, holding a small bottle and a plate. The text



00:12 "PARADOX" appears on screen. She is still holding the small bottle and plate as the scene continues.



00:14 The text "PARADOX" remains visible while she continues presenting the items.



00:16 She remains by the window with the bottle and plate as the shot continues.

Cooking process with evolving observations



Q (00:10): That looks really buttery.

A: Yeah, she's stirring a soft, grainy mixture in the bowl, and there are visible chunks of butter melting through it.



Q (00:17): It's looking smoother now.

A: Yeah, the butter's worked in more and the mixture has turned into a softer, mashed texture.



Q (00:24): Did she just add cream or milk there?

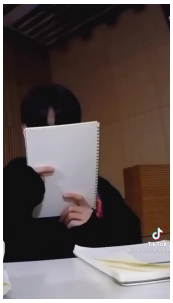
A: Yes, she's pouring in a bit of milk from the white jug, then folding it through the mash.



Q (00:33): She's adding something else now, right?

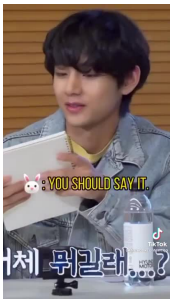
A: Yes—she's adding a dollop of cream or soft cheese to the mash, making it look much silkier as she folds it in.

Social interaction over time



Q (00:09): He was really quiet, and now he's showing his notebook to someone.

A: Yeah, he stops writing and holds the sketchbook out toward the person beside him, like he's asking for a look.



Q (00:17): Wait, did the other guy actually like what he wrote?

A: Yeah, he looks impressed now and keeps asking why he didn't share it earlier.



Q (00:23): Aw, now he's getting shy about it.

A: Yeah, he's hiding behind the notebook while the other guy keeps insisting the idea is his and he should be the one to say it.



Q (00:29): He really won't take credit for it, huh?

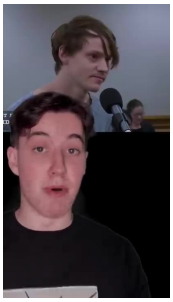
A: Yeah, he's still trying to hand the idea over, and the other guy keeps pushing it back to him.

Tracking a rapidly changing montage



Q (00:10): That cut was abrupt—did it just switch scenes twice?

A: Yeah, it jumped from the microphone setup to someone by a white car, then straight to a pizza close-up.



Q (00:17): It keeps cutting to random examples now—car, pizza, kid, police lights, even a red light.

A: Yeah, it's turned into a rapid montage of examples while the two people by the microphone stay as the main setup.



Q (00:24): Now it's connecting the montage back to the courtroom, right?

A: Yeah, it cut from those example shots back to the person speaking at the microphone, and then wider courtroom views.



Q (00:32): So it's back to the montage again—now tickets, a wedding clip, and saving money?

A: Yeah, it keeps bouncing from the courtroom testimony to these quick illustrative cutaways.