Preprints.org

Review

Not peer-reviewed version

Native Real-Time Interactive Omni Intelligence: A Unified Framework

Yifan Shen †, Zhuoqing Zhong †, Xinzhuo Li †, Jiateng Liu †, Bingxuan Li †, Wei Cao †, Pei Tian, Lin Zhu, Jian Xu, Zewei Chen, Tianjiao Yu, Haichao Zhang, Jiawen Zhang, Yuner Zhang, Brian Nlong Zhao, Sen Fang, Nan Huang, Bowen Fang, Chen Fang, Boyi Li, Onkar Susladkar, Frank Yang, Yuanzhe Liu, Jingyuan Zhu, Cheng Qian, Yijiang Li, Xinyang Han, Yuan Shen, Xiang Li, Bolin Lai, Xu Cao, Chengde Wan, Yiwen Song, Yue Guo, Heng Ji, Xuan Di, Yiwei Wang, Yaoyao Liu, Dimitris N. Metaxas, Yun Fu, James M. Rehg, Chengxiang Zhai, Ismini Lourentzou *

Posted Date: 19 October 2026

Keywords: omni intelligence; omni-models; multimodal foundation models; real-time interaction; full-duplex systems; streaming generation; multi-rate intelligence

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC, OpenAlex.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Review

Native Real-Time Interactive Omni Intelligence: A Unified Framework

Yifan Shen [1,†], Zhuoqing Zhong [2,†], Xinzhuo Li [1,†], Jiateng Liu [1,†], Bingxuan Li [1,†], Wei Cao [1,†], Pei Tian [3], Lin Zhu [1], Jian Xu [1], Zewei Chen [1], Tianjiao Yu [1], Haichao Zhang [4], Jiawen Zhang [1], Yuner Zhang [5], Brian Nlong Zhao [1], Sen Fang [6], Nan Huang [1], Bowen Fang [3], Chen Fang [1], Boyi Li [1], Onkar Susladkar [1], Frank Yang [1], Yuanzhe Liu [1], Jingyuan Zhu [5], Cheng Qian [1], Yijiang Li [7], Xinyang Han [8], Yuan Shen [1], Xiang Li [1], Bolin Lai [1], Xu Cao [1], Chengde Wan [9], Yiwen Song [10], Yue Guo [1], Heng Ji [1], Xuan Di [3], Yiwei Wang [11], Yaoyao Liu [1], Dimitris N. Metaxas [6], Yun Fu [4], James M. Rehg [1], Chengxiang Zhai [1] and Ismini Lourentzou [1,*]

[1] University of Illinois Urbana-Champaign
[2] Carnegie Mellon University
[3] Columbia University
[4] Northeastern University
[5] University of Pennsylvania
[6] Rutgers University
[7] University of California, San Diego
[8] University of California, Berkeley
[9] Independent Researcher
[10] Google
[11] University of California, Merced
* Correspondence: lourent2@illinois.edu
† Core Contributor.

Abstract

The rapid progress of large language models has established strong foundations for reasoning, knowledge, and generation, while the integration of vision, audio, video, and other modalities has extended these capabilities toward increasingly unified intelligence across modalities, which we refer to as Omni Intelligence. Yet, real-world assistants must do more than process completed multimodal prompts. They must perceive evolving streams, respond while new evidence is arriving, handle hesitation, overlap, interruption, and changing intent, and coordinate reasoning or tool use that may proceed asynchronously. We refer to this broader capability as **Interactive Omni Intelligence**: the ability to maintain and exercise omni intelligence under the temporal and adaptive demands of real-time interaction. The question guiding this paper is: *How can foundation-model capabilities be carried into real-time interaction while preserving both intelligence and interactivity?* We develop a conceptual framework connecting two development paths. One adapts pretrained language or omni models through streaming interfaces, modality modules, post-training, and orchestration. The other develops model-native systems that learn temporal structure, continuous multimodal processing, generation, and interaction control. Across both paths, we compare representations, architectures, fusion, data, training objectives, serving, and evaluation. Their comparison reveals an Intelligence–Interactivity Trade-off between preserving reasoning and knowledge, and meeting the timing, coordination, and resource demands of continuous operation. We examine Multi-Rate Interactive Intelligence as a cross-cutting direction, including foreground–background designs that coordinate fast interaction with slower reasoning, retrieval, memory, and tools. Finally, we identify open problems in evolving state, persistent and proactive interaction, joint evaluation, and safe open-world operation. The resulting framework connects capability preservation, interaction control, and multi-rate coordination.

Keywords: omni intelligence; omni-models; multimodal foundation models; real-time interaction; full-duplex systems; streaming generation; multi-rate intelligence

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

1. Introduction

Artificial intelligence has progressed from language-centered models toward multimodal foundation models that can perceive, reason over, and generate content across multiple modalities such as text, images, audio, video, 3D geometry, dynamic 4D representations, and other forms of contextual information [1–10]. Recent omni models integrate several of these modalities within a shared backbone [11–14]. We use **Omni Intelligence** to describe this breadth of multimodal perception, reasoning, knowledge, and generation.

Many current omni models still formulate multimodal interaction as completed prompts or conversational turns [11,12,14]. The model receives a sufficiently complete context and produces a response afterward. However, in real-world interaction, audio and video continue to arrive while the model is reasoning or generating a response, and results from retrieval, tools, or external computation may arrive after the interaction state has changed.

We define **Interactive Omni Intelligence** as the ability to maintain and use omni intelligence while multimodal input, system output, and interaction state continue to evolve. The model must judge when the available evidence is sufficient, whether silence signals hesitation or turn completion, whether speech is addressed to the system, and how new input should affect an active response. Pauses, overlap, backchannels, interruptions, and revisions therefore carry both semantic and control information [15–17].

Current systems support live interaction through three primary ways. Adapted systems add streaming interfaces, modality modules, and interaction controllers to pretrained language or omni models. Cascaded VAD–ASR–LLM–TTS systems are a common instance of this category [18–20]. Model-native systems represent temporal events explicitly and learn speech timing, overlap, interruption handling, or multimodal generation within the model. This group includes speech-native full-duplex models and streaming omni input-output models [15,21–24]. Foreground–background systems connect a low-latency interaction process to slower reasoning, retrieval, memory, or tool execution [25–27]. Such systems can be combined with adapted or model-native systems to coordinate fast interaction with slower computation.

These designs balance intelligence and interactivity differently. Adapted systems retain strong pretrained reasoning and knowledge, although their interaction loop may depend on coordination across several interfaces. Model-native systems incorporate temporal events directly into their state updates. Their requirements include synchronized interaction data, additional streaming computation, and training that preserves existing capabilities [28,29]. Foreground–background systems allow acknowledgements, turn management, and interruption handling to proceed while slower computation continues. Returned results must then be checked against the latest interaction state. These competing demands during training and runtime give rise to the **Intelligence–Interactivity Trade-off** [30–32].

The central challenge is therefore to build systems that can continuously update their understanding of the world, coordinate processes operating at different temporal scales, and express their capabilities at the appropriate moment without sacrificing reasoning quality, responsiveness, or safety. By organizing existing work around this joint intelligence–interactivity perspective, we aim to provide a common conceptual foundation for comparing current systems, identifying missing capabilities, and guiding the development of omni models that can participate naturally and reliably in persistent, open-ended interaction.

We analyze two closely related requirements. **Omni Intelligence** concerns the system's ability to integrate information across modalities, understand the situation, and reason about it. **Omni Interactivity** concerns its ability to respond appropriately as the interaction unfolds: deciding when to respond or act, when to wait, and when to stop or revise an ongoing response. These requirements are coupled because incoming information can change both what the system understands and how it should respond. Figure 1 summarizes the framework, from real-time interaction requirements and streaming representations to model construction, execution, evaluation, and multi-rate coordination.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

We first define the field and position it relative to multimodal language models, spoken dialogue systems, streaming models, and interactive agents in Section 2. With this foundation, we further address three questions. First, how should continuous multimodal streams be represented, and how should models be built and trained to process them? Sections 3 and 4 examine representations, architectures, data, and training objectives. Second, how can a system maintain multimodal understanding and coordinate its responses during ongoing interaction? Sections 5 and 6 cover evolving state, grounding, reasoning, memory, responsiveness, turn-taking, interruption, and serving. Third, how should intelligence and interactivity be evaluated together, and how can their competing demands be managed? Finally, Section 7 covers evaluation, Section 8 analyzes capability and runtime tensions, and Section 9 develops the open research agenda.

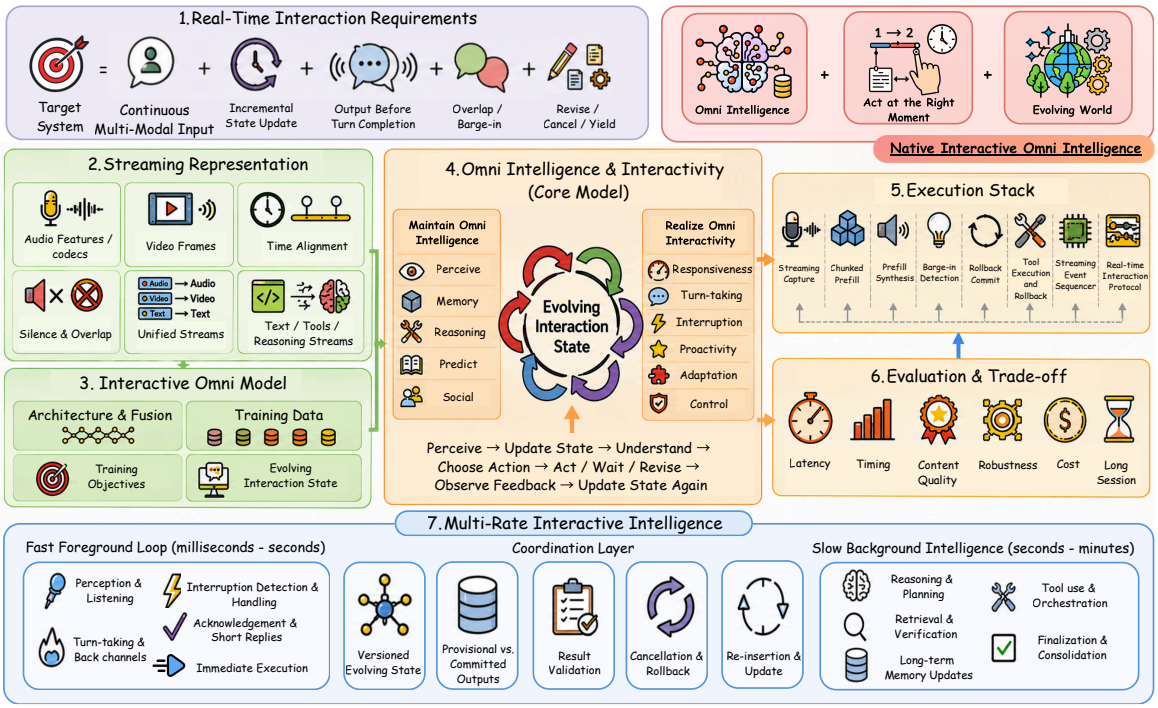


Figure 1. Native Interactive Omni Intelligence architecture. The framework integrates continuous multimodal interaction, streaming representations, an interactive omni-model, and a core intelligence layer that maintains an evolving interaction state. It supports real-time capabilities such as turn-taking, interruption handling, adaptation, tool use, and streaming execution. A multi-rate design coordinates fast foreground interaction with slower background reasoning, retrieval, memory updates, and orchestration, while balancing latency, quality, robustness, cost, and long-session performance.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Contributions

- **A framework for Interactive Omni Intelligence.** We define Interactive Omni Intelligence through multimodal understanding and response control during ongoing interaction, position it relative to adjacent fields, and compare adapted, model-native, and foreground–background implementations.
- **An end-to-end synthesis of native interaction.** We connect streaming representations, architectures, modality fusion, training data and objectives, state maintenance, response control, and real-time serving as successive layers of native interaction.
- **Joint evaluation of intelligence and interactivity.** We organize evaluation around latency, timing, content, robustness, long-session quality, and repair, and map current benchmarks to the combinations of intelligence and interactive behavior they evaluate.
- **The Intelligence–Interactivity Trade-off and a multi-rate research agenda.** We synthesize evidence that real-time adaptation and execution create capability-preservation and runtime coordination challenges, and identify Multi-Rate Interactive Intelligence, evolving state, persistent and proactive interaction, and safe open-world operation as central directions toward systems that can preserve intelligence while supporting natural interaction.

2. A Framework for Interactive Omni Intelligence

As shown in Figure 1, our proposed conceptual framework organizes Interactive Omni Intelligence around two coupled components: Omni Intelligence and Omni Interactivity. This section develops the framework by distinguishing modality coverage from real-time interaction requirements and by positioning related system families according to how they implement interaction control.

2.1. From Omni Intelligence to Real-World Interactivity

Omni Intelligence concerns the breadth and integration of model capabilities across language, vision, audio, and other modalities [2,3]. Interactive Omni Intelligence adds a temporal dimension: the ability to use these capabilities and adapt responses as new information arrives and interaction unfolds. Research toward this goal draws on spoken dialogue systems [33–36], multimodal foundation models [1,2], streaming generation [21,22,37], low-latency serving [38,39], and interactive agents [40,41]. Although these areas often share modalities and components, they differ in their assumptions about input availability, response timing, and whether ongoing responses can be interrupted or revised. We therefore distinguish the modalities a system supports from how it processes incoming information and coordinates its responses over time.

We characterize real-time Interactive Omni Intelligence through three properties. First, **Streaming Input**: input can arrive incrementally through audio and video streams, text updates, changes in screen state, tool feedback, and other signals. Second, **Evolving State**: the system can incorporate new information into its interaction state while processing earlier input or producing a response. Third, **Online Response Control**: it can initiate, revise, or stop outputs while interaction is ongoing, including speech, text, tool calls, and interface actions, without waiting for a fixed end-of-turn boundary. Together, these properties distinguish continuous interaction from conventional turn-based multimodal inference, in which a system receives a completed input before generating a response. Figure 2 illustrates this distinction in the context of spoken interaction.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

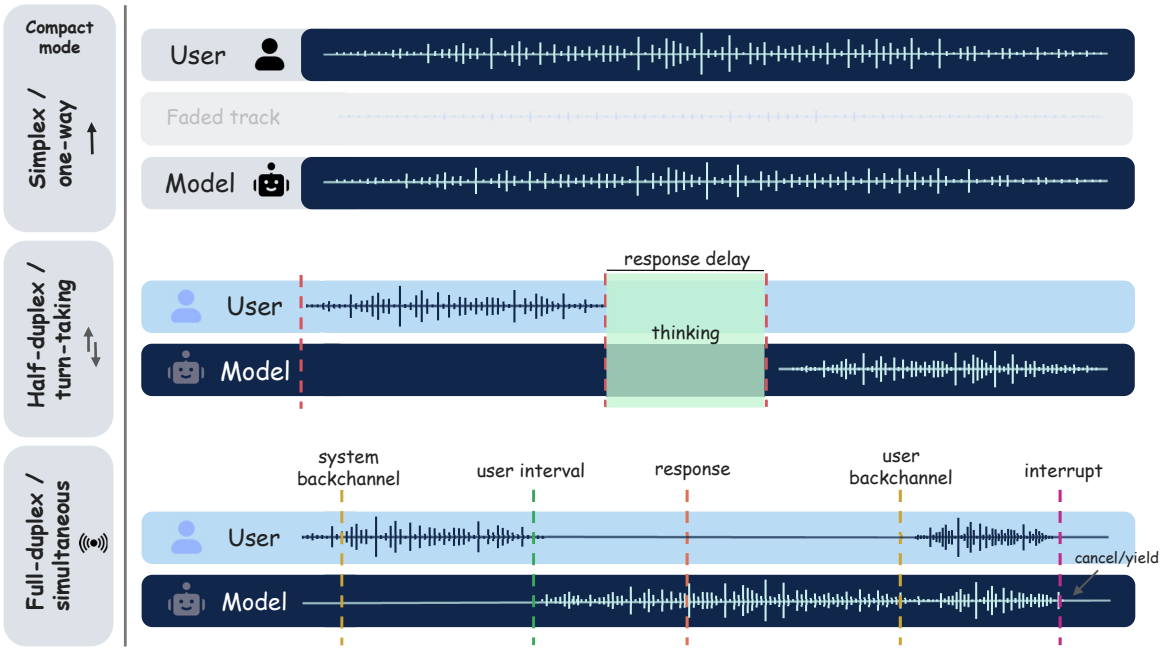


Figure 2. Spoken interaction in simplex, half-duplex, and full-duplex systems. Communication is unidirectional, alternates between participants, or allows overlapping speech and simultaneous listening and speaking, respectively.

These properties describe interactive behavior without specifying how it is implemented. An adapted system can meet these requirements through streaming interfaces, external controllers, or orchestration around a pretrained model [42,43]. We use *native interactive omni model* for models that learn temporal alignment, incremental multimodal processing, and interaction control as integral parts of their architecture and training. Interactivity describes system behavior during ongoing interaction, while nativeness identifies where the mechanisms supporting that behavior are implemented and learned.

This distinction positions the survey at the intersection of two research directions: full-duplex spoken language modeling and general omni and multimodal modeling. Full-duplex spoken language modeling studies temporal coordination, overlapping speech, interruption, and turn-taking, primarily in spoken interaction [34,36]. General omni and multimodal modeling supports a broader range of perceptual and generative capabilities, but many systems still assume completed inputs or discrete conversational turns [1,2]. Interactive Omni Intelligence brings these directions together: it asks how models can maintain broad multimodal understanding and generation while continuously incorporating new information, coordinating their outputs, and adapting to changes in the interaction. This interaction couples updates to the system's understanding with response selection based on the updated state [44].

Within this conceptual framework, characterizing Omni Interactivity requires clarifying the relationship between response latency and continuous interaction. Endpoint detection [33] and inference optimizations such as KV-cache management [38,45], speculative decoding [46], and latency-aware serving [39] can reduce response latency without changing a system's turn-based operation. Conversely, a system can process streaming inputs and support interruption while still requiring substantial computation. We therefore examine which input streams can update the system's state, when outputs can begin or change, and which components control multimodal fusion, response timing, and interruption. This also distinguishes our scope from surveys of spoken dialogue, spoken language models, and full-duplex speech interaction [34–36], and provides a common framework for comparing speech-native, multimodal, modular, and agentic systems.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

2.2. Taxonomy of System Families

We use the conceptual framework to examine four overlapping system families, followed by examples of how deployed systems combine them. They illustrate different combinations of modality coverage, temporal interaction capabilities, and the locus of interaction control. Figure 3 summarizes their relationships.

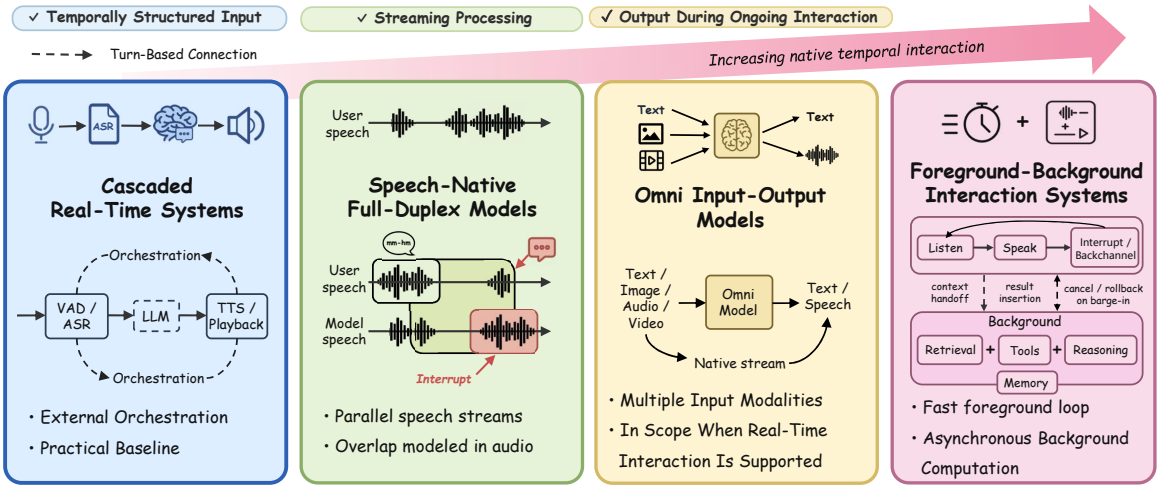


Figure 3. System families for real-time Interactive Omni Intelligence. Cascaded systems use external interaction controllers; speech-native full-duplex models learn speech timing and coordination; omni models support broader modalities with varying interactive capabilities; and foreground–background systems coordinate processes operating at different rates.

Cascaded real-time systems.

A conventional real-time assistant chains separately trained components such as voice activity detection, streaming ASR, an LLM, and incremental TTS [34,36]. Streaming recognition can build on online mechanisms such as RNN-Transducer and strong speech front-ends such as Whisper [18,47]. These systems are modular, deployable, and easy to debug, and sufficiently aggressive streaming can provide low perceived latency. However, timing, turn-taking, interruption, and recovery are typically coordinated outside the language model [36,42]. Textual interfaces between components may also discard information carried by prosody, laughter, hesitation, overlap, and backchannel timing [15,21]. Cascaded systems therefore provide a modular baseline in which real-time interaction is coordinated primarily by components outside the pretrained model.

Speech-native full-duplex models.

Speech-native full-duplex models represent user and model speech as synchronized or parallel streams and learn aspects of turn-taking, overlap, interruption, and backchanneling directly from conversational sequences. dGSLM [15], Moshi [21], and SyncLLM [48] illustrate this progression through two-channel dialogue modeling, parallel user–model speech streams, and language modeling synchronized to a continuous interaction clock. These models directly learn temporal coordination, primarily through speech, and provide methods that can inform broader multimodal interaction.

Omni input and output models.

Omni models broaden perceptual and generative capabilities across text, images, audio, and video. Qwen2.5-Omni [22], for example, uses block-wise audio and visual processing, temporally aligned multimodal position encoding, and a Thinker–Talker architecture supporting streaming text and speech generation; Qwen2-VL provides an important vision–language predecessor [49]. LLaMA-Omni [50], LLaMA-Omni2 [51], AnyGPT [52], Mini-Omni [37], Mini-Omni2 [53], VITA [54], and MiniMind-O [55] further illustrate efforts to unify multimodal perception with text or audio generation. However, modality breadth alone does not imply Interactive Omni Intelligence. Streaming output, for

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

example, does not by itself establish that new input can continuously modify an ongoing response, that overlap and interruption are modeled, or that the system can revise its behavior as the interaction evolves. Our core scope therefore focuses on systems that combine these multimodal capabilities with the real-time interaction properties defined above.

Foreground–background and multi-rate systems.

This system family cuts across the preceding categories and focuses on coordinating components that operate at different computational rates. A fast foreground process remains synchronized with the interaction, while slower processes perform retrieval, reasoning, memory access, planning, or tool use asynchronously. Building on retrieval-augmented generation [56], MoshiRAG [26] and Stream RAG [27] demonstrate how retrieval or tool use can proceed while spoken interaction remains active. STITCH interleaves speech generation with internal reasoning, while EgoMem coordinates asynchronous retrieval, dialogue, and long-term memory processes [57,58].

We view this family as a broader pattern of Multi-Rate Interactive Intelligence. It directly addresses the Intelligence–Interactivity Trade-off developed in this survey: conversational interaction may require decisions on the scale of tens or hundreds of milliseconds, whereas reasoning, retrieval, memory consolidation, or tool execution may require substantially longer computation. Separating these processes allows slower computation to continue without blocking immediate responses, but requires synchronization, validation of returned results, cancellation, and updates to the interaction state. Multi-rate designs can appear in both adapted and model-native systems.

Deployed real-time systems.

Recent industry systems support ongoing multimodal interaction. Google's Gemini Live [59] supports low-latency interaction over speech and visual inputs, allowing the system to respond as new speech and visual information arrives. Thinking Machines Lab's Interaction Model [60] advances this direction by representing interaction as continuous, time-aligned multimodal streams, processing audio, video, and text through fine-grained micro-turns so that pauses, interruptions, overlapping speech, and visual events remain part of the model's temporal context. OpenAI's GPT-Realtime-2 [61] targets real-time conversational applications with native text and audio input/output, image input, tool use, and low-latency streaming, although its currently documented interface does not include native video input. Alibaba's Qwen3.5-Omni-Realtime [62] extends real-time interaction to text, image, audio, and video inputs, supporting streaming spoken responses and multimodal grounding within a persistent interactive session. Similarly, ByteDance's SeedRealtime [63] is designed as a native audio-visual full-duplex model that jointly processes continuous audio, video, and text streams, with an emphasis on natural conversational timing, visual awareness, and integrated tool use. These systems increasingly integrate multimodal perception, response generation, and temporal coordination within real-time models.

3. Streaming Multimodal Representations

Native interactive models must process information as it arrives, without assuming that all relevant input is already available. Audio, video, text, and tool feedback, therefore, require representations that preserve their timing and support incremental processing by the model backbone. This section reviews these representations and four consequences of their design: the token rate the model must sustain, the information it can retain, the minimum delay before new input can influence a response, and the data and objectives available for training. These choices affect both pillars of the survey. Token rate and encoding delay constrain *omni interactivity*, while the preservation of content and temporal structure supports *omni intelligence*, including prosodic understanding and temporal grounding. Training data and objectives shape both. Representation design is therefore an early source of the intelligence–interactivity trade-off, with implications for subsequent architectural and training choices. Each subsection examines how its mechanisms support the two pillars and what computational, informational, or latency costs they introduce.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

We proceed from audio representations in Section 3.1 and streaming video representations in Section 3.2 to representations of time, silence, and overlap in Section 3.3. Section 3.4 then examines how multiple streams are serialized into model input sequences, while Section 3.5 covers the text, tool, and background reasoning streams that accompany ongoing interaction. The last group provides the representations needed to coordinate foreground interaction with background computation in the dual-model pattern discussed in Section 4.

3.1. Audio Representations

An interactive omni model converts continuous microphone input into features or tokens that the language-model backbone can process under a latency constraint. The audio front-end determines the token rate, the paralinguistic information retained, and the minimum encoding delay before the model can respond. Speech representations have been surveyed for spoken dialogue broadly [34,35] and for discrete audio tokens specifically [64,65]. We review mel-spectrogram features, self-supervised speech units, and neural codec tokens, then examine semantic and acoustic information, tokenization granularity, and the choice between discrete tokens and continuous features.

Mel features and their discretization.

Log-mel spectrograms provide frame-based time-frequency features for audio encoders such as Whisper [18] and the block-wise audio towers of omni models [22]. They retain paralinguistic information and are commonly used for speech understanding. dMel directly discretizes mel filterbank energies into intensity bins, producing a training-free tokenizer that supports ASR and TTS within a single decoder [66]. MELLE instead predicts continuous mel frames with a regression head, providing an alternative to vector quantization [67].

Self-supervised speech units.

A second family uses self-supervised learning to obtain representations of linguistic content without transcripts. Contrastive and masked-prediction objectives such as CPC [68], wav2vec 2.0 [69], HuBERT [70], w2v-BERT [71], and WavLM [72] produce frame-level embeddings whose intermediate layers align strongly with phonetic content [73,74]. Quantizing these embeddings, either with learned codes [75] or with offline k-means as popularized by HuBERT, yields *discrete semantic units*. GSLM demonstrated that a language model trained purely on such units can generate intelligible speech, establishing the textless pipeline of speech encoder, unit language model, and vocoder [76]. Semantic units provide compact representations of linguistic content, but their compression may remove speaker and prosodic information needed for expressive full-duplex conversation.

Neural codec tokens.

Neural audio codecs optimize waveform reconstruction at low bitrates, and their discrete code indices can be used as model tokens. SoundStream uses a fully convolutional causal encoder and decoder with residual vector quantization (RVQ), reconstruction and adversarial losses, and quantizer dropout. It compresses 24 kHz audio to 3–18 kbps in a single model and supports real-time streaming [77]. Subsequent codecs refine different parts of this design. EnCodec introduces multi-scale spectrogram discriminators and produces a 75 Hz token stream [78], DAC addresses RVQ codebook underuse with factorized, L_2-normalized codes and supports high-fidelity coding of 44.1 kHz general audio [79], and group-RVQ variants reduce the number of quantizers required [80], while lightweight inverse-STFT decoders accelerate token-to-waveform generation [81].

Because codec indices form a finite vocabulary, they can be modeled autoregressively. AudioLM applies autoregressive modeling to audio tokens [82], and VALL-E formulates zero-shot TTS as codec-token language modeling [83].

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Semantic versus acoustic tokens.

Semantic tokens emphasize linguistic content and achieve relatively low token rates, but retain less information for waveform reconstruction. Acoustic codec tokens prioritize reconstruction fidelity, although their linguistic structure is often less explicit [34,64]. AudioLM combines the two in a coarse-to-fine hierarchy, generating semantic tokens for long-range coherence before acoustic tokens for waveform fidelity [82]. Subsequent tokenizers incorporate semantic information directly into codec representations. SpeechTokenizer distills HuBERT representations into the first RVQ level, leaving later levels to encode residual acoustic detail [84]. Mimi uses a split-RVQ design with a WavLM-distilled semantic codebook and parallel acoustic codebooks [21]. X-Codec incorporates SSL features to strengthen the semantic information carried by acoustic codes [85]. FACodec separates content, prosody, and timbre into quantized subspaces [86], while SemantiCodec represents general audio at fewer than 100 tokens per second [87]. For interactive models, these representations must often support both understanding and generation within the same audio-token interface.

Codebook depth, frame rate, and the latency versus quality trade-off.

When codebooks are serialized, frame rate and codebook depth determine the token rate, while total sequence length additionally depends on audio duration. Deeper RVQ stacks can improve reconstruction fidelity but increase the number of tokens that the model must process. VALL-E, for example, applies autoregressive prediction to the first codebook and non-autoregressive refinement to seven additional codebooks [83]; the serialization of multiple codebooks is discussed in Section 3.4. Other designs reduce token rates by using fewer codebooks. WavTokenizer represents speech with a single codebook at 40–75 tokens per second [88], while BigCodec achieves 1.04 kbps with a single codebook by increasing codec capacity [89]. TS3-Codec provides streaming single-codebook coding without convolutional layers [90], SNAC distributes codebooks across multiple temporal resolutions [91], and GLM-4-Voice uses a single-codebook tokenizer operating at 12.5 Hz for streaming spoken dialogue [92]. Native interaction also requires causal processing and low encoding latency. Mimi, the codec used by Moshi, provides one example through causal convolutions, a 12.5 Hz frame rate at 1.1 kbps, and approximately 80 ms of codec latency [21]. Moshi uses this codec within a full-duplex system designed to respond within approximately 200 ms.

Discussion: discrete tokens or continuous features?

The choice between discrete tokens and continuous features remains open. Controlled comparisons report advantages for continuous features on semantic understanding tasks and attribute the gap to quantization loss [93,94]. Dedicated benchmarks now quantify these gaps along complementary axes: Codec-SUPERB for reconstruction quality [95], DASB for the downstream utility of discrete tokens [96], and STAB for intrinsic tokenizer properties [97]. Discrete tokens support autoregressive speech generation, while continuous output requires an appropriate prediction head [67]. Some omni models therefore use continuous features for understanding and discrete codec tokens for generation [22]. Others tokenize both input and output to model the two speech streams jointly [21]. Continuous input features can preserve semantic and paralinguistic detail, while discrete output tokens support streaming generation. Fully tokenized designs must balance understanding quality with joint speech modeling. Section 4 extends this separation to foreground and background models. How these per-modality choices are arranged across model input and output streams is the subject of Section 3.4.

3.2. Video Representations

Streaming video models process frames as they arrive and may need to respond before the stream ends. This changes the focus from understanding a completed video [98,99] to determining when visual information becomes available and when it should trigger a response [100]. We review frame tokenization, sampling, temporal position encoding, bounded visual memory, and response triggering. Section 3.4 discusses synchronization with audio and text.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Patch tokens and connectors.

A standard video front-end divides each sampled frame into patches, encodes them with a vision transformer, and compresses the resulting tokens before passing them to the language model [101]. A single frame can produce hundreds of patch tokens, so video models use connectors to reduce the number processed by the language model. Cross-attention resamplers map variable-length spatial and temporal features to a fixed number of latent tokens [102–104]. Query-based bottlenecks such as the Q-Former provide another form of token compression [105]. More aggressive per-frame pooling can represent an entire frame with as few as two tokens, making hour-long video inputs more practical [106]. Multiplying the number of tokens per frame by the number of frames processed per second gives the visual token rate that the backbone must process. This is the video counterpart of the frame-rate and codebook-depth calculation discussed for audio in Section 3.1.

Frame sampling.

Uniform sampling is computationally simple but can miss brief events. A common baseline samples approximately one or two frames per second, so events occurring between sampled frames may not be represented. Adaptive keyframe selection prioritizes frames according to query relevance and temporal coverage under a fixed token budget [107]. Streaming systems must make sampling decisions online as frames arrive, which favors lightweight policies that do not depend on future frames. In systems that do not retain raw frames, discarded observations cannot be recovered [100].

Temporal position encoding.

Temporal position encoding represents when each observation occurred. Multimodal rotary position embeddings decompose position indices into temporal, height, and width components [49]. VideoRoPE examines which frequency bands should represent time and how temporal indices should be spaced between consecutive frames [108]. TMRoPE aligns temporal positions across audio and video by assigning the same temporal index to video patches and audio frames that share a timestamp [22]. This allows observations from different modalities to be aligned by absolute time. Section 3.3 examines explicit representations of time, silence, and overlap.

Streaming ingestion, memory, and token compression.

Streaming video is processed incrementally, so stored context grows unless it is bounded by memory or compression. Attention sinks with contiguous position indices support long streams when training follows the same streaming setting [109], while memory modules condense earlier observations into queryable spatiotemporal states [110]. Compression exploits redundancy across frames. Removing unchanged tokens can eliminate more than 80% of visual tokens with little reported accuracy loss [111], while training-free token merging combines redundant visual tokens [112]. These methods support long-running streams while limiting the stored context. Memory and compression determine which past observations remain available and how much memory and computation are required.

When to respond.

Streaming video also requires a decision about when to respond. Methods include per-frame prediction, dedicated response heads, separate perception and response modules, and activation thresholds. VideoLLM-online predicts either silence or the start of a response at each frame [113]. MM-Duet uses a dedicated response head that allows text to be inserted at different points during video playback [114]. Dispider separates perception, decision, and response generation [115], while StreamBridge and LiveStar apply activation or uncertainty thresholds to determine when a response should begin [116,117]. Early triggering can produce irrelevant responses, while conservative triggering can delay useful information. Section 8 examines this distinction through response-onset latency and time-to-informative-answer.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Discussion: coverage versus timeliness.

Video sampling and compression balance information retention against processing cost and response delay. Discarding tokens can remove evidence needed later. Reliable response timing depends on knowing when relevant events occur, making temporal grounding central to proactive interaction. Streaming benchmarks evaluate response timing alongside answer accuracy on timestamp-anchored queries [118,119], and current systems remain behind human performance on these evaluations [120]. Sections 5 and 6 further examine temporal grounding and response timing. The mechanisms reviewed here determine which visual events are encoded and when their representations become available to the model. Section 3.3 next examines how time, silence, and overlap are represented as model inputs.

3.3. Time, Silence, and Overlap as Model Inputs

Cascaded systems commonly use external speech segmentation and endpoint detection before passing a speaker-attributed transcript to the language model. Native models can instead represent pauses, overlapping speech, and timing within their input streams, as illustrated in Figure 4. Turn-taking has a long literature in conversational systems [33,121], and full-duplex spoken language models have recently been surveyed [36]. This subsection examines the information required to learn response timing, while Section 6 discusses turn-taking behavior. In what follows, we first ground the discussion in human conversation science. We then trace the shift from reactive endpointing to predictive turn-taking, examine silence and overlap in turn, and review the mechanisms that make models time-aware; we conclude by discussing how temporal behavior is evaluated.

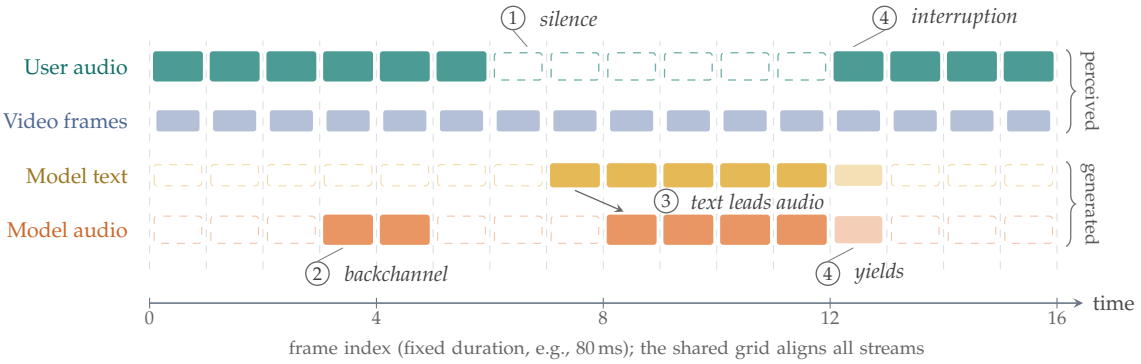


Figure 4. Time-aligned interaction streams. A shared time grid aligns input and output streams. The numbered callouts mark silence, a backchannel, text preceding audio, and interruption-driven yielding. A user interruption stops the model's ongoing speech, and faded blocks indicate generated output that is not played. Section 6 discusses interruption handling.

What human conversation implies for machines.

Human turn-taking is systematic: turns are constructed from recognizable units, and speaker change becomes relevant at projectable completion points [122]. Across ten languages, response offsets typically fall within a narrow window of 0 to 200 ms, with both long gaps and substantial overlap generally avoided [123]. Because utterance planning takes considerably longer than 200 ms, human listeners must anticipate turn completion from prosodic and linguistic cues and prepare responses before the speaker finishes [124]. Interactive models similarly benefit from retaining these cues and preparing responses before an endpoint is detected.

From reactive endpointing to predictive turn-taking.

Cascaded systems commonly reduce turn-taking to endpointing by declaring a turn complete after a fixed silence interval. However, a fixed silence threshold cannot reliably distinguish hesitation from turn completion across all conversational contexts. Continuous prediction provides a more flexible alternative. Recurrent models estimate future speech activity frame by frame [125], multiscale models combine acoustic and lexical cues at their respective rates [126], language models predict turn shifts

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

from pragmatic completeness [127], and generative models learn distributions of response offsets [128]. Voice Activity Projection (VAP) predicts future speech activity for both speakers directly from two-channel audio without requiring turn annotations [16]. Unlike VAD, it predicts upcoming activity instead of detecting current activity and models both speakers jointly. Real-time and multilingual variants extend this approach to additional deployment and language settings [17,129].

Silence as an interaction state.

Acoustically similar silences can serve different interaction functions. A pause within a turn may hold the floor, while a gap between turns may transfer it. The floor-transfer offset measures the interval from the end of one speaker's speech to the start of the other's, with overlap represented by negative values [33]. Speakers can also use fillers such as "uh" to hold the turn during a pause [130]. Systems can use semantic and user-state information to decide whether a pause permits a response [42], a policy considered further in Section 6. In frame-based full-duplex models, silence can be represented as an output state, allowing the model to learn when not to generate speech [21].

Overlap and backchannels as normal input.

Overlapping speech is a normal part of conversation. Backchannels such as "yeah" and "mm-hmm" signal engagement without claiming the floor. Models can predict their timing and type by combining acoustic and language-model cues [131] or by adapting VAP for continuous backchannel prediction [132]. During model speech, the system must distinguish user acknowledgments, interruptions, and background speech [36,42]. This requires representations that distinguish incoming speech from the model's output. Codec-free designs such as SALMONN-Omni explicitly account for the model's own speech during simultaneous listening [133]. dGSLM demonstrates that some overlap and backchannel behavior can be learned from two-channel conversational audio without explicit turn labels [15].

Mechanisms for time awareness.

Models represent temporal information in several ways. Temporal position encodings associate tokens with timestamps. TMRoPE, for example, assigns a temporal index to each fixed audio frame and aligns audio and video using absolute timestamps [22], as discussed in Section 3.2. Fixed-duration chunks provide another way to align input and output over time. SyncLLM inserts synchronization tokens at intervals of approximately 160 to 240 ms and predicts upcoming user chunks to compensate for processing delay [48]. Parallel layouts align user and model tokens at corresponding time steps. Moshi represents both speakers as parallel token streams [21], LSLM integrates a streaming listening channel with its speaking channel [134], and NTPP jointly predicts token pairs across the two dialogue channels [135]. Models can also be trained directly for incremental decoding, as demonstrated by Speech ReaLLM [136]. Synchronization may be learned within the model or implemented by an external controller that determines whether the system should continue listening, respond, or stop [36, 43]. Temporal representations support the reasoning about event timing examined in Section 5 and the response-timing decisions discussed in Section 6. Section 3.4 examines how these streams are serialized.

Discussion: measuring temporal behavior.

Conventional turn-based benchmarks often emphasize response content without evaluating continuous input processing or response timing. Dedicated evaluations provide broader coverage. Full-Duplex-Bench measures pause handling, backchanneling, turn-taking latency, and interruption behavior [137]; later versions add overlap handling [138] and multi-turn interaction with an automated examiner [139]. Talking Turns compares the turn-taking behavior of audio foundation models with conversational corpus statistics [140], while spoken-dialogue surveys summarize measures such as floor-transfer offsets, interruption success, and response delay [34]. Silence is usually studied within turn-taking and full-duplex systems. Evaluations should distinguish response onset from the delivery

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

of useful content, since a fast initial response does not necessarily provide a timely answer. Learning temporal behavior requires representations that retain both speakers' timing, including pauses and overlap. Stream formats that discard either timeline lose this information. Representing both streams at each time step increases token count, including during silence, and requires training data beyond conventional text turns. Section 3.4 examines the resulting trade-offs.

3.4. Unified Stream Formats

The preceding sections reviewed modality-specific streams and their timing. We now examine how these streams are arranged as model inputs and outputs. We group related formats into turn-level concatenation [34,141], time-aligned interleaving [35], and parallel streams [36], and compare their serialization mechanisms and trade-offs in Table 1.

Table 1. Comparison of stream formats by first-output latency, information retention, training requirements, and representative systems. Teal and orange denote user and model tokens, respectively.

Format	First-output latency	Information retention	Training requirements	Representative systems
Turn-level concatenation	high waits for a full segment or text answer	limited speaker-timing detail; explicit text supervision	low text-LM objective on abundant turn data	SpeechGPT [142], Spectron [143], VoxtLM [144], USDM [145]
Time-aligned interleaving 	low–medium chunk granularity plus look-ahead	temporal order kept at alignment granularity; text guidance retained	medium needs forced-aligned or synthetic interleaved data	SUTLM [146], SpiRit-LM [147], GLM-4-Voice [92], IST-LM [148]
Parallel user and model streams 	lowest frame-level, full-duplex native	both timelines, incl. overlap and backchannels; lexical fidelity needs an added text stream	high scarce synchronized two-channel dialogue; architectures adapted beyond text LMs	dGSLM [15], Moshi [21], NTPP [135], PSLM [149]

Turn-level concatenation.

Turn-level concatenation appends complete text or speech segments. SpeechGPT sequences recognized text, a textual response, and speech units [142], while Spectron transcribes speech, generates a textual continuation, and then synthesizes it as audio [143]. VoxtLM combines task tokens with text and speech units in a single decoder [144], while USDM continues pretraining on speech-text data before turn-based dialogue fine-tuning [145]. Complete segments allow these systems to reuse standard text-language-model objectives and turn-based datasets. However, waiting for a segment or textual answer to finish delays speech output and can remove detailed information about speaker timing.

Time-aligned interleaving.

Time-aligned interleaving orders tokens from different streams within a sequence using timestamps or alignment boundaries. SUTLM compares speech-only, text-only, concatenated, and alternating layouts, and reports the strongest cross-modal transfer with word-level alternation [146]. SpiRit-LM interleaves text and speech units at word boundaries [147], while GLM-4-Voice uses chunk-level interleaving for streaming synthesis and trains on large-scale synthetic interleaved data [92,150]. A fixed text-to-speech chunk ratio controls look-ahead and latency, as demonstrated by IST-LM [148]. Interleaved models can be initialized from pretrained text models [151]; recent work also studies their scaling behavior [152] and dynamic turn-level interleaving [153]. These models require aligned training data and an explicit representation of both speakers to support overlap.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Parallel user and model streams.

Parallel layouts represent both participants at aligned time steps, thereby preserving the timing relationship between streams described in Section 3.3. dGSLM uses a dual-tower architecture with cross-attention between two conversation channels [15], while Moshi represents the user's speech, the model's speech, and a time-aligned inner-monologue text stream as parallel sequences [21]. SyncLLM combines interleaving with synchronization of fixed-duration chunks from both speakers [48]. NTPP jointly predicts token pairs across two dialogue channels [135], PSLM uses parallel text and speech sequences to reduce latency [149], Mini-Omni decodes text and audio in parallel, with text guiding audio generation [37], and LSLM integrates a streaming listening channel with its speaking channel [134]. Parallel representations also precede LLM-based dialogue, as pGSLM models speech units, pitch, and duration in separate channels [154]. These categories can overlap. OmniFlatten, for example, progressively converts parallel dialogue streams into a single GPT sequence [155]. Parallel-stream models require synchronized two-channel data and architectures adapted beyond conventional text language models [36].

Serialization mechanics.

Three serialization mechanisms recur across these formats. Multi-codebook outputs require a defined ordering. MusicGen's delay pattern offsets codebook sequences so that they can be generated in a single pass [156], while stack-and-delay changes their ordering to balance latency and reconstruction quality [157]. Moshi instead combines a temporal Transformer with a smaller depth Transformer that predicts tokens across codebooks [21]. Text and audio sequences generally differ in length, requiring padding or delay policies. These include fixed-length padding in LLaMA-Omni [50], trailing padding in Mini-Omni [37], and a learned delay between Moshi's text and audio streams [21]. Chunk ratio and look-ahead control how much text is available during speech generation [148]. Temporal position encodings can additionally align modalities through absolute time, as illustrated by TMRoPE in Section 3.2 [22].

Discussion: serialization trade-offs.

The formats impose different buffering requirements, including complete segments, aligned chunks with look-ahead, and synchronized frames. End-to-end latency also depends on implementation and compute. For example, Moshi operates with 80 ms codec frames, while SyncLLM uses synchronized chunks of approximately 160 to 240 ms [21,48]. Temporal detail depends on alignment granularity, and linguistic consistency can depend on text supervision. Textless parallel models can capture overlap and laughter but may lose semantic information [15]. Text guidance, as in Moshi's inner-monologue stream, can improve linguistic quality within a parallel-stream design [21]. Concatenated formats use conventional turn-based data, interleaved formats require aligned sequences, and parallel formats require synchronized conversational streams [36]. Serialization therefore affects response timing, information retention, and compatibility with pretrained text models. The balance among these factors depends on the model architecture, training data, and implementation.

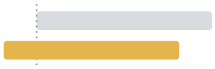
3.5. Speech Synchronization and User Interruption

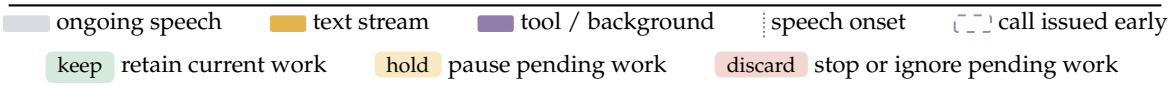
Ongoing interaction can involve partial transcription, internal text generation, tool calls, and retrieval. These topics have dedicated surveys [40,41,158–161]. Here, we examine how these streams are synchronized with speech and handled when the user interrupts. Table 2 summarizes the stream types and their synchronization requirements.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Table 2. Text, tool, and reasoning streams: function, temporal relationship to speech, and handling of user interruptions. Text streams (amber) are speech-aligned; tool and background streams (violet) can run asynchronously, a foreground–background split that Section 4 develops. Legend below the table.

Stream	Function	Temporal relationship to speech	On user interruption
Partial transcript	incremental text of the user's speech	 follows the user's speech with a short lag	keep — continues with the new user speech
Inner monologue	text generated ahead of the model's speech, or unspoken reasoning	 runs slightly ahead of the model's speech	discard — unspoken remainder discarded; context rolled back to what the user heard
Tool call	request to an external tool	 may be issued before the user finishes speaking	discard — cancel if pending; deduplicate if already issued
Tool result	result returned from a tool	 returns asynchronously after call	hold — keep for later use; discard if the request is outdated
Background retrieval	information retrieved asynchronously	 runs alongside speech; results return later	hold — revise the retrieval query; ignore results of the old query



Partial transcript streams.

Incremental transcripts can change as new audio arrives. Transducer models support streaming recognition through monotonic alignment [47,162]. Stabilizing partial recognition results balances transcription accuracy against the delay before results are treated as final [163]. In addition to producing transcripts through a separate recognizer, native models can align text directly with an audio stream. Delayed Streams Modeling pairs input audio with a time-aligned delayed transcript [164]. Moshi uses a related alignment mechanism for a different purpose, pairing inner-monologue text with the model's own speech [21]. Systems must decide when a transcript prefix is reliable enough to use for retrieval or tool selection.

Inner monologue: reasoning while listening and speaking.

Text generated just ahead of audio can improve linguistic consistency, as demonstrated by Moshi [21]. Some models also generate separate, unspoken reasoning sequences while input or audio playback continues. SHANKS generates chain-of-thought while the user is speaking and reports that more than half of its required tool calls can be issued before the user finishes [165]. Think-While-Listening evaluates text-space reasoning for spoken responses and estimates when a question is sufficiently complete to begin reasoning [166]. STITCH alternates unspoken reasoning chunks with spoken chunks, allowing reasoning to continue during audio playback [57]. Other methods reason in hidden states during the user's turn [167] or use pauses in the model's speech for additional reasoning [168]. These methods perform reasoning in text or hidden states while receiving input or playing speech, allowing computation to overlap with ongoing interaction.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Tool-call and tool-result streams.

Tool use adds outgoing call streams and incoming result streams. Text-based agents commonly interleave reasoning, actions, and returned observations [169,170], while Toolformer and ToolLLM train models to decide when and which tools to invoke [171,172]. Tool-learning surveys further organize this process into planning, tool selection, invocation, and result integration [159]. Spoken interfaces introduce additional errors. Speech recognition can produce incorrect tool arguments, and evaluations compare the resulting tool-use performance with text-based inputs. VoiceAgentBench reports that cascaded ASR-plus-LLM systems outperform end-to-end speech models on voice-based agentic tasks [173], while conversions of text tool benchmarks into spoken form reveal a consistent modality gap [174]. Full-duplex benchmarks also evaluate multi-step tool use under realistic speech disfluency [175].

Background retrieval and reasoning.

Retrieval augmentation grounds responses in external knowledge [56]. Retrieval timing determines when external information becomes available. FLARE triggers retrieval when a look-ahead draft is uncertain [176], while Self-RAG uses reflection tokens to decide when to retrieve or critique generated content [177]. MoshiRAG launches retrieval asynchronously and supplies the returned text to the foreground model while speech interaction continues [26]. This can overlap retrieval with early response generation. Section 8 discusses the resulting distinction between response-onset latency and time-to-informative-answer. Stream RAG predicts a retrieval query while the user is still speaking [27]. Speculative interaction agents issue tool calls early and process input and output asynchronously [178], while streaming retrieval methods estimate when a speculative query is sufficiently stable to execute [179]. Other asynchronous designs allow a voice foreground to continue speaking while a background model reasons [180]. These systems coordinate a fast foreground interaction process with slower background retrieval, tool use, or reasoning. The design can reduce blocking delays, but requires a mechanism for validating and incorporating results during ongoing interaction. Section 4 examines this foreground-background pattern [25–27,60].

Interruption: cancel, delay, or roll back.

Interruptions require explicit handling of generated but unplayed speech, running tool calls, and pending retrieval. Systems may cancel pending work, retain useful results, or undo reversible effects. ChipChat tracks which generated tokens have been played, allowing the context to be rolled back to the output that the user actually heard [181]. Interruptible systems may stop spoken output, steer or abandon ongoing reasoning, and update their behavior after new user input [165,180], while full-duplex benchmarks evaluate tool use under interruption [175]. Open problems include preventing duplicate calls and managing state after interrupted execution. Section 6 discusses serving-side operations such as stopping playback and rolling back caches, while Section 9 considers cancellation, deduplication, and reversible side effects as broader research problems.

Discussion: speech-aligned and asynchronous streams.

Table 2 summarizes these streams by function, timing relative to speech, and interruption handling. Text streams can be aligned with speech, while tools and retrieval complete asynchronously. Their requests and results must be tied to the interaction state and inserted at appropriate points. Partial transcripts and inner monologues follow the speech-aligned formats discussed in Section 3.4. Dual-model architectures discussed in Section 4 separate immediate interaction from slower computation, introducing coordination and cancellation requirements. Failures include premature commitment, delayed reasoning, outdated intent, and inconsistent state after interruption. Representation choices determine which activities can be interrupted and resumed consistently. Section 9 examines the remaining cancellation and state-management problems.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

4. Building Native Omni Models for Real-Time Interaction

Native real-time omni models operate while interaction is still unfolding. User speech may continue after response generation begins, visual evidence may become relevant midway through a response, and results from reasoning or tool use may arrive after the conversational state has changed. The central design problem is to update the interaction state with new multimodal evidence while controlling ongoing output, interruption, and revision. This section examines the problem through architecture, data, and training. We first compare architecture families and fusion mechanisms, with attention to where the live state is maintained and how low-latency interaction is coordinated with slower computation. We then discuss the data and objectives required to learn temporally grounded behavior, including silence, overlap, interruption, and foreground–background coordination.

4.1. Architecture Families and Modality Fusion

We compare architectures by which components observe incoming events, maintain the interaction state, and control response timing. Four families organize the current design space: cascaded baselines, speech-native full-duplex models, omni input-output models, and dual-model interaction systems. Figure 5 shows how these four families implement fusion, speech generation, temporal control, and background computation.

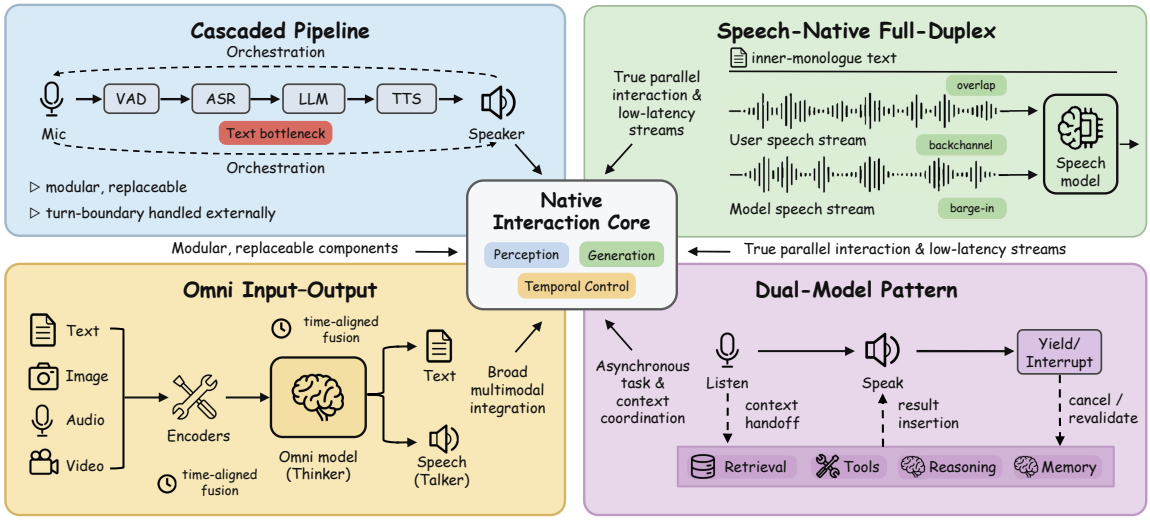


Figure 5. Architecture families for native real-time interaction. The figure maps four architectural patterns by where they place stream fusion, speech generation, temporal control, and background computation.

Cascaded baselines.

The most widely deployed real-time voice architecture chains independently trained modules: voice activity detection (VAD) or endpointing, streaming ASR, an LLM, incremental TTS, and playback control. Online recognition mechanisms such as RNN-Transducer support streaming input [47], while ASR front-ends such as Whisper provide strong transcription quality for speech interfaces [18]. Incremental TTS can begin synthesis from partial text and reduce audible response delay [182,183]. The strength of this design is modularity: each component has a defined interface, can be improved independently, and can be monitored or replaced without retraining the whole interaction system.

These interfaces also determine where the live interaction state can reside. At the ASR-to-LLM boundary, converting audio into text hypotheses and metadata can remove prosody, hesitation, laughter, overlap, and backchannel timing before they reach the language model. At the LLM-to-TTS boundary, generated text becomes an audio queue whose timing is further shaped by synthesis and playback buffering. Interruption is usually handled by runtime logic that stops playback, discards queued speech, and cancels or restarts generation [36,184]. Cascades can therefore be responsive when their components are carefully scheduled, but turn-taking and overlap remain forms of engineered synchronization outside the model [36]. Work such as Voice Activity Projection shows that conversa-

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

tional timing can instead be learned through predictions of future voice activity [16], while full-duplex benchmarks directly evaluate pause handling, backchanneling, turn-taking, and interruption management [137]. A useful middle ground consolidates the front end itself: UAF combines several front-end functions within a single model, including VAD, turn detection, speaker recognition, ASR, question answering, and interruption-related control signals, predicting them within a single streaming autoregressive model while retaining a separate generation module [185]. Native architectures can learn these functions within shared stream representations and interaction state.

Speech-native full-duplex architectures.

Speech-native full-duplex models represent user speech and model speech as time-aligned streams that can be active during the same interval [15,21,48]. Silence, pauses, overlap, backchannels, and interruptions therefore become temporal conditions visible to the model. Their state can update from incoming audio while speech generation continues, allowing the model to condition turn-taking and interruption responses on both streams. Interaction events can consequently be represented as stream events: a backchannel is aligned with ongoing user speech, an interruption occurs while the model-speech stream is active, and yielding changes the model-speech stream in response to the user.

dGSLM provides an early foundation by modeling two-channel dialogue directly from conversational audio, showing that turn-taking can be learned from paired speech streams without first reducing the exchange to text [15]. Moshi jointly models user speech, model speech, and an inner-monologue text stream over neural codec tokens [21], while SyncLLM uses fixed-duration dialogue chunks [48]. Freeze-Omni preserves a text-capable LLM backbone and adds streaming speech modules with chunk-level state prediction [43]. SALMONN-omni uses continuous speech embeddings without codec tokens [186]. Mini-Omni and Mini-Omni2 provide smaller open-source implementations with speech encoders, speech-token decoders, delayed parallel generation, and lightweight interruption signals [37,53]. BayLing-Duplex shows that full-duplex interaction control need not require a redesigned representation: it is learned within an otherwise standard autoregressive speech–language model using only a small set of control tokens, so that a single model decides when to listen, speak, and stop without a separate turn-taking module while retaining the backbone's response capabilities [187]. These designs differ in stream coupling, in the use of text to guide speech generation, and in the choice of speech representation: discrete units, codec tokens, or continuous embeddings. Their main limitation is modality breadth: most remain centered on audio, while video, screen state, tools, memory, and background reasoning are external or absent.

Omni input-output architectures.

Omni input-output systems must coordinate text, visual, and audio information arriving at different rates, extending the temporal problem from speech-centered interaction to images, video, screen context, and spoken response [22–24]. Audio is processed in chunks, video as frames or visual tokens, and text as instructions or intermediate representations. A visual event may change when the model should speak, while an acoustic cue may determine whether it should continue or pause. A central design challenge is to align these inputs while keeping speech generation responsive to new evidence [22,25].

The Qwen-Omni family illustrates how this architecture has evolved. Qwen2-VL introduced dynamic-resolution visual processing and multimodal RoPE [49]. Qwen2.5-Omni extended this approach with block-wise audio and visual encoders, time-aligned multimodal positional encoding, and a Thinker–Talker architecture for streaming text and speech generation [22]. The Thinker produces text-oriented hidden states that condition the Talker, allowing speech to reflect the current multimodal state before a complete text response has been emitted. Qwen3-Omni replaces fixed two-second audiovisual blocks with temporal IDs anchored to absolute time, directly aligning audio and video representations and supporting streaming inputs of arbitrary duration [23]. Qwen3.5-Omni retains the Thinker–Talker structure while introducing Hybrid MoE backbones for long audio-video sequences and ARIA, which adaptively interleaves text and speech units during streaming generation [24].

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Other systems use different interfaces between multimodal perception, language modeling, and speech generation. LLaMA-Omni and LLaMA-Omni2 connect speech encoders and LLM backbones to streaming speech decoders [50,51], while VITA, AnyGPT, and MiniMind-O explore broader or smaller-scale routes to multimodal input and spoken output [52,54,55]. These systems vary in modality coverage, scale, and how speech decoders receive the model's current representations. Talker modules, hidden-state handoffs, codec-token paths, and streaming speech decoders provide different degrees of access to the live model state, and their interfaces affect how quickly new visual or acoustic information can influence generated speech, with corresponding costs in alignment, latency, and cancellation. Recent work also begins to extend this output path beyond text and speech: Ex-Omni-2D generates coordinated text, personalized speech, and reference-conditioned streaming video, using speech units as a shared acoustic–temporal interface between the dialogue and visual-generation pathways [188]. These choices form part of the broader fusion and coupling problem discussed next.

Dual-model interaction systems.

Dual-model interaction systems separate immediate interaction from slower computation [25,60]. A foreground model processes ongoing input and controls speech, pauses, acknowledgments, and interruptions, keeping the user-facing state coherent. A background model handles longer reasoning, retrieval, browsing, tool use, planning, or memory operations without blocking the live exchange. Thinking Machines Lab [60] provides a direct formulation of this pattern. Its time-aware interaction model processes continuous audio, video, and text as micro-turns, while an asynchronous background model handles longer-horizon reasoning and tool use. The foreground remains active as this work proceeds and decides when returned results can enter the conversation.

DuplexOmni provides a recent research implementation of the same separation. Its interaction layer processes streaming audio and video while generating text and speech in real time, whereas a replaceable reasoning component returns reasoning and tool-use results asynchronously [25]. Other systems use specialized background components. MoshiRAG combines a compact full-duplex speech model with selective asynchronous retrieval, allowing response onset to precede the arrival of external evidence [26]. Stream RAG predicts tool queries during ongoing user speech and later fuses retrieved text into spoken responses [27]. STITCH interleaves unspoken reasoning chunks with spoken chunks so that reasoning can continue during speech playback [57], while EgoMem adds asynchronous retrieval and memory management to real-time omnimodal dialogue [58]. StreamArena introduces StreamMind, a streaming architecture that separates latency-critical interaction from persistent multimodal recall and longer-horizon processing over hour-scale video [189]. These systems do not all use two independently scaled models, but they share the same temporal separation between immediate interaction and slower computation.

The defining challenge is the interface between the live and background processes, as illustrated in Figure 6. The foreground must decide when to delegate, what context to send, and whether returned results should be inserted, revised, delayed, or discarded as the live state changes, and must validate them against the current interaction state before incorporating them. Starting audio quickly does not necessarily reduce the time until the user receives a useful answer, so explicit policies for insertion, validation, cancellation, and recovery are required.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

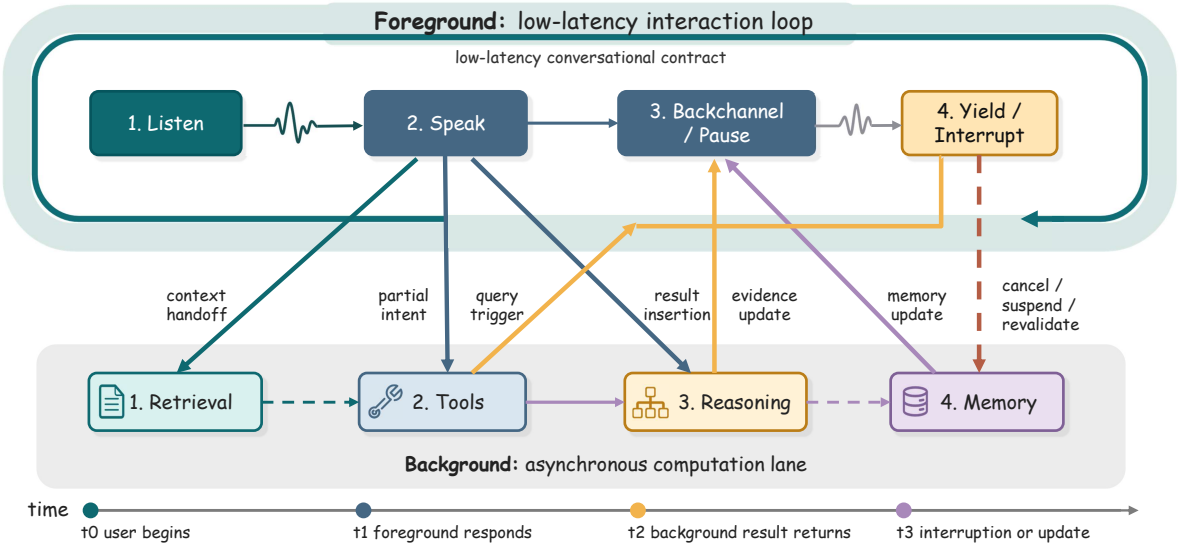


Figure 6. Foreground–background coordination in a dual-model interaction system. A foreground process handles ongoing interaction while retrieval, tools, reasoning, and memory operations run asynchronously. Returned results are incorporated, deferred, revalidated, or discarded according to the current state.

Modality Fusion and Coupling Approaches.

Fusion describes how modality representations are combined, whereas coupling describes how strongly and continuously streams influence one another's content and timing. These design choices can be combined in different architectures. A central question is how much temporal and modality-specific structure remains visible across architectural boundaries.

Encoder-based fusion.

Encoder-based fusion converts speech, audio, images, or video into learned features before projecting them into an LLM or multimodal backbone. Pretrained encoders such as Whisper, CLIP, and SigLIP provide strong semantic representations [18,190,191]. LLaMA-Omni, the Mini-Omni series, and Qwen2.5-Omni follow this pattern [22,37,50,53]. Its main advantage is stable and sample-efficient training. The encoder determines which temporal details are retained and how frequently the model state can update, from once per turn to once per chunk or frame: turn-level features suppress within-turn timing, whereas chunk- or stream-level features preserve more local evidence at greater alignment and memory cost. Separate modality encoders can also process audio and video at mismatched rates. OmniEncoder jointly encodes audio and video at 25 frames per second [192].

Discrete-token and codec-token fusion.

Discrete-token fusion converts continuous signals into token sequences or parallel codebook streams. Neural codecs such as SoundStream and EnCodec provide this interface for speech [77,78]; Moshi uses codec tokens for parallel user and model speech streams [21], while AnyGPT extends discrete-token modeling across several modalities [52]. A common token interface makes autoregressive perception and generation easier to combine, but high token rates and multiple codebooks increase decoding and memory costs as well as first-audio latency. Time-aligned user and model streams can represent silence, overlap, interruption, and yielding within the same representation [15,21]. Codec-token language modeling is not the only route to overlapping conversational audio: Alignment-Free Text-Audiobox instead uses a latent diffusion formulation with flow matching and learns text–speech alignment through cross attention rather than forced alignment, supporting long-form full-duplex dialogue synthesis with turn-taking, backchannels, and emotional dynamics [193].

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Early or encoder-free fusion.

Early fusion uses lightweight projections of lower-level inputs and jointly trains the backbone to learn their perceptual representations, preserving local acoustic structure, visual change, and fine timing. Native multimodal scaling and encoder-free vision-language studies suggest that this approach can remain competitive when supported by sufficient data and joint training [194,195]. Thinking Machines Lab's interaction model provides a real-time example by feeding dMel audio features and patch-level image features through lightweight embeddings into a jointly trained transformer [60], allowing incoming observations to update the model state more directly, while requiring more extensive joint training.

Intermediate fusion and expert routing.

Intermediate fusion can operate on encoder features or discrete tokens, retaining separate computation for each modality while allowing information exchange through shared attention, cross attention, adapters, or shared experts. JavisDiT illustrates this pattern in joint audiovisual generation: audio and video tokens interact through shared self-attention before passing through separate FFNs for each modality [196]. For speech and text, MoST combines dedicated experts with shared experts to support information transfer across modalities [197]. DeepOmni assigns experts by their routing statistics: for each MoE layer it measures the fraction of speech tokens and of text tokens routed to each expert, designates as speech experts those with high speech load and low text load, and keeps the remainder as text experts, alongside a small set of shared experts that both modalities use [198]. Cross-modal exchange occurs through these shared experts and through the attention layers, which all tokens pass through regardless of modality. Training proceeds in stages: the router is frozen while speech and text experts are trained separately on their own modality, then unfrozen for joint training on cross-modal data, which limits the loss of language capability during speech adaptation. These mechanisms preserve modality specialization and support interaction within the backbone, but introduce choices about which layers exchange information, routing balance, and temporal synchronization; related routing mechanisms have also been applied to social evidence in omni models [199].

Coupling strength and speech-output paths.

Loose coupling transfers completed segments, such as ASR text, visual summaries, or tool results. Block-level coupling updates the state as chunks arrive. Shared-temporal-grid coupling aligns multiple active streams that arrive at different rates, such as audio chunks every few tens of milliseconds and video frames at a lower rate, on one time grid, preserving overlap and within-turn events. Qwen3-Omni aligns audio and video through temporal IDs anchored to absolute time [23], while Thinking Machines uses interleaved 200-ms input-output micro-turns [60]. ROMA instead groups frequent audio chunks and less frequent video observations into synchronized multimodal units and adds a speak head that predicts response timing separately from linguistic content [200]. These mechanisms increase temporal control but also increase synchronization, memory, and scheduling costs.

Speech decoders can condition on generated text, hidden states, or codec tokens. Text-to-TTS paths provide modularity but pass through a completed or partial text boundary, whereas hidden-state handoffs, Talker modules, codec-token paths, and streaming speech decoders connect speech more directly to the active model state. Qwen3.5-Omni's ARIA further adapts the interleaving rate between text and speech units during streaming generation [24]. More frequent updates can let new evidence influence speech sooner, while requiring explicit handling of already generated or queued audio after interruption.

Dual-model coupling.

Dual-model coupling defines how the foreground sends context to background processes and incorporates returned results. Context leaves the foreground through a request or context package, while reasoning, retrieval, tool, or memory results return through prompt updates, tool-result streams,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

or other conditioning inputs. Because the live conversation may change before a result returns, the interface must support insertion, validation, cancellation, and revalidation [25,60].

These choices affect temporal detail, modality specialization, training requirements, decoding cost, routing stability, and coordination. Table 3 compares their interfaces, timing, benefits, and risks.

Table 3. Fusion and coupling mechanisms for native real-time omni models.

Mechanism	State interface	Temporal unit	Main benefit	Main risk
Encoder features	Pretrained or unified encoder output	Segment or chunk	Stable, sample-efficient training	Loss of fine temporal detail
Discrete/codec tokens	Token or codebook streams	Frame or token	Unified autoregressive interface	High token and decoding load
Early/encoder-free	Lightweight input projection	Fine-grained signal	Preserves lower-level input detail	Requires extensive joint training
Intermediate/expert-routed	Shared layers or routed experts in the backbone	Layer or token	Specialization with cross-modal exchange	Poor layer placement or routing imbalance
Shared temporal grid	Aligned active streams	Fixed time step	Overlap and event-level control	Misalignment across streams and memory cost
Speech-output coupling	Text, hidden-state, or token handoff	During generation	Speech conditioned on current state	Possible inconsistency and difficult cancellation
Dual-model coupling	Context and result channel	Asynchronous return	Fast onset with slower reasoning	Stale or contradictory results

4.2. Data

Training data determine which interaction behaviors and representations a model can learn. General speech and audiovisual corpora provide abundant supervision for perception and generation [201–204], but native interaction requires temporal supervision such as separate speaker tracks, overlap and silence, event timestamps, response windows, interruption outcomes, and asynchronous execution traces. Figure 7 summarizes this hierarchy, while Table 4 reports representative datasets and training-data pipelines.

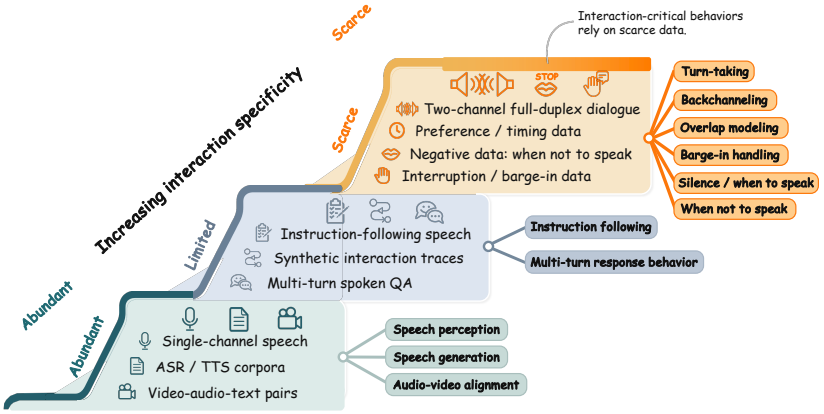


Figure 7. Training data for building native real-time omni models. Temporally annotated data for specialized interaction behaviors is less available than general speech data. These behaviors include turn-taking, acknowledgment, interruption handling, silence, proactive responses, and foreground–background coordination, all of which require temporally aligned interaction traces.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Table 4. Representative datasets and training mixtures for pretraining, instruction tuning, streaming interaction, synthetic dialogue, and preference optimization in omni models.

Dataset	Type	Modality	Language	Size	Date
Fisher [205]	Two-channel telephone speech	Audio, Text	English	1.96k h	2004
Switchboard [206]	Two-channel telephone speech	Audio, Text	English	0.26k h	1992
CallHome [207]	Multilingual two-channel calls	Audio, Text	6 languages	100–120 calls/lang.	1996/1997
DialoSpeech [208]	Dual-track spoken dialogue	Audio, Text	EN, ZH	10k h	2025
Full-Duplex Datasets [209]	Human-recorded dual-track dialogue	Audio, Text	EN, ZH	15 h	2025
HumDial [210]	Human-recorded dual-channel dialogue	Audio, Text	EN, ZH	>100 h	2026
DuplexChat [211]	Podcast-derived dual-track dialogue	Audio	EN, JA	282.6k/132.7k h	2026
LibriSpeech [201]	Read audiobook speech	Audio, Text	English	1.0k h	2015
Libri-Light [212]	Unlabeled audiobook speech	Audio	English	60k h	2020
LibriHeavy [213]	Audiobook speech with punctuation	Audio, Text	English	50k h	2024
LibriTTS [202]	Multi-speaker audiobook speech	Audio, Text	English	0.585k h	2019
LibriTTS-R [214]	Quality-restored audiobook speech	Audio, Text	English	0.585k h	2023
GigaSpeech [215]	Audiobook/podcast/YouTube speech	Audio, Text	English	10k h	2021
GigaSpeech 2 [216]	Multilingual low-resource speech	Audio, Text	TH, ID, VI	30k h	2025
The People's Speech [217]	Web speech	Audio, Text	English	30k h	2021
MLS [218]	Multilingual audiobooks	Audio, Text	8 languages	50k h	2020
Common Voice [219]	Crowdsourced read speech	Audio, Text	Multilingual	Variable	2020
VoxPopuli [220]	Parliamentary speech recordings	Audio, Text	23 languages	400k h unlab.; 1.8k h trans.	2021
YODAS [221]	YouTube speech with transcripts	Audio, Text	100+ languages	500k h	2024
Emilia [222]	In-the-wild speech recordings	Audio, Text	6 languages	101k h	2024
WenetSpeech [223]	YouTube/podcast speech	Audio, Text	Mandarin	10k h	2022
WenetSpeech4TTS [224]	WenetSpeech-derived TTS corpus	Audio, Text	Mandarin	12.8k h	2024
Parler-TTS [225]	Expressive TTS speech	Audio, Text	English	45k h	2024
Seed-TTS subset [226]	Five-speaker adaptation speech	Audio, Text	–	20 h	2024
Fish-Speech data† [227]	Multilingual TTS speech	Audio, Text	8 languages	720k h	2024
Moshi pretrain† [21]	Internal audio pretraining mixture	Audio	Mostly English	~7M h	2024
VoiceAssistant-400K [37]	TTS-rendered GPT-4o dialogues	Audio, Text	English	400k	2024
InstructS2S-200K [50]	TTS-synthesized instructions and responses	Audio, Text	English	200k	2025
SpeechInstruct [142]	Cross-modal instruction data	Audio, Text	English	38k	2023
AnyInstruct-108k [52]	GPT-4-generated multimodal dialogues	Audio, Text, Image, Music	English	108k	2024
UltraChat [228]	LLM-generated text dialogues	Text	English	1.5M	2023
Alpaca [229]	GPT-3.5-generated instructions	Text	English	52k	2023
WavCaps [230]	Audio-caption pairs	Audio, Text	English	403k	2024
Ego4D [203]	Egocentric video with audio	Video, Audio, Text	Multilingual	3.7k h	2022
HowTo100M [204]	YouTube instructional videos	Video, Audio, Text	English	136M clips	2019
AVSpeech [231]	Clean audiovisual speech	Video, Audio	Multilingual	~4.7k h	2018
VoxCeleb [232]	Celebrity interview video	Video, Audio	Multilingual	1k+ spk	2017
VoxCeleb2 [233]	Audiovisual speaker recordings	Video, Audio	Multilingual	1M+ uttr	2018
Perception-Test [234]	Multimodal reasoning video	Video, Audio, Text	English annotations	11.6k clips	2023
AudioSet-Strong [235]	Temporally annotated sound events	Audio, Text	N/A	67k clips	2021
Qwen3-Omni pretrain† [23]	Internal omni pretraining mixture	Audio, Video, Image, Text	Multilingual	~2T tok.; 20M h audio	2025
Qwen3.5-Omni pretrain† [24]	Internal omni pretraining mixture	Audio, Video, Image, Text	Multilingual	>100M h audio-video	2026
ROMA streaming data† [200]	Proactive responses, narration, reactive QA	Video, Audio, Text	English	676k samples	2026
Moshi synthetic† [21]	Internal synthetic multistream dialogue	Audio	English	>20k h	2024
SyncLLM synthetic [48]	Bark-rendered LLM dialogues	Audio	English	212k h	2024
OmniFlatten synthetic [155]	CosyVoice-rendered text dialogues	Audio	ZH, EN	~2k h	2025
LLaMA-Omni/2 synthetic [50,51]	TTS-synthesized InstructS2S dialogues	Audio, Text	English	200k dialogues	2024/2025
OpenHermes-2.5 [236]	Instruction and chat data	Text	English	1M samples	2023
CosyVoice 2 data† [237]	Internal multilingual TTS mixture	Audio, Text	ZH, EN, JA, KO	166.8k h	2024
DuplexOmni† [25]	Writer–Director interaction traces	Video, Audio, Text	ZH, EN	3.02M conv.; 620k seeds	2026
DuplexGen† [238]	Preference-calibrated synthetic dialogue	Audio, Text	English	Not reported	2026
Qwen2.5-Omni DPO† [22]	WER-/pause-ranked speech candidates	Audio, Text	–	Not reported	2025
Qwen2-Audio DPO† [239]	Human-ranked audio responses	Audio, Text	–	Not reported	2024
Align-SLM [240]	AI-feedback preference pairs	Audio	English	~263k pairs	2025
SpeechAlign [241]	Reference/synthetic codec preferences	Audio	English	50k pairs; 250k ablation	2024
ASPIRin [242]	Speak/silence GRPO actions	Audio	English	43 h	2026
Multi-Faceted [243]	RL: pause/turn/backchannel/interruption	Audio	English	≤2k seg./axis/corpus	2026
RLAIF-SDS [244]	Switchboard DPO preferences	Audio, Text	English	166k pairs	2026

† Internal training mixtures or constructed pipelines reported in model papers; sizes follow those papers. "Not reported" means that the total scale is undisclosed, and "–" means that the language composition is unspecified, and "N/A" means that language is not applicable to the dataset.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Speaker-separated dialogue data.

Speaker-separated dialogue provides the most direct supervision for turn-taking, overlap, interruption, backchannels, and conversational silence. Fisher is a widely used example, with more than 16,000 conversations and roughly 2,000 hours of two-channel telephone audio [205]. This dataset has been used by dGSLM, Moshi, and SyncLLM to learn multistream conversational timing [15,21,48]. Switchboard and CallHome provide related telephone conversations [206,207], while the Open-source Full-Duplex Dataset, DialoSpeech, and HumDial add human-recorded or large-scale dual-track dialogue, interaction annotations, and interruption and overlap data [208–210]. DuplexChat increases the available data by automatically extracting and restoring two-speaker tracks from public podcasts, reporting approximately 282,634 hours of English and 132,723 hours of Japanese dialogue [211]. Despite this increase in scale, the category retains important limitations. Classic corpora often contain narrowband, two-person telephone conversations, primarily in English, while automatically recovered tracks are not equivalent to directly recorded multichannel interaction. High-fidelity multilingual and multiparty recordings, together with fine-grained annotations of addressee, interruption intent, backchannel function, and floor negotiation, remain scarce.

Foundational speech and spoken-instruction data.

Single-channel speech supports training for ASR, TTS, audio codecs, and speech representation learning. Representative resources include LibriSpeech [201], Libri-Light [212], LibriHeavy [213], LibriTTS [202], GigaSpeech [215], Common Voice [219], VoxPopuli [220], YODAS [221], Emilia [222], and WenetSpeech [223]. At a larger scale, Moshi reports pretraining its audio stream on approximately seven million hours of mostly English speech transcribed with Whisper [18,21]. These corpora support perception and generation but generally do not preserve separate speaker timelines or explicit turn-control annotations. Spoken-instruction datasets connect these capabilities to assistant behavior. VoiceAssistant-400K [37], InstructS2S-200K [50], SpeechInstruct [142], AnyInstruct-108K [52], and LLaMA-Omni2's multi-turn speech data [51] are largely produced by generating or rewriting text instructions and rendering them with TTS. These datasets support training for spoken instruction following and conversational response generation, but their clean request–response structure provides little direct supervision for overlap, interruption, or deciding not to speak.

Temporally aligned audiovisual data.

Omni models require audio and visual streams whose temporal relationship is preserved. Ego4D [203] supplies egocentric video with audio and annotations, HowTo100M [204] provides narrated instructional video at web scale, and AVSpeech [231] and VoxCeleb [232,233] associate visible speakers with speech. These resources support audiovisual grounding but often rely on clip-level captions or weak temporal alignment. Qwen2.5-Omni uses temporally interleaved audio-video training for synchronized multimodal processing [22], while ROMA curates streaming data for reactive and proactive audiovisual interaction [200]. Nevertheless, these model-specific mixtures do not resolve the broader data gap. Native interaction requires event-level supervision of what occurred, when it occurred, and whether it should trigger, delay, or suppress a response, while large-scale natural audiovisual conversations with such annotations remain scarce.

Synthetic interaction traces.

Synthetic data supplies interaction patterns that are difficult to collect naturally at scale. A common pipeline generates or rewrites dialogue with a text model and renders it as speaker-separated speech with controlled silence, overlap, backchannels, speaker identity, and noise. Moshi reports more than 20,000 hours of synthetic multistream speech [21], while SyncLLM progressively introduces pauses, overlapping speech, and natural Fisher conversations during training [48]. OmniFlatten and LLaMA-Omni similarly synthesize multi-turn spoken dialogues through LLM rewriting and TTS [50,51,155].

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

DuplexOmni extends this approach with a Writer–Director pipeline for continuous multimodal interaction data [25]. The scarcity of synchronized conversational data is compounded by the fact that appropriate turn-taking behavior is scenario-dependent. DuplexGen calibrates synthetic human–AI dialogues using a small amount of human turn-taking preference supervision, producing task-specific timing patterns rather than one universal interaction distribution [238]. The resulting overlap and interruption patterns differ between cooperative and competitive settings, showing why training data should include the context in which a timing behavior is appropriate. More generally, synthetic traces provide scale and control over interaction patterns, but may not reproduce natural disfluency, prosodic coordination, interruption intent, or human timing distributions. Systems such as Moshi and SyncLLM therefore continue to mix synthetic traces with genuine speaker-separated recordings when available [21,48].

Interaction-policy and safety data.

Interaction-policy data supervises when the model should speak, wait, acknowledge, yield, or stop. This layer is smaller and less standardized than the preceding ones. Existing resources include ranked speech candidates for quality and pause control [22], preference pairs for audio-conditioned text responses [239], preference pairs for speech generation [240,241,244], and interaction-specific segments or trajectories for optimizing turn-taking, backchanneling, and interruption [242,243].

Silence is also an explicit positive training target. Moshi, for example, continuously predicts both speech and silence [21]. Training examples should include appropriate silence, brief acknowledgment, yielding after interruption, and abandoning an outdated response, as well as cases where a response must be stopped or corrected when unsafe intent becomes apparent. Safety data must additionally cover streaming refusal, bystander speech, and privacy-sensitive audiovisual input. Most interaction-policy and streaming-safety datasets remain system-specific or unreleased.

Dual-model traces.

Dual-model systems require execution traces that go beyond ordinary dialogue pairs. A trace must record when the foreground delegates, what context is sent, when background reasoning or tool results return, what has already been spoken, and whether the result is inserted, revised, delayed, or discarded. MoshiRAG and Stream RAG provide examples of such coordination traces for asynchronous retrieval and tool use [26,27], while STITCH and EgoMem extend them to background reasoning and long-term memory [57,58]. The scarcity of public traces limits the reproducibility of training for interruption, cancellation, stale-result detection, and recovery.

Cross-cutting data challenges.

Across these categories, three limitations remain. First, speech datasets are growing, but high-quality multilingual and multiparty interaction remains scarce relative to single-channel speech. Second, timestamp and speaker-attribution errors directly corrupt supervision for silence, overlap, interruption, and audiovisual response timing. Third, current data is dominated by English and a small number of conversational settings, leaving other languages, accents, and social norms under-represented; data should also better represent bystander speech and how participants coordinate turns.

4.3. Training Objectives

Training objectives specify how models learn multimodal representations, response generation, and interaction decisions. Native real-time omni models must generate coherent text and speech, align evidence across modalities, control when to speak or remain silent, respond to interruption, and coordinate computation that may finish after the conversational state has changed. We group these objectives into four categories: content and cross-modal grounding, temporal control, interaction-policy alignment, and concurrent reasoning and coordination. Safety and balancing between objectives

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

apply across these categories. Figure 8 maps these objective families onto events in a live user–model interaction.

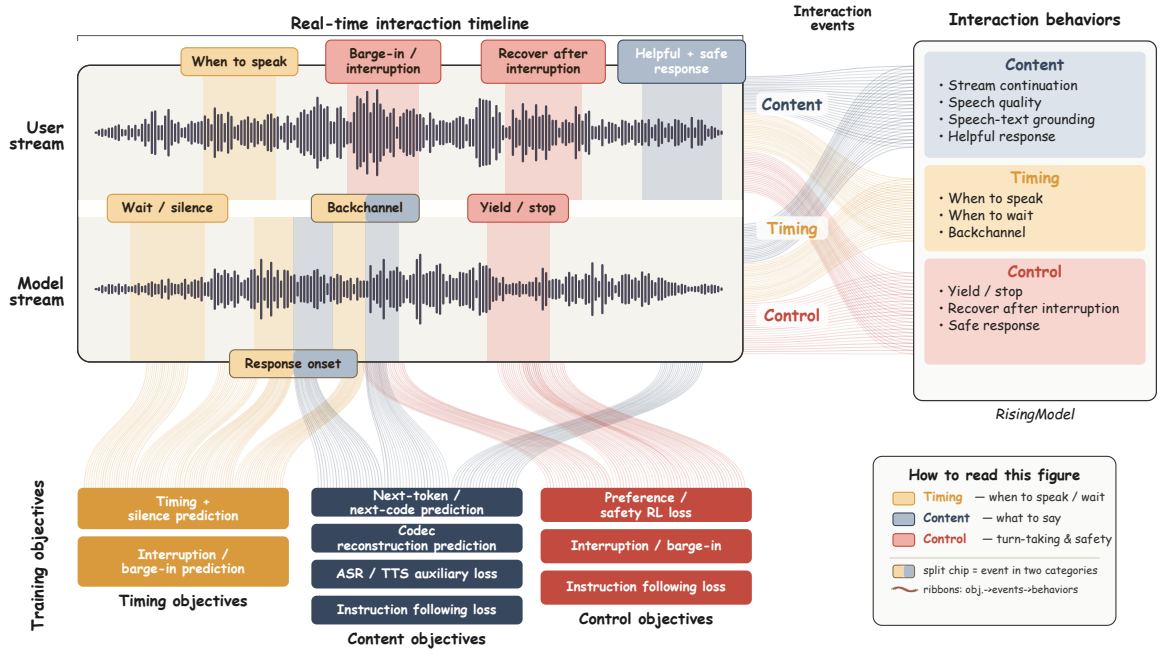


Figure 8. Training objectives for native real-time omni interaction. Objectives for content and cross-modal grounding, temporal control, interaction-policy alignment, and concurrent reasoning and coordination, together with safety objectives, supervise response onset, silence, acknowledgment, interruption handling, yielding, recovery, and safe response.

Content and cross-modal grounding.

Autoregressive likelihood remains the core objective, with targets drawn from text tokens, semantic speech units, codec tokens, acoustic tokens, and interaction-control symbols. Multi-codebook speech requires prediction across both time and codebook depth. Moshi factorizes these dimensions with temporal and depth Transformers [21], while Mini-Omni uses delayed parallel codebook prediction [37,91]. Text representations can condition incremental speech generation, as in text planning streams and Thinker–Talker handoffs [21,22,92].

Reconstruction and speech–text alignment losses support this objective without defining the interaction policy themselves. Codec reconstruction losses train waveform fidelity so that predicted speech units are acoustically usable [77,78], while auxiliary CTC losses align acoustic and linguistic representations in systems such as Freeze-Omni, VITA 1.5, and LLaMA-Omni [43,50,245]. These losses establish perception and generation capability; temporal and policy objectives determine how that capability is used during interaction.

Temporal control.

Temporal objectives supervise when the model should speak, continue, wait, acknowledge, yield, or stop. dGSLM separates content and duration prediction, using cross-entropy for speech units and an L1 loss for their durations [15]. SyncLLM predicts fixed-duration chunks, including silent chunks, while Moshi continuously predicts speech or silence on aligned streams [21,48]. These formulations make pauses, overlap, and response onset part of the prediction target.

Other systems supervise interaction state more explicitly. Freeze-Omni uses chunk-level state classification for continued listening, interruption, and response regeneration, while SALMONN-omni combines speech-boundary tokens with a weighted loss over ambiguous intervals during which the model should delay responding [43,186]. For audiovisual streams, ROMA applies a weighted binary cross-entropy objective to a speak head at each multimodal time step, using event windows or segment

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

boundaries as response-timing labels [200]. These objectives explicitly train response timing and interruption handling, moving them from runtime heuristics into supervised model decisions.

Interaction-policy alignment.

Supervised likelihood encourages imitation of observed interaction sequences, but it does not directly optimize whether a pause, acknowledgment, or interruption response is well timed. Preference optimization first addressed speech quality and response behavior. Qwen2.5-Omni applies DPO to speech candidates ranked by word-error and pause-error measures [22], while Qwen2-Audio, Align-SLM, and SpeechAlign use preference signals for factuality, generation quality, or speech alignment [239–241,246,247].

Recent methods optimize interaction dynamics directly. ASPIRin uses speech–silence actions and timing-oriented GRPO rewards [242], while Multi-Faceted Interactivity Alignment applies reinforcement learning with axis-specific rewards for pause handling, turn-taking, backchanneling, and user interruption, and adds an LLM-based response-quality reward to prevent semantic degradation [243,244]. DuplexPO further factorizes the reward across turn initiation, acknowledgments, yielding, and regularized participation [248]: the turn-initiation term decays with the delay between the reference and predicted response onset, the acknowledgment term rewards agent speech that falls inside the annotated backchannel interval, the yielding term penalizes speech that continues beyond a tolerance after a user interruption, and the participation term penalizes undesirable interaction patterns. Credit is assigned locally by optimizing only the tokens inside short windows around each annotated event, with the preceding history teacher-forced. These methods use separate rewards for interaction behaviors and for response quality, and DuplexPO additionally keeps the policy close to its supervised initialization through a KL term. Without an objective that preserves content quality, timing optimization can reduce response quality.

Concurrent reasoning and coordination.

Training must also support reasoning during ongoing input or output. FLAIR trains a recursively updated latent reasoning state during user speech through an evidence-lower-bound objective and teacher forcing, avoiding the need for explicit reasoning traces [167]. This provides a single-model route to reasoning during perception.

Dual-model systems expose the coordination decisions directly. DuplexOmni uses temporally annotated control events for thinking requests, waiting, interruption, speech cutoff, silence, and delayed result delivery, and trains these events through likelihood objectives over Writer–Director traces [25]. More generally, supervision for dual-model systems covers background requests, the context sent, when returned results are incorporated, and how outdated work is canceled or revalidated. A background result may no longer match the live intent when it returns, so correctness depends on both its content and its validity at the time of insertion. Standardized objectives for stale-result detection and recovery remain limited. Table 5 summarizes the four objective categories by their training signals and the interaction decisions they supervise.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Table 5. Training-objective categories for native real-time omni interaction. Safety and balancing between objectives apply across all four categories.

Objective category	Main training signal	Core decision
Content and cross-modal grounding	Token/code likelihood, reconstruction, and speech–text alignment	What multimodal content should be represented and generated?
Temporal control	Duration, fixed-duration chunks, silence, interaction-state labels, and response windows	When should the model speak, wait, acknowledge, yield, or stop?
Interaction-policy alignment	Ranked candidates and interaction-specific rewards	Which timing and conversational behavior should be preferred?
Concurrent reasoning and coordination	Latent-state objectives and temporally annotated control events	How should reasoning, delegation, result insertion, and cancellation proceed during interaction?

Safety and objective balancing.

Safety training should evaluate partially observed input and partially generated output, because both remain incomplete while interaction unfolds, including when new evidence requires cancellation or correction. More broadly, objectives for content accuracy, acoustic quality, latency, timing, reasoning, and safety may generate competing gradients. Projected action spaces, factorized rewards, and semantic-quality rewards reduce part of this interference [242,243,248], but current systems rarely report objectives that jointly cover continuous multimodal input, partial output, and asynchronous tool results.

5. Maintaining Omni Intelligence

This section examines the capabilities needed to maintain a coherent understanding as new observations, user corrections, and external results arrive during a session. A completed turn gives a model a stable block of context. A live interaction does not. New events continue to arrive, a user may correct an earlier statement, and retrieved evidence may return after the conversation has already moved forward. The model must therefore maintain a coherent representation of events, evidence, and participants as the interaction changes.

We organize this section around five capabilities. Temporal grounding tracks when events occur and how they relate. Memory and session state preserve events, entities, goals, and corrections across the interaction. Reasoning and tool use operate over partial input and continue while speech is being heard or played. Factual grounding ties a claim to the evidence available when that claim is made. Social and pragmatic understanding identifies speakers, intentions, emotional cues, and conversational roles. These capabilities define the intelligence side of native omni interaction. Sections 3 and 4 discuss the supporting representations and architectures; Sections 6–9 address interactive behavior, serving, evaluation, trade-offs, and open problems.

5.1. Temporal Grounding

Event time as an internal model state.

Temporal grounding is the ability to represent when an event occurs and how that event is related to the rest of the interaction [249–252]. The relevant state includes event boundaries, timestamps, duration, ordering, overlap, and relations such as before, during, and after. These quantities are separate from event identity. A model may recognize that a person picked up a cup, that a door closed, or that a user mentioned a meeting while assigning the observation to the wrong interval. In a completed video, this may appear as a localization error [249,251,253]. In a live session, it changes which evidence is valid for the current response.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Earlier video-language models explicitly predict event boundaries and timestamps. VTimeLLM [249] introduces boundary-aware training for video moments, TimeChat [250] introduces a timestamp-aware frame encoder and instruction tuning for time-sensitive long-video understanding, and Grounded-VideoLLM [251] represents temporal locations with dedicated time tokens. In real-time interaction, temporal grounding must update using only observations available so far, rather than relying on a completed clip: at time t, the model can use events observed up to t and cannot assume that the remainder of the stream is known.

Temporal relations across modalities.

An omni model also needs temporal relations across modalities [254]. A spoken reference can overlap a pointing gesture. A sound may begin before its visual source becomes clear. A tool result may arrive several seconds after the request that initiated it. FineMoLA provides a complementary weakly supervised mechanism for learning fine-grained frame–phrase temporal correspondence from clip-level descriptions [255]. The model must retain each observation's time interval and link observations that refer to the same event. This gives temporal grounding three levels of structure: localization within a stream, relations across streams, and whether an event is past, ongoing, or not yet observed.

The third level is the main change from turn-based modeling. Turn-based input preserves sequence order, but the model receives the sequence after the turn is complete. A native model must know whether a relevant event has already happened, is still unfolding, or has not yet become observable. This distinction is needed even before a response policy is applied. It determines whether the current state contains enough information to support a reference, a reasoning step, a tool argument, or a factual claim.

Systems with explicit temporal alignment.

Several recent systems expose this requirement directly. SyncLLM inserts synchronization tokens at fixed temporal intervals so that language-model updates advance with real-world time [48]. MiniCPM-o 4.5 uses Omni-Flow to align continuous multimodal input and output along a shared temporal axis [256]. Aligned timestamps allow audio and visual observations to update a shared interaction state during speech generation. ROMA introduces a lightweight speak head that predicts response initiation from streaming multimodal state [200]. The speak decision belongs to interaction control, but its input must encode whether the event that justifies a response has occurred and whether the relevant evidence is complete enough to act on.

Figure 9 illustrates a time-indexed event representation with event boundaries, cross-modal links, and the current observation time. Temporal grounding identifies when each observation occurred and whether it is relevant to the current response. Section 3.3 discusses how timestamps, silence, and overlap enter the model representation. Section 7.1 defines timing and temporal measurements. Section 8 analyzes premature and delayed commitment. The role of temporal grounding here is narrower: it is the model's internal understanding of when the evidence belongs.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

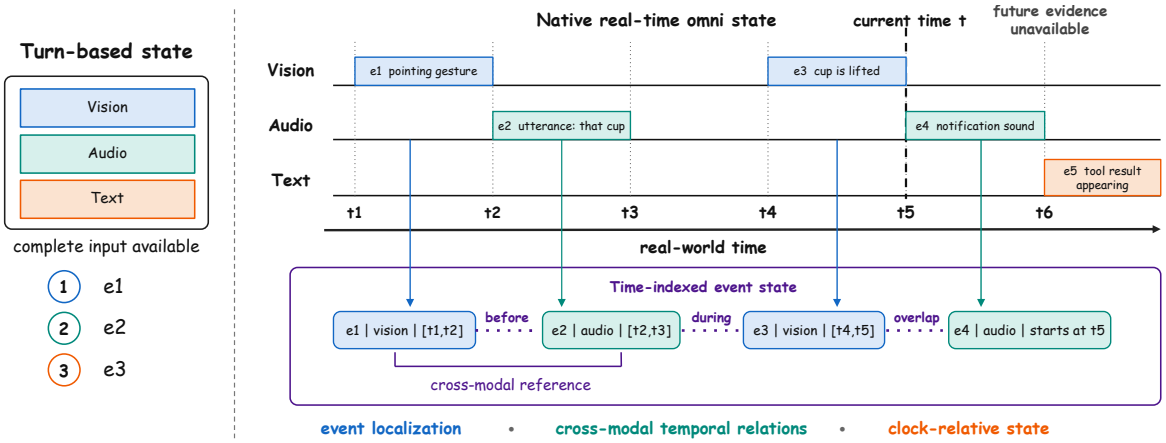


Figure 9. Temporal grounding during ongoing interaction. Observations retain their time intervals and cross-modal links within a time-indexed event state. The current time separates available observations from future input, so event localization and reasoning use only information observed so far.

5.2. Memory and Session State

A transcript alone may not capture recurring entities, changing goals, and corrections in a multimodal session: people and objects recur, events create consequences, user goals change, and later corrections can replace earlier assumptions. The model must track these changes together with the source and time of the supporting observations. This requirement is especially important for continuous audio and video, where an observation may be a short sound, a gesture, a visible object state, or a relation between two modalities.

Three semantic layers.

We describe session state with three semantic layers. The *event state* stores temporally grounded observations. Each event records what occurred, when it occurred, its modality and source, and any confidence or cross-modal links needed by later reasoning [249,251]. The *entity state* connects observations that refer to the same person, object, location, document, or external resource. It preserves identity across time and updates attributes or relations when new evidence arrives [58]. The *versioned task state* records the active goal, constraints, unresolved steps, corrections, tool results, and commitments. A correction creates a new task version and identifies which earlier conclusions depend on the superseded state [257,258].

These layers answer different questions. Event state supports references to what happened at a particular time. Entity state supports continuity across repeated observations. Versioned task state supports revision. The distinction matters because the same new input can have different effects on each layer. A user correction may leave the identity of an object unchanged, add a new event, and replace one task constraint. Treating the entire session as an append-only text history hides this structure.

Working, episodic, and long-term memory.

The semantic layers operate over several stores that differ in retention duration. Working memory keeps the recent multimodal context required for local references and immediate reasoning [259]. Episodic memory preserves events and task transitions across the current session [260]. Long-term memory stores selected user facts, preferences, entities, and recurring goals across sessions [58,261]. The retention duration describes how long an item remains available; the semantic layer describes what the item means. A person's identity may be retained across sessions, while a temporary tool result may be relevant only to the current task.

Each stored item should retain its source, modality, timestamp, and relevant task version. Otherwise, a retrieved item may be semantically relevant and still refer to another user, an earlier time, or an

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

outdated version of the task. Table 6 summarizes the three semantic layers and the update operations they require.

Table 6. Session-state taxonomy for native omni intelligence. The state type specifies what is represented; the retention period specifies how long it is stored.

State type	Stored content	Source and timing information	Main update operations	Retention period
Event state	Time-bounded observations, utterances, actions, sounds, visual changes, and external results	Timestamp or interval, modality, source or speaker, confidence, cross-modal links	Append, merge, split, relabel, revise boundaries	Working; episodic
Entity state	Identities, attributes, and relations of recurring people, objects, locations, documents, and resources	Links to supporting event IDs, identity evidence, last observation time	Re-identify, update attributes, merge or separate identities, revise relations	Working; episodic; long-term
Versioned task state	Current goal, constraints, corrections, unresolved steps, tool results, commitments, and dependencies	Task version, supporting events and entities, tool source, validity status	Create new version, supersede, invalidate dependent state, revalidate, commit	Working; episodic

Queryable state over multimodal streams.

Recent systems implement parts of this memory design. ReKV stores video key-value caches outside GPU memory and retrieves relevant segments when a question arrives [262]. Its contribution is primarily an efficient streaming mechanism, which belongs to the system discussion in Section 6, but it also shows why long visual history must remain queryable by temporal segment. InteractiveOmni extends multi-turn interaction to audio and vision, requiring earlier multimodal observations to remain usable later in the dialogue [263]. EgoMem associates information with users using face and voice cues, maintaining facts, preferences, and social relations across sessions in real-time omnimodal streams [58].

From retrievable history to evolving latent state.

Streaming memory need not remain an external store that is repeatedly retrieved into the context window. FlexMem uses visual key–value caches as memory, applying two compression pathways together with task-dependent retrieval for long and streaming video without additional training [264]. LatentStream instead progressively consolidates historical observations into a compact, fixed-length latent memory for subsequent reasoning [265]. Taken together, these approaches store history as key–value caches, externally retrieved records, or compressed latent representations. Some also retain semantic or personalized information across sessions. StreamArena's StreamMind system illustrates the same separation at the system level, keeping latency-critical interaction distinct from persistent multimodal recall over hour-scale video [189]. Streaming interaction requires memory to remain bounded and revisable while new observations continue to arrive.

Session memory should support retrieval and revision while preserving source information. Sections 6, 8.2, and 9.2 discuss storage, consistency failures, and long-session memory, respectively. The event, entity, and versioned task layers defined here provide the shared vocabulary for those later discussions.

5.3. Reasoning and Tool Use under Real-Time Constraints

Reasoning and tool use share a constraint that does not appear in the same form for a completed prompt. Computation can start while the user is still speaking, continue while the model's audio is being played, and return after the user has changed the task. Intermediate results are therefore tied to the input prefix and task version that produced them. The model needs to reason causally over

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

available evidence and preserve that association when reasoning invokes retrieval or another external tool.

Reasoning while listening.

An internal text stream can support language processing while speech generation continues. Moshi maintains an inner-monologue text stream alongside the speech streams [21]. Qwen3-Omni adopts a Thinker–Talker architecture in which the Thinker processes multimodal inputs and provides semantic states, while the Talker generates streaming speech [23]. Their architectural details belong to Section 4. Semantic computation can retain its own state during speech generation. Cognitive Supersensing illustrates visual latent reasoning in an offline setting [266].

Other methods reason directly during ongoing input or speech. FLAIR updates a latent reasoning state while the user speaks, preserving causal processing [167]. STITCH interleaves internal reasoning with speech playback, using playback time to prepare later reasoning without inserting a full reasoning delay before speech [57]. Can Speech LLMs Think while Listening studies when a speech model should begin reasoning before an utterance is complete, using a question-completeness signal to control the accuracy-latency trade-off [166]. These methods differ in whether reasoning is explicit text or latent state and whether it runs during listening or speaking. Intermediate reasoning must remain consistent with the input observed so far.

Reasoning can also run asynchronously with perception. Video Streaming Thinking processes incoming clips while playback continues, rather than waiting for perception to terminate before deliberation begins [267]. R3-Streaming allocates additional reasoning to difficult queries after memory compression and response-readiness assessment, through a cascaded remember–respond–reason control process [268]. These designs allocate computation adaptively rather than forcing every input through the same reasoning budget, and they connect directly to the multi-rate view developed in Section 9.1.

Reasoning should retain uncertainty when the user's request is incomplete, and revise affected conclusions when constraints change, without treating all earlier observations as invalid. This property connects reasoning directly to the versioned task state in Section 5.2. Such reasoning concerns the model's current understanding; deciding when to speak is a separate interaction-control problem addressed in Section 6.

Tool use before the request is complete.

External tools extend the same causal computation. Stream RAG predicts retrieval queries during ongoing user speech and integrates returned evidence into a later response [27]. Related work on multi-turn agentic scientific literature search uses workflow induction to organize external search across turns [269]. MoshiRAG performs retrieval asynchronously between response onset and the delivery of substantive information [26]. DuplexOmni pairs a low-latency interaction layer with a separate reasoning component that can provide reasoning and tool use in parallel [25]. These systems demonstrate that retrieval and external computation can overlap the spoken interaction.

The intelligence requirement concerns grounding and validity. A tool request should record its arguments, supporting evidence, task version, and launch time. Returned results must be checked against the current task before use. Query prediction and asynchronous execution can reduce idle time, but the returned information is useful only if it still answers the current task. Section 3.5 covers how reasoning, tool, and retrieval streams are represented. Section 8.2 analyzes runtime errors when reasoning or external results become inconsistent with the live state. Section 9.1 treats cancellation, rollback, and multi-rate computation as open design questions.

Figure 10 follows a session in which a correction updates the task version. Events update event and entity state, background results retain their originating version and are checked against the current state before the assistant uses them in a response.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

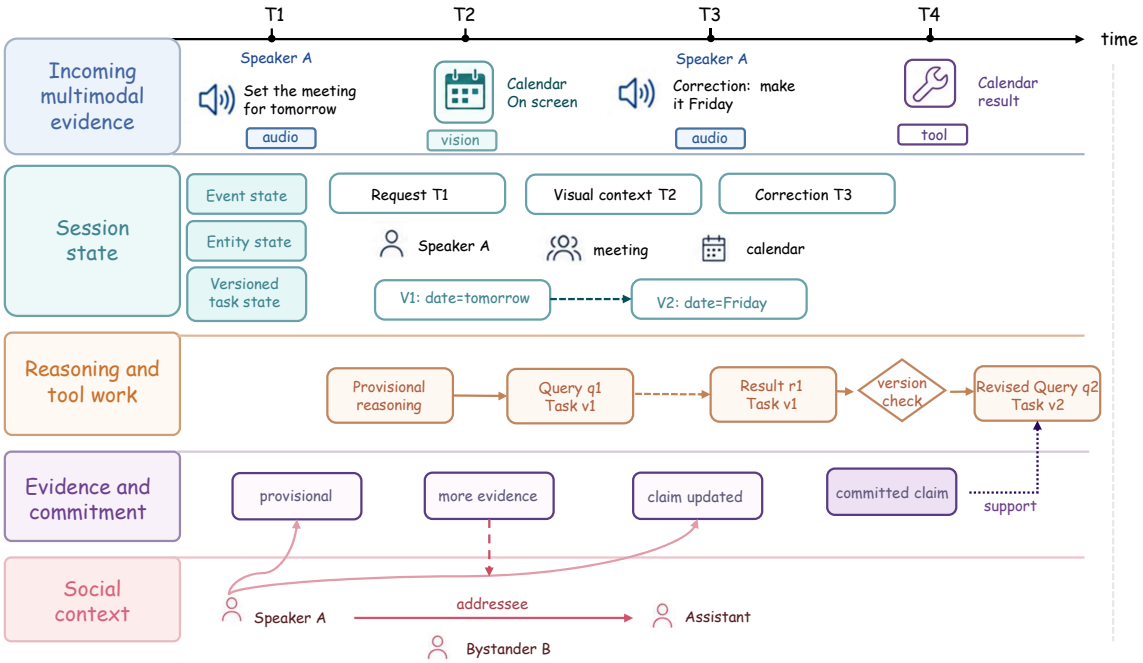


Figure 10. Maintaining a consistent understanding during a session. Multimodal evidence updates event, entity, and versioned task state. Reasoning and tool results proceed concurrently and retain their task version. After a correction, returned evidence is checked before it is incorporated into a response.

5.4. Factual Grounding and Correctness

A model may begin responding before all evidence needed for an answer is available. It must distinguish direct observations, retrieved information, memory, and inference, and identify which claims each source supports at the current time.

Evidence supporting a response.

Before presenting a factual claim, the model should identify its supporting evidence and verify that it remains relevant to the current task [270,271]. Visual evidence can be indexed by object [272], region [273], track [274], or event interval [251]. Acoustic evidence can be indexed by speaker [232], sound source [275], prosodic segment [276], or time range [235]. Retrieved evidence carries the query, source, return time, and task version. Memory-based evidence points back to the event, entity, or earlier task state from which it was recovered [271,277]. This structure allows the model to answer two questions before presenting substantive content: what supports the claim, and whether that support is valid for the current task state.

The strength of evidence can change while the model is already interacting. A partial request may support an acknowledgment before enough information is available for a factual answer. A visible event may resolve a reference after the user's speech has begun, and a retrieval result may arrive after an initial spoken phrase. The system can keep an answer provisional and communicate its uncertainty until supporting evidence arrives.

Perceptual and external grounding.

Streaming retrieval provides a clear example of external grounding. Stream RAG and MoshiRAG allow later response content to depend on evidence retrieved while speech continues [26,27]. Qwen3-Omni provides a complementary perceptual example through its dedicated audio captioning model, which is fine-tuned for detailed audio descriptions with reduced hallucination [23]. The sources differ, but in both cases the model should track the evidence supporting each claim and check its relevance to the current task.

This subsection defines the positive grounding capability. Section 7 measures content quality and the timing of useful evidence. Section 8.1 examines preservation of pretrained language capabilities,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

including the degradation reported by URO-Bench and analyses of the text-speech understanding gap [28,29]; Section 8.2 examines premature claims, loss of grounding, and outdated evidence. Table 7 links these capabilities to evaluation and failure analysis.

Table 7. Capability map for omni intelligence. The final two columns are cross-references to the dedicated measurement and trade-off analyses in later sections.

Capability	Definition	Key mechanisms	Representative systems	Changes required for native interaction	Measurement	Failure or trade-off
Temporal grounding	Event time, boundaries, duration, ordering, overlap, and relation to the current time	Temporal localization; explicit timestamps and temporal position encodings; shared multimodal time axis; event-state prediction	VTimeLLM; TimeChat; SyncLLM; MiniCPM-o 4.5; ROMA	Track events incrementally using only observations available so far	Sec. 7.1 timing and temporal quality	Sec. 8.2 premature or delayed responses and temporal misalignment
Memory and session state	Evolving event, entity, goal, correction, and task-version state	Working, episodic, and long-term memory; temporal and modality indexing; state versioning; cache-based, retrieved, or latent memory	ReKV; InteractiveOmni; EgoMem; FlexMem; LatentStream	Maintain revisable, attributed state over continuous audio-video sessions	Sec. 7.1 long-session quality	Sec. 8.2 evidence and state coordination
Reasoning and tool use	Reasoning and external computation using partially observed, revisable input	Inner-monologue or latent reasoning; background computation; asynchronous retrieval and tools; conditional and overlapping reasoning; task-version binding	Moshi; Qwen3-Omni; FLAIR; STITCH; VST; R3-Streaming; Stream RAG; MoshiRAG; DuplexOmni	Overlap computation with listening and speaking	Sec. 7.1 content, repair, and tool success	Sec. 8.1 capability preservation; Sec. 8.2 runtime coordination
Factual grounding	Claims supported by current perceptual, retrieved, remembered, or inferred evidence	Source tracking; links between claims and evidence; uncertainty estimates; evidence checks before responding	Stream RAG; MoshiRAG; Qwen3-Omni	Ensure claims are correct at the time they are made	Sec. 7.1 content quality and evidence timing	Sec. 8.1 factual capability preservation; Sec. 8.2 loss of grounding
Social and pragmatic understanding	Speaker emotion, intent, addressee, roles, relationships, and conversational context	Prosodic state; audiovisual speaker tracking; gaze and addressee inference; state conditioned on participants	SPIRIT-LM; USDM; AV-Dialog	Interpret multimodal cues from multiple participants	Sec. 7.1 timing and robustness; Sec. 7.2 multi-party coverage	Sec. 8.3 source use and privacy

5.5. Social and Pragmatic Understanding

A live assistant receives social information that is reduced or absent in a text transcript. Prosody can indicate uncertainty, urgency, affect, irony, or conversational stance. Face, gaze, body orientation, and speaker location help determine who produced an utterance and who is being addressed. The lexical form of a request, the speaker's role, and the preceding interaction also affect register, politeness, and intent. Social and pragmatic understanding concerns interpreting these cues to inform response decisions.

Paralinguistic state.

Speech models that preserve prosody provide a basis for this competence. SPIRIT-LM interleaves speech and text within a shared token stream, while its expressive variant augments speech units

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

with pitch and style information [147]. USDM uses paralinguistic speech information to support more natural spoken responses [145]. Prosodic cues inform estimates of speaker emotion, confidence, stance, and intent. They can change the interpretation of the same words and can influence later responses or memory updates.

Speaker and addressee.

Multi-party interaction requires separate estimates of speaker identity and addressee. A person can be audible without addressing the assistant. A speaker can refer to another participant while looking at an object. Addressee recognition therefore combines linguistic context with gaze, orientation, and conversational history. Earlier multimodal work showed that gaze and utterance content can be combined for addressee recognition [278]. More recent work shows that the problem remains challenging in multimodal multi-party dialogue: on a triadic-dialogue benchmark, GPT-4o achieves addressee-recognition accuracy only marginally above chance [279].

Visual cues help distinguish speakers in multiparty settings. AV-Dialog combines audio and visual cues to track the target speaker in noisy multi-speaker dialogue [274]. The model can then track who is speaking, who is being addressed, and which visible entities are relevant to the current utterance. The later decision to take the floor, backchannel, or remain silent is an interaction behavior and is evaluated in Section 7.

Role, register, and pragmatic context.

Social interpretation also considers participants' roles and relationships. The same instruction can carry different authority depending on who speaks, and the same words can convey a request, correction, joke, or refusal depending on prosody and context. Role-conditioned systems such as PersonaPlex demonstrate that duplex behavior can be explicitly conditioned on textual role prompts and speech-based voice prompts [280]. Explicit role prompts differ from inferring roles during an unconstrained interaction, where the relevant role may need to be inferred from multimodal context [278,279].

This subsection focuses on interpreting social cues. Section 7 evaluates response behavior and timing, while Sections 8.3 and 9.2 discuss privacy, personalization, and multiparty coordination.

5.6. Synthesis: Interdependent Capabilities

The capabilities above operate on shared state. Temporal grounding supplies the event structure used by memory and online reasoning. Memory keeps the entities, corrections, and task versions that later reasoning must recover. Reasoning and tools add inferred and external evidence. Factual grounding determines which parts of that evidence support a claim at the current time. Social understanding assigns observations and requests to the correct participants and supplies role and intent information to the task state.

Continuous interaction requires these capabilities to share a consistent state as new input arrives. A completed prompt can be processed after its evidence and intent are fixed, whereas a native omni model must keep time, state, inference, evidence, and participant attribution consistent while the session continues to change. Sections 7 and 8 examine their evaluation and failure modes.

6. Realizing Omni Interactivity

Omni interactivity concerns response timing and behavior during an ongoing session. We examine response latency, turn-taking, interruption recovery, proactive assistance, adaptation across turns, and the serving and client mechanisms that support them.

These behaviors build on the representations of time, silence, overlap, and backchannels described in Section 3.3, together with the temporal grounding, session state, reasoning, and tool validity examined in Section 5. They depend on both the interaction policy and its execution across the model, server, and client. Section 7 covers their metrics and benchmarks, Section 8 analyzes the intelligence–interactivity trade-off, and Section 9 develops the open research agenda.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Figure 11 summarizes the resulting interaction loop. Continuous multimodal input updates the shared state, and the interaction policy selects silence, a backchannel, substantive speech, a proactive intervention, or a control action. Playback and user feedback inform the next decision. The model, server, and client must agree on input timing, playback progress, and which outputs have reached the user.

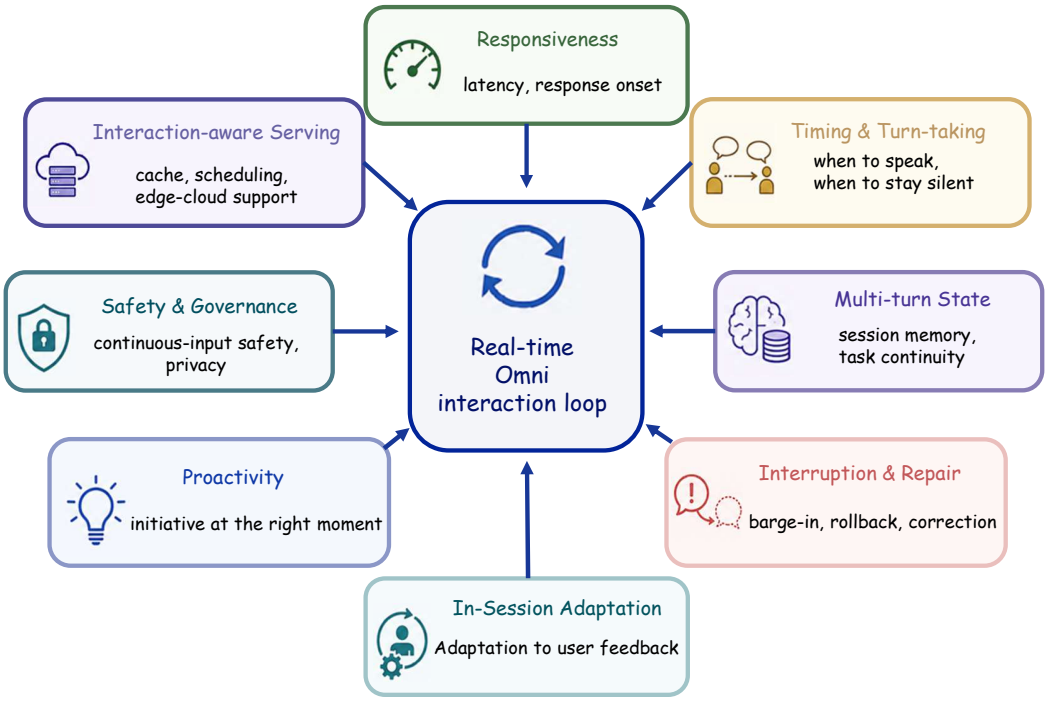


Figure 11. The omni interaction loop. Continuous audio, video, text, and system events update a shared state. The interaction policy controls response timing, floor management, repair, proactivity, and adaptation. Serving and client mechanisms execute each decision and return playback and user feedback to the loop.

6.1. Latency and Responsiveness

End-to-end latency measures the delay from a relevant input event to output reaching the user [141,281]. It includes capture and chunking, audio or visual encoding, inference, speech synthesis, network transport, buffering, and playback delays. Each stage contributes to the total latency, and a stall at any stage delays the observable response.

Cascaded systems serialize several stages [36,141]. Voice activity detection closes the input segment, automatic speech recognition produces text, the language model generates a response, and text-to-speech converts that response into audio. Streaming components reduce individual delays, although module boundaries still introduce buffering and scheduling gaps. Native omni designs can overlap capture, encoding, prefill, decoding, and synthesis [21,22,43]. Section 4 examines their architectural details, while the latency budget records the resulting critical path.

Interruption latency also matters. An interruption must propagate through the client audio controller, server scheduler, generation state, and queued playback. Fast response onset combined with slow stopping still produces a rigid interaction. Systems should therefore specify timing requirements for both starting and stopping output, including input chunking, response onset, per-token generation, playback continuity, and stop propagation. Section 7 defines the corresponding measurements and reporting protocol.

Smaller audio chunks and more frequent video sampling can improve responsiveness but increase processing cost [107,282]. Larger playback buffers tolerate network variation but delay stopping. Larger batches improve throughput but can increase queueing delay. These dependencies determine the deadlines that the serving stack in Section 6.6 must protect. Response onset and the arrival of task-relevant information can occur at different times, an issue examined as part of the intelligence–

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

interactivity trade-off in Section 8. The system should indicate whether it is waiting, acknowledging a request, seeking clarification, or providing an answer. A brief acknowledgment can precede a complete answer, with its delivery timed using turn-taking predictions. In a deployed dialogue robot, recent work prepares a context-aware prefatory response once user intent becomes predictable, while a separate Voice Activity Projection process determines when to deliver it [283]. This separates *content readiness* from *floor-taking readiness*.

6.2. Turn-Taking, Silence, and Backchannels

Turn-taking converts the time-aware representations described in Section 3.3 into an output policy. At each update, the model chooses among continued listening, silence, a backchannel, floor-taking, and yielding. Silence can allow the user to continue, indicate uncertainty, or delay a response until an event occurs. Backchannels acknowledge the user, while substantive speech begins the model's turn. Yielding releases the floor.

A policy can use acoustic, linguistic, visual, and task-state cues to select an action and its timing [16, 33,278,284]. Acoustic cues provide prosody, pause structure, overlap, and speech onset. Linguistic context indicates whether an utterance is complete, while visual context provides gaze, gesture, addressee, and ongoing activity. Task state indicates whether the model has enough information to proceed. Section 5 examines how these signals are interpreted and maintained in the interaction state.

Human conversation constrains how these cues should be mapped to actions. Long gaps can feel unresponsive, premature floor-taking cuts off the user, and excessive backchannels become distracting. A pause may indicate hesitation or turn completion, so the appropriate response depends on linguistic, visual, and social context. Background speech may require silence even when the acoustic channel is active.

Current systems place turn-taking control in different components. Speech-native full-duplex models such as dGSLM and Moshi learn timing through parallel conversational streams [15,21]. Freeze-Omni uses a chunk-level state predictor to control listening and response transitions [43]. SALMONN-omni represents speaking decisions with learned state tokens during generation [186]. BayLing-Duplex learns listen, speak, and stop control within a single autoregressive speech–language model using a small set of control tokens [187]. Analyses of interacting Moshi instances identify internal activations associated with upcoming turn changes [285]. A deployed system can combine learned timing predictions with explicit execution rules shared by the client and scheduler.

Section 3 describes the supporting representations, and Section 7 evaluates response timing, takeover rate, silence, and backchannel quality. The present subsection specifies the action policy that these representations support and the system executes.

6.3. Interruption and Repair

Interruptions can change the user's request while speech or background work continues. The system must stop pending output, update the context and task, cancel outdated work where possible, and resume consistently.

Table 8 separates this process into linked stages. Detection classifies incoming activity during model speech. When an interruption requires the system to yield, it returns the floor and stops pending playback. Rollback then aligns the model context with content delivered to the user, while cancellation stops retrieval, reasoning, and tools associated with a superseded task version. Revision applies the correction to the versioned state defined in Section 5, allowing resumption to proceed from the corrected interaction state.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Table 8. Interruption repair stages. Each stage changes a different part of the live system.

Stage	Required transition	Failure if omitted
Detect	Classify barge-in, backchannel, side speech, or noise	False stop or ignored correction
Yield	Duck speech and flush pending playback	Model talks over the user
Roll back	Align context with played content	References to undelivered output
Cancel	Stop work tied to the old task version	Stale evidence or side effects
Revise	Apply the corrected intent and constraints	Continued execution of the old request
Resume	Continue from the corrected interaction state	Repetition or lost task progress

To support rollback and consistent resumption, the conversational record should distinguish generated, queued, played, and canceled content, using playback progress to track what has been delivered to the user. ChipChat implements this by tracking vocalized tokens and rolling the context back to the point reached by playback [181]. Training can also shape the conversational behavior surrounding these transitions. Multi-Faceted Interactivity Alignment uses separate rewards for pause handling, turn-taking, backchannels, and interruption, together with a response-quality reward [243]. By combining interaction and content objectives, it connects repair behavior to the alignment objectives discussed in Section 4.3.

Repair must also extend to tool operations, which may already have changed external state when an interruption arrives. Recovery therefore needs to account for what was executed and which results remain valid. The versioned task and evidence states in Section 5 help identify outdated work, while recovery may require cancellation, confirmation, or compensation. These requirements also inform the evaluation in Section 7, which examines whether repair succeeds and how much obsolete content is delivered before recovery.

Coordinating this recovery requires information from the client. Echo cancellation and near-end speech detection help identify incoming user speech, while audio ducking reduces playback and cancellation stops pending output. Playback timestamps then help the server distinguish content already delivered to the user from content that should be removed from the shared conversational record. Section 6.6 places these functions within the complete serving and client path.

6.4. Proactive Interaction

Proactive interaction occurs when the system initiates a contribution without a new explicit request. A visual event, acoustic cue, elapsed interval, tool result, or change in task state can trigger this decision [113,116,286–288]. Temporal grounding and event understanding examined in Section 5 determine what happened and when, while the proactive policy determines whether an intervention is appropriate at the current moment. The policy should consider task relevance, urgency, confidence, distraction, privacy, social context, and whether an action can be reversed, together with user permissions and preferences [286,288,289]. A safety warning may justify immediate speech, while a low-confidence suggestion can wait for a pause or appear as a quiet visual cue [290].

Streaming video systems provide early implementations of this policy. VideoLLM-online predicts a response or silence at each frame, MM-Duet allows either participant to contribute during video playback, and systems with separate perception, decision, and reaction components continue observing while selecting a response [113–115]. StreamBridge adds proactive activation to models trained for offline video understanding [116]. Proact-VL trains multimodal language models for continuous video perception, autonomous response triggering, and latency-aware commentary generation. Its Live Gaming Benchmark evaluates response timing and content selection in solo commentary, co-commentary, and user guidance [288]. These systems repeatedly decide whether to speak or continue observing as new input arrives, using model predictions to guide response timing.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

When background results arrive, the model decides whether to use them immediately, wait, notify the user nonverbally, or discard them. It considers whether the result still matches the task and whether the user is currently speaking. Foreground–background architectures provide the concurrent computation, while the proactive policy controls how its results are presented [26,27,60].

Section 7 evaluates proactive response timing, while Sections 8 and 9 discuss decisions based on incomplete evidence, persistent interaction, privacy, and safety.

6.5. Adaptation Across Turns

Within-session adaptation changes response behavior based on user feedback. Subsequent turns should reflect preferences about response length, speaking pace, acknowledgment frequency, interruption tolerance, and initiative [238,291]. A corrected answer demonstrates repair, while a sustained change in behavior demonstrates adaptation. The interaction layer can represent these preferences through policy variables such as response length, speech rate, pause duration, backchannel rate, confirmation threshold, and proactivity threshold [238,291]. Each preference update should specify where it applies and when it expires. A task-specific correction can expire with the task, while a session preference can persist until the conversation ends. Section 5 examines the memory and provenance needed to store these updates, and Section 9 considers persistent personalization across sessions.

Interaction-policy adaptation.

Appropriate turn-taking depends on the task and participant roles. A tutor may interrupt proactively, whereas a counseling assistant may need to avoid interruption. Instruct-FD evaluates whether full-duplex systems can follow instructions about when to listen, backchannel, interrupt, or continue speaking, and reports a substantial gap between fluid and controllable full-duplex behavior [291]. DuplexGen shows that preferences over turn-taking vary across scenarios and can be learned from a small amount of preference supervision [238]. Together, these studies motivate interaction policies controlled by instructions and user preferences, with timing patterns learned from conversational data serving as an initial policy.

Preferences can be stored explicitly or inferred from recent interaction history. Explicit state supports inspection, correction, and deletion [292], while context-conditioned control infers preferences from recent feedback [293]. Both approaches can adapt behavior during a session without changing model weights. Section 4.3 examines offline alignment for timing, turn-taking, backchannels, and interruption.

Preference updates should rely on repeated or explicit feedback. A single interruption may correct content, signal urgency, or result from background noise, so it does not necessarily indicate a lasting preference [293,294]. Repeated interruptions during long answers provide stronger evidence for shorter responses, and rejected proactive suggestions support a higher intervention threshold in the same context [293,294]. Repeated corrections about an entity should increase confidence in the updated interpretation [292,294].

Section 7 evaluates adaptation by comparing behavior before and after feedback under matched conditions. Within-session adaptation changes response timing, style, initiative, or confirmation behavior, while Section 9 considers learning, personalization, and consistency across sessions.

6.6. Full-Duplex Serving and Client Coordination

The behaviors above depend on a serving stack for long-lived, bidirectional sessions. Serving systems must coordinate incoming media, playback, state updates, and cancellation [25,43,60]. Audio and video arrive in frequent small chunks, speech decoding may continue as new input arrives, and an interruption can invalidate part of the current context. Tools and background models also return results on independent schedules [25–27].

Persistent streaming sessions provide the basic execution unit. Per-session encoder state, KV cache, interaction state, and playback state remain available across chunks and workers. Freeze-

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Omni separates user-specific caches from model execution so that a stateless worker pool can process active sessions [43]. Thinking Machines Lab appends 200 ms micro-turns to a persistent GPU-resident sequence and optimizes MoE kernels for the small input shapes produced by bidirectional serving [60]. Section 5 defines the semantic content of session state, while this subsection covers its storage, scheduling, and mutation.

Scheduling must meet speech-generation deadlines while processing new input. Incoming chunks produce frequent prefills that compete with ongoing decoding. Chunked prefill divides new context into bounded units, and continuous batching schedules work at iteration granularity [39,295,296]. Separating prefill and decoding across resources can reduce interference between them [297,298]. Playback progress and user activity can also help interaction-aware schedulers prioritize sessions whose audio buffers are nearly empty and prepare for barge-in [281].

Cache management must limit memory use while retaining timestamps, task versions, corrections, and playback or output-status records needed for recovery. Paged allocation reduces fragmentation, while sliding windows, attention sinks, selective eviction, and retrieval-backed caches limit active memory [38,262,299,300]. Semantic relevance alone may not identify every state required for repair.

MoE backbones introduce additional latency at the small batch sizes common in interactive decoding. Expert-weight movement, routing, and cross-device communication can dominate each decoding step [301–306]. Expert locality, prefetching, phase-specific parallelism, and kernels optimized for small inputs can reduce this cost [60,305,306]. The system should keep decoding time within its latency target under realistic concurrent workloads.

Time-critical functions can run on the client or edge device, while larger models, retrieval, and tools run in the cloud. Edge functions can include wake-word detection [307], VAD [185], echo cancellation [308], audio compression [77,78], floor control [16,33], and interruption detection [43,185]. UAF consolidates several of these functions by predicting VAD, turn detection, speaker recognition, ASR, question answering, and interruption-related control signals as a streaming autoregressive sequence [185]. Network degradation may require reduced media rates, lower codec bitrates, delayed background work, or simpler local behavior [309–312].

The client completes the control loop by capturing media, performing echo cancellation, reducing playback volume during an interruption, maintaining a bounded jitter buffer, and sending playback timestamps to the server. It also shows whether the system is listening, processing, speaking, muted, or waiting for confirmation. These signals support turn coordination and user control. Table 9 maps the main system mechanisms to the behavioral failures they prevent.

Omni interactivity requires timely responses, appropriate turn-taking, coordinated recovery, useful proactive assistance, adaptation to feedback, and serving mechanisms that support these behaviors. Section 7 measures these capabilities, Section 8 examines their relation to intelligence, and Section 9 identifies the remaining research problems.

Table 9. System support for omni interactivity. Each mechanism covers a distinct part of the interaction loop.

Requirement	Representative mechanism	Failure when absent
Continuous media flow	Chunked prefill and deadline-aware scheduling	Audio stalls during new input
Bounded session state	Paged KV and selective retention	Memory growth or lost interaction state
Correct repair	Playback timestamps, queue flush, and rollback	References to unheard output
Valid background work	Task versions, validation, and cancellation	Stale results re-enter the session
Robust deployment	Edge control and adaptive media transport	Network delay breaks the dialogue loop

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

7. Evaluating Intelligence and Interactivity

Real-time evaluation must assess response content together with timing, interruption handling, correction recovery, and consistency across turns. We review evaluation metrics in Section 7.1, existing benchmarks and their coverage in Section 7.2, and limitations of the available protocols in Section 7.3.

7.1. Evaluation Metrics

We organize evaluation into six dimensions: latency, timing quality, content quality, robustness, long-session quality, and repair. Figure 12 summarizes their representative metrics and evaluation settings.

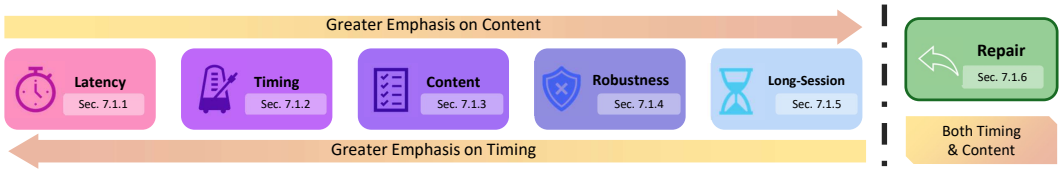


Figure 12. The six evaluation dimensions defined in Section 7.1 for native real-time omni models: latency, timing, content, robustness, long-session quality, and repair.

7.1.1. Latency

Latency captures several distinct delays, including the time to begin generating a response, deliver audible speech, and react to an interruption. Each measure depends on its starting event: first-token and first-audio latency may be measured from user-turn completion or from an annotated point when sufficient input has arrived. The following measures distinguish these stages of the interaction:

- **First-token latency**: elapsed time from declared input event to emission of the first text or semantic token.
- **First-audio latency**: elapsed time from the declared input event to the first audible output at the client, including synthesis, buffering, and playback delays.
- **Interruption-stop latency**: elapsed time from the onset of an interrupting user utterance to cessation of model speech at the client. State whether the start event is annotated speech onset or system detection of the interruption.
- **Response-after-interruption latency**: elapsed time from the interruption event to the start of the subsequent response, using the start and end events specified by the evaluation protocol. Full-Duplex-Bench v1 (FDB-v1) reports this as *Latency After Interruption* [137], which measures response resumption rather than how quickly the model stops its ongoing speech.
- **Turn-taking latency**: the gap between the end of the user's utterance and the onset of the model's response. FDB-v1 measures this as *response latency*, "the time (in seconds) between the end of the user's speech and the start of the model's response," computed only on turns where the model actually takes over the floor [137].

Observed latency depends on both the architecture and the conditions under which it runs. In cascaded VAD–ASR–LLM–TTS systems, delays accumulate across components and their buffers, with additional delays from network transmission and client playback. Interpreting reported values therefore requires information about the hardware, workload, number of concurrent sessions, and buffering configuration. FDB-v3 reports wall-clock latency for multi-step tool-use tasks [175], while SimulEval provides latency measures for simultaneous translation [313].

7.1.2. Timing Quality

Timing quality evaluates whether responses and backchannels occur at appropriate moments. TurnGPT predicts turn completion from text [127], while Voice Activity Projection (VAP) uses self-supervised learning to predict future speech activity from audio [16]. Follow-up work evaluates the

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

effects of prosody [314] and fillers or pauses [130]. Skantze surveys turn-taking in spoken dialogue systems [33]. MuVAP extends prediction to multimodal, multiparty settings [284].

FDB-v1 uses **Takeover Rate (TOR)**, the fraction of probes in which the model begins a speaking turn, across pause handling, backchanneling, turn-taking, and interruption [137]. Its preferred direction depends on the scenario. A low takeover rate is desirable when a pause does not yield the user's turn; a high rate is desirable when the user yields and expects a response. Interruption results must identify who interrupts whom and the intended response behavior rather than assigning one preferred direction to all interruptions. For proactive video responses, ProactiveVideoQA evaluates response timing as well as content using PAUC [118].

7.1.3. Content Quality and Instruction Following

Content quality evaluates correctness, completeness, and instruction following. OmniBench measures static multimodal understanding and does not evaluate streaming interaction or response timing [315]. VoiceBench evaluates instruction following under acoustic and speaker variation [316]. Dynamic-SUPERB and its Phase-2 extension cover classification, regression, and sequence-generation tasks [317,318]. Audio MultiChallenge evaluates instruction retention and recall of audio cues that are absent from transcripts [319]. SpokenWOZ reports task completion and dialogue-state tracking performance in spoken task-oriented dialogue [320].

7.1.4. Interaction Robustness

Robustness evaluates performance under overlapping speech, noise, interruptions, and side conversations. Overlap should be evaluated as a normal conversational condition as well as a source of interference; dGSLM models overlapping speech, laughter, and backchannels in two-channel dialogue [15]. FDB-v1.5 evaluates four overlap scenarios: user interruptions, user backchannels, speech directed to another person, and background speech. It measures response behavior, timing, and prosodic adaptation [138]. FDB-v1 also evaluates interruption handling [137]. VoiceBench varies speakers, acoustic conditions, and content [316]. IRAF addresses interfering speech by adjusting the contribution of incoming audio features; it is a method addressing a robustness failure, rather than a benchmark [321].

7.1.5. Long-Session Quality

Long-session evaluation covers consistency, entity and event tracking, task state, and memory over extended interaction. FDB-v2 uses an automated examiner with staged goals and two pacing conditions, Fast and Slow. Its tasks cover daily interaction, correction, entity tracking, and safety, and its results identify difficulties with overlapping speech, user corrections, and entity tracking [139]. Audio MultiChallenge adapts inference-memory and self-coherence evaluations to spoken multi-turn dialogue [319]. M3-DuplexBench evaluates multilingual dialogue with varying amounts of prior context [322]. SpokenWOZ measures task-state maintenance across spoken turns [320], while StreamingBench tests retention of earlier streamed information [120]. Multi-turn evaluation and continuous hours-long interaction should be distinguished when reporting session length. User simulation offers a complementary approach to long-session evaluation: a simulated user interacts with the system, and performance is assessed from the resulting interaction history [323].

7.1.6. Repair Behavior

Repair evaluation tests whether the model incorporates a correction and stops obsolete output or work. Audio MultiChallenge evaluates mid-utterance and previous-turn edits [319], while FDB-v2 tests correction in streaming multi-turn dialogue [139]. We propose reporting *correction-recovery rate*, the fraction of correction trials whose final response satisfies the revised request, and *stale-content rate*, the fraction that continue using the superseded request after a specified recovery allowance. Reports should define the correction event, scoring rubric, recovery allowance, and trial denominator.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

7.2. Comparative Benchmark Coverage

We group benchmarks by their primary focus on **Omni Intelligence** or **Omni Interactivity**, while comparing both groups using the six dimensions in Section 7.1. These groups describe a benchmark's primary purpose; individual benchmarks may assess capabilities from both groups.

Figure 13 presents the historical development of the benchmark families. The intelligence branch includes multi-turn dialogue evaluation, static omni-modal and spoken understanding, and streaming audiovisual and temporal understanding. The interactivity branch includes full-duplex spoken interaction, streaming and proactive omni interaction, and reusable tools and measures for evaluating turn-taking. Within each family, resources are ordered by first public release year.

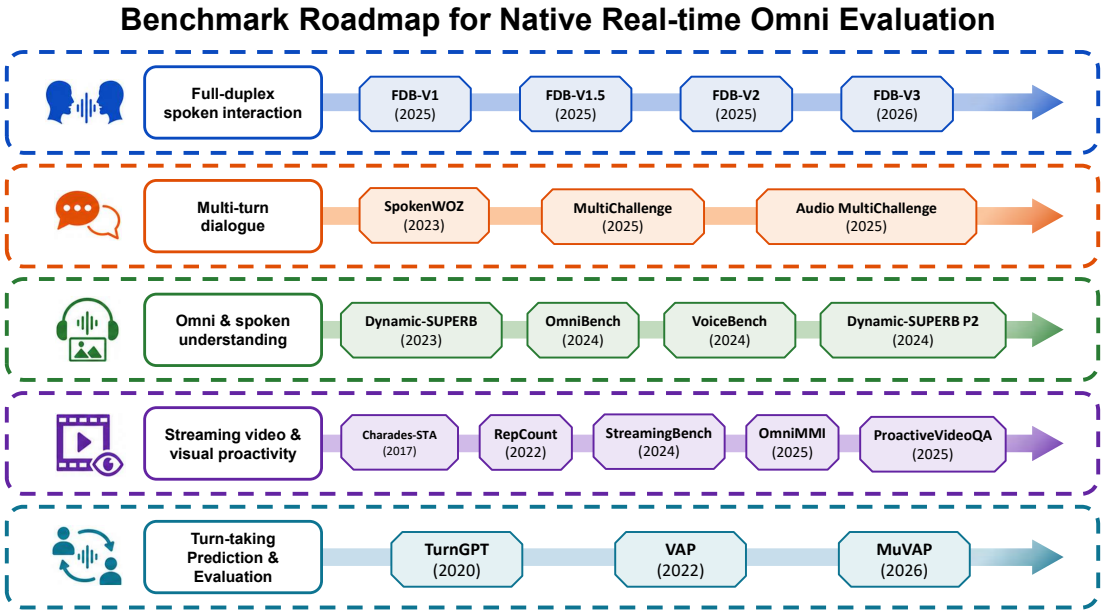


Figure 13. Representative evaluation resources by *first public release year*, grouped by benchmark family. Temporal-grounding and turn-taking evaluations precede benchmarks for end-to-end native interaction, while public full-duplex and online omnimodal benchmarks emerged mainly from 2025 onward.

Table 10 compares coverage across evaluation dimensions. Filled symbols indicate direct coverage through a dedicated task or separately reported metric. Partial symbols indicate limited or indirect coverage: a subset of tasks, or a capability affecting an aggregate score without a separate measurement. Empty symbols indicate absent coverage. These marks characterize benchmark

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Table 10. Benchmarks grouped by whether they primarily assess Omni Intelligence or Omni Interactivity. Coverage is assessed across six dimensions from Section 7.1: **La**=Latency, **Ti**=Timing, **Co**=Content, **Ro**=Robustness, **Ls**=Long-session, and **Re**=Repair. Vendor/internal evaluations are reported separately and must not be co-ranked with public benchmarks. Coverage marks reflect the authors' assessment of each benchmark's stated scope.

Benchmark [ref]	Input modality	Evaluation setting	La	Ti	Co	Ro	Ls	Re
I. Omni Intelligence								
Evaluates the model's multimodal perception, semantic understanding, reasoning, instruction following, dialogue consistency, and long-context comprehension.								
I-A. Multi-turn dialogue understanding and consistency								
Audio MultiChallenge [319]	Speech (E2E)	Multi-turn dialogue	○	○	●	◑	●	●
MultiChallenge (text) [324]	Text	Multi-turn, turn-based	○	○	●	○	●	◑
SpokenWOZ [320]	Speech + text	Task-oriented dialogue	○	○	●	◑	●	○
I-B. Omni and spoken understanding								
VoiceBench [316]	Speech	Voice-assistant tasks	○	○	●	●	○	○
OmniBench [315]	Image + audio + text	Static tri-modal QA	○	○	●	○	○	○
Dynamic-SUPERB [317]	Speech + audio	Zero-shot instruction	○	○	●	◑	○	○
Dynamic-SUPERB Phase-2 [318]	Speech + audio	Zero-shot instruction	○	○	●	◑	○	○
I-C. Streaming visual intelligence								
StreamingBench [120]	Streaming video + audio	Streaming video QA	◑	◑	●	○	◑	○
Charades-STA [325]	Video + language	Temporal localization	○	◑	◑	○	○	○
Charades [326]	Video + language	Temporal understanding	○	◑	◑	○	○	○
RepCount (TransRAC) [327]	Video	Repetition counting	○	◑	●	○	○	○
II. Omni Interactivity								
Evaluates the model's real-time responsiveness, temporal coordination, turn-taking, interruption handling, proactivity, full-duplex communication, and recovery from interaction failures.								
Benchmark [ref]	**Input modality**	**Interaction setting**	**La**	**Ti**	**Co**	**Ro**	**Ls**	**Re**
II-A. Full-duplex spoken interaction								
Full-Duplex-Bench v1 [137]	Speech (duplex)	Single-turn probes	●	●	○	◑	○	○
Full-Duplex-Bench v1.5 [138]	Speech (duplex)	Overlapping speech	●	◑	○	●	○	○
Full-Duplex-Bench v2 [139]	Speech (duplex)	Multi-turn (Fast/Slow)	◑	●	◑	◑	●	●
Full-Duplex-Bench v3 [175]	Speech + tools	Multi-step tool use	●	●	◑	◑	◑	○
M3-DuplexBench [322]	Speech (duplex)	Multi-turn, multilingual	◑	●	◑	◑	●	◑
Instruct-FD [291]	Speech (duplex)	Instruction-conditioned turns	○	●	◑	◑	◑	◑
II-B. Streaming and proactive interaction								
OmniMMI [328]	Streaming video + audio	Streaming interaction	◑	●	●	○	◑	○
ProactiveVideoQA [118]	Streaming video	Proactive interaction	◑	●	◑	○	○	○
StreamArena [189]	Streaming video + audio	Long-horizon agentic	◑	◑	●	○	●	○
Proact-VL [288]	Streaming video	Proactive commentary	◑	●	◑	○	○	○
II-C. Turn-taking and interaction timing								
VAP [16]	Two-channel audio	Turn-taking modelling	○	●	○	○	○	○
TurnGPT [127]	Text	Turn-taking modelling	○	●	○	○	○	○
MuVAP [284]	Audio + video	Multiparty turn-taking	○	●	○	○	○	○
II-D. Vendor and internal interaction evaluations								
TimeSpeak [60]	Speech, reported	Interaction, reported	◑	●	◑	○	○	○
CueSpeak [60]	Speech, reported	Interaction, reported	◑	●	◑	○	○	○

scope rather than model performance.

Public reproducibility is treated separately from capability coverage. Benchmarks with released data, code, a hosted evaluator, or a public leaderboard support independent comparison. Publicly documented benchmarks whose release status cannot yet be independently confirmed are marked with ‡. Vendor and internal evaluations are marked with † and are not co-ranked with public benchmarks.

7.2.1. Omni Intelligence Benchmarks

These benchmarks evaluate multimodal understanding, reasoning, event timing, and retention across turns. We distinguish three benchmark families.

Multi-turn dialogue understanding and consistency.

This family evaluates whether model capabilities persist when relevant information and instructions are distributed across multiple dialogue turns. MultiChallenge measures instruction retention,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

inference memory, self-coherence, and related failures in text-based dialogue [324]. Audio MultiChallenge transfers these evaluation targets to end-to-end speech and introduces audio-specific challenges involving ambient and paralinguistic cues, as well as voice editing and spoken repair [319]. Spoken-WOZ evaluates task-oriented spoken dialogue, including dialogue-state tracking, cross-turn slots, reasoning slots, and task completion [320]. ChildAgentEval compares agent capabilities with age-specific developmental measures without imposing real-time interaction constraints [329]. These benchmarks evaluate content quality and retention across turns, primarily under turn-based protocols without explicit real-time deadlines.

Static multimodal and spoken understanding.

This family measures semantic understanding, reasoning, instruction following, and robustness over static or turn-complete input. OmniBench evaluates integrated image, audio, and text reasoning [315,330]. VoiceBench evaluates knowledge, instruction following, safety, and robustness to variation in speaker, acoustic environment, and spoken content [316]. Dynamic-SUPERB and Dynamic-SUPERB Phase-2 provide broad instruction-conditioned evaluation across speech, music, and environmental sounds, covering classification, regression, and sequence-generation tasks [317,318]. These benchmarks assess baseline capabilities that should be preserved when adding real-time interaction.

Streaming audiovisual and temporal intelligence.

This family evaluates whether a model can understand events as a multimodal stream unfolds and associate observations with their temporal positions. StreamingBench evaluates real-time, omni-source, and contextual understanding at multiple points in a streaming video [120]. Charades-STA provides language-conditioned temporal localization, while Charades supplies temporally annotated activity understanding [325,326]. RepCount and TransRAC provide fine-grained evaluation of repetitive-action counting and temporal structure [327]. Charades-STA, Charades, and RepCount assess event timing and repetitive-action counting. These static datasets assess what happened and when, but do not measure autonomous response timing.

7.2.2. Omni Interactivity Benchmarks

The omni interactivity branch evaluates when and how a model acts during an interaction. It focuses on latency, turn coordination, interruption handling, proactive behavior, and recovery from interaction failures.

Full-duplex spoken interaction.

The Full-Duplex-Bench series evaluates simultaneous listening and speaking. FDB-v1 evaluates pause handling, backchanneling, turn taking, and interruption using controlled behavioral probes [137]. FDB-v1.5 extends the setting to overlapping speech, including user interruptions, listener backchannels, side conversations, and ambient speech [138]. FDB-v2 introduces an automated examiner, Fast and Slow pacing conditions, and multi-turn task families covering daily interaction, correction, entity tracking, and safety [139]. FDB-v3 adds multi-step tool use with real human audio and reports task accuracy, wall-clock latency, and turn-taking behavior [175]. HumDial-FDBench further contributes real dual-channel human conversations containing interruption, overlap, and feedback events [210]. M3-DuplexBench evaluates English and Japanese conversation and question answering with different amounts of dialogue history [322]. Instruct-FD evaluates whether systems follow explicit instructions about listening, acknowledging, interrupting, and continuing speech [291]. Together, these resources provide the strongest existing coverage of latency, interaction timing, and overlap robustness, although their modality coverage remains primarily speech-based.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

Streaming and proactive omni interaction.

This family evaluates systems that operate over continuous audiovisual input and must decide whether and when to respond. OmniMMI combines streaming audiovisual understanding with proactive reasoning across multiple interaction tasks [328]. ProactiveVideoQA isolates the response-initiation decision and evaluates whether a model answers at an appropriate point during video playback [118]. Proact-VL evaluates proactive commentary over continuous game video, jointly scoring response timing and content across solo commentary, co-commentary, and user guidance [288]. StreamArena evaluates extended streaming interaction, including current perception, recall of earlier observations, proactive responses, and tool use [189]. OmniInteract combines online audiovisual input with measures of correctness, timing, interruption handling, invalid outputs, and context retention [119]. These benchmarks connect omni intelligence with interaction timing, but do not yet provide a complete standard for bidirectional spoken overlap, semantic repair, tool use, and long-session operation under a shared protocol.

Turn-taking prediction and evaluation.

VAP, TurnGPT, and MuVAP provide reusable tasks and measures for assessing conversational timing within broader evaluations. VAP predicts future voice activity from two-channel audio and supports evaluation of turn shifts and backchannels [16]. TurnGPT predicts turn completion from textual dialogue context [127]. MuVAP extends future-activity projection to multimodal, multiparty settings using audio and visual speaker information [284]. These tasks and measures can be incorporated into broader omni-interaction evaluations.

Vendor and internal evaluations.

TimeSpeak and CueSpeak are reported as internal Thinking Machines evaluations [60]. TimeSpeak examines time-triggered speech initiation, while CueSpeak evaluates responses triggered by semantic cues during ongoing speech. These vendor-reported evaluations are presented separately because the survey does not identify public datasets and protocols supporting independent reproduction.

7.3. Limitations of Existing Benchmarks

Current benchmarks cover different aspects of content quality, timing, and interaction behavior, but evaluating these capabilities together remains a challenge. The following proposals concern joint evaluation under common episodes and resource budgets, including where existing benchmarks provide partial coverage:

7.3.1. Joint Intelligence–Interactivity Evaluation

Existing static benchmarks can measure whether an answer is correct, while full-duplex benchmarks can measure whether a model yields, interrupts, or takes the floor. Recent benchmarks such as OmniMMI, ProactiveVideoQA, and OmniInteract begin to combine streaming understanding with online response decisions [118,119,328], but a joint protocol should evaluate continuous audio–video reasoning, bidirectional spoken overlap, semantic correction, and end-to-end serving latency using the same episodes and resource budget. One way to narrow the gap between offline replay and closed-loop evaluation is to augment recorded episodes with synthetic observations of plausible future states [331]. User simulation could also support closed-loop evaluation by generating user actions in response to the model's outputs, allowing assessment of how the model adapts to changes in user goals and knowledge [332]. A future protocol should evaluate the same system in at least four conditions: offline content evaluation, causal streaming perception, proactive response generation, and full-duplex interruption. Results should be reported as a Pareto frontier.

Proposed measurements: Evaluate content quality and interactive behavior together under the same episodes and resource budgets, reporting event-level timing and computational cost alongside capability scores. These measurements show how response quality changes under real-time constraints. Report accuracy at specified deadlines and define the input event and output endpoint for response

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

onset and useful-answer time. A useful answer should satisfy a task-specific rubric; report failures to produce one separately. Define accuracy after interruption using the revised request, and replay timestamped episodes without access to future input.

7.3.2. Long-session Continuous Audio and Video

StreamingBench and OmniMMI evaluate bounded video streams and timed probes [120,328]; FDB-v2 and Audio MultiChallenge evaluate multiple turns [139,319]. These protocols are substantially shorter and more controlled than persistent meetings, tutoring sessions, assistive agents, embodied monitoring, and long-running task execution. StreamArena begins to probe hour-scale agentic streaming interaction, but remains video-centric rather than bidirectionally spoken [189]. Long-session evaluation must separate recent-scene perception from long-range memory, cap computation and memory growth, and ensure that the model remains an active participant throughout the session rather than answering only at the end.

Candidate metrics: entity and event retrieval as a function of elapsed time; contradiction rate; task-state recovery after long gaps; accuracy at fixed memory and compute budgets; latency drift over session duration. State the memory budget, compute limit, and session duration for each result.

7.3.3. Multi-party and Addressee-aware Full-duplex Interaction

Most end-to-end interaction benchmarks remain dyadic. MuVAP evaluates multiparty next-speaker prediction but is a predictive primitive rather than an interactive-agent benchmark [284]. HumDial-FDBench provides realistic full-duplex speech but does not establish a broad evaluation suite for three-or-more-speaker coordination [210].

Proposed measurements: Evaluate three or more participants with annotated speaker identity, addressee, and expected responder. Report addressee accuracy, responses to bystanders, speaker-conditioned interruption rates, and recovery from attribution errors. Measure response allocation as each participant's response rate among eligible addressed turns, accounting for differences in opportunities to respond.

7.3.4. Multilingual and Code-switching Full Duplex

Public full-duplex evaluation remains dominated by English, with limited coverage of other languages and little standardized evaluation of code-switching, low-resource languages, and language-dependent timing. M3-DuplexBench begins to address this by evaluating multi-turn full-duplex dialogue in English and Japanese [322], but broader language coverage and code-switching remain open. Static accent or multilingual understanding does not establish multilingual turn-taking competence.

Proposed measurements: Compare content accuracy and response timing by language. Score code-switch recognition against annotated language-switch events, and score recovery by whether the system follows the revised language instruction within a stated interval. Specify backchannel appropriateness in each language.

7.3.5. Repair, Cancellation, and Tool Use During Continuous Input

Audio MultiChallenge evaluates voice editing, FDB-v2 evaluates correction, and FDB-v3 evaluates disfluent multi-step tool use [139,175,319]. What remains missing is their combination: a user revises the request while the model is speaking, a background tool or retrieval process is already running, and the system must cancel, update, or safely reuse partial work without emitting stale results.

Candidate metrics: correction-recovery rate and stale-content rate as defined above; cancellation success among cancellable operations; unnecessary tool execution and its cost; and elapsed time from the correction to the revised action. State the denominator and recovery deadline for each measure.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

7.3.6. Controllable Interaction Policy

Existing benchmarks largely ask whether interaction is *natural*. They rarely ask whether it is *appropriate* for a specified role, or whether it can be changed on instruction. Instruct-FD evaluates instruction-conditioned turn management and finds a large gap between fluid and controllable behavior [291], while DuplexGen shows that the target turn-taking policy is itself scenario dependent [238].

Proposed measurements: instruction adherence for listening, acknowledging, interrupting, and continuing speech. Specify when the instruction changes and measure the delay until compliant behavior begins, using a stated observation interval and scenario-specific scoring rubric.

7.3.7. Safety, Privacy, and Deployment Robustness Under Continuous Input

Turn-based safety evaluation does not cover bystander speech, visible private information, environmental leakage, cross-channel prompt injection, packet loss, delayed frames, or safety actions that must proactively interrupt the user. A system may also appear safe only because network delay prevents it from responding in time.

Proposed measurements: define privacy leakage as unauthorized disclosure of specified private information captured in audio or video. Identify prohibited actions, required interventions, and intervention deadlines, then report missed and false interventions, disclosure rates, and errors under overlapping speech, controlled packet loss, and frame delay.

8. The Intelligence–Interactivity Trade-Off

A central question is whether real-time speech and multimodal capabilities can be added while preserving the reasoning, factual knowledge, grounding, and tool use learned during language pretraining. Evaluations examine how speech and interaction objectives affect these capabilities [28,29,333].

The trade-off also concerns reliable use of those capabilities under real-time constraints and changing evidence. Reasoning, retrieval, grounding, and tool execution may remain incomplete after speech begins, while users may interrupt or revise the task [25,26,57]. Figure 14 distinguishes capability preservation during adaptation from response, evidence, and state coordination during runtime inference.

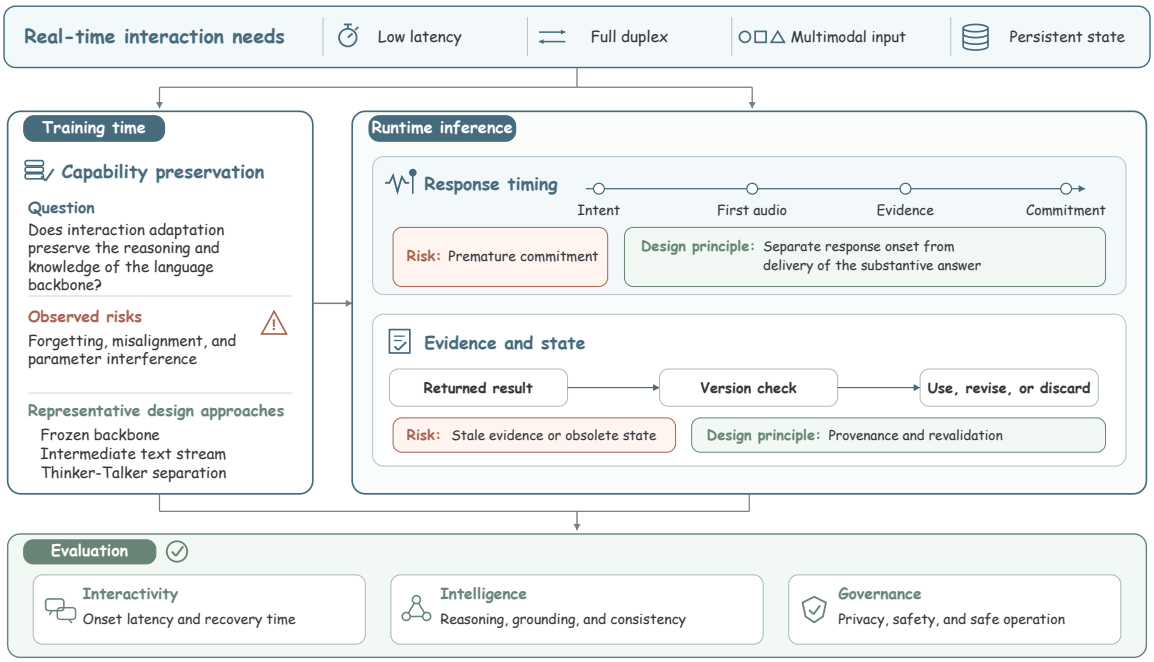


Figure 14. Trade-off between intelligence and interactivity in native real-time omni systems. Real-time interaction requires preserving pretrained capabilities during adaptation and coordinating responses, evidence, and state updates during inference. The diagram summarizes associated risks and representative design approaches.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

8.1. Capability Preservation under Real-Time Omni Adaptation

Evaluations identify capability losses in some models after speech adaptation. URO-Bench reports that several end-to-end spoken dialogue models lag behind their underlying language-model backbones in instruction following and exhibit catastrophic forgetting [28]. Related analysis attributes the text–speech understanding gap to two mechanisms: forgetting of textual capabilities during speech adaptation and misalignment between speech and text representations [29]. Parameter-importance analysis suggests that speech fine-tuning can alter parameters associated with textual reasoning in different layers [333]. The effect depends on training data, optimization, modality interfaces, and whether the language backbone is updated. These findings do not establish that capability loss is inevitable.

A recurring architectural response is to separate interactive speech generation from the representations and computation used for language reasoning. Moshi retains an intermediate linguistic stream within its speech–text foundation model, keeping semantic processing explicit during real-time dialogue [21]. Freeze-Omni freezes the pretrained LLM backbone and attaches separate speech input and output modules [43]. Qwen3-Omni extends this separation through a Thinker–Talker architecture, in which multimodal understanding and reasoning are separated from streaming speech generation [23]. JoyAI-Talker combines a Thinker–Talker architecture with unified speech–text joint training that is designed to mitigate degradation in language understanding and reasoning performance while retaining performance on textual reasoning, STEM tasks, and logic, adding state-driven duplex control and expressive paralinguistic generation [334]. BayLing-Duplex adds listen, speak, and stop control tokens to a speech-language model and reports retained response quality [187].

These designs aim to preserve pretrained reasoning and factual knowledge during speech adaptation. Reported results are consistent with this principle, but they do not yet isolate the effect of individual design choices. Freeze-Omni reports spoken-input performance close to its frozen backbone's text-input performance, while Qwen3-Omni reports performance comparable to same-sized single-modal models across text, image, audio, and video tasks [23,43]. In contrast, broader analyses of speech-adapted LLMs identify textual forgetting, cross-modal misalignment, and disruption of text-critical parameters [29,333]. Because these studies use different backbones, training data, model scales, and evaluation settings, controlled comparisons are still needed to determine when backbone freezing, constrained adaptation, or explicit reasoning–speech separation best preserves the capabilities of the original language model.

Interaction-policy optimization can also reduce response quality unless its objectives explicitly preserve semantic content. Multi-Faceted Interactivity Alignment uses separate rewards for pause handling, turn-taking, backchanneling, and user interruption, together with a response-quality reward [243]. Capability losses can therefore occur both when adding speech during adaptation and when optimizing interaction timing without preserving response quality.

8.2. Runtime Trade-Offs

8.2.1. Response Timing, Reasoning, and Grounding

A model can start responding before reasoning, retrieval, or tool execution is complete. We distinguish *response-onset latency* from *useful-answer time*, when task-relevant information first reaches the user, using the reporting requirements in Section 7.1. Systems that overlap speech with reasoning, retrieval, or tool execution illustrate why these endpoints differ [25–27,57].

Early speech is not necessarily a failure. Brief acknowledgments and backchannels can preserve conversational flow while the system continues to reason, and full-duplex benchmarks distinguish these behaviors from interruptions, side conversations, and ambient speech [138]. The problem arises when early output implies that the answer is already grounded, states unsupported content, or commits to an interpretation that the user later corrects. Full-Duplex-Bench-v2 directly evaluates correction and entity-tracking tasks and reports persistent difficulties under overlapping multi-turn interaction [139]. In tool-augmented settings, a query issued before the user's request is sufficiently clear may retrieve

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

irrelevant information; recent work formalizes this issue as tool-intent stabilization [27,179]. Systems should distinguish early acknowledgments from factual answers and validate later claims or tool results against the current task.

8.2.2. Evidence and State Coordination

Observations and returned results differ in source, timing, reliability, and relevance. These include user and background speech, video, partial transcripts, retrieval outputs, tool results, and memory. This creates three recurring risks: source confusion, in which incidental input is treated as user-directed evidence; temporal misalignment, in which information is attached to the wrong moment; and stale evidence, in which a result associated with an earlier task state is used after the user's intent has changed. Static multimodal evaluation shows that irrelevant modalities can distort model decisions, while full-duplex and streaming benchmarks expose additional difficulties in overlap handling and online visual understanding [120,138,335]. Recent omni systems accordingly treat temporal alignment and synchronized multimodal processing as central architectural requirements [200,256]. Reliable interaction therefore requires evidence to remain associated with its source, time, and current task relevance.

Longer sessions require consistent tracking of people, objects, events, goals, corrections, tool calls, and superseded tasks. Long-context and memory benchmarks show that relevant information may be overlooked and that sustained interaction requires temporal reasoning, knowledge updates, and consistency across sessions [261,336,337]. Recent spoken and audio-visual benchmarks explicitly extend evaluation to correction handling, entity tracking, and multimodal multi-turn memory [139,263]. Memory should retain each observation's source, modality, and time, support consistent entity identification, and allow task updates. Retrieved evidence must be checked for relevance to the current task version [261,262]. Together, these runtime risks show that interactivity depends on coordinating when evidence becomes valid and which version of the interaction it belongs to. Table 11 summarizes these recurring forms of the intelligence–interactivity trade-off.

Table 11. How real-time interaction requirements affect capability preservation and reliable operation. Each interactivity requirement links to the capabilities or coordination mechanisms it depends on and the failures that may occur.

Interactivity requirement	Capability or coordination requirement	Typical manifestation
Speech and omni adaptation	Preserving reasoning and textual capabilities [28,29]	Capability degradation
Interaction-timing alignment	Preserving response content while optimizing timing [243]	Reduced response quality
Immediate response	Completing deliberation and grounding before commitment [25,26]	Premature or unsupported answers
Continuous multimodal input	Selecting source-, time-, and task-relevant evidence [120,138]	Source confusion, temporal misalignment, or stale evidence
Persistent interaction	Maintaining revisable entity and task states [261,337]	Contradictory or obsolete state
Continuous sensing	Use captured information only as permitted [338,339]	Unauthorized disclosure or use of sensitive data

8.3. Safety, Privacy, and Evaluation Implications

Continuous interaction requires rules for capturing, processing, transmitting, and storing information that may not be addressed to the system, including bystander speech and private visual content. Audio and audio–visual safety benchmarks show that unsafe intent can be expressed through speech and distributed across modality combinations, while selective-hearing evaluation reveals substantial leakage of incidental bystander speech [338–341]. Permission checks must apply to each information-handling operation and to any action it triggers. These boundaries may require different

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

decisions for the same input: sensitive bystander speech may need to be detected locally for redaction without being transmitted to an external model or written into persistent memory. Privacy risk also extends to the model's internal state. Analysis of full-duplex systems such as SALM-Duplex and Moshi finds that speaker identity can become easier to infer from hidden states as a conversation progresses. Streaming anonymization can reduce this exposure while adding delay or reducing task performance [289]. Privacy in continuous interaction therefore concerns not only stored media but also what persistent internal representations reveal about the user.

Continuous sensing can improve contextual understanding, but it also requires the system to track where information comes from, whether it remains relevant, and how it may be used, including permission checks. Evaluation therefore needs to assess both whether observations and inferences are used for authorized purposes and whether the system intervenes at appropriate times. These requirements for handling information and actions must also be evaluated.

Existing benchmarks remain fragmented: full-duplex evaluations focus on turn-taking and spoken repair, streaming benchmarks evaluate online perception, long-term memory benchmarks test retrieval and consistency across extended histories, and multimodal safety benchmarks assess jailbreak, refusal, or privacy risks [120,138,139,261,337–339]. First-token latency cannot reveal when useful information becomes available; final-answer accuracy cannot reveal whether stale or misattributed evidence was used; and final refusal cannot reveal whether sensitive information was already transmitted or stored. Evaluate response timing, reasoning quality, evidence validity, state consistency, privacy exposure, and tool or memory actions together. Event-level records can show their trade-offs more clearly than a single aggregate score.

9. Design Principles and Open Problems

Native real-time omni models must coordinate perception, reasoning, generation, and action as new information arrives. Remaining challenges concern maintaining these capabilities jointly within latency and resource limits [25,119,256,342]. We organize the research agenda around computation at different time scales; persistent, proactive, and social interaction; and safe, efficient operation under changing deployment conditions. Figure 15 summarizes these directions.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

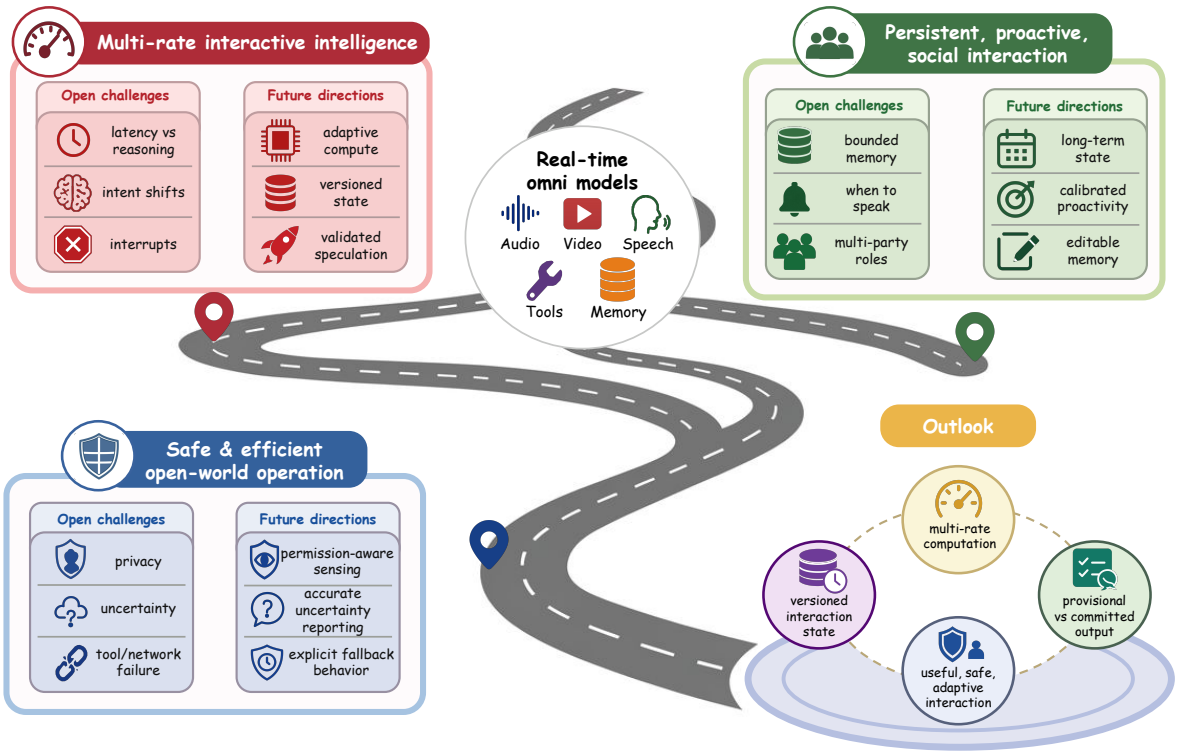


Figure 15. Open problems and outlook. Three directions for native real-time omni models: coordinating interaction and reasoning across different computational rates; supporting persistent, proactive interaction that respects participants' roles, intentions, and preferences; and ensuring safe and efficient operation in open-world settings. Each direction connects current challenges with corresponding research priorities.

9.1. Multi-Rate Interactive Intelligence

Immediate interaction and extended reasoning have different timing requirements. Turn-taking, interruption handling, and short acknowledgments require prompt responses, whereas retrieval, verification, planning, and tool execution may take longer. A system can coordinate multiple models or internal streams, each with different compute, context, and output constraints [25–27,167,342].

Recent streaming systems suggest that this coordination is not a single architectural pattern but a family of mechanisms. *Asynchronous parallelism* runs foreground interaction and background computation concurrently, as in StreamMind, which separates latency-critical interaction from persistent multimodal recall and longer-horizon processing [189]. *Conditional computation* invokes deeper reasoning only when it is needed: R3-Streaming compresses memory, assesses response readiness, and routes only difficult queries toward stronger reasoning [268]. *Perception–reasoning overlap* runs reasoning while new evidence arrives, as in Video Streaming Thinking, which processes incoming clips during video playback [267]. *Multi-rate state evolution* consolidates memory on a slower timescale than immediate perception, as in LatentStream's progressive latent memory [265]. *Decoupled interaction timing* separates content preparation from floor-taking, as in contextual preface generation paired with a Voice Activity Projection onset decision [283]. These mechanisms must share a consistent interaction state even when their inputs come from different task versions.

Background work may begin from an incomplete request and finish after the task changes. Associating each operation with its originating task version allows returned results to be checked against the current request, outdated work to be canceled where possible, and reusable intermediate results to be retained. Speculative retrieval and tool use can reduce delay when the relevant intent stabilizes sufficiently early [178,179]. Recovery also depends on what the operation has done: computation can often be discarded or repeated, whereas an external action that cannot be undone may require a compensating action. Coordinating this background work with ongoing interaction creates additional scheduling demands. Frequent input updates, persistent state, asynchronous results, and cancellation

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

must be handled alongside decoding. Disaggregated execution [298], latency-aware serving [39], and interaction-aware scheduling [281] provide mechanisms for allocating computation. When omni systems generate visual output, component-level acceleration can complement scheduling: rectified-flow auto-coding and optimized flow inference reduce generation cost [343,344], although integrating these methods with real-time interaction scheduling remains open. Efficient scheduling must also account for the current task: even a quickly computed result may no longer be useful if it arrives after it is needed, relies on outdated evidence, or has been invalidated by a correction.

9.2. Persistent, Proactive, and Social Interaction

Persistent memory should distinguish current and superseded information, retain unresolved tasks, and record the evidence supporting stored conclusions. Streaming attention, compression, sparse prefill, and cache retrieval reduce computational growth [262,299,345,346]. Memory must also separate observations from inferences, preserve source and time, and support bounded storage, revision, and deletion [261,337].

Proactive responses should use current observations, task requirements, and user instructions. The system must decide whether an event warrants a response, whether the user is available, and whether waiting would be more helpful. Recent work has begun to model proactive response from streaming sensory context and to evaluate both response content and timing [118,119,256,286]. Proact-VL treats response triggering as part of the interactive policy over an ongoing video stream and evaluates timing and content jointly [288], while Instruct-FD and DuplexGen indicate that the appropriate proactive and turn-taking policy is role- and scenario-dependent and may need to be set by instruction rather than learned as a single fixed prior [238,291]. When intent is uncertain, the model should retain alternative interpretations and defer consequential actions until sufficient evidence is available. Evaluation should distinguish helpful interventions, missed opportunities, unnecessary interruptions, and appropriate decisions to wait.

These decisions become harder in multi-party and personalized settings. Speaker identity, addressee, conversational floor, and authority are related but distinct. A person may be speaking near the system without addressing it, and the person addressing the system may not be authorized to trigger an action. Existing full-duplex data and turn-taking models only begin to capture overlapping speakers, repair, and floor competition [210,284]. Personalization should distinguish temporary behavior from lasting preferences and allow users to inspect, correct, or delete stored information. Systems also need explicit boundaries between shared and private participant context. Systems such as Ex-Omni-2D generate synchronized visual output alongside personalized speech [188], introducing additional questions about attention and social timing. The central challenge is to maintain one coherent interaction while preserving the distinct roles, permissions, and histories of its participants.

9.3. Safe and Efficient Operation in Open-World Settings

Continuous perception requires safety checks throughout the interaction, including for incoming audio and video. Audio and video may contain bystander speech, private screens, or environmental details that are available to the model but not permissible to use. Systems must enforce permissions for capture, processing, transmission, storage, and action. Selective-hearing results show that attending to the intended speaker does not by itself prevent leakage from surrounding speech [339]. Analogous leakage arises from internal state: speaker identity can remain recoverable from the hidden states of full-duplex models and grow more linkable across the first conversational turns, and streaming anonymization trades latency and utility to reduce it [289]. A specific open problem is preserving useful context without retaining unnecessary biometric, identity, or other sensitive attributes. Permission checks also apply to visual evidence, retrieved records, and tool outputs. Safety must also govern action timing. A provisional interpretation may be sufficient to start local computation, but not to send a message, disclose information, or control a device. SafeDreamer uses a learned world model to predict rewards and safety costs for candidate action sequences and incorporates cost constraints

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

into action selection [347]. Greater uncertainty or more consequential actions may require additional confirmation or oversight, including monitoring for high-impact operations [348].

Reliable systems need explicit fallback behavior when network, compute, sensors, or tools degrade. Real-time behavior depends on device memory, network delay, server load, sensor quality, and the availability of external tools. When these conditions deteriorate, the system may need to reduce visual sampling, postpone background reasoning, route computation between edge and cloud, or fall back to a more limited interaction mode. Edge-capable omni models and interaction-aware serving show that such choices can be incorporated into system design [256,281], but fallback behavior should communicate missing information or delays and preserve interaction state. Dropping a modality should lower confidence where that evidence matters, delaying a tool should not be hidden behind an unsupported answer, and a connectivity failure should not erase the interaction state. The open problem is to maintain causal input processing, accurate uncertainty reporting, interruption handling, and action safeguards when full capability is temporarily unavailable.

Training data and evaluations should include these failure conditions. Natural interaction data should include overlap, hesitation, correction, cancellation, silence, and cases in which a proactive response or tool call should be withheld [209,210,319]. Evaluations should replay timestamped streams without access to future input and compare systems under controlled resource and network conditions. They should record when intent becomes clear, when useful information is delivered, which evidence is used, and how tool and memory state change. The protocol should define how intent clarity and useful information are annotated. Recent benchmarks increasingly cover disfluent tool use and online audio-visual interaction [119,175], but they do not yet combine long-session state, multi-party authority, privacy boundaries, and deployment stress in one setting. The resulting systems should be compared on a joint frontier of answer quality, informative latency, repair, cost, and safety rather than reduced to a single aggregate score.

9.4. Outlook

Both integrated and modular architectures can support computation at different time scales. Beyond these architectures, an emerging direction is to pair generative omni models with compact, JEV-like decision modules for streaming and latency-critical control. PlayJev provides an early example by mapping raw visual observations and candidate actions directly to real-time action probabilities, although extending this approach from GUI games to open-ended multimodal interaction remains an open problem [349].

The key requirements are consistent task versions, revisable intermediate reasoning, memory with traceable sources, appropriate decisions to wait, and actions that remain valid after interruptions. Progress should be measured by preserved reasoning and grounding during continuous input, consistent state across sessions and participants, and adaptive computation with accurate uncertainty reporting and privacy protection. Evaluate usefulness, time to a useful answer, recovery from errors, resource cost, and safe action together. The goal is a system that remains useful and correct as evidence and user intent change during its response.

10. Conclusions

Real-time interaction changes the conditions under which multimodal intelligence must operate. A model must reason from incomplete input, decide when to respond, and remain receptive to evidence that may change its interpretation or invalidate an ongoing action. This survey examined how these demands connect multimodal representations, model architectures, training objectives, persistent state, and serving systems. Together, they motivate Interactive Omni Intelligence: the ability to sustain understanding, reasoning, and generation as the world and the conversation continue to unfold. Our conceptual framework comprises two coupled components: Omni Intelligence and Omni Interactivity. The Intelligence–Interactivity Trade-off characterizes how the demands of timely, continuous interaction constrain the multimodal capabilities a system can effectively deploy. Together,

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

the framework, design analysis, and evaluation agenda provide a roadmap for native real-time Interactive Omni Intelligence.

References

1. Caffagni, D.; Cocchi, F.; Barsellotti, L.; Moratelli, N.; Sarto, S.; Baraldi, L.; Cornia, M.; Cucchiara, R. The Revolution of Multimodal Large Language Models: A Survey. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 2024; pp. 13590–13618. https://doi.org/10.18653/v1/2024.findings-acl.807.
2. Bai, T.; Liang, H.; Wan, B.; Xu, Y.; Li, X.; Li, S.; Yang, L.; Li, B.; Wang, Y.; Cui, B.; et al. A survey of multimodal large language model from a data-centric perspective. *arXiv preprint arXiv:2405.16640* **2024**.
3. Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; Chen, E. A survey on multimodal large language models. *National Science Review* **2024**, *11*, nwae403. https://doi.org/10.1093/nsr/nwae403.
4. Yu, T.; Li, X.; Shen, Y.; Liu, Y.; Lourentzou, I. CoRe3D: Collaborative Reasoning as a Foundation for 3D Intelligence. *arXiv preprint arXiv:2512.12768* **2025**, [arXiv:cs.CV/2512.12768]. https://doi.org/10.48550/arXiv.2512.12768.
5. Ye, J.; Wang, Z.; Zhao, R.; Xie, S.; Zhu, J. ShapeLLM-Omni: A Native Multimodal LLM for 3D Generation and Understanding. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38, pp. 99057–99090. https://doi.org/10.52202/085713-2983.
6. Yu, T.; Li, X.; Shen, Y.; Susladkar, O.; Liu, Y.; Zhou, X.; Lourentzou, I. ELSA3D: Elastic Semantic Anchoring for Unified 3D Understanding and Generation, 2026, [arXiv:cs.CV/2607.06565].
7. Yu, T.; Shah, V.; Wahed, M.; Shen, Y.; Nguyen, K.A.; Lourentzou, I. Part^{2}GS: Part-aware Modeling of Articulated Objects using 3D Gaussian Splatting. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2026, pp. 18913–18923.
8. Yu, T.; Li, X.; Wahed, M.; Xiong, J.; Shen, Y.; Shen, Y.; Lourentzou, I. DreamPartGen: Semantically Grounded Part-Level 3D Generation via Collaborative Latent Denoising, 2026, [arXiv:cs.CV/2603.19216].
9. Cao, W.; Luo, C.; Zhang, B.; Nießner, M.; Tang, J. Motion2VecSets: 4D Latent Vector Set Diffusion for Non-rigid Shape Reconstruction and Tracking. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 20496–20506. https://doi.org/10.1109/CVPR52733.2024.01937.
10. Tang, J.; Cao, W.; Zhang, B.; Luo, C.; Liu, Y.; Nießner, M. Motion2VecSets: Non-Rigid Shape Reconstruction and Tracking with 4D Latent Set Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2026**, *48*, 9519–9536. https://doi.org/10.1109/TPAMI.2026.3680779.
11. Gemini Team.; Anil, R.; Borgeaud, S.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; Millican, K.; et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2312.11805* **2023**, [arXiv:cs.CL/2312.11805]. https://doi.org/10.48550/arXiv.2312.11805.
12. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* **2023**.
13. OpenAI. GPT-4o System Card. *arXiv preprint arXiv:2410.21276* **2024**, [arXiv:cs.CL/2410.21276]. https://doi.org/10.48550/arXiv.2410.21276.
14. Wu, S.; Fei, H.; Qu, L.; Ji, W.; Chua, T.S. NExT-GPT: Any-to-Any Multimodal LLM. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning. PMLR, 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 53366–53397.
15. Nguyen, T.A.; Kharitonov, E.; Copet, J.; Adi, Y.; Hsu, W.N.; Elkahky, A.; Tomasello, P.; Algayres, R.; Sagot, B.; Mohamed, A.; et al. Generative Spoken Dialogue Language Modeling. *Transactions of the Association for Computational Linguistics* **2023**, *11*, 250–266. https://doi.org/10.1162/tacl_a_00545.
16. Ekstedt, E.; Skantze, G. Voice Activity Projection: Self-Supervised Learning of Turn-Taking Events. In Proceedings of the Interspeech 2022, 2022, pp. 5190–5194. https://doi.org/10.21437/Interspeech.2022-10955.
17. Inoue, K.; Jiang, B.; Ekstedt, E.; Kawahara, T.; Skantze, G. Real-time and Continuous Turn-taking Prediction Using Voice Activity Projection. *arXiv preprint arXiv:2401.04868* **2024**, [arXiv:cs.CL/2401.04868]. https://doi.org/10.48550/arXiv.2401.04868.
18. Radford, A.; Kim, J.W.; Xu, T.; Brockman, G.; McLeavey, C.; Sutskever, I. Robust Speech Recognition via Large-Scale Weak Supervision. In Proceedings of the Proceedings of the 40th International Conference on Machine Learning. PMLR, 2023, Vol. 202, *Proceedings of Machine Learning Research*, pp. 28492–28518.
19. Ren, Y.; Hu, C.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; Liu, T.Y. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. In Proceedings of the International Conference on Learning Representations (ICLR), 2021.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

20. Kim, J.; Kong, J.; Son, J. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning. PMLR, 2021, Vol. 139, *Proceedings of Machine Learning Research*, pp. 5530–5540.
21. Défossez, A.; Mazaré, L.; Orsini, M.; Royer, A.; Pérez, P.; Jégou, H.; Grave, E.; Zeghidour, N. Moshi: a speech-text foundation model for real-time dialogue. *CoRR* **2024**, *abs/2410.00037*, [2410.00037]. https://doi.org/10.48550/ARXIV.2410.00037.
22. Xu, J.; Guo, Z.; He, J.; Hu, H.; He, T.; Bai, S.; Chen, K.; Wang, J.; Fan, Y.; Dang, K.; et al. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215* **2025**, [arXiv:cs.CL/2503.20215]. https://doi.org/10.48550/arXiv.2503.20215.
23. Xu, J.; Guo, Z.; Hu, H.; Chu, Y.; Wang, X.; He, J.; Wang, Y.; Shi, X.; He, T.; Zhu, X.; et al. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765* **2025**, [arXiv:cs.CL/2509.17765]. https://doi.org/10.48550/arXiv.2509.17765.
24. Qwen Team. Qwen3.5-Omni Technical Report. *arXiv preprint arXiv:2604.15804* **2026**, [arXiv:cs.CL/2604.15804]. https://doi.org/10.48550/arXiv.2604.15804.
25. Huang, M.; Zhang, L.; Yu, X.; Shi, L.; Ma, Z.; Xu, J.; Gao, J.; Hao, J.; He, R.; Liu, J. DuplexOmni: Real-Time Listening, Seeing, Thinking, and Speaking for Full-Duplex Interaction, 2026, [arXiv:cs.HC/2606.09186].
26. Chien, C.M.; Orsini, M.; Kharitonov, E.; Zeghidour, N.; Livescu, K.; Défossez, A. MoshiRAG: Asynchronous Knowledge Retrieval for Full-Duplex Speech Language Models. In Proceedings of the Proceedings of the 43rd International Conference on Machine Learning, 2026.
27. Arora, S.; Khan, H.; Sun, K.; Dong, X.L.; Choudhary, S.; Moon, S.; Zhang, X.; Sagar, A.; Appini, S.T.; Patnaik, K.; et al. Stream RAG: Instant and Accurate Spoken Dialogue Systems with Streaming Tool Usage. In Proceedings of the International Conference on Machine Learning (ICML), 2026.
28. Yan, R.; Li, X.; Chen, W.; Niu, Z.; Yang, C.; Ma, Z.; Yu, K.; Chen, X. URO-Bench: Towards Comprehensive Evaluation for End-to-End Spoken Dialogue Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, 2025; pp. 17211–17242. https://doi.org/10.18653/v1/2025.findings-emnlp.933.
29. Cuervo, S.; Seto, S.; de Seyssel, M.; Bai, R.H.; Gu, Z.; Likhomanenko, T.; Jaitly, N.; Aldeneh, Z. Closing the Gap Between Text and Speech Understanding in LLMs. In Proceedings of the The Fourteenth International Conference on Learning Representations (ICLR), 2026.
30. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.H.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2022, Vol. 35, pp. 24824–24837. https://doi.org/10.52202/068431-1800.
31. Levinson, S.C. Turn-Taking in Human Communication–Origins and Implications for Language Processing. *Trends in Cognitive Sciences* **2016**, *20*, 6–14. https://doi.org/10.1016/j.tics.2015.10.010.
32. Skantze, G.; Irfan, B. Applying General Turn-Taking Models to Conversational Human-Robot Interaction. In Proceedings of the 2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2025, pp. 859–868. https://doi.org/10.1109/HRI61500.2025.10973958.
33. Skantze, G. Turn-taking in Conversational Systems and Human-Robot Interaction: A Review. *Computer Speech & Language* **2021**, *67*, 101178. https://doi.org/10.1016/j.csl.2020.101178.
34. Ji, S.; Chen, Y.; Fang, M.; Zuo, J.; Lu, J.; Wang, H.; Jiang, Z.; Zhou, L.; Liu, S.; Cheng, X.; et al. WavChat: A Survey of Spoken Dialogue Models. *arXiv preprint arXiv:2411.13577* **2024**, [arXiv:cs.CL/2411.13577]. https://doi.org/10.48550/arXiv.2411.13577.
35. Arora, S.; Chang, K.W.; Chien, C.M.; Peng, Y.; Wu, H.; Adi, Y.; Dupoux, E.; Lee, H.y.; Livescu, K.; Watanabe, S. On the Landscape of Spoken Language Models: A Comprehensive Survey. *Transactions on Machine Learning Research* **2025**.
36. Chen, Y.; Yu, H. From Turn-Taking to Synchronous Dialogue: A Survey of Full-Duplex Spoken Language Models. *arXiv preprint arXiv:2509.14515* **2025**, [arXiv:cs.CL/2509.14515].
37. Xie, Z.; Wu, C. Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming. *arXiv preprint arXiv:2408.16725* **2024**, [arXiv:cs.AI/2408.16725]. https://doi.org/10.48550/arXiv.2408.16725.
38. Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C.H.; Gonzalez, J.E.; Zhang, H.; Stoica, I. Efficient Memory Management for Large Language Model Serving with PagedAttention. In Proceedings of the Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles. Association for Computing Machinery, 2023. https://doi.org/10.1145/3600006.3613165.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

39. Agrawal, A.; Kedia, N.; Panwar, A.; Mohan, J.; Kwatra, N.; Gulavani, B.; Tumanov, A.; Ramjee, R. Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve. In Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24), Santa Clara, CA, 2024; pp. 117–134.
40. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A Survey on Large Language Model Based Autonomous Agents. *Frontiers of Computer Science* **2024**, *18*, 186345. https://doi.org/10.1007/s11704-024-40231-1.
41. Wei, T.; Li, T.W.; Liu, Z.; Ning, X.; Yang, Z.; Zou, J.; Zeng, Z.; Qiu, R.; Lin, X.; Fu, D.; et al. A Survey of Agentic Reasoning for Large Language Models: Towards Recursively Self-Improving and Collective Agents. *arXiv preprint arXiv:2601.12538* **2026**, [arXiv:cs.AI/2601.12538]. https://doi.org/10.48550/arXiv.2601.12538.
42. Lin, T.E.; Wu, Y.; Huang, F.; Si, L.; Sun, J.; Li, Y. Duplex Conversation: Towards Human-Like Interaction in Spoken Dialogue Systems. In Proceedings of the Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2022, KDD '22, pp. 3299–3308. https://doi.org/10.1145/3534678.3539209.
43. Wang, X.; Li, Y.; Fu, C.; Zhang, Y.; Shen, Y.; Xie, L.; Li, K.; Sun, X.; Ma, L. Freeze-Omni: A Smart and Low Latency Speech-to-Speech Dialogue Model with Frozen LLM. In Proceedings of the Proceedings of the 42nd International Conference on Machine Learning. PMLR, 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 63345–63354.
44. Zhai, C. Towards a General Intelligent Information Agent: Framework, Models, and Evaluation. In Proceedings of the Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2026, pp. 2780–2789. https://doi.org/10.1145/3805712.3808543.
45. Wang, Y.; Wang, Y.; Yue, Z.; Zeng, H.; Wang, Y.; Lourentzou, I.; Tu, Z.; Chu, X.; McAuley, J. Fasa: Frequency-aware sparse attention. In Proceedings of the International Conference on Learning Representations (ICLR), 2026.
46. Leviathan, Y.; Kalman, M.; Matias, Y. Fast Inference from Transformers via Speculative Decoding. In Proceedings of the Proceedings of the 40th International Conference on Machine Learning. PMLR, 2023, Vol. 202, *Proceedings of Machine Learning Research*, pp. 19274–19286.
47. Graves, A. Sequence Transduction with Recurrent Neural Networks. In Proceedings of the ICML 2012 Workshop on Representation Learning, 2012, [arXiv:cs.NE/1211.3711]. arXiv:1211.3711.
48. Veluri, B.; Peloquin, B.N.; Yu, B.; Gong, H.; Gollakota, S. Beyond Turn-Based Interfaces: Synchronous LLMs as Full-Duplex Dialogue Agents. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024; pp. 21390–21402. https://doi.org/10.18653/v1/2024.emnlp-main.1192.
49. Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191* **2024**, [arXiv:cs.CV/2409.12191]. https://doi.org/10.48550/arXiv.2409.12191.
50. Fang, Q.; Guo, S.; Zhou, Y.; Ma, Z.; Zhang, S.; Feng, Y. LLaMA-Omni: Seamless Speech Interaction with Large Language Models. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
51. Fang, Q.; Zhou, Y.; Guo, S.; Zhang, S.; Feng, Y. LLaMA-Omni 2: LLM-Based Real-Time Spoken Chatbot with Autoregressive Streaming Speech Synthesis. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 18617–18629.
52. Zhan, J.; Dai, J.; Ye, J.; Zhou, Y.; Zhang, D.; Liu, Z.; Zhang, X.; Yuan, R.; Zhang, G.; Li, L.; et al. AnyGPT: Unified Multimodal LLM with Discrete Sequence Modeling. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 2024; pp. 9637–9662. https://doi.org/10.18653/v1/2024.acl-long.521.
53. Xie, Z.; Wu, C. Mini-Omni2: Towards Open-Source GPT-4o with Vision, Speech and Duplex Capabilities. *arXiv preprint arXiv:2410.11190* **2024**, [arXiv:eess.AS/2410.11190]. https://doi.org/10.48550/arXiv.2410.11190.
54. Fu, C.; Lin, H.; Long, Z.; Shen, Y.; Dai, Y.; Zhao, M.; Zhang, Y.F.; Dong, S.; Li, Y.; Wang, X.; et al. VITA: Towards Open-Source Interactive Omni Multimodal LLM, 2024, [arXiv:cs.CV/2408.05211]. https://doi.org/10.48550/arXiv.2408.05211.
55. Gong, J. MiniMind-O Technical Report: An Open Small-Scale Speech-Native Omni Model. *arXiv preprint arXiv:2605.03937* **2026**, [arXiv:cs.SD/2605.03937]. Code: https://github.com/jingyaogong/minimind-o.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

56. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems, 2020, Vol. 33, pp. 9459–9474.
57. Chiang, C.H.; Wang, X.; Li, L.; Lin, C.C.; Lin, K.; Liu, S.; Wang, Z.; Yang, Z.; Lee, H.y.; Wang, L. STITCH: Simultaneous Thinking and Talking with Chunked Reasoning for Spoken Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), 2026, [arXiv:cs.CL/2507.15375].
58. Yao, Y.; Yu, N.; Li, X.; Jiang, X.; Fang, X.; Ma, W.; Meng, X.; Li, J.; Sun, A.; Wang, Y. EgoMem: Lifelong Memory Agent for Full-Duplex Omnimodal Models. *arXiv preprint arXiv:2509.11914* **2025**, [arXiv:cs.AI/2509.11914]. https://doi.org/10.48550/arXiv.2509.11914.
59. Google. Gemini Live. Google Gemini product page, 2024. Gemini Live began rolling out on August 13, 2024.
60. Thinking Machines Lab. Interaction Models: A Scalable Approach to Human–AI Collaboration. Thinking Machines Lab, Connectionism, 2026. Accessed September 20, 2026, https://doi.org/10.64434/tml.20260511.
61. OpenAI. GPT-Realtime-2 Model. https://developers.openai.com/api/docs/models/gpt-realtime-2, 2026. OpenAI API Documentation, accessed September 7, 2026.
62. Alibaba Cloud. Qwen3.5-Omni-Realtime: Real-Time Omni-Modal Interaction. Alibaba Cloud Model Studio Documentation, 2026.
63. ByteDance Seed. SeedRealtime Audio-Visual Full-Duplex LLM Released: Toward Omni-Modal Natural Interaction. ByteDance Seed Blog, 2026.
64. Guo, Y.; Li, Z.; Wang, H.; Li, B.; Shao, C.; Zhang, H.; Du, C.; Chen, X.; Liu, S.; Yu, K. Recent Advances in Discrete Speech Tokens: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2026**, *48*, 4184–4204. https://doi.org/10.1109/TPAMI.2025.3643619.
65. Mousavi, P.; Maimon, G.; Moumen, A.; Petermann, D.; Shi, J.; et al. Discrete Audio Tokens: More Than a Survey! *Transactions on Machine Learning Research* **2025**.
66. Bai, H.; Likhomanenko, T.; Zhang, R.; Gu, Z.; Aldeneh, Z.; Jaitly, N. dMel: Speech Tokenization Made Simple. *arXiv preprint arXiv:2407.15835* **2024**.
67. Meng, L.; Zhou, L.; Liu, S.; Chen, S.; Han, B.; Hu, S.; Liu, Y.; Li, J.; Zhao, S.; Wu, X.; et al. Autoregressive Speech Synthesis without Vector Quantization. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 1287–1300. https://doi.org/10.18653/v1/2025.acl-long.65.
68. van den Oord, A.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748* **2018**, [arXiv:cs.LG/1807.03748]. https://doi.org/10.48550/arXiv.1807.03748.
69. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In Proceedings of the Advances in Neural Information Processing Systems, 2020, Vol. 33, pp. 12449–12460.
70. Hsu, W.N.; Bolte, B.; Tsai, Y.H.H.; Lakhotia, K.; Salakhutdinov, R.; Mohamed, A. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2021**, *29*, 3451–3460. https://doi.org/10.1109/TASLP.2021.3122291.
71. Chung, Y.A.; Zhang, Y.; Han, W.; Chiu, C.C.; Qin, J.; Pang, R.; Wu, Y. w2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2021, pp. 244–250. https://doi.org/10.1109/ASRU51503.2021.9688253.
72. Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing* **2022**, *16*, 1505–1518. https://doi.org/10.1109/jstsp.2022.3188113.
73. Mohamed, A.; yi Lee, H.; Borgholt, L.; Havtorn, J.D.; Edin, J.; Igel, C.; Kirchhoff, K.; Li, S.W.; Livescu, K.; Maaløe, L.; et al. Self-Supervised Speech Representation Learning: A Review. *IEEE Journal of Selected Topics in Signal Processing* **2022**, *16*, 1179–1210. https://doi.org/10.1109/JSTSP.2022.3207050.
74. wen Yang, S.; Chi, P.H.; Chuang, Y.S.; Lai, C.I.J.; Lakhotia, K.; Lin, Y.Y.; et al. SUPERB: Speech processing Universal PERformance Benchmark. In Proceedings of the Interspeech, 2021. https://doi.org/10.21437/Interspeech.2021-1775.
75. Baevski, A.; Schneider, S.; Auli, M. vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations. In Proceedings of the International Conference on Learning Representations, 2020.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

76. Lakhotia, K.; Kharitonov, E.; Hsu, W.N.; Adi, Y.; Polyak, A.; Bolte, B.; Nguyen, T.A.; Copet, J.; Baevski, A.; Mohamed, A.; et al. On Generative Spoken Language Modeling from Raw Audio. *Transactions of the Association for Computational Linguistics* **2021**, *9*, 1336–1354. https://doi.org/10.1162/tacl_a_00430.
77. Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; Tagliasacchi, M. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2022**, *30*, 495–507. https://doi.org/10.1109/TASLP.2021.3129994.
78. Défossez, A.; Copet, J.; Synnaeve, G.; Adi, Y. High Fidelity Neural Audio Compression. *Transactions on Machine Learning Research* **2023**.
79. Kumar, R.; Seetharaman, P.; Luebs, A.; Kumar, I.; Kumar, K. High-Fidelity Audio Compression with Improved RVQGAN. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). Curran Associates, Inc., 2023. https://doi.org/10.52202/075280-1214.
80. Yang, D.; Liu, S.; Huang, R.; Tian, J.; Weng, C.; Zou, Y. HiFi-Codec: Group-Residual Vector Quantization for High Fidelity Audio Codec. *arXiv preprint arXiv:2305.02765* **2023**, [arXiv:cs.SD/2305.02765]. https://doi.org/10.48550/arXiv.2305.02765.
81. Siuzdak, H. Vocos: Closing the Gap between Time-Domain and Fourier-Based Neural Vocoders for High-Quality Audio Synthesis. In Proceedings of the International Conference on Learning Representations (ICLR), 2024.
82. Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; Roblek, D.; Teboul, O.; Grangier, D.; Tagliasacchi, M.; et al. AudioLM: A Language Modeling Approach to Audio Generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2023**, *31*, 2523–2533. https://doi.org/10.1109/TASLP.2023.3288409.
83. Chen, S.; Wang, C.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; et al. Neural Codec Language Models Are Zero-Shot Text-to-Speech Synthesizers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2025**, *33*, 705–718. https://doi.org/10.1109/TASLPRO.2025.3530270.
84. Zhang, X.; Zhang, D.; Li, S.; Zhou, Y.; Qiu, X. SpeechTokenizer: Unified Speech Tokenizer for Speech Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), 2024, [arXiv:cs.CL/2308.16692].
85. Ye, Z.; Sun, P.; Lei, J.; Lin, H.; Tan, X.; Dai, Z.; Kong, Q.; Chen, J.; Pan, J.; Liu, Q.; et al. Codec Does Matter: Exploring the Semantic Shortcoming of Codec for Audio Language Model. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2025, Vol. 39, pp. 25697–25705. https://doi.org/10.1609/aaai.v39i24.34761.
86. Ju, Z.; Wang, Y.; Shen, K.; Tan, X.; Xin, D.; Yang, D.; Liu, E.; Leng, Y.; Song, K.; Tang, S.; et al. NaturalSpeech 3: Zero-Shot Speech Synthesis with Factorized Codec and Diffusion Models. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning. PMLR, 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 22605–22623.
87. Liu, H.; Xu, X.; Yuan, Y.; Wu, M.; Wang, W.; Plumbley, M.D. SemantiCodec: An Ultra Low Bitrate Semantic Audio Codec for General Sound. *IEEE Journal of Selected Topics in Signal Processing* **2024**, *18*, 1448–1461. https://doi.org/10.1109/JSTSP.2024.3506286.
88. Ji, S.; Jiang, Z.; Wang, W.; Chen, Y.; Fang, M.; Zuo, J.; Yang, Q.; Cheng, X.; Wang, Z.; Li, R.; et al. WavTokenizer: An Efficient Acoustic Discrete Codec Tokenizer for Audio Language Modeling. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
89. Xin, D.; Tan, X.; Takamichi, S.; Saruwatari, H. BigCodec: Pushing the Limits of Low-Bitrate Neural Speech Codec. *arXiv preprint arXiv:2409.05377* **2024**, [arXiv:eess.AS/2409.05377]. https://doi.org/10.48550/arXiv.2409.05377.
90. Wu, H.; Kanda, N.; Eskimez, S.E.; Li, J. TS3-Codec: Transformer-Based Simple Streaming Single Codec. In Proceedings of the Interspeech 2025, 2025, pp. 604–608. https://doi.org/10.21437/Interspeech.2025-921.
91. Siuzdak, H.; Grötschla, F.; Lanzendörfer, L.A. SNAC: Multi-Scale Neural Audio Codec. In Proceedings of the Audio Imagination: NeurIPS 2024 Workshop on AI-Driven Speech, Music, and Sound Generation, 2024.
92. Zeng, A.; Du, Z.; Liu, M.; Wang, K.; Jiang, S.; Zhao, L.; Dong, Y.; Tang, J. GLM-4-Voice: Towards Intelligent and Human-Like End-to-End Spoken Chatbot, 2024, [arXiv:cs.CL/2412.02612].
93. Wang, D.; Cui, M.; Yang, D.; Chen, X.; Meng, H. A Comparative Study of Discrete Speech Tokens for Semantic-Related Tasks with Large Language Models. *arXiv preprint arXiv:2411.08742* **2024**, [arXiv:cs.CL/2411.08742]. https://doi.org/10.48550/arXiv.2411.08742.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

94. Wang, D.; Li, J.; Cui, M.; Yang, D.; Chen, X.; Meng, H.M. Speech Discrete Tokens or Continuous Features? A Comparative Analysis for Spoken Language Understanding in SpeechLLMs. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 2025; pp. 24913–24924. https://doi.org/10.18653/v1/2025.emnlp-main.1266.
95. Wu, H.; Chen, X.; Lin, Y.C.; Chang, K.; Du, J.; Lu, K.H.; Liu, A.H.; Chung, H.L.; Wu, Y.K.; Yang, D.; et al. Codec-SUPERB @ SLT 2024: A Lightweight Benchmark for Neural Audio Codec Models. In Proceedings of the 2024 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2024, pp. 570–577. https://doi.org/10.1109/SLT61566.2024.10832364.
96. Mousavi, P.; Duret, J.; Petermann, D.; Ploujnikov, A.; Della Libera, L.; Kuznetsova, A.; Subakan, C.; Ravanelli, M. DASB: Discrete Audio and Speech Benchmark. *Transactions on Machine Learning Research* **2026**.
97. Vashishth, S.; Singh, H.; Bharadwaj, S.; Ganapathy, S.; Asawaroengchai, C.; Audhkhasi, K.; Rosenberg, A.; Bapna, A.; Ramabhadran, B. STAB: Speech Tokenizer Assessment Benchmark. *arXiv preprint arXiv:2409.02384* **2024**, [arXiv:cs.CL/2409.02384]. https://doi.org/10.48550/arXiv.2409.02384.
98. Tang, Y.; Bi, J.; Xu, S.; Song, L.; Liang, S.; Wang, T.; Zhang, D.; An, J.; Lin, J.; Zhu, R.; et al. Video Understanding with Large Language Models: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology* **2026**, *36*, 1355–1376. https://doi.org/10.1109/TCSVT.2025.3566695.
99. Cao, W.; Zhang, H.; Tian, F.; Wu, Y.; Li, Y.; Wang, S.; Yu, N.; Liu, Y. FreeOrbit4D: Training-Free Arbitrary Camera Redirection for Monocular Videos via Foreground-Complete 4D Reconstruction. In Proceedings of the Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers, 2026, pp. 1–12. https://doi.org/10.1145/3799902.3811122.
100. Yang, Z.; Zhang, K.; Liu, Q.; Liu, T.; Ying, L.; Xue, D.; Hou, Q.; Qian, S.; Xu, C. Towards Online Interactors: A Comprehensive Survey on Streaming Video Understanding. *preprints202606.1674.v1* **2026**. https://doi.org/10.20944/preprints202606.1674.v1.
101. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, 2021.
102. Alayrac, J.B.; Donahue, J.; Luc, P.; Miech, A.; et al. Flamingo: A Visual Language Model for Few-Shot Learning. In Proceedings of the Advances in Neural Information Processing Systems, 2022, Vol. 35, pp. 23716–23736.
103. Jaegle, A.; Gimeno, F.; Brock, A.; Vinyals, O.; Zisserman, A.; Carreira, J. Perceiver: General Perception with Iterative Attention. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning. PMLR, 2021, Vol. 139, *Proceedings of Machine Learning Research*, pp. 4651–4664.
104. Jaegle, A.; Borgeaud, S.; Alayrac, J.B.; Doersch, C.; Ionescu, C.; Ding, D.; Koppula, S.; Zoran, D.; Brock, A.; Shelhamer, E.; et al. Perceiver IO: A General Architecture for Structured Inputs and Outputs. In Proceedings of the International Conference on Learning Representations, 2022.
105. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models. In Proceedings of the International Conference on Machine Learning (ICML), 2023.
106. Li, Y.; Wang, C.; Jia, J. LLaMA-VID: An Image Is Worth 2 Tokens in Large Language Models. In Proceedings of the Computer Vision – ECCV 2024. Springer, 2024, Vol. 15104, *Lecture Notes in Computer Science*, pp. 323–340. https://doi.org/10.1007/978-3-031-72952-2_19.
107. Tang, X.; Qiu, J.; Xie, L.; Tian, Y.; Jiao, J.; Ye, Q. Adaptive Keyframe Sampling for Long Video Understanding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025, pp. 29118–29128.
108. Wei, X.; Liu, X.; Zang, Y.; Dong, X.; Zhang, P.; Cao, Y.; Tong, J.; Duan, H.; Guo, Q.; Wang, J.; et al. VideoRoPE: What Makes for Good Video Rotary Position Embedding? In Proceedings of the Proceedings of the 42nd International Conference on Machine Learning. PMLR, 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 66118–66136.
109. Xu, R.; Xiao, G.; Chen, Y.; He, L.; Peng, K.; Lu, Y.; Han, S. StreamingVLM: Real-Time Understanding for Infinite Video Streams. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
110. Zhang, H.; Wang, Y.; Tang, Y.; Liu, Y.; Feng, J.; Jin, X. Flash-VStream: Efficient Real-Time Understanding for Long Video Streams. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, pp. 21059–21069.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

111. Yao, L.; Li, Y.; Wei, Y.; Li, L.; Ren, S.; Liu, Y.; Ouyang, K.; Wang, L.; Li, S.; Li, S.; et al. TimeChat-Online: 80% Visual Tokens Are Naturally Redundant in Streaming Videos. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia. ACM, 2025, pp. 10807–10816. https://doi.org/10.1145/3746027.3754839.
112. Bolya, D.; Fu, C.Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; Hoffman, J. Token Merging: Your ViT But Faster. In Proceedings of the International Conference on Learning Representations (ICLR), 2023.
113. Chen, J.; Lv, Z.; Wu, S.; Lin, K.Q.; Song, C.; Gao, D.; Liu, J.W.; Gao, Z.; Mao, D.; Shou, M.Z. VideoLLM-online: Online Video Large Language Model for Streaming Video. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 18407–18418. https://doi.org/10.1109/CVPR52733.2024.01742.
114. Wang, Y.; Meng, X.; Wang, Y.; Liang, J.; Wei, J.; Zhang, H.; Zhao, D. VideoLLM Knows When to Speak: Enhancing Time-Sensitive Video Comprehension with Video-Text Duet Interaction Format. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, 2025; pp. 6338–6359. https://doi.org/10.18653/v1/2025.findings-emnlp.336.
115. Qian, R.; Ding, S.; Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Lin, D.; Wang, J. Dispider: Enabling Video LLMs with Active Real-Time Interaction via Disentangled Perception, Decision, and Reaction. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025, pp. 24045–24055.
116. Wang, H.; Feng, B.; Lai, Z.; Xu, M.; Li, S.; Ge, W.; Dehghan, A.; Cao, M.; Huang, P. StreamBridge: Turning Your Offline Video Large Language Model into a Proactive Streaming Assistant. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38. https://doi.org/10.52202/085713-4406.
117. Yang, Z.; Zhang, K.; Hu, Y.; Wang, B.; Qian, S.; Wen, B.; Yang, F.; Gao, T.; Dong, W.; Xu, C. LiveStar: Live Streaming Assistant for Real-World Online Video Understanding. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38. https://doi.org/10.52202/085713-1050.
118. Wang, Y.; Meng, X.; Wang, Y.; Zhang, H.; Zhao, D. ProactiveVideoQA: A Comprehensive Benchmark Evaluating Proactive Interactions in Video Large Language Models. *CoRR* **2025**, *abs/2507.09313*, [2507.09313]. https://doi.org/10.48550/ARXIV.2507.09313.
119. Lu, X.; Li, X.; Wang, A.; Bo, Y.; Chen, J.; Li, Z.; Yang, N.; Liu, R.; Yang, X.; Hou, J.; et al. OmniInteract: Benchmarking Real-World Streaming Interaction for Real-Time Omnimodal Assistants. *arXiv preprint arXiv:2605.26485* **2026**. https://doi.org/10.48550/arXiv.2605.26485.
120. Lin, J.; Fang, Z.; Chen, C.; Cheng, H.; Wan, Z.; Luo, F.; Wang, Z.; Li, P.; Liu, Y.; Sun, M. StreamingBench: Assessing the Gap for MLLMs to Achieve Streaming Video Understanding. In Proceedings of the 2026 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2026, pp. 12147–12151. https://doi.org/10.1109/ICASSP55912.2026.11463959.
121. Castillo-López, G.; de Chalendar, G.; Semmar, N. A Survey of Recent Advances on Turn-taking Modeling in Spoken Dialogue Systems. In Proceedings of the Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology, Bilbao, Spain, 2025; pp. 254–271.
122. Sacks, H.; Schegloff, E.A.; Jefferson, G. A Simplest Systematics for the Organization of Turn-Taking for Conversation. *Language* **1974**, *50*, 696–735. https://doi.org/10.2307/412243.
123. Stivers, T.; Enfield, N.J.; Brown, P.; Englert, C.; Hayashi, M.; et al. Universals and Cultural Variation in Turn-Taking in Conversation. *Proceedings of the National Academy of Sciences* **2009**, *106*, 10587–10592. https://doi.org/10.1073/pnas.0903616106.
124. Levinson, S.C.; Torreira, F. Timing in Turn-Taking and Its Implications for Processing Models of Language. *Frontiers in Psychology* **2015**, *6*, 731. https://doi.org/10.3389/fpsyg.2015.00731.
125. Skantze, G. Towards a General, Continuous Model of Turn-Taking in Spoken Dialogue Using LSTM Recurrent Neural Networks. In Proceedings of the Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Saarbrücken, Germany, 2017; pp. 220–230. https://doi.org/10.18653/v1/W17-5527.
126. Roddy, M.; Skantze, G.; Harte, N. Multimodal Continuous Turn-Taking Prediction Using Multiscale RNNs. In Proceedings of the Proceedings of the 20th ACM International Conference on Multimodal Interaction. Association for Computing Machinery, 2018, pp. 186–190. https://doi.org/10.1145/3242969.3242997.
127. Ekstedt, E.; Skantze, G. TurnGPT: a Transformer-based Language Model for Predicting Turn-taking in Spoken Dialog. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020; Cohn, T.; He, Y.; Liu, Y., Eds., Online, 2020; pp. 2981–2990. https://doi.org/10.18653/v1/2020.findings-emnlp.268.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

128. Roddy, M.; Harte, N. Neural Generation of Dialogue Response Timings. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020; pp. 2442–2452. https://doi.org/10.18653/v1/2020.acl-main.221.
129. Inoue, K.; Jiang, B.; Ekstedt, E.; Kawahara, T.; Skantze, G. Multilingual Turn-taking Prediction Using Voice Activity Projection. In Proceedings of the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia, 2024; pp. 11873–11883.
130. Jiang, B.; Ekstedt, E.; Skantze, G. What Makes a Good Pause? Investigating the Turn-Holding Effects of Fillers. In Proceedings of the Proceedings of the 20th International Congress of Phonetic Sciences – ICPhS 2023; Skarnitzl, R.; Volín, J., Eds., Prague, Czech Republic, 2023; pp. 3512–3516.
131. Wang, J.; Chen, L.; Khare, A.; Raju, A.; Dheram, P.; He, D.; Wu, M.; Stolcke, A.; Ravichandran, V. Turn-Taking and Backchannel Prediction with Acoustic and Large Language Model Fusion. In Proceedings of the ICASSP 2024 – 2024 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2024, pp. 12121–12125. https://doi.org/10.1109/ICASSP48485.2024.10447196.
132. Inoue, K.; Lala, D.; Skantze, G.; Kawahara, T. Yeah, Un, Oh: Continuous and Real-time Backchannel Prediction with Fine-tuning of Voice Activity Projection. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Albuquerque, New Mexico, 2025; pp. 7171–7181. https://doi.org/10.18653/v1/2025.naacl-long.367.
133. Yu, W.; Wang, S.; Yang, X.; Chen, X.; Tian, X.; Zhang, J.; Sun, G.; Lu, L.; Wang, Y.; Zhang, C. SALMONN-omni: A Standalone Speech LLM without Codec Injection for Full-Duplex Conversation. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38, pp. 27885–27911. https://doi.org/10.52202/085713-0829.
134. Ma, Z.; Song, Y.; Du, C.; Cong, J.; Chen, Z.; Wang, Y.; Wang, Y.; Chen, X. Language Model Can Listen While Speaking. *Proceedings of the AAAI Conference on Artificial Intelligence* **2025**, *39*, 24831–24839. https://doi.org/10.1609/aaai.v39i23.34665.
135. Wang, Q.; Meng, Z.; Cui, W.; Zhang, Y.; Wu, P.; Wu, B.; King, I.; Chen, L.; Zhao, P. NTPP: Generative Speech Language Modeling for Dual-Channel Spoken Dialogue via Next-Token-Pair Prediction. In Proceedings of the Proceedings of the 42nd International Conference on Machine Learning. PMLR, 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 63912–63928.
136. Seide, F.; Shi, Y.; Doulaty, M.; Gaur, Y.; Jia, J.; Wu, C. Speech ReaLLM – Real-time Speech Recognition with Multimodal Language Models by Teaching the Flow of Time. In Proceedings of the Interspeech 2024, 2024, pp. 1900–1904. https://doi.org/10.21437/Interspeech.2024-571.
137. Lin, G.T.; Lian, J.; Li, T.; Wang, Q.; Anumanchipalli, G.; Liu, A.H.; Lee, H.Y. Full-Duplex-Bench: A Benchmark to Evaluate Full-Duplex Spoken Dialogue Models on Turn-Taking Capabilities. In Proceedings of the 2025 IEEE Automatic Speech Recognition and Understanding Workshop. IEEE, 2025, pp. 1–8. https://doi.org/10.1109/ASRU65441.2025.11433838.
138. Lin, G.T.; Kuan, S.Y.S.; Wang, Q.; Lian, J.; Li, T.; Watanabe, S.; Lee, H.Y. Full-Duplex-Bench v1.5: Evaluating Overlap Handling for Full-Duplex Speech Models. In Proceedings of the 2026 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2026, pp. 19447–19451. https://doi.org/10.1109/ICASSP55912.2026.11463576.
139. Lin, G.T.; Kuan, S.Y.S.; Shi, J.; Chang, K.W.; Arora, S.; Watanabe, S.; Lee, H.y. Full-Duplex-Bench-v2: A Multi-Turn Evaluation Framework for Duplex Dialogue Systems with an Automated Examiner. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), San Diego, California, United States, July 2026; pp. 27–36, [2510.07838]. https://doi.org/10.18653/v1/2026.acl-short.4.
140. Arora, S.; Lu, Z.; Chiu, C.C.; Pang, R.; Watanabe, S. Talking Turns: Benchmarking Audio Foundation Models on Turn-Taking Dynamics. In Proceedings of the International Conference on Learning Representations (ICLR), 2025.
141. Cui, W.; Yu, D.; Jiao, X.; Meng, Z.; Zhang, G.; Wang, Q.; Guo, S.Y.; King, I. Recent Advances in Speech Language Models: A Survey. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 13943–13970.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

142. Zhang, D.; Li, S.; Zhang, X.; Zhan, J.; Wang, P.; Zhou, Y.; Qiu, X. SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023. Association for Computational Linguistics, 2023, pp. 15757–15773. https://doi.org/10.18653/v1/2023.findings-emnlp.1055.
143. Nachmani, E.; Levkovitch, A.; Hirsch, R.; Salazar, J.; Asawaroengchai, C.; Mariooryad, S.; Rivlin, E.; Skerry-Ryan, R.J.; Tadmor Ramanovich, M. Spoken Question Answering and Speech Continuation Using Spectrogram-Powered LLM. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
144. Maiti, S.; Peng, Y.; Choi, S.; Jung, J.w.; Chang, X.; Watanabe, S. VoxtLM: Unified Decoder-Only Models for Consolidating Speech Recognition, Synthesis and Speech, Text Continuation Tasks. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2024, pp. 13326–13330. https://doi.org/10.1109/ICASSP48485.2024.10447112.
145. Kim, H.; Seo, S.; Jeong, K.; Kwon, O.; Kim, S.; Kim, J.; Lee, J.; Song, E.; Oh, M.; Ha, J.W.; et al. Paralinguistics-Aware Speech-Empowered Large Language Models for Natural Conversation. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 131072–131103. https://doi.org/10.52202/079017-4166.
146. Chou, J.C.; Chien, C.M.; Hsu, W.N.; Livescu, K.; Babu, A.; Conneau, A.; Baevski, A.; Auli, M. Toward Joint Language Modeling for Speech Units and Text. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 2023; pp. 6582–6593. https://doi.org/10.18653/v1/2023.findings-emnlp.438.
147. Nguyen, T.A.; Muller, B.; Yu, B.; Costa-jussa, M.R.; Elbayad, M.; Popuri, S.; Ropers, C.; Duquenne, P.A.; Algayres, R.; Mavlyutov, R.; et al. SpiRit-LM: Interleaved Spoken and Written Language Model. *Transactions of the Association for Computational Linguistics* **2025**, *13*, 30–52. https://doi.org/10.1162/tacl_a_00728.
148. Yang, Y.; Liu, S.; Li, J.; Wang, H.; Meng, L.; et al. Interleaved Speech-Text Language Models for Simple Streaming Text-to-Speech Synthesis. *arXiv preprint arXiv:2412.16102* **2024**, [arXiv:eess.AS/2412.16102]. https://doi.org/10.48550/arXiv.2412.16102.
149. Mitsui, K.; Mitsuda, K.; Wakatsuki, T.; Hono, Y.; Sawada, K. PSLM: Parallel Generation of Text and Speech with LLMs for Low-Latency Spoken Dialogue Systems. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, 2024; pp. 2692–2700. https://doi.org/10.18653/v1/2024.findings-emnlp.151.
150. Zeng, A.; Du, Z.; Liu, M.; Zhang, L.; Jiang, S.; Dong, Y.; Tang, J. Scaling Speech-Text Pre-Training with Synthetic Interleaved Data. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
151. Hassid, M.; Remez, T.; Nguyen, T.A.; Gat, I.; Conneau, A.; Kreuk, F.; Copet, J.; Défossez, A.; Synnaeve, G.; Dupoux, E.; et al. Textually Pretrained Speech Language Models. In Proceedings of the Advances in Neural Information Processing Systems, 2023, Vol. 36.
152. Maimon, G.; Hassid, M.; Roth, A.; Adi, Y. Scaling Analysis of Interleaved Speech-Text Language Models. In Proceedings of the The 2nd Conference on Language Modeling, 2025.
153. Cui, W.; Zhu, L.; Li, X.H.; Guo, Z.; Bai, H.; Hou, L.; King, I. TurnGuide: Enhancing Meaningful Full Duplex Spoken Interactions via Dynamic Turn-Level Text-Speech Interleaving. In Proceedings of the Interspeech, 2026.
154. Kharitonov, E.; Lee, A.; Polyak, A.; Adi, Y.; Copet, J.; Lakhotia, K.; Nguyen, T.A.; Riviere, M.; Mohamed, A.; Dupoux, E.; et al. Text-Free Prosody-Aware Generative Spoken Language Modeling. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 2022; pp. 8666–8681. https://doi.org/10.18653/v1/2022.acl-long.593.
155. Zhang, Q.; Cheng, L.; Deng, C.; Chen, Q.; Wang, W.; Zheng, S.; Liu, J.; Yu, H.; Tan, C.H.; Du, Z.; et al. OmniFlatten: An End-to-end GPT Model for Seamless Voice Conversation. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 14570–14580. https://doi.org/10.18653/v1/2025.acl-long.709.
156. Copet, J.; Kreuk, F.; Gat, I.; Remez, T.; Kant, D.; Synnaeve, G.; Adi, Y.; Defossez, A. Simple and Controllable Music Generation. In Proceedings of the Advances in Neural Information Processing Systems. Curran Associates, Inc., 2023, Vol. 36, pp. 47704–47720. https://doi.org/10.52202/075280-2066.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

157. Le Lan, G.; Nagaraja, V.; Chang, E.; Kant, D.; Ni, Z.; Shi, Y.; Iandola, F.; Chandra, V. Stack-and-Delay: A New Codebook Pattern for Music Generation. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024, pp. 796–800. https://doi.org/10.1109/ICASSP48485.2024.10447392.
158. Mialon, G.; Dessì, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Rozière, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; et al. Augmented Language Models: A Survey. *Transactions on Machine Learning Research* **2023**.
159. Qu, C.; Dai, S.; Wei, X.; Cai, H.; Wang, S.; Yin, D.; Xu, J.; Wen, J.R. Tool Learning with Large Language Models: A Survey. *Frontiers of Computer Science* **2025**, *19*, 198343. https://doi.org/10.1007/s11704-024-40678-2.
160. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* **2023**, [arXiv:cs.CL/2312.10997].
161. Ning, X.; Tieu, K.; Fu, D.; Wei, T.; Li, Z.; Bei, Y.; Zou, J.; Ai, M.; Liu, Z.; Li, T.W.; et al. Code as agent harness. *arXiv preprint arXiv:2605.18747* **2026**. https://doi.org/10.48550/arXiv.2605.18747.
162. Prabhavalkar, R.; Hori, T.; Sainath, T.N.; Schlüter, R.; Watanabe, S. End-to-End Speech Recognition: A Survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2024**, *32*, 325–351. https://doi.org/10.1109/TASLP.2023.3328283.
163. Shangguan, Y.; Knister, K.; He, Y.; McGraw, I.; Beaufays, F. Analyzing the Quality and Stability of a Streaming End-to-End On-Device Speech Recognizer. In Proceedings of the Proceedings of Interspeech 2020, 2020, pp. 591–595. https://doi.org/10.21437/Interspeech.2020-1194.
164. Zeghidour, N.; Kharitonov, E.; Orsini, M.; Volhejn, V.; de Marmiesse, G.; Grave, E.; Pérez, P.; Mazaré, L.; Défossez, A. Streaming Sequence-to-Sequence Learning with Delayed Streams Modeling. *arXiv preprint arXiv:2509.08753* **2025**, [2509.08753]. https://doi.org/10.48550/arXiv.2509.08753.
165. Chiang, C.H.; Wang, X.; Li, L.; Lin, C.C.; Lin, K.; Liu, S.; Wang, Z.; Yang, Z.; Lee, H.y.; Wang, L. SHANKS: Simultaneous Hearing and Thinking for Spoken Language Models. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, California, United States, 2026; pp. 8951–8972. https://doi.org/10.18653/v1/2026.acl-long.404.
166. Shih, Y.J.; Raj, D.; Wu, C.; Zhou, W.; Bong, S.; Gaur, Y.; Mahadeokar, J.; Kalinli, O.; Seltzer, M. Can Speech LLMs Think while Listening?, 2025, [arXiv:cs.CL/2510.07497].
167. Wu, D.; Zhang, T.; Li, Y.; Liu, H.; Chen, C.; Chng, E.S.; Bengio, Y. The Silent Thought: Modeling Internal Cognition in Full-Duplex Spoken Dialogue Models via Latent Reasoning, 2026, [arXiv:eess.AS/2603.17837].
168. Du, X.; Yan, Q.; Li, W.; Jiang, B.; Xiao, C.; Shu, H.; Chen, X. Thinking-While-Speaking: A Controlled, Interleaved Reasoning Method for Real-Time Speech Generation. *arXiv preprint arXiv:2605.20946* **2026**.
169. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
170. Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. WebGPT: Browser-Assisted Question-Answering with Human Feedback. *arXiv preprint arXiv:2112.09332* **2021**, [arXiv:cs.CL/2112.09332]. https://doi.org/10.48550/arXiv.2112.09332.
171. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; et al. Toolformer: Language Models Can Teach Themselves to Use Tools. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2023. https://doi.org/10.52202/075280-2997.
172. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-World APIs. In Proceedings of the International Conference on Learning Representations (ICLR), 2024.
173. Jain, D.; Shukla, H.; Rajeev, G.; Kulkarni, A.; Khatri, C.; Agarwal, S. VoiceAgentBench: Are Voice Assistants Ready for Agentic Tasks? *arXiv preprint arXiv:2510.07978* **2025**, [arXiv:cs.AI/2510.07978]. https://doi.org/10.48550/arXiv.2510.07978.
174. Laskar, M.T.R.; Fu, X.Y.; Sarfjoo, S.S.; McNamara, Q.; Robertson, J.; TN, S.B. From Text to Voice: A Reproducible and Verifiable Framework for Evaluating Tool Calling LLM Agents. *arXiv preprint arXiv:2605.15104* **2026**. https://doi.org/10.48550/arXiv.2605.15104.
175. Lin, G.; Chen, C.; Chen, Z.; Lee, H. Full-Duplex-Bench-v3: Benchmarking Tool Use for Full-Duplex Voice Agents Under Real-World Disfluency. *CoRR* **2026**, *abs/2604.04847*, [2604.04847]. https://doi.org/10.48550/ARXIV.2604.04847.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

176. Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active Retrieval Augmented Generation. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 2023; pp. 7969–7992. https://doi.org/10.18653/v1/2023.emnlp-main.495.
177. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the International Conference on Learning Representations (ICLR), 2024.
178. Hooper, C.; Kang, M.; Moon, S.; Lee, N.; et al. Speculative Interaction Agents: Building Real-Time Agents with Asynchronous I/O and Speculative Tool Calling. *arXiv preprint arXiv:2605.13360* **2026**.
179. Galbraith, E. When Does Streaming Tool Use Help? Characterizing Tool-Intent Stabilization in Streaming Retrieval-Augmented Generation, 2026, [arXiv:cs.CL/2606.20113]. https://doi.org/10.48550/arXiv.2606.20113.
180. Lin, Y.; Hu, Z.; Subramanian, J.; Wang, Q.; Vlassis, N.; Li, H.H.; Chen, Y. AsyncVoice Agent: Real-Time Explanation for LLM Planning and Reasoning. In Proceedings of the 2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2025, pp. 1–4. https://doi.org/10.1109/ASRU65441.2025.11434747.
181. Likhomanenko, T.; Carlson, L.; Bai, R.H.; Gu, Z.; Tran, H.; Aldeneh, Z.; Zhang, Y.; Zhang, R.; Zheng, H.; Jaitly, N. ChipChat: Low-Latency Cascaded Conversational Agent in MLX. In Proceedings of the 2025 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2025, pp. 1–4. https://doi.org/10.1109/ASRU65441.2025.11434783.
182. Ma, M.; Zheng, B.; Liu, K.; Zheng, R.; Liu, H.; Peng, K.; Church, K.; Huang, L. Incremental Text-to-Speech Synthesis with Prefix-to-Prefix Framework. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 2020; pp. 3886–3896. https://doi.org/10.18653/v1/2020.findings-emnlp.346.
183. Saeki, T.; Takamichi, S.; Saruwatari, H. Low-Latency Incremental Text-to-Speech Synthesis with Distilled Context Prediction Network. In Proceedings of the 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2021, pp. 749–756. https://doi.org/10.1109/ASRU51503.2021.9687904.
184. Singh, Y.R. CrossOracle: An Agentic Framework for Real-Time Expert Personas with Parallelized Acoustic Synthesis. In Proceedings of the Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Atlanta, Georgia, USA, 2026; pp. 254–257.
185. Li, Y.; Wu, G.; Hou, H.; Li, B. UAF: A Unified Audio Front-end LLM for Full-Duplex Speech Interaction, 2026, [2604.19221].
186. Yu, W.; Wang, S.; Yang, X.; Chen, X.; Tian, X.; Zhang, J.; Sun, G.; Lu, L.; Wang, Y.; Zhang, C. SALMONN-omni: A Codec-Free LLM for Full-Duplex Speech Understanding and Generation. *arXiv preprint arXiv:2411.18138* **2024**, [2411.18138]. https://doi.org/10.48550/arXiv.2411.18138.
187. Fang, Q.; Guo, S.; Feng, Y. BayLing-Duplex: Native Full-Duplex Speech Dialogue with a Single Autoregressive LLM, 2026, [arXiv:cs.CL/2606.14528].
188. Zhang, H.; Li, Z.; Tang, X.; Yu, T.; Guo, Y. Ex-Omni-2D: Expressive Omni-Modal Dialogue Models with Native Visual Presence, 2026, [2608.10720].
189. Zhang, X.; Li, G.; Zhu, Y.; Wang, S.; Wu, S.; Yu, S.; Chu, M.; Lu, Y.; Jia, J. StreamArena: Toward Continuous, Interactive, and Long-Horizon Agentic Streaming Video Understanding, 2026, [arXiv:cs.CV/2608.05703].
190. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. In Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 8748–8763.
191. Zhai, X.; Mustafa, B.; Kolesnikov, A.; Beyer, L. Sigmoid Loss for Language Image Pre-Training. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11975–11986.
192. Bai, D.; Yao, S.; Chen, W.; Lai, C.; Li, Y.; Ma, Z.; Wei, X. OmniEncoder: See, Hear, and Feel Continuous Motion Like Humans With One Encoder. *arXiv preprint arXiv:2605.01506* **2026**.
193. Chen, S.; Hwang, M.J.; Inoue, S.; Sun, A.; Yu, B.; Kant, D.; Hyun, D.; Desblancs, D.; Antonovsky, G.; Repin, O.; et al. Alignment-Free Text-Audiobox for Voice Dubbing and Full-Duplex Dialogue Synthesis, 2026, [2609.03992].
194. Shukor, M.; Fini, E.; Turrisi da Costa, V.G.; Cord, M.; Susskind, J.; El-Nouby, A. Scaling Laws for Native Multimodal Models. *arXiv preprint arXiv:2504.07951* **2025**. https://doi.org/10.1109/ICCV51701.2025.00009.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

195. Diao, H.; Cui, Y.; Li, X.; Wang, Y.; Lu, H.; Wang, X. Unveiling Encoder-Free Vision-Language Models. In Proceedings of the Advances in Neural Information Processing Systems, 2024, Vol. 37, pp. 52545–52567.
196. Liu, K.; Zheng, Y.; Wang, K.; Wu, S.; Zhang, R.; Luo, J.; Hatzinakos, D.; Liu, Z.; Fei, H.; Chua, T.S. JavisDiT++: Unified Modeling and Optimization for Joint Audio-Video Generation. *arXiv preprint arXiv:2602.19163* **2026**, [arXiv:cs.CV/2602.19163].
197. Lou, Y.; Yang, K.; You, Y. MoST: Mixing Speech and Text with Modality-Aware Mixture of Experts. *arXiv preprint arXiv:2601.10272* **2026**, [arXiv:cs.CL/2601.10272].
198. Shao, H.; Gao, H.; Shen, Y.; Chen, J.; Long, Z.; Yang, D.; Li, K.; Sun, X. DeepOmni: Towards Seamless and Smart Speech Interaction with Adaptive Modality-Specific MoE. *arXiv preprint arXiv:2506.21864* **2025**, [arXiv:cs.CL/2506.21864].
199. Shen, Y.; Tian, P.; Li, X.; Fang, B.; Xia, S.; Li, B.; Jojic, A.; Ye, W.; Cao, X.; Rehg, J.M.; et al. CogniRoute: Learning to Route Social Evidence in Omni-Modal Models. *arXiv preprint arXiv:2606.20970* **2026**, [arXiv:cs.CV/2606.20970]. https://doi.org/10.48550/arXiv.2606.20970.
200. Tian, X.; Li, W.; Xu, B.; Dong, H.; Wang, Y.; Shen, H. ROMA: Real-time Omni-Multimodal Assistant with Interactive Streaming Understanding. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, San Diego, California, United States, July 2026; pp. 23018–23039, [2601.10323]. https://doi.org/10.18653/v1/2026.findings-acl.1153.
201. Panayotov, V.; Chen, G.; Povey, D.; Khudanpur, S. LibriSpeech: An ASR Corpus Based on Public Domain Audio Books. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2015, pp. 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964.
202. Zen, H.; Dang, V.; Clark, R.; Zhang, Y.; Weiss, R.J.; Jia, Y.; Chen, Z.; Wu, Y. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In Proceedings of the Interspeech 2019, 2019, pp. 1526–1530. https://doi.org/10.21437/Interspeech.2019-2441.
203. Grauman, K.; Westbury, A.; Byrne, E.; Chavis, Z.; Furnari, A.; Girdhar, R.; Hamburger, J.; Jiang, H.; Liu, M.; Liu, X.; et al. Ego4d: Around the world in 3,000 hours of egocentric video. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 18995–19012.
204. Miech, A.; Zhukov, D.; Alayrac, J.B.; Tapaswi, M.; Laptev, I.; Sivic, J. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In Proceedings of the ICCV, 2019. https://doi.org/10.1109/ICCV.2019.00272.
205. Cieri, C.; Miller, D.; Walker, K. The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text. In Proceedings of the Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04); Lino, M.T.; Xavier, M.F.; Ferreira, F.; Costa, R.; Silva, R., Eds., Lisbon, Portugal, 2004.
206. Godfrey, J.J.; Holliman, E.C.; McDaniel, J. SWITCHBOARD: Telephone Speech Corpus for Research and Development. In Proceedings of the Proceedings of ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1992, Vol. 1, pp. 517–520. https://doi.org/10.1109/ICASSP.1992.225858.
207. Linguistic Data Consortium. CALLHOME Collection in Six Languages. LINGUIST List 8.1209, 1997. Language-specific telephone speech, transcript, and lexicon releases for American English, Egyptian Arabic, German, Japanese, Mandarin Chinese, and Spanish.
208. Xie, H.; Guo, D.; Wang, C.; Li, Y.; Tian, W.; Zhu, X.; Wang, X.; Li, X.; Miao, G.; Liu, B.; et al. DialoSpeech: Dual-Speaker Dialogue Generation with LLM and Flow Matching. In Proceedings of the 2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference. IEEE, 2025, pp. 1–6. https://doi.org/10.1109/APSIPAASC65261.2025.11249327.
209. Zhou, Z.; Zhang, Q.; Luo, L.; Liu, J.; Zhou, R. Open-Source Full-Duplex Conversational Datasets for Natural and Interactive Speech Synthesis. *Data Intelligence* **2026**, *8*, 309–316. https://doi.org/10.3724/2096-7004.di.2026.00d2.
210. Wang, C.; Xue, H.; Li, G.; Zhao, Z.; Wang, S.; Wang, S.; Xu, X.; Bu, H.; Xie, L. Full-Duplex Interaction in Spoken Dialogue Systems: A Comprehensive Study from the ICASSP 2026 HumDial Challenge, 2026, [arXiv:eess.AS/2604.21406].
211. Nakata, W.; Saito, Y.; Saruwatari, H. DuplexChat: Constructing Speaker-Separated Full-Duplex Dialogue Speech at Scale for Spoken Dialogue Language Modeling. *arXiv preprint arXiv:2607.04941* **2026**. https://doi.org/10.48550/arXiv.2607.04941.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

212. Kahn, J.; Riviere, M.; Zheng, W.; Kharitonov, E.; Xu, Q.; Mazare, P.; Karadayi, J.; Liptchinsky, V.; Collobert, R.; Fuegen, C.; et al. Libri-Light: A Benchmark for ASR with Limited or No Supervision. In Proceedings of the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, p. 7669–7673. https://doi.org/10.1109/icassp40776.2020.9052942.
213. Kang, W.; Yang, X.; Yao, Z.; Kuang, F.; Yang, Y.; Guo, L.; Lin, L.; Povey, D. LibriHeavy: A 50,000 Hours ASR Corpus with Punctuation Casing and Context. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024, pp. 10991–10995. https://doi.org/10.1109/ICASSP48485.2024.10447120.
214. Koizumi, Y.; Zen, H.; Karita, S.; Ding, Y.; Yatabe, K.; Morioka, N.; Bacchiani, M.; Zhang, Y.; Han, W.; Bapna, A. LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus. In Proceedings of the Interspeech 2023, 2023, pp. 5496–5500. https://doi.org/10.21437/Interspeech.2023-1584.
215. Chen, G.; Chai, S.; Wang, G.B.; Du, J.; Zhang, W.Q.; Weng, C.; Su, D.; Povey, D.; Trmal, J.; Zhang, J.; et al. GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio. In Proceedings of the Interspeech 2021. ISCA, 2021, p. 3670–3674. https://doi.org/10.21437/interspeech.2021-1965.
216. Yang, Y.; Song, Z.; Zhuo, J.; Cui, M.; Li, J.; Yang, B.; Du, Y.; Ma, Z.; Liu, X.; Wang, Z.; et al. GigaSpeech 2: An Evolving, Large-Scale and Multi-domain ASR Corpus for Low-Resource Languages with Automated Crawling, Transcription and Refinement. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2025, pp. 2673–2686. https://doi.org/10.18653/v1/2025.acl-long.135.
217. Galvez, D.; Diamos, G.; Torres, J.; Achorn, K.; Cerón, J.; Gopi, A.; Kanter, D.; Lam, M.; Mazumder, M.; Reddi, V.J. The People's Speech: A Large-Scale Diverse English Speech Recognition Dataset for Commercial Usage. In Proceedings of the Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, 2021, Vol. 1.
218. Pratap, V.; Xu, Q.; Sriram, A.; Synnaeve, G.; Collobert, R. MLS: A Large-Scale Multilingual Dataset for Speech Research. In Proceedings of the Interspeech 2020. ISCA, 2020, p. 2757–2761. https://doi.org/10.21437/interspeech.2020-2826.
219. Ardila, R.; Branson, M.; Davis, K.; Kohler, M.; Meyer, J.; Henretty, M.; Morais, R.; Saunders, L.; Tyers, F.; Weber, G. Common Voice: A Massively-Multilingual Speech Corpus. In Proceedings of the Proceedings of the Twelfth Language Resources and Evaluation Conference; Calzolari, N.; Béchet, F.; Blache, P.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Isahara, H.; Maegaard, B.; Mariani, J.; et al., Eds., Marseille, France, 2020; pp. 4218–4222.
220. Wang, C.; Riviere, M.; Lee, A.; Wu, A.; Talnikar, C.; Haziza, D.; Williamson, M.; Pino, J.; Dupoux, E. VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 2021; pp. 993–1003. https://doi.org/10.18653/v1/2021.acl-long.80.
221. Li, X.; Takamichi, S.; Saeki, T.; Chen, W.; Shiota, S.; Watanabe, S. YODAS: YouTube-Oriented Dataset for Audio and Speech. In Proceedings of the 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2023, pp. 1–8. https://doi.org/10.1109/ASRU57964.2023.10389689.
222. He, H.; Shang, Z.; Wang, C.; Li, X.; Gu, Y.; Hua, H.; Liu, L.; Yang, C.; Li, J.; Shi, P.; et al. Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation. In Proceedings of the 2024 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2024, pp. 885–890. https://doi.org/10.1109/SLT61566.2024.10832365.
223. Zhang, B.; Lv, H.; Guo, P.; Shao, Q.; Yang, C.; Xie, L.; Xu, X.; Bu, H.; Chen, X.; Zeng, C.; et al. WenetSpeech: A 10000+ Hours Multi-domain Mandarin Corpus for Speech Recognition. In Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 6182–6186. https://doi.org/10.1109/ICASSP43922.2022.9746682.
224. Ma, L.; Guo, D.; Song, K.; Jiang, Y.; Wang, S.; Xue, L.; Xu, W.; Zhao, H.; Zhang, B.; Xie, L. WenetSpeech4TTS: A 12,800-Hour Mandarin TTS Corpus for Large Speech Generation Model Benchmark. In Proceedings of the Interspeech 2024, 2024, pp. 1840–1844. https://doi.org/10.21437/Interspeech.2024-2343.
225. Lacombe, Y.; Srivastav, V.; Gandhi, S. Parler-TTS. https://github.com/huggingface/parler-tts, 2024.
226. Anastassiou, P.; Chen, J.; Chen, J.; Chen, Y.; Chen, Z.; Chen, Z.; Cong, J.; Deng, L.; Ding, C.; Gao, L.; et al. Seed-TTS: A Family of High-Quality Versatile Speech Generation Models, 2024, [arXiv:eess.AS/2406.02430].

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

227. Liao, S.; Wang, Y.; Li, T.; Cheng, Y.; Zhang, R.; Zhou, R.; Xing, Y. Fish-Speech: Leveraging Large Language Models for Advanced Multilingual Text-to-Speech Synthesis, 2024, [arXiv:cs.SD/2411.01156].
228. Ding, N.; Chen, Y.; Xu, B.; Qin, Y.; Hu, S.; Liu, Z.; Sun, M.; Zhou, B. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 2023; pp. 3029–3051. https://doi.org/10.18653/v1/2023.emnlp-main.183.
229. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Stanford Alpaca: An Instruction-following LLaMA model. GitHub repository, 2023.
230. Mei, X.; Meng, C.; Liu, H.; Kong, Q.; Ko, T.; Zhao, C.; Plumbley, M.D.; Zou, Y.; Wang, W. WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2024**, *32*, 3339–3354. https://doi.org/10.1109/taslp.2024.3419446.
231. Ephrat, A.; Mosseri, I.; Lang, O.; Dekel, T.; Wilson, K.; Hassidim, A.; Freeman, W.T.; Rubinstein, M. Looking to listen at the cocktail party: a speaker-independent audio-visual model for speech separation. *ACM Transactions on Graphics* **2018**, *37*, 1–11. https://doi.org/10.1145/3197517.3201357.
232. Nagrani, A.; Chung, J.S.; Zisserman, A. VoxCeleb: A Large-Scale Speaker Identification Dataset. In Proceedings of the Interspeech 2017, 2017, pp. 2616–2620. https://doi.org/10.21437/Interspeech.2017-950.
233. Chung, J.S.; Nagrani, A.; Zisserman, A. VoxCeleb2: Deep Speaker Recognition. In Proceedings of the Interspeech 2018. ISCA, Sept 2018, p. 1086–1090. https://doi.org/10.21437/interspeech.2018-1929.
234. Pătrăucean, V.; Smaira, L.; Gupta, A.; Recasens Continente, A.; Markeeva, L.; Banarse, D.; Koppula, S.; Heyward, J.; Malinowski, M.; Yang, Y.; et al. Perception Test: A Diagnostic Benchmark for Multimodal Video Models. In Proceedings of the Advances in Neural Information Processing Systems, 2023, Vol. 36, pp. 42748–42761. https://doi.org/10.52202/075280-1852.
235. Hershey, S.; Ellis, D.P.W.; Fonseca, E.; Jansen, A.; Liu, C.; Moore, R.C.; Plakal, M. The Benefit of Temporally-Strong Labels in Audio Event Classification. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 366–370. https://doi.org/10.1109/ICASSP39728.2021.9414579.
236. Teknium. OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants, 2023.
237. Du, Z.; Wang, Y.; Chen, Q.; Shi, X.; Lv, X.; Zhao, T.; Gao, Z.; Yang, Y.; Gao, C.; Wang, H.; et al. CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models, 2024, [arXiv:cs.SD/2412.10117].
238. Kim, T.; wook Kim, K.; Woo, S.H.; Hirschberg, J.; Kim, G.; Hakkani-Tür, D. DuplexGen: Adaptive Synthesis of Human-AI Turn-Taking Dialogues, 2026, [arXiv:cs.CL/2607.26178].
239. Chu, Y.; Xu, J.; Yang, Q.; Wei, H.; Wei, X.; Guo, Z.; Leng, Y.; Lv, Y.; He, J.; Lin, J.; et al. Qwen2-Audio Technical Report, 2024, [arXiv:eess.AS/2407.10759].
240. Lin, G.T.; Shivakumar, P.G.; Gourav, A.; Gu, Y.; Gandhe, A.; Lee, H.y.; Bulyko, I. Align-SLM: Textless Spoken Language Models with Reinforcement Learning from AI Feedback. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 20395–20411. https://doi.org/10.18653/v1/2025.acl-long.997.
241. Zhang, D.; Li, Z.; Li, S.; Zhang, X.; Wang, P.; Zhou, Y.; Qiu, X. SpeechAlign: Aligning Speech Generation to Human Preferences. In Proceedings of the Advances in Neural Information Processing Systems, 2024, Vol. 37, pp. 50343–50360. https://doi.org/10.52202/079017-1594.
242. Hsiao, C.Y.; Lu, K.H.; Fu, Y.K.; Lin, G.T.; Hung, H.T.; yi Lee, H. ASPIRin: Action Space Projection for Interactivity-Optimized Reinforcement Learning in Full-Duplex Speech Language Models, 2026, [arXiv:cs.CL/2604.10065].
243. Ohashi, A.; Zeghidour, N.; Défossez, A.; Kharitonov, E. Multi-Faceted Interactivity Alignment in Full-Duplex Speech Models, 2026, [arXiv:cs.CL/2606.11167].
244. Arora, S.; Tian, J.; Shi, J.; Futami, H.; Kashiwagi, Y.; Tsunoo, E.; Watanabe, S. Optimizing Conversational Quality in Spoken Dialogue Systems with Reinforcement Learning from AI Feedback. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, San Diego, California, United States, 2026; pp. 41049–41066. https://doi.org/10.18653/v1/2026.findings-acl.2040.
245. Fu, C.; Lin, H.; Wang, X.; Zhang, Y.F.; Shen, Y.; Liu, X.; Cao, H.; Long, Z.; Gao, H.; Li, K.; et al. VITA-1.5: Towards GPT-4o Level Real-Time Vision and Speech Interaction. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38. https://doi.org/10.52202/085713-2529.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

246. Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C.D.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In Proceedings of the Advances in Neural Information Processing Systems, 2023, Vol. 36, pp. 53728–53741. https://doi.org/10.52202/075280-2338.
247. Shen, Y.; Liu, Y.; Zhu, J.; Cao, X.; Zhang, X.; He, Y.; Ye, W.; Rehg, J.M.; Lourentzou, I. Fine-Grained Preference Optimization Improves Spatial Reasoning in VLMs. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38, pp. 20406–20437. https://doi.org/10.52202/085713-0606.
248. Li, Y.; Wu, D.; Lin, G.T.; Lee, H.y.; Qin, C.; Chen, Z.; Chen, C. Decoupling Conversational Dynamics in Full-Duplex Spoken Models through Reinforcement Learning. *arXiv preprint arXiv:2607.07148* **2026**. https://doi.org/10.48550/arXiv.2607.07148.
249. Huang, B.; Wang, X.; Chen, H.; Song, Z.; Zhu, W. VTimeLLM: Empower LLM to Grasp Video Moments. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, pp. 14271–14280.
250. Ren, S.; Yao, L.; Li, S.; Sun, X.; Hou, L. TimeChat: A Time-sensitive Multimodal Large Language Model for Long Video Understanding. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, pp. 14313–14323.
251. Wang, H.; Xu, Z.; Cheng, Y.; Diao, S.; Zhou, Y.; Cao, Y.; Wang, Q.; Ge, W.; Huang, L. Grounded-VideoLLM: Sharpening Fine-grained Temporal Grounding in Video Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 959–975. https://doi.org/10.18653/v1/2025.findings-emnlp.50.
252. Shen, Y.; Xu, J.; Li, B.; Zhang, Y.; Yu, T.; Li, B.; Yang, H.; Wang, R.; Cao, X. ChronoVision: Temporal Reasoning via Latent State Reconstruction. *arXiv preprint arXiv:2608.05631* **2026**, [arXiv:cs.CV/2608.05631]. https://doi.org/10.48550/arXiv.2608.05631.
253. Holla, M.; Lourentzou, I. Commonsense for zero-shot natural language video localization. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 2166–2174.
254. Cao, X. Enhancing Video Vision Language Model with Hippocampal Sensing. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026, pp. 33682–33692.
255. Wang, T.; et al. FineMoLA: Towards Fine-Grained Motion-Language Alignment from Clip-Level Supervision. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2026, pp. 5510–5519.
256. Cui, J.; Xu, B.; Wang, C.; Yu, T.; Sun, W.; Xu, Y.; Wang, T.; He, Z.; Ma, W.; Cai, T.; et al. MiniCPM-o 4.5: Towards Real-Time Full-Duplex Omni-Modal Interaction, 2026, [arXiv:cs.CL/2604.27393].
257. Williams, J.D.; Raux, A.; Henderson, M. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse* **2016**, *7*, 4–33. https://doi.org/10.5087/dad.2016.301.
258. Chao, H.; Bai, Y.; Sheng, R.; Li, T.; Sun, Y. STALE: Can LLM Agents Know When Their Memories Are No Longer Valid?, 2026, [arXiv:cs.CL/2605.06527]. https://doi.org/10.48550/arXiv.2605.06527.
259. Hu, M.; Chen, T.; Chen, Q.; Mu, Y.; Shao, W.; Luo, P. HiAgent: Hierarchical Working Memory Management for Solving Long-Horizon Agent Tasks with Large Language Model. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vienna, Austria, 2025; pp. 32779–32798. https://doi.org/10.18653/v1/2025.acl-long.1575.
260. Pink, M.; Wu, Q.; Vo, V.A.; Turek, J.; Mu, J.; Huth, A.; Toneva, M. Position: Episodic Memory is the Missing Piece for Long-Term LLM Agents, 2025, [arXiv:cs.AI/2502.06975]. https://doi.org/10.48550/arXiv.2502.06975.
261. Wu, D.; Wang, H.; Yu, W.; Zhang, Y.; Chang, K.W.; Yu, D. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
262. Di, S.; Yu, Z.; Zhang, G.; Li, H.; ZhongTao.; Cheng, H.; Li, B.; He, W.; Shu, F.; Jiang, H. Streaming Video Question-Answering with In-context Video KV-Cache Retrieval. In Proceedings of the International Conference on Learning Representations; Yue, Y.; Garg, A.; Peng, N.; Sha, F.; Yu, R., Eds., 2025, Vol. 2025, pp. 42115–42127.
263. Tong, W.; Guo, H.; Ran, D.; Chen, J.; Lu, J.; Wang, K.; Li, K.; Zhu, X.; Li, J.; Li, K.; et al. InteractiveOmni: A Unified Omni-modal Model for Audio-Visual Multi-turn Dialogue, 2025, [arXiv:cs.CV/2510.13747].
264. Chen, T.; Zhang, K.; Wu, Q.; Chen, X.; Chang, C.; Sun, X.; Zhou, Y.; Ji, R. Scaling the Long Video Understanding of Multimodal Large Language Models via Visual Memory Mechanism, 2026, [arXiv:cs.CV/2603.29252].

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

265. Qu, H.; Yao, G.; Xing, L.; Hu, X.; Ding, R.; Zhang, G.; Zhang, F.; Yuan, Y.; Shu, X.; Yan, S. Beyond Retrieval: Progressive Latent Memory Evolution for Streaming Video Understanding, 2026, [2609.04131].
266. Li, B.; Shen, Y.; Liu, Y.; Xu, Y.; Liu, J.; Li, X.; Li, Z.; Zhu, J.; Zhong, Y.; Lan, F.; et al. Toward cognitive supersensing in multimodal large language model. *arXiv preprint arXiv:2602.01541* **2026**. https://doi.org/10.48550/arXiv.2602.01541.
267. Guan, Y.; Yin, L.; Liang, D.; Ju, J.; Luo, Z.; Luan, J.; Liu, Y.; Bai, X. Video Streaming Thinking: VideoLLMs Can Watch and Think Simultaneously, 2026, [arXiv:cs.CV/2603.12262].
268. Liu, J.; Huang, J.; Jia, Z.; Li, J.; Zhang, X.; Guo, Z.; Li, B.; Zeng, W.; Lu, Y.; Jin, X. An Efficient Streaming Video Understanding Framework with Agentic Control, 2026, [2605.17921].
269. Li, J.; Li, B.; Jiang, N.; Ning, X.; Wang, X.; Shen, Y.; Wang, H.; Jian, Y.; Wu, X.; Athiwaratkun, B.; et al. Multi-Turn Agentic Scientific Literature Search via Workflow Induction. *arXiv preprint arXiv:2607.00597* **2026**. https://doi.org/10.48550/arXiv.2607.00597.
270. Gao, T.; Yen, H.; Yu, J.; Chen, D. Enabling Large Language Models to Generate Text with Citations. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 2023; pp. 6465–6488. https://doi.org/10.18653/v1/2023.emnlp-main.398.
271. Banerjee, P.; Moshtaghi, M.; Subramanian, S.; Misra, A.; Chadha, A. APEX-MEM: Agentic Semi-Structured Memory with Temporal Reasoning for Long-Term Conversational AI. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, California, United States, 2026; pp. 16470–16489. https://doi.org/10.18653/v1/2026.acl-long.749.
272. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, 2023.
273. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In Proceedings of the Computer Vision – ECCV 2024. Springer, 2024, pp. 38–55. https://doi.org/10.1007/978-3-031-72970-6_3.
274. Chen, T.; Veluri, B.; Gong, H.; Gollakota, S. AV-Dialog: Spoken Dialogue Models with Audio-Visual Input. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Liakata, M.; Moreira, V.P.; Zhang, J.; Jurgens, D., Eds., San Diego, California, United States, 2026; pp. 42208–42225. https://doi.org/10.18653/v1/2026.acl-long.1954.
275. Adavanne, S.; Politis, A.; Nikunen, J.; Virtanen, T. Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks. *IEEE Journal of Selected Topics in Signal Processing* **2019**, *13*, 34–48. https://doi.org/10.1109/JSTSP.2018.2885636.
276. Roll, N.; Graham, C.; Todd, S. PSST! Prosodic Speech Segmentation with Transformers. In Proceedings of the Proceedings of the 27th Conference on Computational Natural Language Learning, Singapore, 2023; pp. 476–487. https://doi.org/10.18653/v1/2023.conll-1.31.
277. Van, H.P.; Hieu, N.M.; Tuan, K.P.T.; Hai, N.L.; Van, L.N.; Diep, N.T.N.; Le, T. MemORAI: Memory Organization and Retrieval via Adaptive Graph Intelligence for LLM Conversational Agents. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2026, San Diego, California, United States, 2026; pp. 28235–28253. https://doi.org/10.18653/v1/2026.findings-acl.1408.
278. Minh, T.L.; Shimizu, N.; Miyazaki, T.; Shinoda, K. Deep Learning Based Multi-modal Addressee Recognition in Visual Scenes with Utterances. In Proceedings of the Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 1546–1553. https://doi.org/10.24963/ijcai.2018/214.
279. Inoue, K.; Lala, D.; Elmers, M.; Ochi, K.; Kawahara, T. An LLM Benchmark for Addressee Recognition in Multi-modal Multi-party Dialogue. In Proceedings of the Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology; Torres, M.I.; Matsuda, Y.; Callejas, Z.; del Pozo, A.; D'Haro, L.F., Eds., Bilbao, Spain, 2025; pp. 330–334.
280. Roy, R.; Raiman, J.; Lee, S.G.; Ene, T.D.; Kirby, R.; Kim, S.; Kim, J.; Catanzaro, B. PersonaPlex: Voice and Role Control for Full Duplex Conversational Speech Models. In Proceedings of the ICASSP 2026 – 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2026, pp. 16137–16141. https://doi.org/10.1109/ICASSP55912.2026.11463413.
281. Zhi, X.; Yin, P.; Guan, S.; Zheng, C.; Cheng, J.; Yan, X. LiveServe: Interaction-Aware Serving for Real-Time Omni-Modal LLMs. *arXiv preprint arXiv:2606.22983* **2026**, [arXiv:cs.DC/2606.22983]. https://doi.org/10.48550/arXiv.2606.22983.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

282. Ellinas, N.; Vamvoukakis, G.; Markopoulos, K.; Chalamandaris, A.; Maniati, G.; Kakoulidis, P.; Raptis, S.; Sung, J.S.; Park, H.; Tsiakoulis, P. High Quality Streaming Speech Synthesis with Low, Sentence-Length-Independent Latency. In Proceedings of the Proceedings of Interspeech 2020, 2020, pp. 2022–2026. https://doi.org/10.21437/Interspeech.2020-2464.
283. Okafuji, Y.; Inoue, K.; Ohira, Y. Low-Latency Turn-Taking via Context-Aware Preface Generation in a Real-World Dialogue Robot, 2026, [arXiv:cs.RO/2607.23204].
284. Qi, H.; Skantze, G. MuVAP: Multimodal Multiparty Voice Activity Projection for Turn-taking Prediction in the Wild. *arXiv preprint arXiv:2606.16731* **2026**. https://doi.org/10.48550/arXiv.2606.16731.
285. Riera, P.; Brusco, P.; Kuo, C.; Sancinetti, M.; Branavan, S.R.K. Synchronization and Turn-Taking in Full-Duplex Speech Dialogue Models, 2026, [2605.20356].
286. Yang, B.; Xu, L.; Zeng, L.; Liu, K.; Jiang, S.; Lu, W.; Chen, H.; Jiang, X.; Xing, G.; Yan, Z. ContextAgent: Context-Aware Proactive LLM Agents with Open-World Sensory Perceptions. In Proceedings of the Advances in Neural Information Processing Systems, 2025, Vol. 38, pp. 185637–185671. https://doi.org/10.52202/085713-5582.
287. Shen, Y.; Bis, D.; Lu, C.; Lourentzou, I. Elba: Learning by asking for embodied visual navigation and task completion. In Proceedings of the Proceedings of the Winter Conference on Applications of Computer Vision, 2025, pp. 5177–5186.
288. Yan, W.; Dai, Y.; Ran, Q.; Li, H.; Lin, W.; Jin, T.; Xie, X.; Liao, H.; Lian, J. Proact-VL: A Proactive VideoLLM for Real-Time AI Companions. In Proceedings of the Forty-third International Conference on Machine Learning, 2026.
289. Kuzmin, N.; Zhong, T.; Deng, J.; Zhu, Y.; Tsoi, T.; Cao, T.; Lui, S.; Lee, K.A.; Chng, E.S. Privacy-Preserving End-to-End Full-Duplex Speech Dialogue Models, 2026, [2603.08179].
290. Horvitz, E. Principles of Mixed-Initiative User Interfaces. In Proceedings of the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, 1999, pp. 159–166. https://doi.org/10.1145/302979.303030.
291. Tang, Y.; Ma, W.; Zhao, X.; Salimi, A.; Moridani, S.H.; Shen, D.; Wang, J.; Abdulrazzag, A.; Aubry, M.; Chen, Y.H.; et al. Instruct-FD: Can Your Full-Duplex Speech System Follow Turn-Taking Instructions?, 2026, [arXiv:cs.CL/2607.20460].
292. Chhikara, P.; Khant, D.; Aryan, S.; Singh, T.; Yadav, D. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. In Proceedings of the ECAI 2025. IOS Press, 2025, Vol. 413, *Frontiers in Artificial Intelligence and Applications*, pp. 2993–3000. https://doi.org/10.3233/FAIA251160.
293. Lau, A.; Choi, Y.; Balazadeh Meresht, V.; Chidambaram, K.; Syrgkanis, V.; Krishnan, R.G. Personalized Adaptation via In-Context Preference Learning, 2024, [arXiv:cs.LG/2410.14001]. https://doi.org/10.48550/arXiv.2410.14001.
294. Hao, Y.; Mehri, S.; Zhai, C.; Hakkani-Tür, D. User Preference Modeling for Conversational LLM Agents: Weak Rewards from Retrieval-Augmented Interaction, 2026, [arXiv:cs.CL/2603.20939]. https://doi.org/10.48550/arXiv.2603.20939.
295. Yu, G.I.; Jeong, J.S.; Kim, G.W.; Kim, S.; Chun, B.G. Orca: A Distributed Serving System for Transformer-Based Generative Models. In Proceedings of the Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation, Carlsbad, CA, 2022; pp. 521–538.
296. Agrawal, A.; Panwar, A.; Mohan, J.; Kwatra, N.; Gulavani, B.S.; Ramjee, R. SARATHI: Efficient LLM Inference by Piggybacking Decodes with Chunked Prefills. *arXiv preprint arXiv:2308.16369* **2023**.
297. Patel, P.; Choukse, E.; Zhang, C.; Shah, A.; Goiri, Í.; Maleki, S.; Bianchini, R. Splitwise: Efficient Generative LLM Inference Using Phase Splitting. In Proceedings of the 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture. IEEE, 2024, pp. 118–132. https://doi.org/10.1109/ISCA59077.2024.00019.
298. Zhong, Y.; Liu, S.; Chen, J.; Hu, J.; Zhu, Y.; Liu, X.; Jin, X.; Zhang, H. DistServe: Disaggregating Prefill and Decoding for Goodput-Optimized Large Language Model Serving. In Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation, Santa Clara, CA, 2024; pp. 193–210.
299. Xiao, G.; Tian, Y.; Chen, B.; Han, S.; Lewis, M. Efficient streaming language models with attention sinks. In Proceedings of the International Conference on Learning Representations, 2024, Vol. 2024, pp. 21875–21895.
300. Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; et al. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems, 2023. https://doi.org/10.52202/075280-1506.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

301. Fedus, W.; Zoph, B.; Shazeer, N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research* **2022**, *23*, 1–39.
302. Jiang, A.Q.; Sablayrolles, A.; Roux, A.; et al. Mixtral of Experts. *arXiv preprint arXiv:2401.04088* **2024**, [arXiv:cs.LG/2401.04088]. https://doi.org/10.48550/arXiv.2401.04088.
303. DeepSeek-AI. DeepSeek-V3 Technical Report. *arXiv preprint arXiv:2412.19437* **2024**.
304. Gale, T.; Narayanan, D.; Young, C.; Zaharia, M. MegaBlocks: Efficient Sparse Training with Mixture-of-Experts. In Proceedings of the Proceedings of Machine Learning and Systems, 2023, Vol. 5, pp. 288–304.
305. Hwang, R.; Wei, J.; Cao, S.; Hwang, C.; Tang, X.; Cao, T.; Yang, M. Pre-Gated MoE: An Algorithm-System Co-Design for Fast and Scalable Mixture-of-Expert Inference. In Proceedings of the 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture, 2024, pp. 1018–1031. https://doi.org/10.1109/ISCA59077.2024.00078.
306. Yu, H.; Cui, X.; Zhang, H.; Wang, H.; Wang, H. Taming Latency-Memory Trade-Off in MoE-Based LLM Serving via Fine-Grained Expert Offloading. In Proceedings of the Proceedings of the 21st European Conference on Computer Systems. Association for Computing Machinery, 2026, pp. 176–191. https://doi.org/10.1145/3767295.3769319.
307. Sainath, T.N.; Parada, C. Convolutional Neural Networks for Small-Footprint Keyword Spotting. In Proceedings of the Proceedings of Interspeech 2015, 2015, pp. 1478–1482. https://doi.org/10.21437/Interspeech.2015-352.
308. Sridhar, K.; Cutler, R.; Saabas, A.; Pärnamaa, T.; Loide, M.; Gamper, H.; Braun, S.; Aichner, R.; Srinivasan, S. ICASSP 2021 Acoustic Echo Cancellation Challenge: Datasets, Testing Framework, and Results. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 151–155. https://doi.org/10.1109/ICASSP39728.2021.9413457.
309. Kim, S.; Mangalam, K.; Moon, S.; Malik, J.; Mahoney, M.W.; Gholami, A.; Keutzer, K. Speculative Decoding with Big Little Decoder. In Proceedings of the Advances in Neural Information Processing Systems, 2023. https://doi.org/10.52202/075280-1705.
310. Liu, Z.; Zhao, C.; Iandola, F.; Lai, C.; Tian, Y.; Fedorov, I.; Xiong, Y.; et al. MobileLLM: Optimizing Sub-Billion Parameter Language Models for On-Device Use Cases. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2024.
311. Jin, H.; Wu, Y. CE-CoLLM: Efficient and Adaptive Large Language Models Through Cloud-Edge Collaboration. In Proceedings of the Proceedings of the IEEE International Conference on Web Services, 2025. https://doi.org/10.1109/ICWS67624.2025.00046.
312. Qu, G.; Chen, Q.; Wei, W.; Lin, Z.; Chen, X.; Huang, K. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. *IEEE Communications Surveys & Tutorials* **2025**, *27*, 3820–3860. https://doi.org/10.1109/COMST.2025.3527641.
313. Ma, X.; Dousti, M.J.; Wang, C.; Gu, J.; Pino, J. SIMULEVAL: An Evaluation Toolkit for Simultaneous Translation. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; Liu, Q.; Schlangen, D., Eds., Online, 2020; pp. 144–150. https://doi.org/10.18653/v1/2020.emnlp-demos.19.
314. Ekstedt, E.; Skantze, G. How Much Does Prosody Help Turn-taking? Investigations using Voice Activity Projection Models. In Proceedings of the Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue; Lemon, O.; Hakkani-Tur, D.; Li, J.J.; Ashrafzadeh, A.; Garcia, D.H.; Alikhani, M.; Vandyke, D.; Dušek, O., Eds., Edinburgh, UK, 2022; pp. 541–551. https://doi.org/10.18653/v1/2022.sigdial-1.51.
315. Li, Y.; Zhang, G.; Ma, Y.; Yuan, R.; Zhu, K.; Guo, H.; Liang, Y.; Liu, J.; Wang, N.; Yang, J.; et al. OmniBench: Towards The Future of Universal Omni-Language Models. In Proceedings of the Advances in Neural Information Processing Systems 38 (NeurIPS 2025), Datasets and Benchmarks Track, 2025, Vol. 38, [2409.15272]. https://doi.org/10.52202/085713-1036.
316. Chen, Y.; Yue, X.; Zhang, C.; Gao, X.; Tan, R.T.; Li, H. VoiceBench: Benchmarking LLM-Based Voice Assistants. *Transactions of the Association for Computational Linguistics* **2026**, *14*, 378–398. https://doi.org/10.1162/tacl.a.628.
317. Huang, C.Y.; Lu, K.H.; Wang, S.H.; Hsiao, C.Y.; Kuan, C.Y.; Wu, H.; Arora, S.; Chang, K.W.; Shi, J.; Peng, Y.; et al. Dynamic-SUPERB: Towards a Dynamic, Collaborative, and Comprehensive Instruction-Tuning Benchmark for Speech. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing, 2024, pp. 12136–12140. https://doi.org/10.1109/ICASSP48485.2024.10448257.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

318. Huang, C.; Chen, W.; Yang, S.; Liu, A.T.; Li, C.; Lin, Y.; Tseng, W.; Diwan, A.; Shih, Y.; Shi, J.; et al. Dynamic-SUPERB Phase-2: A Collaboratively Expanding Benchmark for Measuring the Capabilities of Spoken Language Models with 180 Tasks. In Proceedings of the The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net, 2025.
319. Gosai, A.; Vuong, T.; Tyagi, U.; Li, S.; You, W.; Bavare, M.; Uçar, A.; Fang, Z.; Jang, B.; Liu, B.; et al. Audio MultiChallenge: A Multi-Turn Evaluation of Spoken Dialogue Systems on Natural Human Interaction. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, California, United States, July 2026; pp. 35740–35770, [2512.14865]. https://doi.org/10.18653/v1/2026.acl-long.1654.
320. Si, S.; Ma, W.; Gao, H.; Wu, Y.; Lin, T.E.; Dai, Y.; Li, H.; Yan, R.; Huang, F.; Li, Y. SpokenWOZ: A Large-Scale Speech-Text Benchmark for Spoken Task-Oriented Dialogue Agents. In Proceedings of the Advances in Neural Information Processing Systems; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds. Curran Associates, Inc., 2023, Vol. 36, pp. 39088–39118.
321. Zhong, T.; Deng, J.; Kuzmin, N.; Zhu, Y.; Cao, T.; Tsoi, T.; Tan, Z.; Lui, S.; Liu, X. IRAF: Interference-Resilient Adaptive Fusion for Noise-Robust End-to-End Full-Duplex Spoken Dialogue Systems. *arXiv preprint arXiv:2606.06559* **2026**, [2606.06559]. https://doi.org/10.48550/arXiv.2606.06559.
322. Fukuda, R.; Ando, A.; Kanagawa, H.; Kano, T.; Delcroix, M.; Tawara, N.; Chiba, Y. M3-DuplexBench: A Multi-Turn, Multilingual, Multidomain Benchmark for Full-Duplex Spoken Dialogue Models, 2026, [2607.29125].
323. Balog, K.; Zhai, C. User Simulation for Evaluating Information Access Systems. *Foundations and Trends in Information Retrieval* **2024**, *18*, 1–261. https://doi.org/10.1561/1500000098.
324. Deshpande, K.; Sirdeshmukh, V.; Mols, J.B.; Jin, L.; Hernandez-Cardona, E.Y.; Lee, D.; Kritz, J.; Primack, W.E.; Yue, S.; Xing, C. MultiChallenge: A Realistic Multi-Turn Conversation Evaluation Benchmark Challenging to Frontier LLMs. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025; Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 18632–18702. https://doi.org/10.18653/v1/2025.findings-acl.958.
325. Gao, J.; Sun, C.; Yang, Z.; Nevatia, R. TALL: Temporal Activity Localization via Language Query. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017, pp. 5277–5285. https://doi.org/10.1109/ICCV.2017.563.
326. Sigurdsson, G.A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; Gupta, A. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In Proceedings of the Computer Vision – ECCV 2016. Springer, 2016, Vol. 9905, *Lecture Notes in Computer Science*, pp. 510–526. https://doi.org/10.1007/978-3-319-46448-0_31.
327. Hu, H.; Dong, S.; Zhao, Y.; Lian, D.; Li, Z.; Gao, S. TransRAC: Encoding Multi-scale Temporal Correlation with Transformers for Repetitive Action Counting. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 18991–19000. https://doi.org/10.1109/CVPR52688.2022.01843.
328. Wang, Y.; Wang, Y.; Chen, B.; Wu, T.; Zhao, D.; Zheng, Z. OmniMMI: A Comprehensive Multi-Modal Interaction Benchmark in Streaming Video Contexts. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025, pp. 18925–18935. https://doi.org/10.1109/CVPR52734.2025.01763.
329. Shen, Y.; Zhang, J.; Xu, J.; Kim, J.; Lourentzou, I.; Cao, X.; Huang, M. Evaluating Cognitive Age Alignment in Interactive AI Agents. *arXiv preprint arXiv:2605.17894* **2026**, [arXiv:cs.AI/2605.17894]. https://doi.org/10.48550/arXiv.2605.17894.
330. Cao, X.; Shen, Y.; Lai, B.; Ye, W.; Ma, Y.; Heintz, J.; Chen, J.; Huang, M.; Cao, J.; Zhang, A.; et al. What is the Visual Cognition Gap between Humans and Multimodal LLMs? In Proceedings of the Proceedings of the Conference on Language Modeling (COLM), 2025.
331. Cao, W.; Hallgarten, M.; Li, T.; Dauner, D.; Gu, X.; Wang, C.; Miron, Y.; Aiello, M.; Li, H.; Gilitschenski, I.; et al. Pseudo-Simulation for Autonomous Driving. In Proceedings of the Proceedings of the 9th Conference on Robot Learning; Lim, J.; Song, S.; Park, H.W., Eds. PMLR, 2025, Vol. 305, *Proceedings of Machine Learning Research*, pp. 4709–4722.
332. Balog, K.; Zhai, C. The Indispensable Role of User Simulation in the Pursuit of AGI. *Communications of the ACM* **2026**, *69*, 45–48. https://doi.org/10.1145/3768170.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org | NOT PEER-REVIEWED | Posted: 19 October 2026 https://doi.org/10.20944/preprints202610.1649.v1

333. Wang, C.; Zheng, R.C.; Ai, Y.; Ling, Z.H. Understanding Textual Capability Degradation in Speech LLMs via Parameter Importance Analysis. In Proceedings of the ICASSP 2026 – 2026 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2026, pp. 16282–16286. https://doi.org/10.1109/ICASSP55912.2026.11461684.
334. Bai, Y.; Chen, J.; Chen, Y.; Deng, W.; Dong, B.; Duan, N.; Gu, Y.; Han, W.; Huang, Y.; Ke, M.; et al. JoyAI-Talker: Full-Duplex Speech Interactive Large Model Built for Empathetic Voice Agents, 2026, [2608.01119].
335. Cai, R.; Li, B.; Wen, X.; Chen, M.; Zhao, Z. Diagnosing and Mitigating Modality Interference in Multimodal Large Language Models, 2026, [arXiv:cs.LG/2505.19616].
336. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* **2024**, *12*, 157–173. https://doi.org/10.1162/tacl_a_00638.
337. Maharana, A.; Lee, D.H.; Tulyakov, S.; Bansal, M.; Barbieri, F.; Fang, Y. Evaluating Very Long-Term Conversational Memory of LLM Agents. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand, 2024; pp. 13851–13870. https://doi.org/10.18653/v1/2024.acl-long.747.
338. Pan, L.; Fu, Z.; Zhai, Y.; Tao, S.; Guan, S.; Huang, S.; Zhang, L.; Liu, Z.; Ding, B.; Henry, F.; et al. Omni-SafetyBench: A Benchmark for Safety Evaluation of Audio-Visual Large Language Models. *arXiv preprint arXiv:2508.07173* **2025**, [arXiv:cs.CL/2508.07173]. https://doi.org/10.48550/arXiv.2508.07173.
339. Zhan, X.; Sun, G.; Such, J.; Woodland, P. Protecting Bystander Privacy via Selective Hearing in Audio LLMs. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, California, United States, 2026; pp. 15180–15192. https://doi.org/10.18653/v1/2026.acl-long.693.
340. Peng, Z.; Liu, Y.; Sun, Z.; Li, M.; Luo, Z.; Zheng, J.; Dong, W.; He, X.; Wang, X.; Xue, Y.; et al. JALMBench: Benchmarking Jailbreak Vulnerabilities in Audio Language Models. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
341. Song, Z.; Jiang, Q.; Cui, M.; Li, M.; Gao, L.; Zhang, Z.; Xu, Z.; Wang, Y.; Ouyang, G.; Chen, Z.; et al. Audio Jailbreak: An Open Comprehensive Benchmark for Jailbreaking Large Audio-Language Models. In Proceedings of the Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, California, United States, 2026; pp. 27294–27308. https://doi.org/10.18653/v1/2026.acl-long.1259.
342. Zhang, H.; Chen, J.; Wu, D.; Li, Y.; Zhang, Y.; Zhang, X.T.; Liu, C.; Lin, Q.; Peng, Y.; Liu, H.; et al. Duplexsla: A full-duplex spoken language model with synchronized speech, language, and action. *arXiv preprint arXiv:2605.20755* **2026**, [2605.20755]. https://doi.org/10.48550/arXiv.2605.20755.
343. Fang, S.; Feng, Y.; Zhang, Y.; Metaxas, D.N. RAC: Rectified Flow Auto Coder, 2026, [arXiv:cs.CV/2603.05925].
344. Fang, S.; Zhong, H.; Feng, Y.; Zhang, Y.; Metaxas, D.N. StreamFlow: Theory, Algorithm, and Implementation for High-Efficiency Rectified Flow Generation. In Proceedings of the Forty-third International Conference on Machine Learning, 2026.
345. Munkhdalai, T.; Faruqui, M.; Gopal, S. Leave No Context Behind: Efficient Infinite Context Transformers with Infini-Attention. *arXiv preprint arXiv:2404.07143* **2024**. https://doi.org/10.48550/arXiv.2404.07143.
346. Jiang, H.; Li, Y.; Zhang, C.; Wu, Q.; Luo, X.; Ahn, S.; Han, Z.; Abdi, A.H.; Li, D.; Lin, C.Y.; et al. Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. *Advances in Neural Information Processing Systems* **2024**, *37*, 52481–52515. https://doi.org/10.52202/079017-1663.
347. Huang, W.; Ji, J.; Xia, C.; Zhang, B.; Yang, Y. SafeDreamer: Safe Reinforcement Learning with World Models. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
348. Autio, C.; Schwartz, R.; Dunietz, J.; Jain, S.; Stanley, M.; Tabassi, E.; Hall, P.; Roberts, K. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1, National Institute of Standards and Technology, Gaithersburg, MD, 2024. https://doi.org/10.6028/NIST.AI.600-1.
349. OmniJev. PlayJev: A Multimodal JEV-Like Model for Small Games. GitHub repository, 2026. Accessed September 23, 2026.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2026 by the author(s). Distributed under a Creative Commons CC BY license.