Wan-Streamer v0.1: End-to-end Real-time Interactive Foundation Models

Wan Team, Alibaba Group

See Contributions and Acknowledgements for the full author list.

Abstract

We present **Wan-Streamer**, a native-streaming, end-to-end interactive foundation model designed from the ground up for real-time, low-latency, full-duplex audio-visual interaction. Wan-Streamer seamlessly models language, audio, and video as both input and output within **a single Transformer**, where the sequence is represented as interleaved visual, audio, and text input tokens together with visual, audio, and text output tokens, coordinated by block-causal attention for incremental streaming. Unlike cascaded interactive systems that rely on separate VAD, ASR, language, TTS, audio-driven animation, or video-generation modules, Wan-Streamer does not rely on external language, speech, avatar, or video-generation modules: perception, reasoning, generation, response timing, turn management, and cross-modal synchronization are learned jointly within one unified model, reducing pipeline latency and error accumulation. To support natural audio-visual responsiveness, we redesign the entire stack around streamability, including causal encoders, causal decoders, block-causal attention, and low-latency multimodal token scheduling, enabling streaming units as short as 160 ms at 25 fps. Wan-Streamer achieves approximately **200 ms** model-side response latency and approximately 550 ms total interaction latency when combined with 350 ms bidirectional network latency, supporting sub-second duplex audio-visual communication. These results position Wan-Streamer as a unified, end-to-end, multimodal interactive foundation model for low-latency streaming interaction.

Website: https://wan-streamer.com/

1 Introduction

Human interaction with the physical world is fundamentally streaming and full-duplex. People do not first finish perceiving, then reason in isolation, and only afterwards produce a response. Instead, they continuously watch, listen, speak, gesture, react, pause, and interrupt, with perception and expression overlapping at audio-visual timescales. Building artificial systems with the same interaction pattern is becoming increasingly important for embodied assistants, real-time digital humans, live broadcasting, interactive entertainment, and world models that can be explored or controlled online [14, 22, 39]. These applications require more than a model that can understand an image, generate a clip, or answer a text prompt. They require a real-time interactive foundation model: a model that continuously consumes audio-visual observations, maintains a persistent world and dialogue state, decides when and how to respond, and expresses that response through synchronized language, speech, and video with very low latency.

Recent progress has advanced several pieces of this goal. Multimodal language models can reason over visual and acoustic inputs [3, 15, 35], while large-scale video generation models can synthesize increasingly realistic motion and appearance from text, audio, camera trajectories, or actions [2, 14, 31, 37, 39, 44]. Causal and streaming generation methods further make incremental synthesis feasible by replacing offline bidirectional generation with cached context and rolling prediction [7, 17, 25]. However, these advances are still usually assembled as asymmetric or cascaded systems. Some systems perceive audio and video but respond only in text or speech; others generate audio-visual behavior but rely on external language, ASR, TTS, animation, or

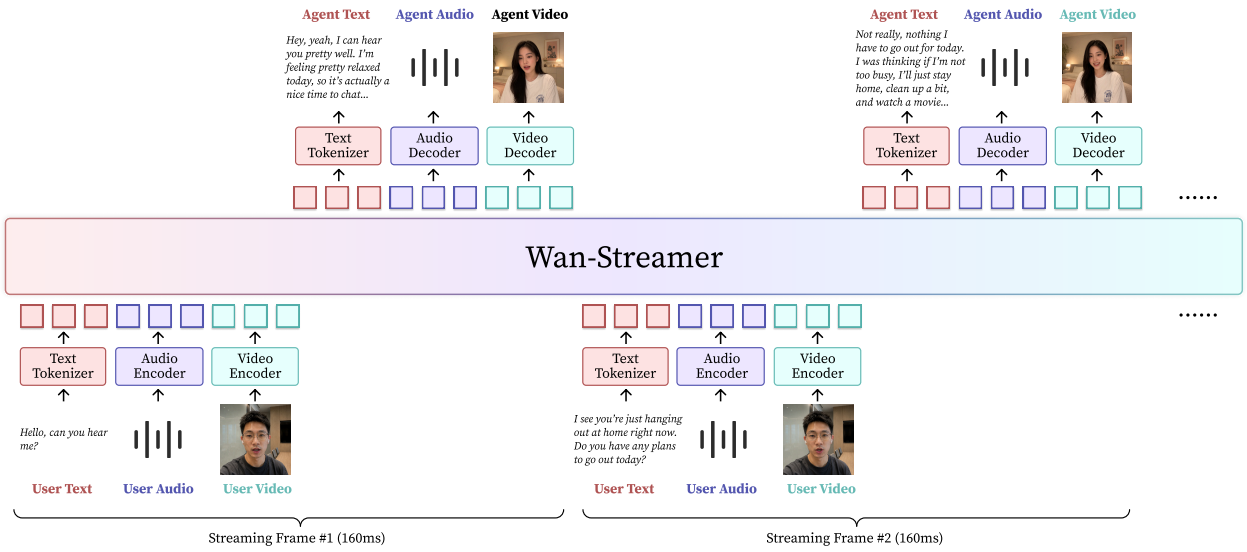

Figure 1 Overview of Wan-Streamer. It models language, audio, and video as both input and output within **a single Transformer**, using block-causal attention for incremental streaming generation.

rendering modules. Even when the interface appears multimodal, text is often used as a hidden intermediate representation between separately trained components [6, 28]. Such pipelines introduce waiting time at module boundaries, accumulate recognition and synchronization errors, and make response timing, turn management, identity preservation, and long-horizon consistency difficult to learn as part of one behavior.

The core difficulty is that real-time audio-visual interaction is not simply the union of multimodal understanding and multimodal generation. It is intrinsically full-duplex rather than a simple alternation between input and output: when the user is speaking, the agent should still produce visible listening behavior, and when the agent is responding, it should still perceive the user's audio-visual feedback for interruption and adaptation. Different modalities have different token rates, representations, objectives, and latency constraints, yet they must be causally aligned within a single ongoing process. Incoming speech and video should immediately affect outgoing speech and motion; generated audio and visual states should be coupled before decoding, not repaired afterwards; and every emitted unit should become part of the full interaction history so that the model can preserve identity, scene state, speaking rhythm, and user intent over long sessions. These requirements make streamability a modeling constraint rather than a serving optimization. A system designed around offline encoders, bidirectional video decoders, round-based dialogue, or post-hoc audio-visual synchronization cannot recover truly low-latency full-duplex behavior by engineering alone.

To address this challenge, we design **Wan-Streamer** from the ground up as a native-streaming, end-to-end model for real-time full-duplex audio-visual interaction, as illustrated in Fig. 1. Wan-Streamer is built around one streaming contract: every component must operate causally, every newly observed unit must be usable immediately, and every generated unit must be emitted and committed back into the interaction history. Wan-Streamer represents language, audio, and video, on both the input and output sides, as an interleaved causal sequence processed by **a single Transformer**. The model does not rely on external VAD, ASR, language, TTS, audio-driven animation, or video-generation modules. Instead, audio-visual perception, semantic reasoning, response planning, speech generation, visual generation, response timing, and turn-taking behavior are optimized jointly within one persistent interaction state. To make this possible, the entire stack is designed for causality from the beginning: strictly causal audio and video variational autoencoders (VAEs) for streaming latent coding, causal audio-visual encoders, causal audio and video decoders, and a temporally causal Transformer coordinated by block-causal attention.

During inference, Wan-Streamer remains a single end-to-end model, but we deploy it as a thinker-performer streaming pipeline to maximize overlap and hardware utilization. At streaming step k, the thinker first consumes the current user audio-visual observations, applies the causal audio-visual encoders, and runs the short token-causal Transformer pass for language prediction and state update. This produces the new KV-cache slice for the current interaction state. At the communication boundary, the thinker receives the

audio and video latents generated by the performer for the previous response unit, and sends the current KV-cache slice to the performer. The thinker then decodes those previous latents into output audio and video for immediate emission, while the performer uses the newly received full-history KV context to run only the flow-matching solver for the next audio-visual latent unit. The resulting latents are kept on the performer and returned to the thinker at the next streaming step. This schedule preserves the unified causal state through KV exchange, but places expensive latent generation on the performer and allows current-frame perception/state update, previous-frame output decoding, next-frame latent denoising, and KV/latent communication to overlap across adjacent streaming units. Together with CUDA graph capture, compilation, and optimized kernels, Wan-Streamer reaches approximately **200 ms** model-side response latency. With 350 ms bidirectional network latency, the total interaction latency is approximately 550 ms, supporting sub-second audio-visual communication without the module-boundary waiting time of typical cascaded real-time dialogue systems.

Our contributions are summarized as follows:

- We introduce Wan-Streamer, a native-streaming, end-to-end interactive foundation model that supports language, audio, and video as both inputs and outputs within a single Transformer, without relying on external language, speech, animation, or video-generation modules.
- We develop a fully causal multimodal architecture for real-time interaction, including strictly causal audio and video VAEs, causal audio-visual encoders and decoders, block-causal multimodal attention, and full-history autoregressive streaming.
- We present a low-latency thinker-performer inference system that preserves the unified model state through KV-cache exchange while overlapping understanding and generation, achieving approximately 200 ms model-side response latency and approximately 550 ms total interaction latency.

2 Method

2.1 Overview

Wan-Streamer models interaction as a continuous causal stream in which user observations and agent responses jointly update the ongoing context. At the k-th streaming unit, let $u_k = (u_k^{\mathrm{t}}, u_k^{\mathrm{a}}, u_k^{\mathrm{v}})$ denote the user's language, audio, and video observations, and let $y_k = (y_k^{\mathrm{t}}, y_k^{\mathrm{a}}, y_k^{\mathrm{v}})$ denote the agent response. The model encodes the currently available user observations and predicts the next response from the complete causal history across both sides of the interaction:

$$p_\theta(y_{1:K} \mid u_{1:K}) = \prod_{k=1}^{K} p_\theta\big(y_k^{\mathrm{t}}, y_k^{\mathrm{a}}, y_k^{\mathrm{v}} \mid u_{\le k}^{\mathrm{t}}, u_{\le k}^{\mathrm{a}}, u_{\le k}^{\mathrm{v}}, y_{<k}^{\mathrm{t}}, y_{<k}^{\mathrm{a}}, y_{<k}^{\mathrm{v}}\big)\,, \tag{1}$$

where K denotes the number of streaming units. Once generated, the response unit is appended together with the corresponding user observations to the history state and becomes context for the next unit. The language response is represented as a sequence of discrete tokens and optimized with next-token prediction using cross-entropy loss. The audio and video responses are represented in continuous latent spaces and generated jointly with conditional flow matching. For the current response unit, clean states and noisy states play different roles. For modality $m \in \{\mathrm{a}, \mathrm{v}\}$, let z_0^m be the clean target latent of the response, and let $\epsilon^m \sim \mathcal{N}(0, I)$ be Gaussian noise. At flow time τ, we construct a noisy latent z_τ^m using

$$z_\tau^m = (1-\tau) z_0^m + \tau \epsilon^m, \qquad \frac{\partial z_\tau^m}{\partial \tau} = \epsilon^m - z_0^m. \tag{2}$$

Let $c_k = \{u_{\le k}^{\mathrm{t}}, u_{\le k}^{\mathrm{a}}, u_{\le k}^{\mathrm{v}}, y_{<k}^{\mathrm{t}}, y_{<k}^{\mathrm{a}}, y_{<k}^{\mathrm{v}}\}$ denote the clean streaming context, consisting of user observations that have arrived and agent responses that have already been committed to history. The current audio and video responses are the noisy variables being denoised, while the context remains available as causal history. We train the model to estimate the velocity fields for both noisy audio and video latents conditioned on c_k and noise level τ:

$$\mathcal{L}_{\mathrm{FM}}^m = \mathbb{E}_{\epsilon^m} \left\| f_\theta(z_\tau^{\mathrm{a}}, z_\tau^{\mathrm{v}}, c_k, \tau) - \frac{\partial z_\tau^m}{\partial \tau} \right\|_2^2, \tag{3}$$

where f_θ denotes the unified diffusion transformer. The same clean context conditions both velocity predictions, allowing speech, motion, appearance, and scene evolution to be optimized as a coupled response. After denoising, the estimated clean latents are appended directly to the history as clean context for subsequent streaming units, while the causal decoders render them into external audio and video outputs.

2.2 Data

Wan-Streamer is trained on a broad mixture of understanding, generation, and end-to-end interaction data. For understanding-oriented learning, we include image, audio, and video understanding data, text dialogue, ASR, TTS, audio dialogue, and related language-audio-visual supervision. These data teach the model to convert streaming multimodal observations into a shared causal context and to preserve dialogue competence under multimodal inputs. For generation-oriented learning, we include image generation, audio generation, video generation, and joint audio-visual generation tasks, covering both single-modality and cross-modality conditions. Finally, we use end-to-end duplex interaction data in which text, audio, and video can appear on both the input side and the output side. This end-to-end data component exposes the model to the target setting of simultaneous multimodal perception and expression.

2.3 Training

Training proceeds in three stages. The first stage is independent-task pretraining. We initialize the unified Transformer from a language model [42, 43] and train the multimodal interface around it with the mixture described above. On the understanding side, the causal audio and video encoders are trained together with the Transformer so that image, audio, video, and dialogue understanding metrics approach those of dedicated multimodal understanding models, while conversational ability remains comparable to turn-based dialogue models of similar scale. On the generation side, the same Transformer is trained with the causal audio and video latent spaces on image, audio, video, and joint audio-visual generation tasks. These understanding and generation tasks are mixed during pretraining so that perception, language reasoning, and latent generation are aligned in one sequence model rather than optimized as isolated modules.

The second stage is end-to-end interaction training. We train on duplex interaction data where user text/audio/video inputs and agent text/audio/video outputs are interleaved in the same causal stream. This stage adapts the pretrained model from independent tasks to the target real-time setting: the model must update its state from current user observations, generate synchronized language, audio, and video responses, and commit the generated clean latents back into history for subsequent streaming units. As a result, response timing, active listening behavior, interruption handling, and long-context consistency are learned under the same causal format used at inference time.

The third stage is distillation for low-latency streaming. A stronger teacher with classifier-free guidance (CFG) and more flow-matching solver steps is distilled into an efficient student used at deployment. This distillation absorbs the effect of CFG into the student and reduces the number of solver steps while preserving audio-visual quality. We also use rolling distillation to mitigate long-horizon degradation: the student is rolled out over consecutive streaming units and trained on its own generated history, using a self-forcing strategy [17] with distribution matching [45, 46] to align the student trajectory with the teacher under realistic rollout conditions. This substantially reduces train-test mismatch and improves long-form generation quality.

2.4 Inference

Although Wan-Streamer is trained as a single end-to-end model, we deploy it as a separated thinker-performer pipeline to maximize overlap and hardware utilization. The *thinker* hosts the causal audio and video encoders, the short token-causal Transformer path for language prediction and state update, KV-cache construction, and the causal audio and video decoders. The *performer* hosts only the latent generation path. After system prefill, the thinker broadcasts the initial KV cache to the performer so that both sides share the same full-history state.

At streaming step k, the thinker consumes the current user audio-visual observations, applies the causal encoders, and runs token-causal decoding over the language and state slots to produce the current KV-cache slice. Around the same communication boundary, the thinker receives from the performer the clean audio and video latents produced in the preceding step, sends the newly produced KV slice to the performer, and

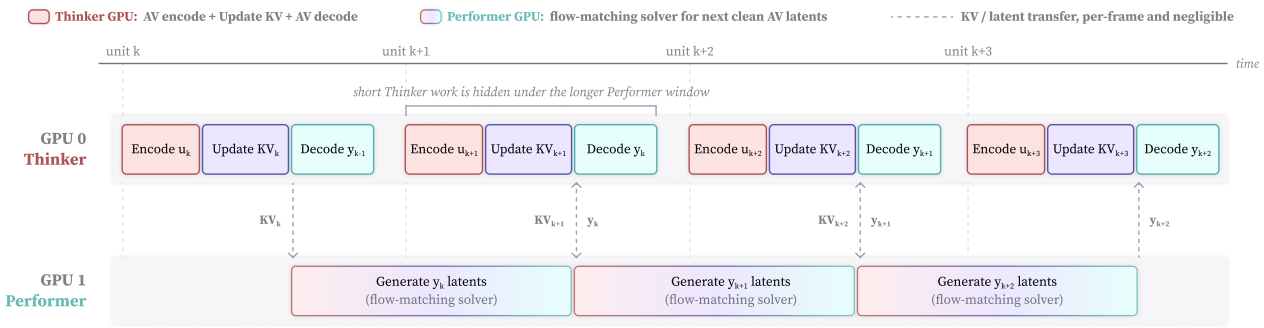

Figure 2 Thinker-performer overlap during streaming inference. At unit k, the thinker encodes the current user observations u_k, updates the KV cache, and decodes the previous response latents y_{k-1} for immediate emission. The performer receives the current KV slice and runs only the flow-matching solver to generate the next clean audio-visual latents y_k, which are returned to the thinker at the following unit. For clarity, performer windows are drawn as one full 160 ms streaming unit; in practice, real-time throughput only requires the performer time plus KV/latent communication to be below this unit duration.

decodes the returned latents into the audio-visual output that is emitted immediately. The performer appends the received KV slice into its own full-history cache and runs the flow-matching solver only for the next audio-visual latent unit. The resulting clean latents remain on the performer and are sent back to the thinker at the next streaming step for decoding and emission.

As shown in Fig. 2, this schedule pipelines current-frame perception and state update, previous-frame audio-visual decoding, KV/latent communication, and next-frame latent denoising across adjacent streaming units. After overlap, per-frame throughput is determined mainly by the performer wall time: the system can run in real time as long as the performer time, plus the small KV-cache and latent communication overhead, fits within one 160 ms streaming unit. This throughput condition is distinct from model-side response latency, which measures the full signal-to-signal path from receiving a user unit to emitting the corresponding response unit. That latency is the sum of encoding, thinker state update, performer latent generation, and decoding, and is currently approximately **200 ms**. Since the performer does not run decoders and the thinker does not run the expensive flow-matching solver, the deployment preserves the semantics of one unified model while overlapping most latency-critical work. In practice, we further use CUDA graph capture, compilation, optimized kernels, and KV-cache exchange to improve throughput.

3 Experiments

Latency and runtime comparison. For real-time interaction, the most relevant latency is the delay from the user's latest signal to the first perceptible assistant response. For Wan-Streamer, we measure the two-GPU thinker-performer serving path described in Sec. 2.4. Model-side signal-to-signal latency starts when a 160 ms user streaming unit is available to the thinker and ends when the corresponding audio-video response unit has been decoded for emission at 25 FPS. Under this protocol, Wan-Streamer reaches about 200 ms model-side latency; adding a 350 ms bidirectional network budget gives about 550 ms total interaction latency for a remote user.

Public systems do not always expose the same endpoint. Tab. 1 compares speech and omni-modal dialogue systems by the closest available user-visible response latency, while keeping model-only, first-packet, and API/product measurements separate when they are not directly comparable.

Table 1 Response-latency comparison for real-time speech and omni-modal interaction systems. The shaded row highlights Wan-Streamer; "N/R" means no aligned absolute response latency is publicly reported.

System	Interaction	User-visible response	Other reported metric	Comparison boundary
Doubao Realtime Voice [4, 36]	speech-to-speech	∼1 s overall	∼700 ms bare-model latency	Official speech-only product numbers; no visual agent output.
Seeduplex [5]	speech-to-speech	N/R absolute	−250 ms endpoint, −300 ms interruption latency vs. previous Doubao	Relative production improvement; speech-only.
GPT-4o / Realtime API [18, 20, 26]	speech-to-speech, audio/vision input	protocol-dependent	232/320 ms official audio response; ∼500 ms API TTFB; ∼800 ms target voice-to-voice	Reported numbers mix model response, API TTFB, endpointing, and network.
Hume EVI 3 [18]	speech-to-speech	0.9–1.4 s web-app benchmark	under 300 ms model response	Vendor benchmark; no visual output stream.
Gemini Live API [18]	speech-to-speech	1.2–3.6 s API benchmark	N/R model-side	Vendor benchmark; not an official model breakdown.
Sesame web app [18]	speech-to-speech	0.8–1.2 s web-app benchmark	N/R model-side	Vendor benchmark; speech-only.
Moshi [12]	speech-to-speech	N/R product path	160 ms theoretical; 200 ms practical model latency	Native full-duplex speech model; no visual agent.
Qwen3/3.5-Omni [29, 30]	audio-video-text in, speech/text out	N/R interaction loop	first-packet: 234/547 ms; Qwen3.5 Flash 235/426 ms, Plus 435/651 ms	First-packet metric; no synchronized visual avatar generation.
MiniCPM-o 4.5 [27]	audio-video in, speech/text out	N/R interaction loop	0.58 s first-token; RTF 0.20–0.27	First-token/RTF metric; no visual avatar generation.
Wan-Streamer (ours)	text/audio/video in/out	∼**550 ms total** including 350 ms network	∼**200 ms model-side**; 25 FPS video output	One end-to-end model; text I/O, speech, and synchronized visual response share one causal stream.

Tab. 1 should be read by measurement boundary rather than by the smallest raw number. Some public systems report model-internal, first-packet, first-token, endpointing, or API time-to-first-byte latency, which is useful for engineering but does not always correspond to the delay perceived by a remote user. Several omni-modal systems further accept audio or video input but do not close the loop with synchronized visual agent output. We therefore report both the ∼200 ms model-side streaming latency and the ∼550 ms total interaction latency for Wan-Streamer, and keep these numbers separate from partial or speech-only measurements.

For visually embodied systems, the available measurements are less uniform. Some systems are full interaction loops, while others are renderers or audio-visual generators driven by external dialogue and speech modules. Tab. 2 therefore records the runtime metric reported by each source, such as FPS, first-frame delay, chunk latency, or audio-to-visual delay, together with the part of the interaction stack that the system covers.

To make this comparison readable, we separate full-loop or interactive digital-human systems from avatar rendering and joint audio-visual generation components. The first group is closer in interaction scope, but often reports only real-time operation, FPS, or qualitative bounded-latency claims instead of absolute signal-to-signal response latency. The second group usually provides clearer rendering-side metrics, but assumes external dialogue, speech, or perception modules. This is why Tab. 2 reports both runtime and covered scope, rather than treating every FPS, first-frame, or audio-to-visual number as an end-to-end interaction latency.

Under this convention, the key comparison is whether the reported path includes user perception, response timing, speech generation, and synchronized visual output. A renderer can be fast once clean audio or text is available, but its perceived interaction latency still depends on the upstream dialogue and speech stack. Conversely, an interactive system can claim real-time operation while leaving the latency boundary unspecified. Wan-Streamer is evaluated against both axes: it reports the full remote audio-visual response path and also specifies the model-side streaming runtime.

Taken together, these comparisons emphasize why raw speed alone is insufficient. Speech-only systems can report very low model or first-packet latency, but they do not generate a synchronized visual response. Avatar and audio-visual generation systems can run at 20–40 FPS, but many rely on external dialogue, speech, or perception modules, and their published runtime does not include the whole conversational path. Wan-Streamer combines these parts in one causal stream: text I/O, user audio-video perception, response timing, speech generation, and 25 FPS visual expression are produced by the same end-to-end model. Thus, its 550 ms total latency covers the full audio-visual response path.

Table 2 Runtime comparison with visual agents, streaming avatars, and audio-visual generators. Most numbers here are component-level runtime metrics rather than the aligned response latency in Tab. 1; the shaded row highlights Wan-Streamer.

System	Visual interaction scope	Reported runtime	Main difference from Wan-Streamer
Full-loop or interactive digital-human systems			
Body of Her [1]	end-to-end humanoid agent	next frame within 42 ms at 24 FPS	Preliminary unified agent; no deployed signal-to-signal latency.
MIDAS [8]	multimodal digital-human video synthesis	real-time frame-by-frame generation	Does not disclose absolute response latency.
U-Mind [13]	text, speech, motion, and video interaction loop	real-time video rendering claimed	Text-first pipeline; latency breakdown not public.
X-Streamer [40]	open-ended video chat from a portrait	25 FPS multimodal streaming on two A100 GPUs	Absolute response latency is not disclosed.
LPM 1.0 [47]	online character performance engine	low-latency real-time causal streaming	Visual engine coupled to external A2A systems; latency is not intrinsic to LPM alone.
MAViD [28]	audio-visual dialogue framework	no absolute latency reported	Modular framework; useful for capability comparison, not latency comparison.
M.I.O [6]	interactive omni-avatar system	bounded-latency design discussed	Multi-module embodied system; no public signal-to-signal number.
Avatar rendering or joint audio-visual generation components			
VASA-1 [41]	audio-driven talking face	40 FPS with 170 ms preceding latency	Renderer only; no dialogue reasoning or user visual perception.
TalkingMachines [25]	FaceTime-style audio-driven video	real-time chunk generation by TTBC	Relies on an external audio LLM for dialogue and speech.
StreamAvatar [34]	streaming talking/listening avatar	FFD 0.33–0.39 s; video latency ∼1.20 s	Avatar renderer driven by speech/audio; no unified dialogue model.
Avatar Forcing (Ki et al.) [19]	interactive head-avatar reactions	∼500 ms reaction latency; 6.8× speedup	Reacts to user audio/motion, but does not generate dialogue speech.
AvatarForcing (Cui et al.) [11]	one-step streaming talking avatar	34 ms/frame; 0.51 s audio-to-visual delay	Strong visual streaming metric, not perceptual dialogue.
LiveTalk [10]	multimodal interactive avatar video	24.82 FPS; 0.33 s first-frame latency	Uses Qwen3-Omni for speech reasoning; video latency is separate.
Hallo-Live [21]	text-driven joint audio-video avatar	20.38 FPS with 0.94 s latency	Text-driven; does not continuously perceive user audio-video.
OmniForcing [32]	text-to-audio-video streaming generation	TTFC ∼0.7 s; ∼25 FPS	First-chunk generation latency, not user response latency.
Wan-Streamer (ours)	text/audio/video perceptual dialogue with synchronized speech and video output	**25 FPS**; **∼550 ms total**; **∼200 ms model-side**	Single causal Transformer learns text I/O, perception, speaking, listening behavior, interruption, and visual response together.

Naturalness. Beyond response speed, Wan-Streamer improves interaction naturalness by continuously generating visible behavior during non-speaking intervals. In the idle state, the agent does not collapse into a frozen portrait; it maintains identity, gaze, posture, breathing, and subtle facial motion over the streaming history. In the listening state, the model can produce responsive non-verbal feedback such as gaze shifts, nods, micro-expressions, and posture changes that are temporally coupled with the user's speech and visual cues. Since speech and video latents are predicted from the same causal context before decoding, lip motion, facial dynamics, and prosody are synchronized natively rather than repaired by post-hoc alignment. These properties make the generated agent feel closer to a real interlocutor: it is visually present while waiting, attentive while the user speaks, and coherent when transitioning between listening, thinking, and speaking.

Interruption and proactive speaking. The full-duplex behavior of Wan-Streamer is learned from interleaved interaction data instead of being implemented only as hand-crafted turn-taking rules. During training, user inputs and agent outputs from text, audio, and video are placed on the same causal timeline, so the model observes when humans continue, pause, overlap, interrupt, yield, or resume. At inference time, the model keeps consuming user audio-video observations even while generating its own response, allowing it to stop, shorten, or redirect its speech when the user naturally interrupts. The same unified context also enables proactive speaking: when salient visual events, objects, expressions, or user actions appear in the input stream, the model can initiate a relevant comment or question based on what it sees, rather than waiting for an explicit spoken request. This turns interaction from a passive question-answer loop into a more human-like continuous exchange.

4 Related Works

Full-duplex spoken dialogue. Recent spoken dialogue models move beyond turn-based speech pipelines by modeling listening and speaking on a shared streaming timeline [9]. Moshi models user and assistant speech as parallel streams and removes explicit turn segmentation [12]; OmniFlatten adapts a GPT backbone into an end-to-end full-duplex speech-text dialogue model [49]; SALM-Duplex directly models continuous user speech inputs and codec-based assistant outputs [16]; and DuplexSLA further adds a synchronized action stream for planning and tool use during speech [48]. Commercial systems such as Seeduplex have also reported native end-to-end full-duplex speech interaction with always-on listening and interference suppression [5]. These works establish that real-time dialogue should not be formulated as alternating ASR–LLM–TTS turns. However, they are primarily speech or speech-text systems: they do not generate a visual agent, do not consume streaming video observations, and therefore cannot learn visual listening behavior, facial/body feedback, or audio-visual response timing as part of the same model.

Interactive digital humans and audio-visual avatars. Audio-driven avatar generation has progressed from portrait and talking-head synthesis toward real-time full-body and long-duration character animation [14, 23, 41]. Streaming methods such as TalkingMachines, StreamAvatar, LiveTalk, Hallo-Live, and OmniForcing improve latency and temporal consistency for audio-visual generation [10, 21, 25, 32, 34]. Recent interactive digital-human systems further connect multimodal understanding or dialogue modules with real-time visual generation, including MIDAS, MAViD, M.I.O, U-Mind, LPM, and X-Streamer [6, 8, 13, 28, 40, 47]. These systems are important steps toward visual conversational agents, but most of them still assemble separate modules: a speech or language model decides what to say, while an audio-driven or instruction-driven visual generator renders the character. As a result, duplex behavior is often handled by system-level control, VAD, turn-taking logic, or external audio models, rather than learned end to end together with visual perception and visual expression.

End-to-end audio-visual interaction. A smaller set of works has explored more unified audio-visual agents. Body of Her is a notable preliminary study of an end-to-end humanoid agent that integrates audio and visual inputs and models speech, full-body behavior, idling, response, and manipulation in real time [1]. FlowAct-R1 and related humanoid video models also point toward action-level interactive generation [38], while causal video and world-modeling methods provide tools for streaming rollouts and long-horizon consistency [7, 17, 24, 33, 39]. Wan-Streamer follows this end-to-end direction but targets a stricter setting: language, audio, and video appear on both the input and output sides, are modeled by a single Transformer, and are served with fully causal encoders, decoders, and full-history streaming inference. This allows perception, reasoning, speaking, visible listening, interruption handling, and synchronized audio-visual generation to be learned as one native full-duplex process rather than connected as a cascade.

5 Conclusion

We presented Wan-Streamer, a native-streaming, end-to-end foundation model for real-time full-duplex text, audio, and video interaction. Unlike cascaded systems that alternate among perception, language modeling, speech synthesis, and visual generation modules, Wan-Streamer represents user inputs and agent outputs across all modalities as one causal stream processed by a single Transformer. With fully causal audio and video VAEs, causal encoders and decoders, and a block-causal Transformer, the model can perceive current observations, generate synchronized audio-visual responses, emit each streaming unit, and commit the generated latents back into history with minimal delay. Together with the thinker-performer serving design, Wan-Streamer reaches sub-second interactive latency while preserving full-history context. The current v0.1 results are validated at a preliminary 192p output resolution; this serves as a proof of concept for the end-to-end streaming design, and scaling to higher resolutions is straightforward and left to future work. These results suggest that real-time multimodal agents should be designed from the ground up as native full-duplex systems, where listening, seeing, speaking, and visible response are learned jointly rather than assembled as post-hoc modules.

References

[1] Tenglong Ao. Body of her: A preliminary study on end-to-end humanoid agent. *arXiv preprint arXiv:2408.02879*, 2024.

[2] Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14834–14844, 2025.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] ByteDance Seed Team. Doubao realtime voice model. https://seed.bytedance.com/en/realtime_voice, 2025. Model page, January 20, 2025.

[5] ByteDance Seed Team. Introducing seed full-duplex speech llm: Attentive listening, robust interference suppression, enabling more natural interaction. ByteDance Seed Blog, 2026. Blog post, April 9, 2026.

[6] Yiyi Cai, Xuangeng Chu, Xiwei Gao, Sitong Gong, Yifei Huang, Caixin Kang, Kunhang Li, et al. Towards interactive intelligence for digital humans. *arXiv preprint arXiv:2512.13674*, 2025.

[7] Boyuan Chen, Diego Marti Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, 2024.

[8] Ming Chen, Liyuan Cui, Wenyuan Zhang, Haoxian Zhang, Yan Zhou, Xiaohan Li, Songlin Tang, Jiwen Liu, Borui Liao, Hejia Chen, et al. Midas: Multimodal interactive digital-human synthesis via real-time autoregressive video generation. *arXiv preprint arXiv:2508.19320*, 2025.

[9] Yuxuan Chen and Haoyuan Yu. From turn-taking to synchronous dialogue: A survey of full-duplex spoken language models. *arXiv preprint arXiv:2509.14515*, 2025.

[10] Ethan Chern, Zhulin Hu, Bohao Tang, Jiadi Su, Steffi Chern, Zhijie Deng, and Pengfei Liu. Livetalk: Real-time multimodal interactive video diffusion via improved on-policy distillation. *arXiv preprint arXiv:2512.23576*, 2025.

[11] Liyuan Cui, Wentao Hu, Wenyuan Zhang, Zesong Yang, Fan Shi, and Xiaoqiang Liu. Avatarforcing: One-step streaming talking avatars via local-future sliding-window denoising. *arXiv preprint arXiv:2603.14331*, 2026.

[12] 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. *arXiv preprint arXiv:2410.00037*, 2024.

[13] Xiang Deng, Feng Gao, Yong Zhang, Youxin Pang, Xu Xiaoming, Zhuoliang Kang, Xiaoming Wei, and Yebin Liu. U-mind: A unified framework for real-time multimodal interaction with audiovisual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10874–10886, 2026.

[14] Yikang Ding, Jiwen Liu, Wenyuan Zhang, Zekun Wang, Wentao Hu, Liyuan Cui, Mingming Lao, Yingchao Shao, Hui Liu, Xiaohan Li, Ming Chen, Xiaoqiang Liu, Yu-shen Liu, and Pengfei Wan. Kling-avatar: Grounding multimodal instructions for cascaded long-duration avatar animation synthesis. *arXiv preprint arXiv:2509.09595*, 2025.

[15] Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

[16] Ke Hu, Ehsan Hosseini-Asl, Chen Chen, Edresson Casanova, Subhankar Ghosh, Piotr Želasko, Zhehuai Chen, Jason Li, Jagadeesh Balam, and Boris Ginsburg. Salm-duplex: Efficient and direct duplex modeling for speech-to-speech language model. *arXiv preprint arXiv:2505.15670*, 2025.

[17] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[18] Hume AI. Introducing evi 3: The world's most realistic and instructible speech-language model. https://www.hume.ai/blog/introducing-evi-3, 2025. Blog post, 2025.

[19] Taekyung Ki, Sangwon Jang, Jaehyeong Jo, Jaehong Yoon, and Sung Ju Hwang. Avatar forcing: Real-time interactive head avatar generation for natural conversation. *arXiv preprint arXiv:2601.00664*, 2026.

[20] Latent.Space. Openai realtime api: The missing manual. https://www.latent.space/p/realtime-api, 2024. Technical blog, December 2024.

[21] Chunyu Li, Jiaye Li, Ruiqiao Mei, Haoyuan Xia, Hao Zhu, Jingdong Wang, and Siyu Zhu. Hallo-live: Real-time streaming joint audio-video avatar generation with asynchronous dual-stream and human-centric preference distillation. *arXiv preprint arXiv:2604.23632*, 2026.

[22] Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[23] Gaojie Lin, Jianwen Jiang, Jiaqi Yang, Zerong Zheng, and Chao Liang. Omnihuman-1: Rethinking the scaling-up of one-stage conditioned human animation models. *arXiv preprint arXiv:2502.01061*, 2025.

[24] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025.

[25] Chetwin Low and Weimin Wang. Talkingmachines: Real-time audio-driven facetime-style video via autoregressive diffusion models. *arXiv preprint arXiv:2506.03099*, 2025.

[26] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024. Blog post, May 13, 2024.

[27] OpenBMB Team. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

[28] Youxin Pang, Jiajun Liu, Lingfeng Tan, Yong Zhang, Feng Gao, Xiang Deng, Zhuoliang Kang, Xiaoming Wei, and Yebin Liu. Mavid: A multimodal framework for audio-visual dialogue understanding and generation. *arXiv preprint arXiv:2512.03034*, 2025.

[29] Qwen Team. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

[30] Qwen Team. Qwen3.5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

[31] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

[32] Yaofeng Su, Yuming Li, Zeyue Xue, Jie Huang, Siming Fu, Haoran Li, Ying Li, et al. Omniforcing: Unleashing real-time joint audio-visual generation. *arXiv preprint arXiv:2603.11647*, 2026.

[33] Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

[34] Zhiyao Sun, Ziqiao Peng, Yifeng Ma, Yi Chen, Zhengguang Zhou, Zixiang Zhou, Guozhen Zhang, Youliang Zhang, Yuan Zhou, Qinglin Lu, and Yong-Jin Liu. Streamavatar: Streaming diffusion models for real-time interactive human avatars. *arXiv preprint arXiv:2512.22065*, 2025.

[35] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

[36] Volcengine. Doubao end-to-end realtime voice model. Volcengine product page, 2025. Product page, 2025.

[37] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[38] Lizhen Wang, Yongming Zhu, Zhipeng Ge, Youwei Zheng, Longhao Zhang, Tianshu Hu, Shiyang Qin, et al. Flowact-r1: Towards interactive humanoid video generation. *arXiv preprint arXiv:2601.10103*, 2026.

[39] Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

[40] You Xie, Tianpei Gu, Zenan Li, Chenxu Zhang, Guoxian Song, Xiaochen Zhao, Chao Liang, Jianwen Jiang, Hongyi Xu, and Linjie Luo. X-streamer: Unified human world modeling with audiovisual interaction. *arXiv preprint arXiv:2509.21574*, 2025.

[41] Sicheng Xu, Guojun Chen, Yu-Xiao Guo, Jiaolong Yang, Chong Li, Zhenyu Zang, Yizhong Zhang, Xin Tong, and Baining Guo. Vasa-1: Lifelike audio-driven talking faces generated in real time. *arXiv preprint arXiv:2404.10667*, 2024.

[42] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[43] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

[44] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

[45] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024.

[46] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2024.

[47] Ailing Zeng, Casper Yang, Chauncey Ge, Eddie Zhang, Garvey Xu, Gavin Lin, Gilbert Gu, et al. Lpm 1.0: Video-based character performance model. *arXiv preprint arXiv:2604.07823*, 2026.

[48] Haoyang Zhang, Jun Chen, Donghang Wu, Yuxin Li, Yuxin Zhang, Xiangyu Tony Zhang, Che Liu, Qingjian Lin, Yizhou Peng, Hexin Liu, Eng Siong Chng, Chao Yan, Boyong Wu, Yechang Huang, Xuerui Yang, and Fei Tian. Duplexsla: A full-duplex spoken language model with synchronized speech, language, and action. *arXiv preprint arXiv:2605.20755*, 2026.

[49] Qinglin Zhang, Luyao Cheng, Chong Deng, Qian Chen, Wen Wang, Siqi Zheng, Jiaqing Liu, Hai Yu, Chaohong Tan, Zhihao Du, and Shiliang Zhang. Omniflatten: An end-to-end gpt model for seamless voice conversation. *arXiv preprint arXiv:2410.17799*, 2024.

Appendix

A Contributions and Acknowledgements

A.1 Core Contributors

Lianghua Huang, Zhifan Wu, Wei Wang, Yupeng Shi, Mengyang Feng, Junjie He, Chenwei Xie, Yu Liu, and Jingren Zhou.

A.2 Contributors

Contributors are listed alphabetically by first name: Ang Wang, Bang Zhang, Baole Ai, Chen Liang, Cheng Yu, Chongyang Zhong, Jinwei Qi, Kai Zhu, Pandeng Li, Peng Zhang, Wenyuan Zhang, Xinhua Cheng, Yitong Huang, Yun Zheng, and Zoubin Bi.