Wan-Streamer v0.2: Higher Resolution, Same Latency

Wan Team, Alibaba Group

See Contributions and Acknowledgements for the full author list.

Abstract

We present **Wan-Streamer v0.2**, a latency-preserving upgrade of the native-streaming, end-to-end audio-visual interaction model. v0.2 keeps the v0.1 modeling formulation, but raises the interactive output stream from 192×336 to **640×368** while preserving approximately **200 ms** model-side signal-to-signal latency at 25 FPS. The higher-resolution stream supports **scene-grounded mid-shot agents** whose posture, gaze, hands, nearby objects, and local scene layout remain legible during real-time conversation. To support the larger visual stream without adding user-visible delay, v0.2 keeps the thinker as a single-GPU low-latency path for streaming perception, the short language/state Transformer pass that builds the generation cache, and final decoding. The performer becomes a multi-GPU Ulysses-style context-parallel group for the expensive next-unit latent generation. Each performer rank writes incoming K/V into a pre-sharded local cache. The long high-resolution latent video sequence is split across ranks for denoising and gathered through Ulysses communication, while the much shorter audio latent sequence is generated without sequence sharding. In this split, the thinker's language/state computation reaches the performer only as K/V conditioning, so no separate language sequence has to be communicated inside the performer group. This concentrates additional hardware on visual generation while preserving the compact thinker-performer boundary, keeping total remote interaction latency at approximately 550 ms when a 350 ms bidirectional network budget is included.

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

1 Introduction

Wan-Streamer v0.2 directly upgrades Wan-Streamer v0.1 [1]. Real-time audio-visual interaction sits at the intersection of full-duplex spoken dialogue, multimodal perception, streaming video generation, and interactive digital humans. Full-duplex speech systems show that natural dialogue should not be reduced to alternating ASR–LLM–TTS turns [2, 3]. Omni-modal models extend perception to image, video, and audio inputs [4, 5], while video generation and causal rollout methods provide the visual synthesis and streaming foundations needed for interactive output [6–8]. In parallel, real-time avatars and digital-human systems have advanced audio-driven faces, streaming visual agents, and end-to-end embodied interaction [9–12].

Wan-Streamer v0.1 established a native-streaming formulation for this setting: user and agent text, audio, and video are represented on one causal timeline and modeled by a single Transformer. Unlike cascaded visual-agent systems, this formulation keeps perception, response timing, speech, visible listening behavior, and synchronized video response inside one causal interaction state. It closes the audio-visual interaction loop, but the preliminary 192p output limits the visual range. Close-up video-call framing preserves facial response and speaking behavior, while wider compositions leave body posture, nearby objects, and scene context too compressed for scene-grounded interaction.

Wan-Streamer v0.2 is a latency-preserving resolution upgrade. It raises the interactive output stream from

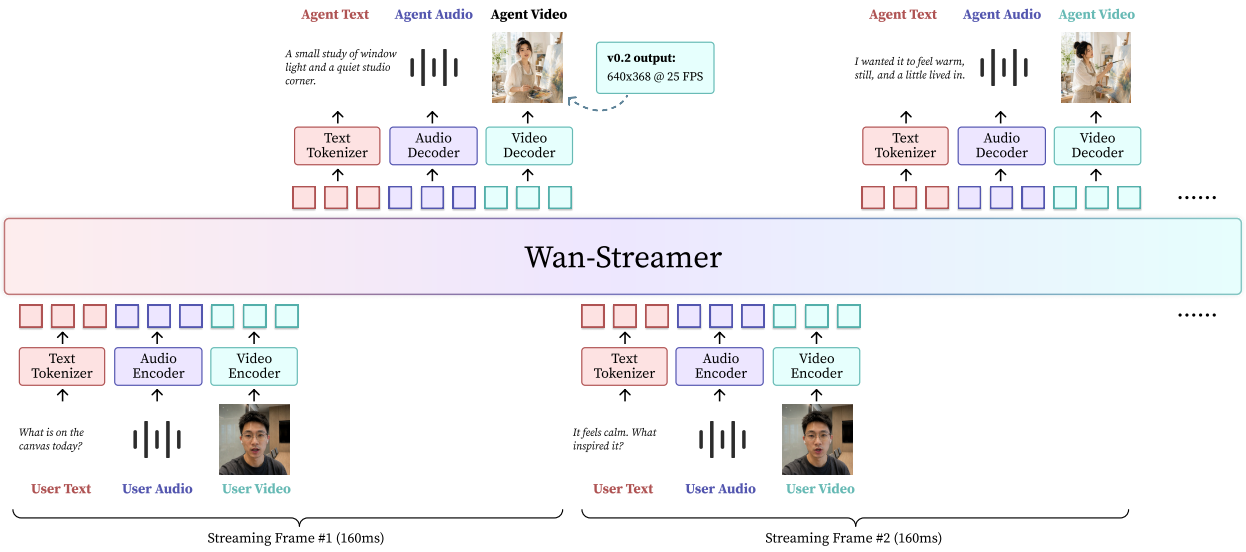

Figure 1 Wan-Streamer remains one native-streaming model: language, audio, and video inputs and outputs are represented on a shared causal timeline and coordinated by block-causal attention. v0.2 keeps this formulation while increasing the output resolution and changing the deployment strategy described in Fig. 2.

192×336 to 640×368 at 25 FPS while keeping approximately 200 ms model-side response latency. This target is constrained by streaming causality rather than offline rendering quality: every 160 ms unit must process current user observations, update the shared interaction state, generate synchronized speech and video latents, decode the previous unit, and emit the response without stretching the interaction cadence. With a 350 ms bidirectional network budget, the resulting remote interaction latency remains approximately 550 ms.

The larger stream changes the usable visual composition. v0.2 improves close-up video-call fidelity and supports scene-grounded mid-shot agents whose posture, gaze, hands, nearby objects, and local scene layout remain legible during real-time conversation. This expands the visual format from portrait-like calls toward situated conversations in which the agent remains visibly grounded in its surroundings.

To meet the same latency budget, v0.2 changes the serving topology while keeping the native-streaming formulation fixed. The thinker remains a single-GPU, low-latency path for streaming perception, language/state update, KV-cache construction, and final causal decoding; this language/state pass is the part that produces the K/V cache used by generation. The 640×368 latent generation path moves into a multi-GPU Ulysses-style context-parallel performer group [13]. The thinker broadcasts compact performer-compatible K/V slices, each performer rank writes them into its pre-sharded local cache, and denoising for the long latent video sequence is split across ranks with Ulysses all-to-all/gather communication. Audio latents for each streaming unit contain far fewer tokens, so they are generated without sequence sharding. This isolates the added visual-generation cost from the latency-critical control path while preserving the compact thinker-performer boundary.

Figure 1 shows the native-streaming formulation inherited from v0.1. Against this baseline, v0.2 changes three axes: the output stream increases to 640×368, the high-cost latent generation path moves into a Ulysses-style context-parallel performer, and the supported visual composition expands from close-up calls to scene-grounded mid-shot agents. The rest of the paper follows these axes: Sec. 2 summarizes the version comparison, Sec. 3 describes latency-preserving serving, and Sec. 4 describes the experiments.

Our contributions are:

- We upgrade Wan-Streamer from 192×336 to 640×368 video output while keeping approximately 200 ms model-side response latency.
- We introduce a v0.2 serving topology with a single-GPU thinker and a Ulysses-style context-parallel multi-GPU performer, using pre-sharded performer-side K/V caches and sequence parallelism for the high-resolution latent video denoising path.
- We expand the visual interaction scope from close-up calls to higher-fidelity close-up interactions and

scene-grounded mid-shot agents with readable body and scene context.

2 Upgrade Design

2.1 Stable native-streaming formulation

v0.2 keeps the core Wan-Streamer formulation stable. The model is still trained as one end-to-end causal stream: user text, audio, and video observations update the same history that conditions agent text, speech, and video responses. Generated audio-video latents are committed back into history after each unit, so the next response can depend on both the user's recent behavior and the agent's own previous expression. This formulation is inherited from v0.1 and serves as the baseline for the v0.2 upgrade.

v0.2 expands the visual range around this formulation. The visual target moves from 192p to 640×368, and the data emphasis moves from mostly close-up call framing to wider, scene-grounded conversational settings. In mid-shot composition, the model must preserve identity, gaze, hand and torso posture, local objects, and scene layout while continuing to listen and speak in real time.

2.2 Version comparison

Table 1 summarizes the version-level changes in v0.2: the end-to-end streaming formulation and latency budget stay fixed, while output resolution, visual format, and serving topology change.

Table 1 Summary of the Wan-Streamer v0.1 to v0.2 upgrade. Emphasized cells indicate changed components or preserved latency targets.

Aspect	v0.1	v0.2
Output resolution	192×336	**640×368**
Frame rate	25 FPS	25 FPS
Model-side latency	∼200 ms	∼200 ms, **unchanged**
Total interaction latency	∼550 ms with a 350 ms bidirectional network budget	∼550 ms with the same 350 ms bidirectional network budget
Thinker	Streaming perception, state update, KV construction, and decoding	Same role, **kept on one GPU**
Performer	Single-GPU latent generation	**Multi-GPU Ulysses-style context-parallel** latent generation
Communication	Thinker-performer K/V and latent exchange	Thinker broadcasts performer-compatible K/V slices; Ulysses all-to-all/gather for the latent video sequence stays inside the performer group
Visual presence	Close-up video-call framing	Higher-fidelity close-up interactions plus **scene-grounded mid-shot agents with readable body and scene context**

3 Latency-Preserving Serving

The serving challenge in v0.2 is to allocate the additional 640×368 generation cost without slowing down the interactive loop. We split the deployed model into two roles:

- **Thinker.** A single GPU hosts the causal audio/video encoders, the token-causal Transformer path for language and state update, KV-cache construction, and the causal decoders that turn returned latents into output audio and video. The language/state path is reflected in the K/V cache that conditions generation.
- **Performer.** A Ulysses-style context-parallel GPU group hosts the expensive flow-matching latent generation path. Performer ranks keep pre-sharded K/V caches, split the long latent video sequence across ranks, and communicate through Ulysses all-to-all/gather collectives around attention. The audio latents are short enough that sequence sharding would add overhead rather than useful parallelism, so they are generated without sequence sharding.

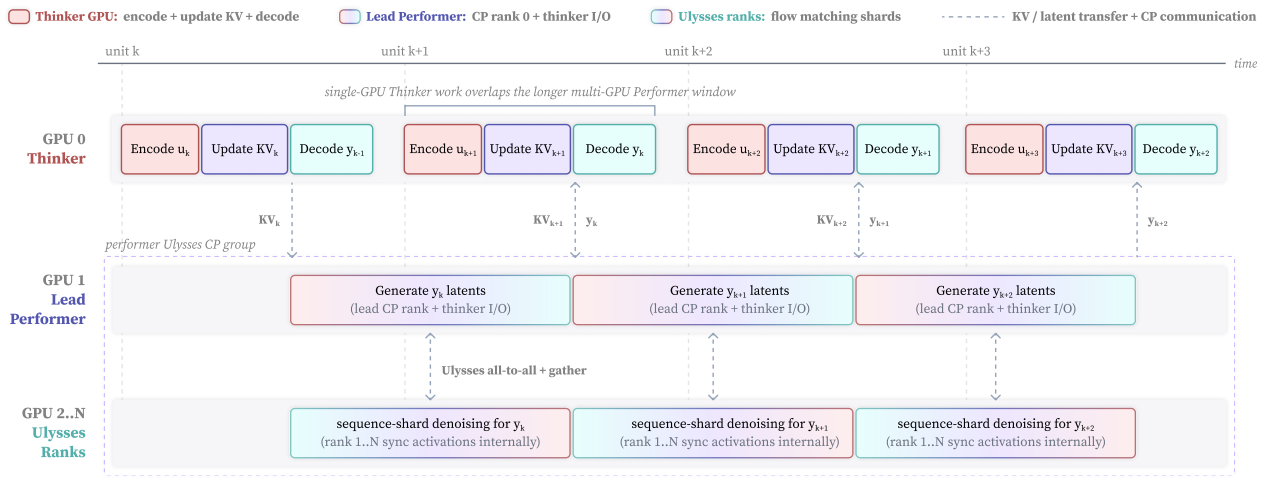

Figure 2 v0.2 latency-preserving serving. The thinker stays on one GPU and owns perception, language/state update, K/V construction, and final decoding. The performer uses Ulysses-style context parallelism for the 640×368 latent generation path: K/V slices are written into pre-sharded performer caches, the high-resolution latent video sequence is split across ranks for denoising, and Ulysses all-to-all/gather communication stays inside the performer group. Audio latents are short and are not split across ranks.

As shown in Fig. 2, at streaming unit k, the thinker consumes the current user observations and produces a new performer-compatible K/V slice. Around the same boundary, it receives the previous unit's generated latents, decodes them, and emits the response. The performer ranks receive the current K/V slice, update their local shards of the full-history cache, and run Ulysses context-parallel denoising for the next unit. The high-resolution latent video is the main sequence-parallel path. The audio latents are much shorter, so they stay unsharded; the language/state computation has already been folded into the K/V slice produced by the thinker.

This schedule separates throughput from response latency. Real-time throughput requires the performer group time plus thinker-performer K/V and latent transfer plus intra-performer Ulysses communication to fit inside one 160 ms unit. The model-side response latency is the signal-to-signal path through encoding, state update, latent generation, and decoding; this remains approximately 200 ms. The key constraint is that the additional v0.2 work is concentrated in the context-parallel performer, while the thinker remains a compact low-latency interaction path.

4 Experiments

Latency and runtime protocol. We use the same response boundary as Wan-Streamer v0.1. 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. Under the serving path in Sec. 3, v0.2 keeps approximately 200 ms model-side latency while producing 640×368 video at 25 FPS. With the same 350 ms bidirectional network budget used in v0.1, the total remote interaction latency remains approximately 550 ms. We keep this network term as an external deployment assumption, so the comparison isolates the v0.2 model-side and serving changes; bandwidth-limited transport effects are outside the model-side latency measurement reported here.

Public real-time systems report different endpoints, including first-packet speech latency, first-frame delay, FPS, audio-to-visual delay, or product-level response time [10, 14–17]. We therefore keep the v0.1 measurement convention and report the v0.2 runtime at the same response boundary.

Qualitative visual observations. We use generated 640×368 conversations as qualitative observations of the upgraded output format. The inspection focuses on visual stability and legibility during both listening and speaking intervals, including facial detail, gaze, mouth motion, hands, posture, nearby objects, and local scene layout.

These observations characterize the v0.2 output format: clearer close-up calls and scene-grounded mid-shot agents under the same low-latency streaming setting.

5 Conclusion

Wan-Streamer v0.2 keeps the native full-duplex formulation of v0.1 while raising the interactive stream from 192×336 to 640×368 at approximately 200 ms model-side latency. The single-GPU thinker preserves the latency-critical loop, while the Ulysses-style context-parallel performer absorbs the added visual latent-generation cost through pre-sharded K/V caches and sequence parallelism for the high-resolution latent video denoising path. This yields clearer video-call interaction and scene-grounded mid-shot agents under the same low-latency streaming setting.

References

[1] Lianghua Huang, Zhi-Fan Wu, Yupeng Shi, Wei Wang, Mengyang Feng, Junjie He, Chen-Wei Xie, Yu Liu, Jingren Zhou, 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. Wan-Streamer v0.1: End-to-end Real-time Interactive Foundation Models, June 2026. URL https://arxiv.org/abs/2606.25041. Submitted on 23 Jun 2026.

[2] 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.

[3] 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.

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

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

[6] 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.

[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] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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. *arXiv preprint arXiv:2309.14509*, 2023.

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

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

[16] 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.

[17] 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.

Appendix

A Contributions and Acknowledgements

A.1 Core Contributors

Lianghua Huang, Zhi-Fan Wu, Yupeng Shi, Wei Wang, Mengyang Feng, Junjie He, Chen-Wei 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, Yuxiang Bao, Yuzheng Wang, and Zoubin Bi.