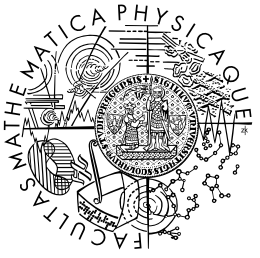

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

**MASTER THESIS**

Bc. Eliáš Cizl

# A Text-Guided Full-Duplex Spoken Dialogue System

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: Mgr. et Mgr. Ondřej Dušek, Ph.D.

Study programme: Computer Science – Artificial Intelligence

Prague 2026

I declare that I carried out this master thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....  .....

Author's signature

Title: A Text-Guided Full-Duplex Spoken Dialogue System

Author: Bc. Eliáš Cizl

Institute: Institute of Formal and Applied Linguistics

Supervisor: Mgr. et Mgr. Ondřej Dušek, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Full-duplex spoken dialogue systems, i.e. systems that can listen and speak simultaneously, remain difficult to build with strong linguistic capabilities. We present Warmblood, a fine-tuned variant of the Moshi full-duplex speech large language model (LLM), extended with a novel guide mechanism: an additional token channel through which short, externally generated text is injected at runtime. During inference, a backbone textual LLM receives a transcription of the ongoing conversation and produces concise steering text that is fed into the guide channel, grounding Warmblood's responses without altering its duplex behavior. The main speech LLM is fine-tuned with LoRA on synthetic and natural conversational data annotated with guide signals. Internal evaluations confirm that the model learns to exploit the guide, yielding significant improvements in answer correctness. On VoiceAssistant-Eval, Warmblood outperforms the base Moshi across all categories, including speech consistency, robustness, and safety. Results on Full-Duplex-Bench are mixed: pause handling and backchanneling improve, while smooth turn-taking and response latency regress slightly. A 100-participant user study reports moderate overall satisfaction, with stronger scores for response coherence and truthfulness and lower scores for naturalness.

Keywords: dialogue systems, large language models, audio language models, spoken dialogue systems, natural language processing, chatbots, dialogue

Název práce: Plně duplexní hlasový dialogový systém s textovým vedením

Autor: Bc. Eliáš Cizl

Institut: Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: Mgr. et Mgr. Ondřej Dušek, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Plně duplexní hlasové dialogové systémy, tedy systémy, které umí současně naslouchat i mluvit, je obtížné vybudovat s dostatečnými jazykovými schopnostmi. Představujeme Warmblood, přetrénovanou variantu plně duplexního mluveného velkého jazykového modelu (LLM) Moshi rozšířenou o nový mechanismus vedení: dodatečný tokenový kanál, jehož prostřednictvím je modelu za běhu vkládán krátký externě generovaný text. Při inferenci přijímá základní textové LLM přepis probíhající konverzace a generuje stručný návodný text vkládaný do kanálu vedení, čímž ukotvuje odpovědi Warmblood bez narušení jeho duplexního chování. Hlavní mluvené LLM je přetrénováno pomocí LoRA na syntetických a přirozených konverzačních datech anotovaných signály guide. Interní evaluace potvrzují, že model se naučí guide využívat, což přináší výrazné zlepšení správnosti odpovědí. Na benchmarku VoiceAssistant-Eval Warmblood překonává základní model Moshi ve všech kategoriích, včetně konzistence řeči, robustnosti a bezpečnosti. Výsledky na Full-Duplex-Bench jsou smíšené: zpracování pauz a backchanneling se zlepšují, zatímco plynulé střídání mluvčích a latence odpovědí se mírně zhoršují. Uživatelská studie se 100 účastníky vykazuje mírnou celkovou spokojenost s vyššími hodnoceními koherence a pravdivosti odpovědí a nižšími hodnoceními přirozenosti.

Klíčová slova: dialogové systémy, velké jazykové modely, zvukové jazykové modely, hlasové dialogové systémy, zpracování přirozeného jazyka, chatboti, dialog

# Contents

# Introduction

## Background

Spoken dialogue systems (SDSs) have been studied for some time, but the field has recently gained significant traction, owing to advances in natural language generation (NLG). While both text and speech dialogue research have progressed substantially, text-based dialogue systems are currently the most mature and widely deployed, with large language models (LLMs) dominating the field. Speech is considerably more difficult to capture, as it conveys information at a finer granularity: content is communicated not only through linguistic but also through paralinguistic features (emotions, prosody, etc.), and both parties may communicate simultaneously.

The ideal goal is to build an artificial assistant that speaks as naturally as a person would, while possessing the intelligence of contemporary AI systems. When a user interacts with such a system, it should operate in real time with minimal latency. The assistant's speech should be generated incrementally, and the system should operate in full-duplex mode, capable of simultaneously listening and speaking.

Much of current research is focused on extending LLMs into the speech modality. The conventional approach is to convert user speech into text with an ASR (automatic speech recognition) module, process the text (e.g., with an LLM), and convert the output back to speech with a TTS (text-to-speech) module. These separate stages introduce higher latencies, however. Therefore, models are often trained to handle direct speech representations or to accept multimodal content (text, audio, video) in general. Another principal drawback of conventional models is that they are often designed in a turn-based fashion, in which the user and assistant exchange turns. While they may not require explicit turn-taking from the user (e.g., by using VAD—voice activity detection—and incorporating an explicit end-of-turn mechanism), they remain inherently half-duplex where only one side can speak at a time.

Even setting aside the assistant's behavior, the user's speech remains unpredictable. Users may wish to interrupt, change topics, address a third party, introduce background noise, or produce backchannels during the agent's speech. If the system cannot adapt to such occurrences, the interaction will feel unnatural and lead to user dissatisfaction. Figure 1 illustrates a short but highly dynamic exchange with overlaps, backchanneling, and minimal inter-turn gaps.

| | |
|---|---|
| User | Hi, where is the lab room? |
| Assistant | It is on the second floor, corridor B, right after— |
| User | [overlap] Sorry, is there an elevator? |
| Assistant | [continues] —the stairs. Yes, there is one near reception. |
| User | mm-hmm (0.2 s) and can I enter with a guest? |
| Assistant | [short gap: 0.1 s] Yes, but the guest must check in first. |
| User | [overlap] Okay, great, thanks. |
| Assistant | Sure, I can also send you the wayfinding map. |

**Figure 1** Short example of a dynamic spoken dialogue with overlap, backchanneling, and minimal response gaps.

We believe that some aspects of an SDS are more important than others. People may change their behavior when talking to a robot [71], but their experience may still be positive. A robotic-sounding voice may be acceptable if the system demonstrates strong understanding of the user's requests. Another important aspect is readiness for rapid turn exchange. When a user interrupts an assistant, this does not necessarily mean they want the system to stop immediately—they may simply be making a brief interjection or signaling that they would like to respond once the assistant finishes its current sentence. Abruptly halting output generation upon detecting any speech from the other side may cause confusion for both parties. Some systems, such as GPT-4o [111] voice mode, exhibit this behavior owing to their inherently half-duplex design.

In recent research, Moshi [41] stands out because it aims to be a full-duplex, low-latency dialogue model built on an end-to-end architecture. It explicitly captures both the user and the model channel, allowing it to listen and speak simultaneously. No artificial turn boundaries are necessary. The authors released an online demo in which the system's capabilities can be tested. It can converse in real time and often appears very natural in its interactive behavior. However, its speech intelligence falls short of systems that incorporate a standard textual LLM [28], and the system is prone to instability. Our experiments showed that it can conduct a very natural conversation in one instance while being nearly incomprehensible in the next.

Many authors of other models focus on scores from artificial benchmarks, which do not necessarily translate to performance in a real conversational setting. No current evaluation framework, besides real user testing, can fully capture the human experience of conversing with an SDS [38]. Speech intelligence, correct turn-taking, and latency are but individual aspects that collectively shape the overall experience. If any one of these aspects is poor, it can cause considerable dissatisfaction even when all others are of high quality. Therefore, assessing the true quality of a system without a publicly available demo, a released model, or user study scores is nearly impossible.

### Aims

Our aim is to take a system with strong interaction capabilities and improve its linguistic intelligence. The key properties we sought for the system are as follows. First and foremost, the system should handle both channels in parallel and allow either side to speak at any time. It should provide grounded information which it receives from a more reliable source. The system should prefer external guidance over its own speculation. It should retain interaction capabilities like giving priority to the user's speech (but not necessarily cutting off too abruptly) and being able to determine when to start speaking.

### Contributions

We chose Moshi as our base model and present *Warmblood*, an improved variant. We add an additional text channel to Moshi's input called *guide* through which the model can receive external information or instructions. During fine-tuning, the model is taught to use the guide to its benefit. During inference,

additional modules run in parallel and send a transcription of the conversation to a fast LLM that generates the guide.

We evaluate Warmblood on our own internal datasets; we then assess it on established benchmarks (VoiceAssistant-Eval [151] and Full-Duplex-Bench [86]) and conduct a user study in which participants rate their experience with the system.

Our internal evaluations confirm that the model learns to use the guide, yielding meaningful improvements in question-answering quality. On VoiceAssistant-Eval, Warmblood outperforms the base Moshi model across all categories, including speech consistency, robustness, and safety. Results on Full-Duplex-Bench are mixed: Warmblood improves pause handling and backchannel behavior, but at the cost of some smooth turn-taking and response latency relative to Moshi. The user study produced moderate scores overall, with higher ratings for response coherence and intelligence and lower ratings for naturalness and general impression.

### Why Warmblood?

The name *Warmblood* is inspired by the Czech Warmblood horse breed, common in Czechia where we are based. The metaphor reflects our architecture: like a horse, the main model has "a mind of its own," while a rider can steer its behavior. In our case, this steering signal is the *guide* channel, which provides additional information or instructions; however, the final decision always belongs to the main model (or the horse).

### Thesis Structure

The remainder of this thesis is organized as follows. Chapter 1 provides the necessary background, covering two-party spoken dialogue and turn-taking research, human–AI interaction, and the landscape of spoken language modeling from early systems through cascaded and end-to-end architectures, including speech representations, training strategies, evaluation methods, and notable recent full-duplex models. Chapter 2 describes the Warmblood architecture—specifically the guide mechanism and the real-time inference pipeline. Chapter 3 presents the experimental setup: the training data survey and preparation pipeline, the evaluation datasets and scenarios, the metrics, and the fine-tuning configuration. Chapter 4 presents the evaluation: internal ablation experiments, observations from live inference, results on the VoiceAssistant-Eval and Full-Duplex-Bench benchmarks, and findings from a 100-participant user study. The Conclusion summarizes contributions, limitations, and directions for future work.

# 1 Background

This chapter surveys the background relevant to building a full-duplex spoken dialogue system. Section 1.1 introduces the properties of spoken dialogue, covering turn-taking, conversation dynamics, human–AI interaction, and potential deployment use cases. Section 1.2 reviews spoken language modeling, tracing the evolution from early systems through cascaded and end-to-end architectures. Section 1.3 discusses system capabilities and requirements. Section 1.4 covers speech representations. Section 1.5 describes the common building blocks shared across systems. Section 1.6 discusses training strategies and data, and Section 1.7 surveys evaluation techniques and benchmarks. The chapter concludes in Section 1.8 with a review of the models most directly relevant to our work.

## 1.1 Dialogue

When discussing dialogue in the context of software agents, we can distinguish two main branches: text and audio/speech. A dialogue involves two or more participants. As the number of participants increases, the interaction becomes more complex and harder to analyze. Therefore, we focus solely on two-party dialogue, where the roles are easier to define.

Text dialogue usually occurs in messaging environments: texting, chatting, emails, and similar settings. It is inherently turn-based, with clear markers of whose turn it is. Brief interjections native to spoken conversation are relatively uncommon. By contrast, longer messages are prevalent, and the recipient often does not need to be online when the message is sent. Both sides may also prepare their messages simultaneously, with the sending timestamp determining their order.

Spoken dialogue has also received considerable research attention. However, capturing all the aspects that make everyday conversation feel natural is not easy. Beyond the semantic content of words, spoken dialogue carries information through prosody (pitch, intonation, and rhythm), speaking rate, voice quality, and timbre, as well as conversational signals such as filled pauses (“um,” “uh”), laughter, and backchannels (“mm-hmm,” “right”). None of these have a direct textual equivalent, yet they profoundly shape the meaning and flow of an exchange.

Text is comparatively easier to analyze. It carries information primarily through the words themselves and their linguistic features. There is little room to express emotion beyond devices such as emojis, which remain only a distant proxy. Text is also much more convenient for machine processing because it is a compact representation and there is much more textual data available.

For these reasons, recent research, especially in AI and natural language processing, has been much more text-oriented. Text-based systems are considerably more prevalent than audio ones.

In this section, we discuss the progress that has been made in spoken dialogue. The ultimate goal is to construct an agent whose conversational abilities are as close to human behavior as possible. Such an agent should actively use the interactive elements mentioned above, appear intelligent, and understand where the dialogue should be heading.

### 1.1.1 Turn-Taking

A straightforward way to analyze spoken dialogue is through *turn-taking*. Although it is more complex in speech than in text, it is an aspect shared by both modalities. When two people talk, one of them usually *has the floor* while the other is listening, possibly with brief confirmations. At some point, the main speaker changes. This change can be initiated by either side, either by one speaker finishing their utterance or by the other interrupting. As a result, the gap between turns can vary substantially. For example, after a question is asked, the other person may need additional time to formulate a response before speaking. This delay can be as long as about two seconds [121]. In other cases, the respondent may already know what to say before the question has finished, leading to an overlap of up to about two seconds.

The gap between speakers varies across languages [139]. In English, it centers around 200 ms on average. However, affirmative responses, such as answers to questions, tend to be faster than other types of responses. This is well below the silence threshold of 700–1,000 ms commonly used in traditional dialogue systems, which shows that natural turn-taking relies on *predicting* turn completion in advance rather than reacting only after the dialogue counterpart has finished speaking [134].

Nguyen et al. [104] use several key terms: *Inter-Pausal Units* (IPUs) are defined as continuous stretches of speech in one speaker's channel, delimited by a silence of more than 200 ms on both sides. They further define *silence* as sections of the recording with no voice signals on either channel and *overlap* as sections where voice signals are present on both channels. Silences can be subdivided into *gaps*, when they occur between two IPUs produced by different speakers, and *pauses*, when they occur within the speech of the same speaker. Successive IPUs by the same speaker separated by a pause are regrouped into a *turn*. The authors also distinguish two overlap types in theory, although they are harder to measure: *backchannels*, short utterances by the other side that usually serve as confirmations, and true interruptions, where the turn passes to the other side even though they started speaking earlier [134].

These properties make a transcription of spoken dialogue very different even from a fast-paced textual online chat. Whereas a text chat shows clean turns composed of complete sentences, a spoken dialogue transcript records overlap markers, pause durations, filled pauses, and incomplete utterances, producing a much richer but also noisier representation of the same exchange [104]. This difference will be important when we examine how speech is represented (Section 1.4) and how spoken dialogue agents are trained (Section 1.6).

Stephens and Beattie [138] investigated how well people can distinguish turn-final and turn-medial utterances. They found that this was possible only when participants listened to audio recordings rather than reading textual transcripts, and only when the conversation involved disagreement rather than agreement. Even in those settings, however, the judges did not achieve very high accuracy. This suggests that determining the correct time to take the turn may be difficult even for humans and that there may be multiple appropriate moments.

Research has also addressed meetings [20], but we focus on two-person conversation because it best represents the interactions of our interest. Meetings usually follow a different format from conversations between two people, even though

some elements are shared. For this reason, the authors of Moshi [41], for instance, deliberately fine-tune their dialogue system on two-party telephone conversations rather than on multi-party meetings.

Beyond explicit turn-taking, conversations also differ in how "dynamic" they are: some involve frequent speaker changes and interruptions, while others consist of longer stretches where one participant holds the floor with occasional backchannels. This notion is inherently context-dependent and not captured by turn-taking alone; an action can be formally well-timed yet pragmatically inappropriate given the preceding interaction.

### 1.1.2 Human–AI Interaction

The interaction between humans and AI systems has been studied through many different systems. In the text domain, large language models are now prevalent. They provide an easy-to-use interface with low accessibility requirements and strong reasoning and language-understanding capabilities. They are trained on massive amounts of textual data available on the Internet. Competition has grown rapidly, accompanied by many objective evaluation methods that compare the quality and usability of these systems across areas such as knowledge and reasoning [56], mathematical problem solving [35], and open-ended instruction following.

For audio, we turn to spoken dialogue systems. These take audio as input and generate audio as output. They may also incorporate additional input sensors, such as a camera or text interface, and physical outputs such as gestures, an avatar, or text. Open audio data, especially dialogue data, are much scarcer. Evaluation is possible in this domain, but it usually focuses on specific components, for example by applying metrics only to textual transcriptions, measuring audio quality, or examining turn behavior. There is still no benchmark that can comprehensively determine the quality of a conversational system.

The fluency of a dialogue may also be disrupted by factors related to the practical realization of an SDS. Boland et al. [16] found that Internet-based conversations exhibited delays between speakers that are not present in face-to-face interaction. This was measured in person-to-person conversations. However, higher latency, and thus reduced fluency, can also be expected in modern conversational systems.

Branigan et al. [18] showed that people tend to coordinate their syntactic structure in dialogue. They use phrasing similar to that of the other side and spontaneously adopt some of the same linguistic behaviors. This means that people can adapt, to some extent, to their conversation partner. Kennington et al. [71] presents several patterns in which people adjust their communicative behavior and their expectations when interacting with a robot. These findings are important because the system does not need to be perfectly natural in every respect for the interaction to remain workable; the user can partially adapt to it. Some aspects of speech may therefore be less important for fluency, for example selecting the "correct" syntactic structure.

Some early attempts of practical human–AI interaction are worth noting. Building a full-fledged system has long been difficult. Addlesee et al. [1] placed a robot in a hospital. Their system has a physical presence and uses audio and video

as input, which are processed by an LLM. Its benchmark performance and overall naturalness remain far from perfect. Another robotic hospital receptionist was deployed by Gunson et al. [54], who built a complex system with many modules, including ASR and TTS. Other attempts include connecting an LLM with external tools to construct a virtual chatbot [162] or combining an LLM with a Furhat [4] robotic head [30]. Some early systems focus on specific tasks, such as an agent operating in a simulated household environment [141].

#### Use Cases

There are several areas of society in which an automated dialogue agent could be deployed. These use cases matter because a conversational system should always be designed with a specific purpose in mind. General chitchat and knowledge-retrieval assistants are also possible, but a clearly defined task helps guide development more effectively.

Typical examples include hotel reception, where an agent can handle check-in/check-out workflows and common guest requests [30]; retail assistance, where it can support product questions, orders, and returns; and restaurant service, where it can simplify table-side ordering and follow-up requests. Online companionship or chatbot use is a related category, where the value lies in broad availability and natural interaction rather than strict task completion. Public-facing local guidance is another practical case: an always-available assistant can answer repetitive navigation and information queries for visitors and residents.

Healthcare is especially important, because communication quality directly affects user comfort and outcomes. Prior work has demonstrated this domain's significance: Addlesee et al. [2] discuss the importance of incremental dialogue for people with dementia; DeVault et al. [42] show that extra-verbal utterances can be helpful in this setting; and Gunson et al. [54] built a virtual hospital receptionist.

Across all of these settings, the main advantage is that routine conversations can be handled automatically at scale, while more complex or unusual situations can still be handed over to a human when needed.

## 1.2 Spoken Language Modeling

Several surveys have covered spoken language modeling [116, 38, 65, 10, 29]. We drew inspiration from them, especially WavChat [65], when structuring the following sections.

The naming of these systems is very inconsistent in the contemporary literature. Terms such as *speech language models* (SLMs), *(large) audio language models*, and SDSs all appear, and their meanings often overlap. We therefore do not enforce a strict distinction either. We mainly use the term SDSs because dialogue is the core motivation of our work. We also use the more general terms *model*, *system* or occasionally *assistant* or *agent*.

A key term in dialogue modeling is *duplexity*. A *fully duplex* system is one in which either side can speak and listen at the same time, their speech may overlap, and arbitrary gaps may occur—this best reflects real human dialogue. The alternative is a *half-duplex* or *semi-duplex* system. In this mode, only one side may be active at a time and the other has to listen. The turn is decided

by some mechanism. By design, it is impossible for both sides to speak and be listened to at the same time.

Many contemporary studies lack a clear goal beyond "building an audio system." Even when a stated objective such as "achieving natural dialogue" is given, it is difficult to quantify. There are no objective metrics for measuring dialogue quality or naturalness. The benchmarks that are used vary widely, and most rely on properties of the audio transcription, question answering, and similar tasks. There is also no consensus on which datasets should be used for training. Examples include text data, audio data, and text synthesized into audio. Many of the datasets used are private, or the details of the training data are withheld. Many authors also do not release source code or provide demos, which makes manual analysis of the final systems more difficult or impossible. The challenges of training data and evaluation are discussed in more detail in Sections 1.6 and 1.7.

Current surveys make navigation easier, but they often stay at the surface level, listing figures and characteristics from the original papers without analyzing the systems more deeply. For example, many agents lack the ability to determine turn-taking on their own or operate with higher response latency. This makes comparisons unfair when one system has substantially richer interactive abilities, such as full duplexity, and another does not. Ideally, surveys would go further in assessing the practical usefulness of the reported results from multiple viewpoints. Spoken dialogue remains a young area, and the needed categorization is still incomplete.

In the following subsections, we list the various approaches from both the past and present.

### 1.2.1 Early Systems

The traditional spoken dialogue system pipeline consists of three components: a spoken language understanding (SLU) module that converts the user's speech into a semantic representation, a dialogue manager that updates an internal state and selects an action according to a handcrafted policy (typically a finite-state machine or flowchart), and a natural language generation module that synthesizes the spoken response [170]. While widely deployed in call-center and information-retrieval applications, such deterministic systems were expensive to build and fragile under the recognition errors common in real-world environments.

To address this, partially observable Markov decision processes (POMDPs) were proposed as a principled statistical framework for dialogue management [170]. A POMDP maintains a *belief state*—a probability distribution over all possible dialogue states—updated by Bayesian inference after each utterance, and a policy optimized via reinforcement learning maps it directly to a system action, making the dialogue manager substantially more robust to recognition errors.

Before the introduction of Transformers [145], LSTM networks [58] were widely used in dialogue management. State tracking was employed for incremental dialogue [183]. Direct turn prediction—predicting when the system should begin speaking rather than simply reacting to silence—was also explored in task-oriented and situated systems [36, 96]. Roddy et al. [123] further show that acoustic features are preferable to linguistic ones for turn-taking.

Kalatzis et al. [70] attempted to reduce reliance on large amounts of data by

extracting more information from a single dialogue.

With the advent of pre-trained language models (PLMs) and large language models, the text-based components of dialogue systems improved substantially in language understanding and generation, paving the way for the cascaded and end-to-end approaches described in the following sections.

### 1.2.2 Cascaded Audio LLM Systems

The most straightforward way to implement a spoken dialogue system at present is to connect several pre-existing modules: a VAD component to determine turn-taking, an ASR component to convert speech to text, a textual LLM to produce an answer, and finally a TTS component to synthesize the spoken response. Figure 1.1 shows this architecture.

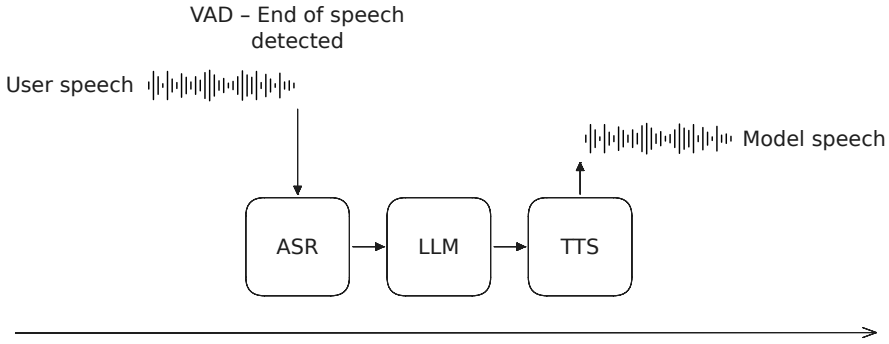


**Figure 1.1** An overview of a cascaded dialogue system. VAD detects end of user speech which is then converted to text, and processed by an LLM. The response is converted back to speech and played to the user.

The earliest prototypes of this kind, such as AudioGPT [63], simply chained an ASR model, ChatGPT, and a TTS module, using speech merely as an input–output interface. Open-source pipelines later added a VAD module to separate speech from silence and to distinguish speakers. Because a plain transcription discards all paralinguistic information (see Section 1.1.1), some systems try to preserve it: ParalinGPT [87], E-chat [161], and Spoken-LLM [83] feed emotion or style embeddings extracted from speech alongside the text so that the LLM can react to *how* something was said. A complementary direction keeps the text output but also passes the raw speech to the LLM via a pre-trained speech encoder plus a small learned "bridge" module (an adapter), as in Qwen-Audio [33] or SALMONN [143], allowing the system to "listen" without a separate ASR stage. More recent cascaded models, such as Llama 3.1 [100], even integrate a trainable TTS decoder, but since they still generate text first and convert it to speech only afterward, they remain cascaded in nature.

Despite their simplicity, cascaded systems remain a strong baseline. Because the core reasoning happens in the text domain, they inherit the full strength of modern LLMs and tend to perform well on content-oriented benchmarks such as question answering, while losing on tasks that depend on acoustic cues [10]. For this reason, they are frequently used as a reference point against which end-to-end systems are compared [10, 65]. Their main drawbacks are the loss of paralinguistic information, the accumulation of errors across the three stages, and the high latency caused by running several components in sequence.

GPT-4o [111] (and newer versions like `gpt-realtime-2.1` [110]) with its voice mode is one of the flagship commercial systems. Its implementation details have not been released, but it operates on a semi-duplex basis and exhibits the behavior of a cascaded system.

A recent open-source example is AURA [94], which combines open-weight ASR, a text LLM with ReAct-style [168] reasoning, tool-use APIs (calendar, contacts, web search, email), and TTS in a cascaded multi-turn pipeline. The authors report competitive results on content-oriented tasks. This reinforces the practical strength of cascaded designs for service-oriented assistants, especially when reliable task completion and tool orchestration are prioritized over full-duplex interaction.

Recent work suggests that half-duplex ASR–LLM–TTS pipelines can approximate full-duplex interaction via external control modules. FireRedChat [23], for instance, adds turn-taking control, streaming personalized VAD for barge-in, end-of-turn detection, and tool-aware dialogue management, while supporting cascaded and semi-cascaded back ends.

Wang et al. [152] similarly propose a cascaded system where the LLM decides whether to speak or listen, but the pipeline remains text-based and thus falls short of true full duplexity.

### 1.2.3 End-to-End Audio LLM Systems

The key difference in end-to-end systems is that the core language model directly understands and generates speech representations instead of relying on text as the central intermediary. This brings such systems conceptually closer to our goal of natural conversation: there is no lossy transcription step, acoustic information can be preserved throughout, and the assistant can in principle react to the user while the user is still speaking. The category is broad and hard to group cleanly, however—the systems differ in their representations, architectures, and training, and far from all of them are actually full-duplex. Usually, the audio is tokenized or otherwise encoded on the input side and decoded on the output side. Most of the systems in use today are based on Transformers originally trained on text and later adapted to operate on audio. Some models produce only text as output. To our knowledge, there is no fully end-to-end model that operates directly on raw waveforms [65, 10].

The clear advantage of end-to-end systems is the low latency at which they can operate. This is crucial in spoken dialogue, where speaker changes happen very quickly. However, many systems do not aim for full duplexity and can be used only in a standard request-answer mode, where a recording is provided first and an answer is produced afterward. Some works do claim theoretical full duplexity, or what the authors describe as full duplex, but they do not provide examples or metrics that would clearly demonstrate this ability in practice [49].

Adapting a text-trained model to work with audio naturally weakens some of its original abilities. This is one reason why end-to-end systems tend not to outperform cascaded approaches on traditional benchmarks such as question answering [116, 65].

Representative examples range from early textless systems such as GSLM [75] and AudioLM [17], through speech-extended text LLMs such as SpeechGPT [175], to the more recent full-duplex systems described in Section 1.8. A notable

conceptual distinction within this class is whether the model learns a conditional distribution $p(S^b|S^a)$—predicting one speaker's channel given the other, as in Moshi (Section 1.8.2) and LSLM [92]—or the joint distribution $p(S^a, S^b)$ of both channels simultaneously, as in dGSLM and NTPP [153]. In the conditional case, training typically bakes in an implicit directionality: one stream is treated as "context" and the other as "to be generated," so performance can degrade if the channels are swapped or if the desired output speaker changes. Modeling the joint distribution instead treats both channels symmetrically and allows generation to be conditioned on either side (or on partial observations of both), which supports speaker-independent behavior and more robust role swapping.

A recent direction narrows the gap between duplex end-to-end and cascaded systems by adding explicit reasoning to streaming speech generation. Streaming chain-of-thought (SCoT) predicts an intermediate ASR transcript and text response on fixed-duration audio blocks without VAD turn segmentation [11]. STITCH similarly incorporates explicit reasoning into streaming speech generation, reporting improvements on reasoning tasks without increasing latency [31].

A related architectural question is how to route the user stream into the LLM during generation. Lu et al. [91] compare channel fusion (injecting the user stream at each step) with cross-attention routing (keeping it as external memory). Channel fusion improves semantic grounding and QA (question answering), but cross-attention is more robust to context corruption; under interruptions, fusion can mix overlapping user speech into the generation and derail it.

## 1.3 System Capabilities

An audio language model may possess the following capabilities, the list is based on Ji et al. [65]:

- **Text intelligence**: Text understanding typically comes from audio fine-tuning of an originally text-based model. The system may still be able to process and generate text. By *intelligence*, we mean general capabilities such as answering questions, reasoning, following instructions, and maintaining an engaging conversation.

- **Speech intelligence**: Speech intelligence is the counterpart of the previous capability. In theory, the same linguistic abilities should also be available in the speech modality. In addition, the system should take acoustic information into account. For example, the same sentence can convey different meanings depending on how it is said. This includes understanding prosody, accent, tone of voice, and speaking rate.

- **Music understanding and generation**: A dialogue system is usually trained to accept and produce speech, but this could in principle be extended to music as well.

- **Multilingual capability**: Most models today are trained on English or Chinese data. The ability to communicate in multiple languages is still relatively uncommon, mainly because of the scarcity of data in other languages.

- **Context and multi-turn capabilities**: Real conversations can last for hours and span multiple contexts. A system should be able to retain the relevant history and respond appropriately to recurring or evolving topics.
- **Interactivity**: This comprises full duplexity and the ability of the model to take turns at appropriate moments, react correctly to interruptions and backchannels, and potentially interrupt the user itself.
- **Latency**: Responses should be timely and ideally begin within one second, although shorter or longer delays may be appropriate depending on the interaction.
- **Multimodal capability**: Instead of having a separate model for each modality (text, audio, images, video), it is desirable to have a system that can accept any modality, or several at once, and respond appropriately to the given instruction.

For Warmblood's target use cases, **speech intelligence**, **interactivity**, and **low latency** are the most critical capabilities: the system must understand requests correctly, handle turns naturally, and respond without noticeable delay. Context and multi-turn capability are needed for any service exchange spanning more than a few turns. Music understanding and extensive multimodal input are largely out of scope for a service-oriented agent, although multilingual support would widen the deployment range. In terms of current coverage, text intelligence and basic speech quality are handled reasonably well by state-of-the-art models, whereas full-duplex interactivity and robust long-context behavior remain the main open challenges [29].

### 1.3.1 Tool Usage

Tool usage—letting the model call external functions, query databases, or trigger actions—is still largely underexplored in spoken dialogue models [65]. The idea is well established in older modular task-oriented systems and text LLMs, where function calling allows the agent to complete concrete tasks such as making a reservation, and it maps naturally onto our use cases (Section 1.1.2). For spoken systems, the main difficulty is doing this without sacrificing responsiveness; improvements have nonetheless been shown with respect to latency [182]. Previous work has also coupled spoken dialogue models with retrieval-augmented generation to answer factual questions from an external knowledge base [10], which is directly relevant to the information needs of our target scenarios.

### 1.3.2 Dialogue Requirements

Following WavChat [65], we summarize the properties that distinguish spoken dialogue from text-based interaction and that an ideal system should therefore satisfy: streaming, duplexity, and interaction.

**Streaming.** A real system cannot wait for an entire utterance to finish before it starts processing; it must consume the input and produce the output chunk by chunk, in a causal manner that relies only on past information. In a cascaded system, every component—the ASR and the TTS in particular—must itself be streaming for the whole pipeline to remain responsive.

**Duplexity.** As introduced above, a full-duplex system can listen and speak at the same time. In practice, this means the model must continue perceiving the user even while generating its own response, so that it can be interrupted, place a backchannel, or interrupt the user when appropriate.

**Interaction.** Building on the turn-taking concepts from Section 1.1.1, the interactions a system has to handle can be grouped into normal turn exchanges, interruptions, and backchannels. The earliest full-duplex systems relied on a simple VAD to decide whether the user wanted to take the turn, but such a mechanism cannot distinguish a genuine interruption from a backchannel and therefore leads to frequent premature cutoffs [65, 134].

In early research, Skantze and Hjalmarsson [135] used the Wizard-of-Oz method to compare an incremental system with its non-incremental counterpart. The former emerged as a clear winner among participants. Additional research supports this finding [6].

There is also early research on handling overlaps [51], where the authors built a robot that can replicate human-like behavior during overlap situations. Rather than treating overlap as an error, the system classifies overlap according to its temporal position relative to turn boundaries, dialogue history, and utterance priority, then selects an appropriate conversational strategy.

Both cascaded and end-to-end systems have attempted to model these real-time interactions. On the cascaded side, turn management is usually made explicit, for example through a neural finite-state machine [152]. On the end-to-end side, the behavior is learned directly from data; notable examples include Parrot [99], SyncLLM [146], OmniFlatten [178], and Freeze-Omni [154], while Mini-Omni2 [159] trains an explicit interruption mechanism on a specially constructed dataset with dedicated interrupt and non-interrupt markers. A recurring problem across all of these systems is the lack of a standardized benchmark for interaction, which makes fair comparison difficult.

Recent work has also begun integrating explicit reasoning mechanisms into full-duplex architectures rather than relying solely on turn-management strategies, suggesting that interaction quality and semantic reasoning need not be treated as competing objectives [11]. Wu et al. [157] push this idea further by arguing that a full-duplex agent should not remain semantically idle while listening. Instead of repeatedly predicting silence tokens during the user's turn, their model performs compact, strictly causal "chronological thinking" throughout the listening phase and then switches to speech generation without additional delay. This highlights a more demanding view of interaction: beyond taking turns correctly, a strong spoken dialogue system should also use the listening window productively while remaining interruptible and latency-efficient.

## 1.4 Representations

Language models are trained to predict the next token in a sequence, which presupposes a discrete, compact vocabulary—exactly what text provides through subword tokenization. Speech has no such natural tokenization: it is a continuous waveform that mixes linguistic content with acoustic detail (timbre, prosody, emotion) and is far less information-dense than text. The choice of speech representation is therefore one of the central design questions of a spoken dialogue model, since it determines how the system perceives, processes, and generates speech. The literature distinguishes two broad families of representations—*semantic* and *acoustic* tokens—and the following subsections discuss them from the input and output perspectives, together with the main trade-offs.

### 1.4.1 Input

**Semantic tokens.** Semantic representations are produced by prediction-based models that are trained to capture the linguistic content of speech. They are usually obtained from pretrained self-supervised encoders such as wav2vec 2.0 [12], HuBERT [59] and WavLM [27], or the supervised Whisper [118], often the same models that were trained with text as the target. Their advantage is that they map naturally into the embedding space of a text LLM, which simplifies alignment with the text modality, and they offer high compression rates (frame rates of 25–50 Hz are common), keeping the sequences the language model has to process short.

**Acoustic tokens.** Acoustic representations come from neural codecs (EnCodec [40], SpeechTokenizer [180], Mimi [41]) trained for compression and faithful reconstruction. Their added value is that they preserve the paralinguistic detail—timbre, prosody, emotion—that semantic tokens discard, and they are straightforward to apply to any kind of audio, not only speech.

AudioLM [17] combines both semantic and acoustic tokens and shows that they complement each other in terms of phonetic discriminability and reconstruction quality.

### 1.4.2 Output

The same two families appear on the output side. When the model generates semantic tokens, these lack acoustic conditioning and therefore need a separately trained vocoder or decoder to synthesize the final waveform. When the model generates acoustic tokens, they can be fed directly into the frozen codec decoder—the neural network component of the codec that maps discrete codebook indices back to a continuous audio waveform—giving a simpler end-to-end path to the waveform. Another practical advantage of discrete acoustic outputs is that they can be fed straight back into the model as input for the next dialogue turn, as done in OmniFlatten [178].

### 1.4.3 Semantic or Acoustic

There is no consensus on which family to use, and many systems mix them—for instance, by using a semantic representation on the input side and an acoustic one on the output side, as in Mini-Omni [158]. The trade-off is summarized well by WavChat [65]: semantic tokens are stronger on the comprehension side, align better with the LLM, and compress better, but fall short on naturalness and expressiveness; acoustic tokens carry richer emotional and acoustic information, generalize to music and general audio, and connect directly to a codec decoder, at the cost of lower compression and weaker LLM alignment. Because dialogue ultimately needs both meaning and expressive speech, a promising direction is to combine the two. SpeechTokenizer [180] and Mimi [41] already do this by distilling semantic information (from HuBERT or WavLM) into the first quantizer level of an otherwise acoustic codec. In duplex systems the mixing can also be applied per stream rather than per modality axis: SALM-Duplex [61] routes the user stream through a pretrained streaming ASR encoder (continuous semantic features) while generating discrete acoustic codec tokens for the agent, which allows the codec to be fine-tuned toward the agent's target voice independently of user-side perception.

### 1.4.4 Discrete or Continuous

A second axis is whether the representation is discrete or continuous. Discrete tokens are by far the more typical choice, because language models trained with a next-token-prediction objective naturally favor a finite vocabulary, and because discrete acoustic outputs plug directly into a codec decoder [65, 10]. Continuous features (such as mel-spectrograms or hidden states from a neural encoder) avoid the information loss of quantization and can preserve finer acoustic detail, but their use is still in its infancy. Continuous inputs are common in speech-aware models that attach a pretrained encoder and an adapter to the LLM (such as Mini-Omni [158] or LLaMA-Omni [48]), which requires an extra modality-alignment training phase; whether continuous truly outperforms discrete in this setting remains an open question [65]. Whisper-GPT [147] combines both types in their architecture.

### 1.4.5 Text Guidance

Because direct speech-to-speech generation is difficult, many end-to-end systems first generate an intermediate text stream and use it to guide the speech output [65]. This text guidance leverages the strong text-modeling abilities of the LLM and improves the quality and factuality of the generated speech, but it comes at the cost of latency, especially when the entire text answer has to be produced before speaking can start. Parallel schemes that emit text and speech tokens at the same time, such as Moshi's inner monologue (Section 1.8.2), Mini-Omni [158], or LLaMA-Omni [48], mitigate this delay. Looking further ahead, WavChat [65] envisions speech tokenizers that align text and speech already during the encoding phase, which would allow the model to drop explicit text guidance altogether and give rise to a new paradigm for conversational systems.

## 1.5 Existing Components

Most spoken dialogue systems are not built from scratch. Instead, researchers usually assemble them from a handful of established building blocks: a speech encoder (or tokenizer) on the input side, a language model as the backbone, and a vocoder or decoder on the output side. Because of this, even "end-to-end" systems are rarely end-to-end in the strict sense—the three components are typically trained separately and only then connected. The selection of these components is inconsistent across the literature; every researcher group tends to make its own choice, and since the rest of the system differs as well, it is hard to measure the influence of any single component [65, 10]. Systematic comparisons are rare: GSLM [75] is one of the few works that explicitly compared several speech tokenizers (CPC [108], wav2vec 2.0 [12], and HuBERT [59]), finding HuBERT to perform best. In the following subsections, we briefly review the common choices for each component; the existing systems assembled from them are described in Section 1.8.

### 1.5.1 Encoder

On the input side, most systems reuse a pretrained encoder rather than training their own. Self-supervised models such as wav2vec 2.0 [12], HuBERT [59] and WavLM [27], or the supervised Whisper encoder [118], are popular choices for extracting high-level features from the waveform. Training an encoder from scratch is less common and is mostly done when a streaming-capable codec is required, as with the Mimi codec used in Moshi (1.8.2).

A related design decision is the number of quantizer layers. A *quantizer* maps continuous encoder outputs to a discrete codebook of learned embedding vectors, replacing each frame with the index of its nearest entry and thereby producing the integer tokens a language model can consume. A *residual* quantizer (RVQ) stacks multiple such codebooks in series: the first codebook quantizes the raw encoder output, and each subsequent codebook quantizes the residual error left by all previous ones, progressively recovering detail. Multi-layer residual quantizers (EnCodec [40], SpeechTokenizer [180], SNAC [133], Mimi [41]) reconstruct high-fidelity audio but introduce latency, since each layer depends on the output of the previous one. Recent single-layer quantizers such as WavTokenizer [66] achieve competitive results while drastically reducing the burden of autoregressive modeling and therefore look promising for dialogue systems.

### 1.5.2 Language Model

The language model is the core of the system and is almost always a decoder-only Transformer [38]. The dominant strategy is to start from an existing pretrained text LLM—most commonly from the Llama [100] or Qwen [164] families—so that the system inherits the linguistic knowledge and reasoning abilities learned from large text corpora. Training the backbone from scratch (cold initialization) is possible, but it generally performs worse and converges more slowly; Hassid et al. [55] showed that initializing from a text model clearly outperforms cold initialization. Moshi (1.8.2) is a notable example that pretrains its own text backbone (Helium) before adding the speech modality.

### 1.5.3 Vocoder

On the output side, a vocoder (token-to-speech synthesizer) converts the generated tokens back into a waveform. GAN-based vocoders, especially HiFi-GAN [73] and its unit-based variant, are by far the most common because of their fast and high-fidelity synthesis. When the system emits acoustic codec tokens, the matching codec decoder can be reused directly without any additional training. As with the other components, off-the-shelf vocoders are preferred over training a new one. The choice is, however, tied to the type of tokens produced: acoustic tokens carry enough detail for direct synthesis, whereas semantic tokens usually have to be first enriched into an intermediate representation, such as a mel-spectrogram [65].

## 1.6 Training

### 1.6.1 Main Strategy

At the highest level, a spoken dialogue model takes a sequence of audio (and optionally text) tokens as input, updates an internal state, and predicts the next token(s) at each step—either for the model's own speech stream, for a joint text stream, or for both simultaneously. Because audio tokens are continuous-time and arrive in real time, the model must process them causally and, in the full-duplex case, do so for both the user and itself in parallel. The training objective follows the same autoregressive next-token-prediction scheme used in text LLMs, extended to multiple token streams and, in duplex systems, to two speaker channels at once.

Adapting a text LLM into a spoken dialogue model requires aligning the speech and text modalities. WavChat [65] distinguishes five architectural paradigms for how text and speech generation are coupled, ordered roughly from the simplest to the most ambitious:

**Text-output only.** The LLM keeps its text-only output and only learns to "listen" through an audio encoder and adapter. This is essentially the speech-aware cascaded approach (1.2.2): it excels at audio understanding but cannot speak on its own and needs an external TTS, so it is of limited interest for our goal.

**Chain-of-modality.** The model first generates the whole text answer autoregressively and then converts it into speech tokens (e.g. SpeechGPT [175] or EMOVA [24]). The text guides the speech and improves quality, but the model cannot start speaking before the text is finished, which causes high latency and makes it unsuitable for live interaction.

**Interleaving text and speech tokens.** Text and speech tokens are concatenated into a single stream using their temporal alignment, as in SpiRit-LM [105] and USDM [72]. This improves both understanding and generation, but the discrete units used tend to lose some paralinguistic information.

**Parallel generation of text and speech.** The model emits text and speech tokens simultaneously on the assistant side, transferring the LLM's reasoning into

the speech stream while keeping latency low. This is the approach of Moshi (1.8.2), Mini-Omni [158], LLaMA-Omni [48] and PSLM [101]. TurnGuide [39] extends this idea specifically by segmenting the assistant channel into turns and interleaving turn-level text chunks with speech chunks, demonstrating that turn-granularity text guidance substantially outperforms Moshi's token-level approach on semantic coherence.

**Speech-to-speech.** The most ambitious paradigm removes the intermediate text entirely, which minimizes latency and is conceptually closest to human conversation. SyncLLM [146], IntrinsicVoice [179], Align-SLM [88] and OmniFlatten [178] take steps in this direction, but text-free generation is still hard and the available conversational data is scarce.

### 1.6.2 Multiple Stages

The training itself is usually split into several stages [65]:

**Text LLM pre-training.** The first stage produces a text-intelligent backbone. As noted above, most systems simply reuse an existing pretrained LLM rather than training one themselves, which is both cheaper and more effective; cold initialization is the alternative.

**Modality adaptation and alignment.** The backbone is then post-trained to understand and generate speech, typically through cross-modal tasks such as ASR and TTS that tie the two modalities together in a shared space. Providing the system with explicit temporal alignment between speech and text—word-level interleaving as in SpiRit-LM [105], or word-level alignment priors as in Moshi (1.8.2)—has been shown to improve this modality alignment [65]. The main risk at this stage is catastrophic forgetting: the comparatively small amount of paired speech–text data can erode the model's original text abilities [150], so careful parameter selection and data balancing are needed. In practice, many pipelines therefore rely on parameter-efficient fine-tuning methods such as low-rank adaptation (LoRA) [60], which inject low-rank adapter matrices into selected linear layers (often attention projections) so that only a small number of additional parameters are trained while the backbone weights remain frozen or nearly frozen. An alternative is to sidestep this stage for the user stream entirely: SALM-Duplex [61] routes user input through a pretrained streaming encoder with a lightweight adapter, requiring no dedicated speech–text pretraining and demonstrating that competitive duplex performance is achievable with significantly less speech data.

**Instruction tuning and dialogue fine-tuning.** Next, the model is fine-tuned on instruction-following or dialogue data to make it conversational. Much of the interesting work happens here: Moshi (1.8.2) builds a realistic multi-stream dialogue dataset that includes noise and overlap, while OmniFlatten [178] and SyncLLM [146] use multi-stage procedures that progressively teach half- and full-duplex behavior.

**Preference optimization.** A final, optional and still largely unexplored stage aligns the model with human preferences. Direct preference optimization (DPO) [119] has so far mostly been applied to TTS and text-based LLMs rather than to spoken dialogue models; Align-SLM [88] is a pioneering attempt that uses AI-generated feedback to apply DPO to a textless spoken language model.

Alignment techniques such as reinforcement learning from human feedback (RLHF) and DPO are now beginning to be applied directly to spoken dialogue systems. Wu et al. [156] propose an offline preference-learning framework for Moshi (1.8.2) using more than 150,000 preference pairs derived from real user interactions. Unlike text alignment, their framework explicitly addresses timing-related behaviors such as interruptions and delayed responses. The resulting model demonstrates improvements in both question answering and safety, suggesting that preference optimization is becoming an important component of spoken dialogue training pipelines.

### 1.6.3 Resources

Training a spoken dialogue model draws on data from many areas. The modality-alignment stage relies on large ASR and TTS corpora—LibriSpeech [112], Libri-Light [69], GigaSpeech [22], VoxPopuli [149], Common Voice [9], MLS [117], and others—which provide paired speech and text. The instruction and dialogue stages reuse text instruction or conversation datasets such as Alpaca [95], UltraChat [43], and Open-Orca [78], as well as the few real conversational corpora, most notably Fisher [34].

Because suitable spoken data are scarce, most such data are generated rather than recorded. Text dialogue or instruction data are first filtered, removing code, formulae, URLs, and overly long passages, and then synthesized into speech using a TTS system, often with a fixed voice for the assistant and varied voices for the user. The synthesized streams are then arranged to simulate natural timing—turn-taking, pauses, and well-timed interruptions—and augmented with gain changes, background noise (e.g. from MUSAN [136] or the DNS challenge [120]), and simulated echo to better match real recordings.

Beyond data, compute is the other major barrier. Large-scale SLMs such as Moshi (1.8.2) and SpiritLM [105] required hundreds or thousands of GPU-days to train. Maimon et al. [97] challenge this assumption directly, showing that a competitive generative SLM can be trained on a single A5000 GPU within 24 hours by carefully selecting the base model family, using TTS-synthesized data, and applying a short DPO fine-tuning phase—results that exceed the performance predicted by SLM scaling laws for the same compute budget. This suggests that the barrier to SLM research is lower than previously assumed.

A striking gap is that, to our knowledge, there is no large dataset of natural human data in which a person plays the role of the "robot" in the kind of service-oriented conversations described in Section 1.1.2. Collecting such data is expensive; Araki et al. [8] use the Wizard-of-Oz method to gather dialogue data, which is one way around the problem but does not scale easily.

## 1.7 Evaluation Techniques

Evaluating spoken dialogue models is difficult, and the field still lacks agreed-upon test sets, metrics, and benchmarks. Evaluation is often piecemeal—each work picks a few capabilities to measure rather than assessing the system as a whole, which makes cross-model comparison difficult [10]. Seyssel and Dhekane [131] provide a taxonomy for speech model evaluation. Following WavChat [65], we review the main dimensions along which these systems are evaluated.

Evaluations are either objective or subjective. Objective, automatic metrics (accuracy, WER, BLEU, RTF, etc.) are preferred because they are cheap and reproducible [10]. Subjective evaluation relies on human listeners and remains indispensable for qualities that resist automation, above all, naturalness; MOS (mean opinion score) and its variants are the standard tools, and naturalness MOS (N-MOS) in particular is a common subjective measure of conversational naturalness [104]. In practice, a combination of automated benchmarks and human studies is needed to judge a dialogue system fairly.

### 1.7.1 Text Intelligence

Text intelligence concerns only the semantic content of the answer, ignoring timbre, emotion, and style; the generated speech is transcribed and the resulting text is scored. Two kinds of metric are used. *Acc-metrics* measure accuracy on closed-ended questions with short answers, typically using LLM benchmarks such as MMLU [56] or GSM-8K [35]. *MT-metrics* target open-ended questions that have no single correct answer; they borrow measures of similarity to a reference from machine translation (BLEU [113], METEOR [13], ROUGE [81]). However, some of the basic automatic NLG metrics have been shown to perform poorly compared to human judges [106]. To get an improved correlation with human preference, recent works use a strong textual LLM as a judge [181]: the model is prompted to score or compare responses according to a rubric, providing a scalable alternative to human evaluation that has been shown to correlate well with human preference on open-ended tasks.

### 1.7.2 Speech Intelligence

Speech intelligence is what distinguishes a spoken model from a text-based one: the ability to understand and generate acoustic information beyond the words. On the *understanding* side, evaluation usually targets paralinguistic recognition—since the classes are limited (e.g., negative, neutral, positive sentiment), accuracy or F-score are the natural metrics, and several emotion datasets are available. Whether a system can generate an *appropriate* response conditioned on the acoustic input, for example by answering differently to a cheerful versus a sad tone, is far less explored and is usually measured by transcribing the answer and comparing its content to a reference. On the *generation* side, the focus is controllability—producing speech in a requested style, accent, speed, or energy—and voice cloning, for which speaker-similarity metrics borrowed from zero-shot TTS are used [65, 10].

### 1.7.3 Speech Quality

Speech quality has two common dimensions. The clarity and naturalness of the audio are judged subjectively with a mean opinion score, or can be estimated automatically, e.g., using UTMOS [125], a machine learning system trained to predict MOS values. The robustness of the audio—missing or extra words with respect to the text output (if present)—is measured objectively with the word error rate (WER) or character error rate (CER) computed on an ASR transcription.

### 1.7.4 Latency

Latency captures how quickly the system responds. The two usual measures are the time to the first generated token of speech—that is, the wait after the user stops speaking—and the overall real-time factor (RTF), the ratio of the model's generation time to the duration of the produced audio [65]. For reference, the average human response latency is around 200 ms [139], and reported figures vary largely with hardware and configuration [10].

### 1.7.5 Interaction and Naturalness

Interaction and naturalness are the hardest dimensions to quantify and the least standardized [29]. Basic interactive ability is the capacity to be interrupted at any moment and to react to the new input, commonly measured through the accuracy of detecting and handling the interruption. Beyond interruptions, systems are evaluated on how often they place discourse markers such as "okay" or "haha," and on turn-taking statistics derived from voice activity (gaps, pauses, overlaps) that compare the model's behavior with human conversations [104]. A further, mostly unexplored aspect is whether the system can revise its responses as the conversation evolves. Recent benchmarks such as Full-Duplex-Bench [86] have begun to assess pause handling, backchanneling, smooth turn transitions, and interruption handling in a reproducible way.

Recent work also suggests that interaction should not be evaluated in isolation from response quality. For example, Wu et al. [157] evaluate a full-duplex model with explicit reasoning during listening not only on turn-taking latency and barge-in handling, but also on response-quality benchmarks and human preference judgments. This is a useful direction for the field: improvements in dialogue intelligence are only convincing if they do not come at the cost of slower or less natural interaction.

### 1.7.6 Long-Term Capabilities

Long-term, multi-turn ability—effectively a memory that keeps a conversation coherent over a long time and across many contexts—is important but not yet well explored [65, 10]. It is usually evaluated on dedicated long-duration test sets with the same MT- or Acc-metrics as text intelligence, for example by asking context-dependent questions. A particularly relevant variant is when the user revises information mid-conversation and the system must react accordingly; how to evaluate this kind of understanding is still an open problem.

SpeechSSM [115] is directly aimed at this area: it is a hybrid state-space spoken language model specifically designed to generate coherent speech over multiple minutes without running out of memory, using an architecture with recurrent layers that maintain a fixed-size state rather than attending to all previously generated tokens. To evaluate this capability, the authors introduce the LibriSpeech-Long benchmark and propose time-stratified metrics—semantic coherence over length (SC-L) and naturalness MOS over time (MOS-T)—designed to quantify how well a model sustains coherence and acoustic quality as the generation grows longer.

### 1.7.7 Multilingual Capabilities

Multilingual ability is required in principle, but most current models cover only English and/or Chinese [65, 10]. A naive evaluation measures speech-to-speech or speech-to-text translation with BLEU or BERTScore, but translation alone does not capture conversational multilingual ability: in a real dialogue, the system is expected to reply automatically in the user's language and accent rather than to translate on command. Evaluating this—for example through language identification of the generated speech combined with human judgment—remains largely unexplored.

### 1.7.8 Compound Benchmarks

Several benchmarks bundle many of the above dimensions into a single suite. VoiceBench [28] focuses on general knowledge, instruction following, and robustness to noise and speaker variation; SUPERB [167] covers a wide range of speech-processing tasks; AudioBench [148], AIR-Bench [166], and MMAU [126] target audio understanding and reasoning; SpokenWOZ [132] addresses task-oriented dialogue; and SD-EVAL [7] evaluates understanding beyond words, including emotion, accent, age, and environment. On the pure language-modeling side, likelihood-based suites such as sWUGGY, sBLIMP [45], and spoken StoryCloze [55] probe lexical, syntactic, and semantic competence, and SALMon [98] adds acoustic and prosodic consistency. Interestingly, scores on different benchmarks are not always correlated—a system strong on content can be weak on acoustic detail—which is a strong argument for evaluating from several angles at once [10].

## 1.8 Notable Audio Language Models

In this section, we review the models we consider most important for our work, grouped by the way they approach the dialogue problem. Exhaustive tables of open-source models can be found in the surveys we follow [65, 10].

### 1.8.1 Textless Dual-Tower Models

A key step was made by dGSLM [104], which uses a dual-tower Transformer (one "tower" for each conversation side) connected by cross-attention to predict the continuation of a spoken dialogue. It is trained on the Fisher corpus (2,000 hours of two-channel telephone conversations [34]) and reconstructs the audio of

both channels with a HiFi-GAN vocoder. dGSLM achieves significant results in terms of output naturalness (N-MOS), but it falls far behind in meaningfulness: the system produces utterances that sound like interpersonal conversation, yet the individual words and phonemes are essentially babbling in which no distinct content can be distinguished. The reason is that it works purely from speech units and is not connected to any text LLM, so it has no real linguistic intelligence. Nonetheless, it represents a major step toward natural turn-taking, being the first fully end-to-end system to model spontaneous overlaps and backchannels directly from raw audio. SLIDE [90] directly targets this semantic gap: an LLM first generates the textual content of the dialogue, which is then converted into a phoneme sequence with a two-tower Transformer duration predictor, and finally dGSLM is conditioned on that spoken phoneme sequence to vocalize the result. The hybrid scheme preserves dGSLM's naturalistic turn-taking statistics (IPUs, overlaps, gaps, pauses) while reducing perplexity of the generated speech from 1,229 to 421—within 12% of real conversation—demonstrating that text-level semantics can be injected into a textless model without sacrificing paralinguistic fidelity.

NTPP [153] revisits the same dual-channel setup but replaces the two-tower encoder–decoder architecture with a single decoder-only transformer. Its central idea—*next-token-pair prediction*—is to jointly predict one token from each speaker channel at every time step, learning the joint distribution $p(S^a, S^b)$ directly rather than a conditional one. Like dGSLM, it is trained on the Fisher corpus without text supervision and extends naturally to RVQ tokenizers through cyclic depth embeddings, while a pair-wise causal masking scheme preserves conditional independence within each step.

### 1.8.2 Parallel Speech–Text Full-Duplex Models

#### Moshi

Moshi [41] is an open-source model that is closest to our goal and the one on which Warmblood builds (2). Its architecture follows the three-component structure described in Section 1.5—encoder, language model, and vocoder—but all three are designed from scratch rather than reusing off-the-shelf pieces.

**Helium.** The language model backbone is Helium, a 7B-parameter decoder-only Transformer pretrained from scratch on 2.1T tokens of English web text. Unlike most contemporaries that initialize from an existing LLM (Section 1.5), Moshi trains its own backbone to retain full control over the architecture before adding speech.

**Mimi.** The encoder and codec decoder are both provided by Mimi, a streaming neural audio codec. Mimi encodes a 24 kHz mono waveform at a frame rate of 12.5 Hz using a split RVQ with $Q = 8$ codebooks, each with a vocabulary of 2,048 entries. As discussed in Section 1.4, the two token families—semantic and acoustic—carry complementary information. Mimi resolves the tension between them by design: its first codebook is a plain VQ trained with knowledge distillation from WavLM [27] to capture *semantic* content (phonetic and linguistic structure),

while the remaining seven codebooks form a standard RVQ that encodes *acoustic* detail (timbre, prosody, fine reconstruction quality). The first codebook therefore acts as a compact semantic proxy, and the remaining codebooks handle high-fidelity reconstruction via the codec decoder. All convolutions in Mimi are causal, so both encoding and decoding operate in a streaming fashion with a frame stride of 80 ms.

**RQ-Transformer.** The generative model is an RQ-Transformer that processes the token streams produced by Mimi. A large *Temporal Transformer* (initialized from Helium) runs at the 12.5 Hz frame rate and models context across frames; a smaller *Depth Transformer* (six layers, 1,024-dimensional) then models the $Q$ codebook levels within each frame, following the residual quantization order. Separate linear projection heads per codebook level and an acoustic delay of one step between the semantic and acoustic codebooks stabilize training and improve audio quality.

Figure 1.2 summarizes these components and the multi-stream token flow.

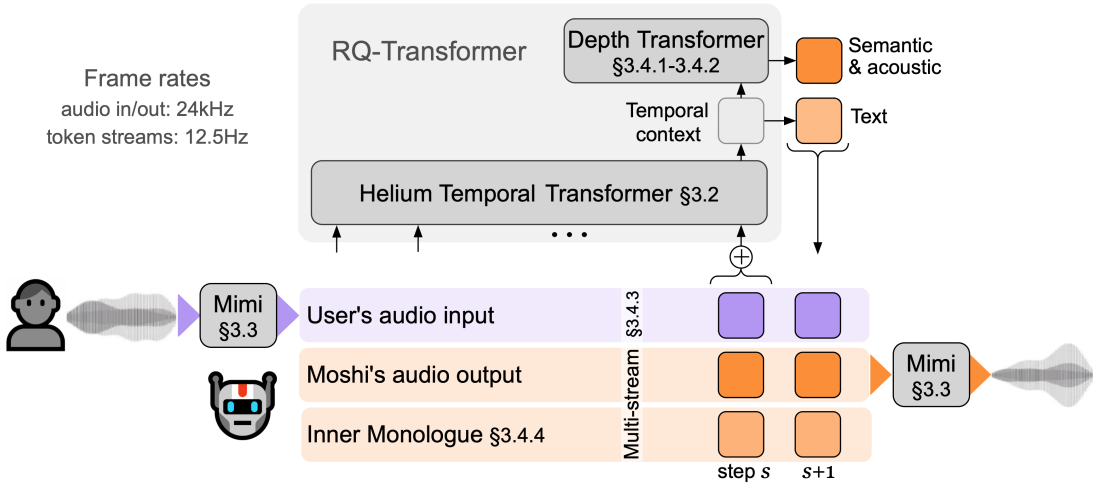


**Figure 1.2** Moshi architecture overview with Mimi-based audio tokenization, multi-stream inputs (user audio, assistant audio, and inner monologue), and RQ-Transformer generation. Adapted from the original figure in Défossez et al. [41], section numbers shown refer to the original paper.

The full-duplex capability comes from *multi-stream modeling*: Moshi represents both the user and the assistant as independent RVQ streams processed jointly by the same Temporal Transformer. At each time step the model processes $2Q + 1 = 17$ discrete tokens—eight audio tokens for the user, eight audio tokens for the assistant and one text token—with no concept of a speaker turn. When the assistant is silent, its audio stream decodes to a near-silent "natural silence" waveform rather than a special padding symbol. This design allows arbitrary overlaps, backchannels, and interruptions to emerge naturally from the data rather than from an explicit finite-state controller.

Moshi's most impactful idea is the *inner monologue*. At each 80 ms frame, instead of predicting a semantic audio token in isolation, the model first predicts a text token representing the word being spoken at that moment—acting as an explicit textual "thought" that guides the audio generation. Word-level timestamps

derived from Whisper transcriptions align text tokens to audio frames during training; two special tokens (PAD and EPAD) fill the gaps between words. Not only does this drastically improve linguistic quality and factual accuracy of generated speech[1] but the same mechanism, with a different text–audio delay, can be repurposed as a streaming ASR or TTS system without any architectural change. Forcing the sampling of an EPAD token at inference time also causes Moshi to start speaking immediately, reducing response latency.

**Extensions and Related Systems**

PersonaPlex [124] is a Moshi-based extension that addresses the inability of existing full-duplex models to adopt different roles or voices. Its *Hybrid System Prompt* prepended to each conversation consists of two segments: a *text prompt segment* conditions the model on a role description, while a *voice prompt segment* supplies a short reference speech sample for zero-shot voice cloning. Neither the architecture nor the training objective is changed; the intervention is purely at the data level.

MoshiRAG [32] is a modular extension of Moshi that improves factual accuracy by asynchronously injecting retrieved information during generation. It exploits the natural *end-to-end keyword delay*—the two to five seconds between the end of the user's turn and the point where factual content is needed—to complete a retrieval call in the background; the result is encoded and injected into Moshi's Temporal Transformer input without interrupting the conversation. This is a concurrent work to Warmblood's guide mechanism (Section 2.1), which similarly injects externally generated text into Moshi during inference; the key difference is that Warmblood's guide originates from an LLM rather than a retrieval back end.

Moshi's architecture has also inspired several adaptations to new languages and tasks. J-Moshi [107] extends Moshi to Japanese, demonstrating how the system can acquire language-specific dialogue characteristics such as increased speech overlaps when trained on suitable data. Hibiki [74] adapts the multistream architecture to simultaneous speech-to-speech translation, showing that the framework generalizes beyond dialogue to real-time interpretation tasks. F-Actor [184] takes the parallel-stream design in a different direction, focusing on *controllable conversational behavior*: a text instruction prepended to each dialogue specifies the target speaker voice, the conversation topic, the desired counts of system backchannels and interruptions, and which participant should initiate the dialogue. Unlike PersonaPlex, which addresses role and voice but not proactive dialogue behaviors, F-Actor is an open, fully released instruction-following full-duplex model; it uses a NanoCodec—a lightweight audio codec based on finite scalar quantization (FSQ), which replaces the lookup-table codebooks of RVQ with a simple per-dimension rounding scheme and achieves comparable audio quality with a smaller model and shorter sequences—rather than Mimi, and requires only 2,000 hours of data and two days on four A100 GPUs to train. These examples illustrate Moshi's versatility as a foundation for developing specialized spoken language systems.

Sesame's CSM [127] aims to improve voice naturalness with a single-stage model directly on RVQ tokens. Two autoregressive Transformers operate in tandem: a larger Llama-style backbone predicts the first codebook from interleaved text

[1] It nearly triples spoken question-answering accuracy compared to a model without it.

and audio tokens, while a smaller decoder models the remaining codebook levels; a compute amortization scheme trains the decoder on only 1/16 of frames to keep training tractable. The publicly available demo shows impressively natural interaction, though only a small 1B variant has been released, whose quality falls short of the demo.

### 1.8.3 Time-Multiplexed Full-Duplex Models

An alternative model architecture achieves full duplexity not through parallel streams but by chunking time and interleaving the two speakers in a single sequence, which allows it to retain a standard decoder-only backbone. SyncLLM [146] divides the conversation into fixed time chunks and predicts the user's upcoming speech before each system chunk in order to stay synchronized with the real-world clock, handling overlaps, interruptions, and backchannels. OmniFlatten [178] flattens user and assistant, speech and text, into one stream and is trained in three progressive stages (half-duplex, then removing the user's and finally the assistant's text stream) to reach text-free full-duplex dialogue. Freeze-Omni [154] keeps the LLM frozen and uses a chunk-wise state predictor to decide when to listen, interrupt, or respond, shifting the dialogue logic to the encoder and decoder.

DuplexCascade [165] upgrades a classical ASR-LLM-TTS pipeline by LoRA-fine-tuning the LLM on text-only dialogues to emit conversational special tokens governing turn-taking without any VAD, preserving the backbone's reasoning ability while achieving competitive Full-Duplex-Bench results. FlexDuo [79] acts as a plug-and-play controller that wraps any half-duplex LLM in a three-state finite-state machine (Listen, Speak, Idle), where the Idle state suppresses backchannels and noise before they can pollute the dialogue context. SALMONN-omni [172] is a codec-free full-duplex speech model that can keep listening (to itself and the background) while speaking, using a duplex dialogue setup to coordinate asynchronous text and speech generation.

### 1.8.4 Text-Guided Parallel Generation

Several open models reach low latency by emitting text and speech in parallel with a delay pattern, while still relying on an intermediate text stream. Mini-Omni [158] applies text-instructed delayed parallel decoding over SNAC tokens, together with a batch-parallel trick that transfers the backbone's text reasoning to speech; Mini-Omni2 [159] adds multimodal input and a command-based interruption mechanism trained on a specially constructed dataset. LLaMA-Omni [48] attaches a streaming, non-autoregressive speech decoder to a LLaMA backbone to generate text and discrete units simultaneously. These systems focus on a traditional turn-taking structure and therefore have only a limited ability to model spontaneous turn-taking.

### 1.8.5 Other Notable Systems

`gpt-realtime-2.1` [110] is the flagship commercial system (the audio mode was first introduced in GPT-4o [111]); its voice mode feels close to human conversation, but its implementation and training data are undisclosed, which makes it

impossible to reproduce or analyze rigorously. MinMo [26] and GLM-4-Voice [174] are large, recent voice-interaction models; GLM-4-Voice in particular demonstrates fixed-ratio interleaved text–speech generation without forced alignment, scaled up with synthetic interleaved data.

Qwen2.5-Omni [160] extends the Qwen family to a full omni-modality model (text, audio, images, video) with a *Thinker-Talker* architecture: a standard LLM generates text while a lightweight dual-track decoder consumes its hidden states to produce speech tokens concurrently. Step-Audio 2 [62] integrates a latent audio encoder with chain-of-thought reasoning and reinforcement learning, achieving strong results on paralinguistic understanding, speech translation, and spoken QA while also supporting RAG (retrieval-augmented generation) and tool calling.

SpeechSSM [115] replaces the Transformer with a state-space model to produce more natural and semantically consistent long-form (minutes-long) speech, targeting the long-term capabilities discussed above. TurnGuide [39] takes a different direction and builds on GLM-4-Voice fine-tuned on Fisher, using dynamic turn segmentation to interleave turn-level text guidance into the duplex audio stream; it achieves state-of-the-art results on Full-Duplex-Bench.

# 2 Architecture

We chose Moshi [41] (described more in Section 1.8.2) as the core model on which to build Warmblood. Most importantly, Moshi's architecture with two parallel audio channels that are supplied to the model at all times influenced our key decision because it matches our desired characteristics. The token-level multi-stream structure that motivated this choice is illustrated in Figure 2.1. The Moshi team released their entire Python codebase online, including the main model code, a runnable server, and fine-tuning code, along with English documentation. The authors experimented with many different setups and thoroughly documented all their techniques and findings. These made it possible to begin development immediately.

Relying on benchmark results alone can be problematic for full-duplex speech. Moshi's advantage was that it had an online demo on which we could test its readiness for natural speech. Among the open-source models, it seemed the most prepared for live conversation.

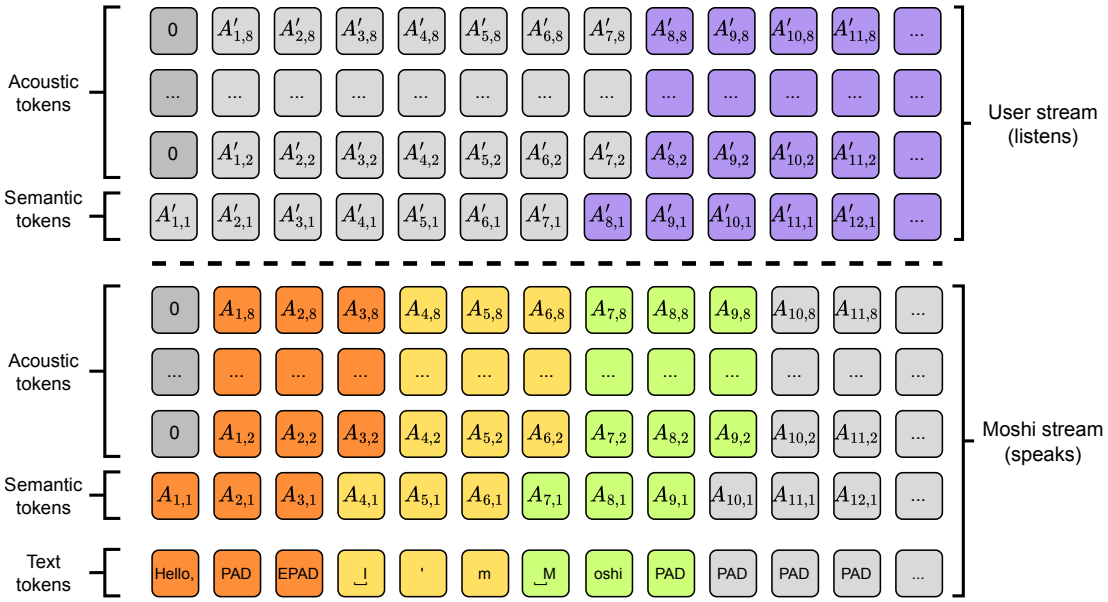


**Figure 2.1** Token-level view of Moshi's multi-stream setup: user audio stream, assistant audio stream, and inner monologue text stream feeding the temporal transformer.

At the time of Moshi's release, there were not many projects that used and improved existing models. Almost all research focused on developing new models, preparing data, and evaluating them.

The situation has changed since then. Some prominent works similar to ours also chose Moshi as their base, which confirmed our reasoning [124, 32]. As of June 2026, Moshi is among the most cited models in this area of research.

Section 2.1 introduces the guide mechanism that forms the core architectural contribution of this work and explains how guides are incorporated during training and inference. Section 2.2 describes the inference framework: the real-time loop that combines the main model thread with external ASR and guide generation, and the prompting strategies.

## 2.1 Guide

A **guide** is a short text that comes from an external source and carries important information; for example, factual data, exact numbers or instructions on what to say. Moshi (1.8.2) uses multiple token channels in the input of its temporal transformer: user audio and model audio—several channels coming from RVQ, both semantic and acoustic—and model text (the "inner monologue"). At each time step, the tokens are summed together. The model is trained to produce a text token while simultaneously generating audio via a depth transformer (which also uses the text token). We extend this functionality by adding a new token channel into which the guide text tokens are exclusively fed. This channel is then summed with the others during generation, as illustrated in Figure 2.2.

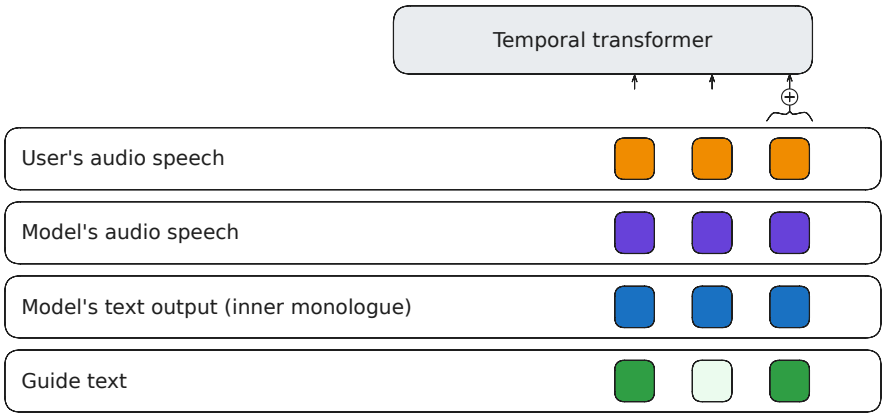


**Figure 2.2** Simplified token-stream architecture with the additional guide channel injected into Moshi's multi-stream temporal input. The guide may also be empty as indicated.

The idea of an external guide came from our initial observations. When testing Moshi, we noticed that it could sometimes get "distracted"—the model would latch onto some words that it had already generated and keep repeating either the word or a sentence, with no way for the user to escape this situation (apart from restarting the session). Because current transformer outputs are non-deterministic, the model could randomly deteriorate into a state in which it misbehaves and cannot be recovered by any means other than a complete restart. Furthermore, its long-term capabilities were limited, and the output quality would degrade over longer conversations. The key observation is that these issues can be detected but not corrected from the outside while the system is running. This led us to the idea of an external guide that would tell the model what to do and keep it on track.

Our contribution was inspired by Toolformer [129]. Text-based LLMs used to struggle with tasks that required factual data and precise information. In Toolformer, the LLM was fine-tuned to use external tools for specific tasks (e.g., information retrieval, the current time, etc.). This approach has since been incorporated directly into LLM designs, and contemporary models leverage tool calling by default. Similarly, we decided to use an external textual source to guide the real-time spoken dialogue model (Moshi). A closely related concurrent work is TurnGuide [39], which also injects text into a Moshi-style duplex audio stream, but derives that text automatically from within-corpus ASR transcriptions

segmented by VAD-based turn boundaries. Our approach differs in that the guide originates from an external source, making it task-agnostic and capable of supplying information the base model has never seen.

The guide offers many potential benefits. It can give the agent a purpose: instead of defaulting to a generic chatbot, it can be instructed to act as a receptionist, a customer-support agent, a research assistant, or to fulfill any of the other use cases discussed in Section 1.1.2. Beyond role selection, the guide can also steer the conversation toward an explicit goal, maintain a lightweight dialogue state (e.g., which stage of a workflow the system is in), enforce constraints (policy, tone, and verbosity), and prompt for targeted clarifications when user input is ambiguous or incomplete.

There are many possible sources for the guide. The most straightforward is a text-based LLM that directs the output, but it could also be a more elaborate system with an explicit decision process. For instance, it could combine a fast rule-based component (to provide immediate steering signals such as the current state and next required question) with slower external processes (knowledge-base retrieval, database lookup, or other tools) whose results are injected into the guide text when they become available.

The guide does not have to be used at all times. We assume that the agent can maintain a conversation on its own when no guide is available.

For the guide text, we use the same tokenizer that the base model uses for its inner text. This brings the benefit that the model can directly understand the meaning of the newly added tokens. However, it also carries the drawback that the original tokens and the new ones must be distinguished, since they essentially appear on top of each other with no clear distinction.

The guide is inserted at specific moments, as described in the following sections. Whenever a guide text appears, its tokens are inserted directly one after another with no padding. In contrast, the model's "regular" text tokens are spread out because they are padded to align with the higher-frequency stream of audio tokens. Removing padding makes the guide stand out and enables denser information packing (more words per second).

### 2.1.1 Guide in Training Data

The datasets that include the guide (described in Section 3.1) are all synthetically generated without dynamic speech overlap. Therefore, it is clear when a user turn starts and when an agent turn starts. The guide is inserted after the user's speech and before the assistant's response. The guide can be longer, so its tokens may be fed alongside the assistant's answer once it starts speaking, as shown in Figure 2.3.

Since multiple pieces of information may need to be included at once, the guide was kept as short as possible. It contains no filler words and often omits verbs—just the essential pieces of information that, for example, a human could read from a cue card. It is the responsibility of the model to attend to these words and understand how to use them when constructing full sentences.

Although it would be possible to insert an initial guide at the beginning of a conversation, we did not use one in training, because we assumed that the model had to start producing output right away, when no guide would be ready yet.


**Figure 2.3** Guide placement during training: after user tokens and before/alongside the assistant response tokens.

## 2.2 Inference

Several options are available during inference. We leave the model's sampling parameters at Moshi's defaults and instead focus on guide management. Even though the model weights are fixed after fine-tuning, we can theoretically improve performance on future benchmarks through strategic guide placement at runtime.

Text-based LLMs are among the most suitable systems for guide generation. When running the model in real time, the guide has to be generated with sufficiently low latency, and an external system must process the conversation and decide which words to insert into the next input. LLMs can start producing output tokens within a very short time while maintaining high quality and coherence, and they are versatile across a wide variety of scenarios. Unlike a complex decision-based system, an LLM is very easy to set up—all that is needed is a natural-language prompt.

To align the guide timing with training, the guide is placed as soon as possible after the user has finished their utterance.

Choosing LLMs for this task brings several advantages. They can keep track of what has happened in the conversation by including the textual history in the user prompt. The textual conversation is much more compact than the history of audio tokens that Moshi has to attend to. The LLM can maintain a much larger and more stable context window, which can be used in longer-form dialogues to help the assistant remember. Modern LLMs also support function calling, which can extend their range of capabilities. However, external calls may introduce additional delays and negatively affect the overall latency, so they should be approached with caution.

A natural extension of this design would be a two-tier LLM setup: a smaller, faster model continuously reads the dialogue and produces guides with minimal latency, while a larger model is invoked only for requests that require deeper reasoning or external tool use. A related two-tier philosophy appears in Zhang et al. [176], where a lightweight LLM acts as a dedicated dialogue manager producing turn-taking control tokens, while a larger model handles response generation—though in their case the small LLM drives the primary output rather than generating an auxiliary guide.

The core principle of Warmblood's inference is to convert the conversation to text, send the transcription to an LLM, and return the response to the model as a guide. The framework is designed as a real-time loop that continuously receives user audio, updates the conversational state (see below), and streams model output back to the client. At a high level, it combines three cooperating

parts: the main model thread, automatic speech recognition, and guide generation logic.

**Main model thread.** The central thread handles interaction with the main model. Incoming user audio is processed frame by frame, transformed into model inputs, and decoded into output speech that is immediately sent to the client. In parallel, text fragments generated by the model are tracked as part of the conversation state.

**The conversation state** is not a complicated data structure; it is simply a text log of the whole conversation history (with speaker annotations).

**ASR integration.** ASR converts spoken user input into text so that the external LLM can react to the actual dialogue content. No transcription is needed for the assistant's speech, because it already produces text by default.

We use a voice activity detection model before ASR. Once the end of speech is detected, ASR is run on the most recent input.

**Guide generation.** The guide is generated by an external LLM. It receives the whole conversation so far (including previously generated guides—to prevent repetition) and produces a new guide. It can also opt to leave the output empty if it determines that no guide is necessary at that moment. This process is triggered at a specific time. There are multiple trigger strategies available:

- **End-of-turn trigger:** when the user finishes their utterance (detected by VAD) and the ASR transcription is complete, a request is sent and a guide is generated as soon as possible. This is the approach we selected and used.
- **Periodic guidance:** generate prompts at fixed intervals, independently of user turn boundaries. This strategy can be somewhat unpredictable, since it can work both ways. Sometimes it may request a guide after the user has already said the important part—lowering guide latency. At other times, the interval may fall well after the user has finished, increasing latency; and the previous request might have been sent when an important detail had not yet been stated by the user, causing the LLM to produce an incorrect guide.
- **Model-driven triggering:** this is an alternative approach that would require further orchestration during training. It would require training the model itself to produce a guide request (e.g. via a special token). Our key decision was not to put this extra strain on the model; we want to retain external control over this.

We settled on the end-of-turn strategy because it is closest to the fine-tuning setup, is the most stable, is easy to control, and does not send many unnecessary requests.

If a new guide is generated and a previous one has not been fully fed to the model, the rest of the older one is discarded and the new one is processed instead.

The overall design makes it possible to tune the system for either stronger conversational control (more frequent guidance) or more spontaneous behavior (less intervention). For experimental work, the key advantage is modularity: the prompting policy can be changed independently while keeping the same core model loop.

### 2.2.1 Guide LLM Prompts

System prompt design for the guide LLM acts as a lightweight control policy. The main practical caveat is the trade-off between control and naturalness. The LLM itself can decide not to produce any response (leaving control to the main model). If guidance is emitted too frequently, the conversation may become over-steered and less spontaneous. If guidance is emitted too rarely, it has little effect.

Another caveat is error propagation from ASR. Since the guide LLM prompt depends on transcribed turns, transcription errors (or a mismatch between the model's audio and text) can lead to guidance that is coherent with the transcript but misaligned with the original speech intent. In addition, strict output constraints (very short guides, constrained wording, and format restrictions) improve consistency but may occasionally suppress useful nuance in ambiguous turns.

For the experiments in this thesis, we selected and tuned a general-purpose prompt that should be ready for most situations (included in Appendix A.3.4). This variant was chosen as a compromise between generality and control strength: it supports both factual and conversational guidance, includes explicit rules for concise outputs suitable for real-time use, and remains usable across heterogeneous dialogue intents without requiring immediate prompt switching. It tends to produce somewhat more talkative guides in order to clearly steer the model in the right direction and avoid making it misunderstand concepts.

At the same time, the architecture keeps prompt selection modular. When interaction goals are known in advance, specialized variants can be used as drop-in alternatives to prioritize a specific behavior profile (for example, more focus on factual precision or open-ended chitchat engagement).

# 3 Experimental Setup

This chapter describes the experimental setup used to develop and evaluate Warmblood. Section 3.1 covers the training data: the survey and selection of candidate datasets, the preparation pipeline, the final dataset composition, and potential improvements. Section 3.2 presents the fine-tuning configuration, including the LoRA setup and training hyperparameters. Section 3.4 defines the internal evaluation datasets and the test scenarios. Section 3.5 describes the metrics used to assess model outputs.

## 3.1 Data

With the architecture defined, a key consideration was how to structure the training data used to fine-tune the model to:

- Use the information from the guide. The model should be able to answer according to the instructions and information from the LLM.
- Lead a natural conversation without the guide and interleave the guide contents into its ongoing output. The model should be highly interactive, able to determine whether the user has finished speaking or is merely pausing, and react to interruptions correctly.
- The model does not need to be smart or to store large amounts of factual data; it should rely on the external guide for that.

### 3.1.1 Existing Datasets

We present a selection of existing datasets that we considered for our purposes in Table 3.1.

There are some initial requirements we set:

- It is essential that the data is in dialogue form. Warmblood is trained on situations in which it speaks with a single user.
- The audio has to be separated into two channels. An alternative is to apply speech separation [140, 67] to single-channel audio; however, this introduces additional processing and a possibility of errors, so we chose to avoid this approach as long as alternatives were available.
- We rejected meeting recordings, because they capture a different domain from the one our agent is intended for.

When filtering sources, we assumed that the base model was trained well enough to understand speech data in general. Therefore, we discarded TTS and ASR datasets, since their main purpose would be to align the model's speech and text understanding.

We list other possible sources that could be gathered in practice but are not easily available or carry other caveats:

| Dataset | Size | Type | Description |
|---|---|---|---|
| Fisher [34] | 2k hours | Phone conversations | Natural 10-minute dialogues between pairs of participants, separated into 2 channels. |
| CANDOR [121] | 850 hours | Video-chat conversations | Natural ~30-minute dialogues between pairs of participants, separated into 2 channels. |
| Switchboard [52] | 260 hours | Phone conversations | Natural short dialogues between pairs of participants, separated into 2 channels. |
| InstructS2S-200K [48] | 100 hours | Synthetic audio conversations | A fine-tuning dataset for a similar model. Instructions and responses. |
| UltraChat [43] | 1.5M dialogues | Synthetic text conversations | Short-to-middle length questions and longer LLM responses. |
| TriviaQA [68] | 95k questions | Gathered texts | Trivia questions and answers. |
| CANARD [47] | 40k questions | Crowdsourced texts | Multi-turn question and answer dialogues. |
| VoiceAssistant-400k [158] | 400k questions | Synthetic texts and audio | Audio and textual questions with short text answers. |
| Behavior-SD [77] | 2.1k hours | Synthetic audio and text | Short dialogues that are meant to sound natural. Made using TTS. |
| MultiDialog [114] | 340 hours | Recorded readings | Several people independently recorded reading scripted open-domain dialogues. |
| ABCD [21] | 10k dialogues | Manually created texts | Task-oriented dialogues focused on customer service. |
| MetalWOZ [76] | 38k dialogues | Crowdsourced texts | Task-oriented dialogues of 10 turns from 47 domains collected using Wizard-of-Oz[a]. |
| MultiWOZ [19] | 10k dialogues | Crowdsourced texts | Task-oriented multi-turn dialogues from 7 common domains collected using Wizard-of-Oz[a]. |
| FusedChat [171] | 10k dialogues | Modified texts | MultiWOZ with prepended/appended open domain conversations that do not require the task-specific interface. |
| ClariQ [5] | 11k question-answer pairs | Crowdsourced texts | Clarifying questions to ambiguous user requests in open domain. |
| FaithDial [46] | 5.5k dialogues | Crowdsourced texts | Knowledge-grounded conversations that are faithful to a source of information aimed to prevent hallucinations. |

*Note:* [a] Wizard-of-Oz (WoZ) data collection means that a human operator secretly plays the role of the system/assistant while users believe they are interacting with an automated system.

**Table 3.1** Selection of existing dialogue datasets considered for our use case.

- **Podcast recordings** offer natural one-on-one conversations. Interview-style podcasts are better suited, because they involve more interaction and are asymmetrical (similar to the robot-human dynamic). Many podcasts feature two people discussing a topic without acknowledging each other much, though. Their other downsides are copyright issues and the fact that the audio is typically not separated into two channels.
- **Help-desk recordings.** If a large company records the phone calls of its customer-support department and has the consent of the customers, it could use this data to train a natural dialogue system that would act in place of its staff. The public availability of such data is very rare, however.
- **Radio recordings** are often publicly available, but the copyright status is not always straightforward. Broadcasts also frequently include music, advertisements, and other non-speech audio, so substantial filtering is needed to isolate the interview segments. Even then, the extracted material would likely not be separated into two clean speaker channels.

### 3.1.2 Preparation Pipeline

The preparation pipeline converts dialogue data into training-ready stereo audio samples paired with word-level transcript alignments and guide annotations. We implemented two variants of the pipeline: the main pipeline operates on textual dialogue data and synthesizes audio from scratch, while the alternative pipeline processes existing conversational audio recordings and skips the synthesis stages. An overview of both paths is shown in Figure 3.1.

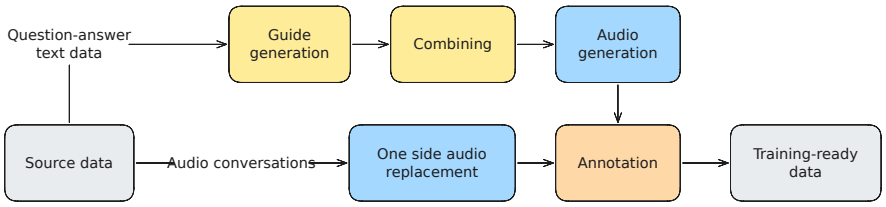


**Figure 3.1** Schema of the data preparation pipeline, including the synthetic-text branch (guide generation, combining, audio generation) and the natural-audio branch with one-side voice replacement and shared annotation/output stages. Text processing is shown in yellow, audio generation in blue, and audio processing in orange.

#### Main Pipeline

The main pipeline applies to purely text-based dialogue datasets. It begins either with single-turn question-answer pairs that are concatenated, or with complete textual dialogues (with defined user and agent roles).

**Guide generation.** We generated guides using GPT-4o-mini [111] and GPT-4.1-mini [109]. The exact format and prompt depend on the dataset. For Q&A, the guide is usually a comma-separated list of keywords and key phrases that captures the essential facts of the gold-standard agent's response from the dataset. If the answer is not present or needs to be discarded because it is too long or

unsuitable, it can also be generated with an LLM first. For task-oriented dialogues (TOD), it is better to feed the entire conversation to the LLM as context in order to generate a guide for the final agent turn. The prompt templates used for this stage are listed in Appendix A.3.1; legacy variants are provided in Appendix A.3.2.

**Combining.** For Q&A datasets, individual single-turn pairs that already carry guide annotations are concatenated into multi-turn dialogues. The number of Q&A pairs per dialogue is sampled from a specified range, and multiple source datasets can be mixed according to configured weight ratios. This step produces complete, guide-annotated dialogues ready for audio synthesis.

**Audio generation.** Speech is synthesized for each turn using CosyVoice 2 [44], a zero-shot text-to-speech system that clones a speaker's voice from a short reference recording. User turns and agent turns are generated with separate reference voices: a single fixed voice is used for all agent turns across the dataset, while for the user a voice is drawn at random from a pool of reference recordings, with one voice remaining consistent throughout a single dialogue. The per-turn audio segments are concatenated and assembled into a single stereo file, with the agent occupying channel 0 and the user occupying channel 1.

We used LibriTTS [173] to source all voices. We selected 186 voices from the *train-clean-100* subset for the synthesized training users, with an additional 20 and 20 voices (from the same subset) set aside for the dev and test splits, respectively. For the agent, we chose a single, fixed voice.[1] We selected one female sample from the *test-clean* subset of LibriTTS; none of the voices from this set were used for any user-side inputs.

We note that CosyVoice sometimes failed to generate any audio for a sentence or a few words. These occurrences were quite rare (below 1% of dialogues), and we left them as they were generated. The rest of the conversation stayed the same. This happened only with user voices, never with the assistant's.

**Annotations.** We follow Moshi's original annotation workflow. The agent channel is transcribed using Whisper [118] with word-level timestamp resolution. The resulting annotation records two elements: the word-level alignments of the agent speech within the full stereo file, and the timestamp at which each agent turn begins (and thus at which its guide becomes active).

### Alternate Pipeline

Datasets that already consist of conversational audio recordings need to be handled differently. The early steps of the pipeline are omitted. Each sample is already an audio file in which the two participants occupy separate channels. We always designate one channel as the *assistant channel* and the other as the *user channel*. Where necessary (e.g. when one participant joined the call later), leading silence is trimmed before processing begins. The assistant channel is then transcribed with Whisper to produce word-level alignments, which serve the same

[1] The authors of Moshi did not release the original recordings of the voices they used for fine-tuning the base model, so we had to select a new one to give the agent a stable voice (Moshi outputs used for voice cloning sounded too robotic).

role as in the main pipeline. Voice conversion is applied to the assistant channel using CosyVoice 2: the original speaker's voice (timbre) is replaced by the chosen reference voice while the timing and linguistic content of the speech are preserved. The resulting stereo audio and text annotations match the same format as in the main pipeline, with the exception that the guide list is left empty, since no guide annotations are available for naturally occurring speech—and these are the situations in which the agent should communicate on its own.

### 3.1.3 Chosen Datasets

We selected some of the previously mentioned datasets (Section 3.1.1) and processed them further. We focused on several aspects, each covered by its own dataset. First, the model needs to learn to use the guide. The best-suited data for this seem to be user questions followed by an agent answer that is guided by external text. Then there are task-oriented dialogues, which almost always require some kind of external system on the assistant's side; this information can be fed precisely through the guide text. Since these datasets are mostly in textual form and the audio had to be synthesised, we also looked for a way to help the model maintain its natural interaction capabilities. For this, we focused on in-the-wild data containing conversations between real people with no artificial turn boundaries. With this combination, we aim for the model both to learn to use the guide when it is present and to maintain a dialogue on its own.

This choice is also supported by recent evidence from a related, though not identical, area. Mizumoto et al. [102] study speech language model training for ASR and AST (automatic speech translation) and report that synthetic data alone can degrade performance, whereas combining synthetic and real data can improve it. A more directly comparable result comes from Maimon et al. [97], who train a generative SLM and find that a model trained exclusively on synthetic (TTS-generated) data scores substantially lower on speech modeling benchmarks and exhibits much higher validation loss on real data compared to a model trained on a mixture of real and synthetic speech. Although their setup targets generative SLM pre-training rather than full-duplex spoken dialogue generation, the finding still motivates our decision to combine synthetic guided datasets with natural conversational audio.

For simplicity, we use abbreviations to distinguish between the different dataset groups.

- Category ***QA*** consists of Q&A datasets. The answers were newly generated so that they would resemble spoken dialogue, and a guide was added to each answer. The expectation is that the model needs external guidance, which it learns to use in these scenarios.

- Category ***Nat*** includes exclusively natural spoken conversations, where one side is meant to resemble the assistant and the other a user. No guide was added here. The goal is for the model to maintain its natural capabilities (quick turn-taking, correct backchannel handling, sounding human-like).

- Category ***TOD*** consists of task-oriented dialogues, where the guide was added to those agent turns that were deemed to require it.

- Category ***Ext*** comprises extensions of the previous categories—mostly of category *QA*—where the guide is placed differently.

A detailed description of the specific datasets can be found in Table 3.2. *Size* refers to the number of individual conversations in the dataset. For categories *QA* and *TOD*, each conversation usually lasts about 2 to 4 minutes. During training, their ends are trimmed to use a defined length (2 minutes). The original conversations in *Nat* tend to be longer. Instead of keeping only their beginnings, we chunk each conversation into multiple separate inputs. This increases the number of actual training samples obtained from the dataset. For example, a 10-minute recording becomes 5 training elements. Their size also doubles relative to the originals, because either of the two sides can be assigned the assistant role.

| **Dataset** | **Size** | **Description** |
|---|---|---|
| *QA-UC* | 20k | Randomly sampled from UltraChat (3/4 weight) and TriviaQA (1/4 weight). Only the first question is used (in the case of UltraChat; TriviaQA is not multi-turn), and the answer and guide are then generated. Multiple question-answer pairs are concatenated one after another in each sample, with the same voice used to generate the user side. |
| *QA-IS2S* | 200k | Encapsulates InstructS2S-200k. This dataset was eventually not used, because it internally relies on data from UltraChat, which would overlap with *QA-UC*. |
| *Nat-Can* | 3k | CANDOR, where one side is always voiced by the agent. The ~30-minute conversations are chunked into smaller segments. |
| *Nat-Fis* | 11.5k | Fisher (part 2), where one side is always voiced by the agent. The 10-minute conversations are chunked into smaller segments. |
| *TOD-FC* | 8k | FusedChat original conversations, with a guide additionally generated. Most conversations are between 1.5 and 3 minutes long. |
| *Ext-UC-Delay* | 10k | Similar to *QA-UC*, but includes only UltraChat sources. The guide is intentionally delayed by a few seconds rather than fed to the model immediately after the user finishes their utterance. Typically, the first sentence of the answer does not yet include the actual information from the guide; it appears only in the following sentence, once we can assume the guide has been loaded. |
| *Ext-UC-Miss* | 10k | Similar to *QA-UC*, but includes only UltraChat sources. The guide is removed. The responses begin similarly to those in *Ext-UC-Delay* but deliberately provide no factual answer (so that the model does not hallucinate), because the guide never arrives. The model talks around the topic but admits that it cannot provide any further information. |

**Table 3.2** Chosen dataset groups used in our experiments.

### 3.1.4 Possible Improvements

Possible improvements include adding interaction elements to the synthetic datasets, such as overlapping speech, gaps or pauses, filler words, and backchanneling. Omniflatten [178] implements some of these aspects using half-duplex dialogues, where a user question is followed by an immediate answer, a user interruption causes the assistant to stop speaking abruptly (which is not necessarily realistic), and once the assistant finishes, it remains silent while waiting for the user. This may not be the most desirable behavior, since it removes any opportunity for initiative from the model, but it is a step in the natural-speaking direction.

One way to make the system more robust is to include various sources of noise in the user audio (for example, as a background to existing speech). MUSAN [136] is one data source that could be used. In our final selection, we mostly employ clean audio generated by TTS. The only way noise could have entered the training data is through the natural-conversation datasets, if it was present in the original recordings.

Since Moshi was not trained on data that would make it proficient at recognizing emotions in the user's speech, we elected not to explore this area either, focusing instead on the textual and semantic aspects of guide usage and on accurate turn-taking.

Moshi has been trained to disclose that it is an audio AI model. Our datasets do not include this tuning component; instead, we rely on the provided text guide to supply the identity. There is no mechanism, such as a system prompt, to adjust this. This area is explored in much greater depth by Roy et al. [124].

## 3.2 Training Setup

In this section, we outline the training setup and LoRA configuration; the evaluation datasets and scenarios are defined in Section 3.4 and the metrics in Section 3.5. The main experimental results that build on this setup are presented in Section 4.1.

We followed Moshi in using the same first codebook loss weight multiplier. This upweights the semantic part of audio (the first codebook) relative to the acoustic parts (other codebooks). We also keep the same text padding weight. This downweights loss contributions from padded text positions, keeping actual text more impactful. The other parameters used can be found in Table 3.3.

All training runs were performed on two NVIDIA H100 NVL GPUs. The batch size reported in Table 3.3 (4) is the per-GPU batch size; with two GPUs, the effective total batch size is 8.

We used LoRA [60] for efficient fine-tuning. Instead of updating all parameters of the pretrained model, the base weights are kept frozen and we train a small number of additional low-rank matrices (adapters) injected into selected layers. This dramatically reduces the number of trainable parameters and GPU memory/optimizer state, while still allowing the model to adapt to the target task. It also means that the base model weights are kept unchanged.

Unless stated otherwise, we ran our experiments for 20k optimization steps. We also chose to feed the model 120-second-long samples at a time instead of the

| Hyperparameter | Value |
|---|---|
| Batch size (per GPU) | 4 |
| Batch size (total; 2 GPUs) | 8 |
| First codebook loss weight multiplier | 100.0 |
| Text padding weight | 0.5 |
| LoRA rank | 512 |
| LoRA scaling ($\alpha$) | 4.0 |
| Fine-tune embeddings with LoRA | false |
| Learning rate | $8.0 \times 10^{-6}$ |
| Warmup fraction (`pct_start`) | 0.05 |
| Weight decay | 0.1 |

**Table 3.3** Training hyperparameters used in our experiments.

full five-minute ones used by Moshi. This decision was motivated by the scarcity of dialogue data and the fact that two minutes should be enough to maintain multi-turn capabilities; and most of our data does not include context longer than this.

During training, in each step, we randomly sample the batch from the available datasets. The sampling probabilities are the same for all sets unless specified otherwise.

We periodically calculated loss on evaluation datasets (described below in Section 3.4). We used the same loss function as Moshi, where the text tokens and audio tokens have equal weights. The audio token weights are further split into semantic and acoustic, where the first codebook (semantic) has an increased weight according to the parameter mentioned above. Concretely, following Défossez et al. [41], the training loss at each sequence position $s$ is

$$L(V, l) = \frac{1}{S} \sum_{s=1}^{S} \left( \mathrm{CE}(l_{s,1},\, V_{s,1}) \;+\; \frac{1}{\sum_{k=2}^{K} \alpha_k} \sum_{k=2}^{K} \alpha_k \, \mathrm{CE}(l_{s,k},\, V_{s,k}) \right), \tag{3.1}$$

where $V_{s,1}$ is the text token at step $s$, $V_{s,k}$ for $k > 1$ are the audio codebook tokens, and $\alpha_k$ is a per-codebook weight: $\alpha_2 = 100$ for the semantic token and $\alpha_k = 1$ for all acoustic codebooks $k > 2$. The formulation gives equal importance to the text stream and the combined audio stream while emphasising the semantic codebook within the audio.

In practice, the evaluation-loss curves followed expected generalization behavior. When the evaluation set matched data types present in training, loss typically decreased during fine-tuning. For evaluation sets representing data types not covered by the training mixture, loss tended to increase instead. In low-data settings (few training datasets or limited diversity), we often observed an initial decrease followed by a later increase, which is consistent with overfitting in later training stages.

## 3.3 Inference Setup

In this section, we describe some of the choices made for the inference mode of Warmblood building on what we described in Section 2.2.

For ASR, we use faster-whisper [142], a reimplementation of OpenAI's Whisper model [118]. For the VAD model used before ASR is run, we chose Silero [144].

### 3.3.1 Guide Prompt

We prepared several prompt variants for different interaction styles. A general-purpose variant provides balanced factual and conversational guidance—this is the one we used primarily. Other specialized variants focus on casual chitchat, exploratory Socratic-style dialogue, question answering with factual emphasis, and a task-oriented restaurant style. The exact runtime prompt template used in the final system is provided in Appendix A.3.4.

### 3.3.2 Guide LLM Choice

For the guide generating LLM, many options appear. A model could be hosted on the same servers as the dialogue system. While this would bring lower latencies, it also means fewer options to choose from (open-source only) and additional management and computational overhead. Therefore, we chose to use a cloud-based LLM, where many alternatives are available and switching is possible within moments. With a strong internet connection, the latency difference from a local model may be about 200 to 300 ms.

We settled on Llama-3.1-70b-instruct [100] because of its high response quality and low latency. Other comparable alternatives were GPT-4o [111], GPT-4.1 [109] and Gemini-2.0-flash-lite [53]. While all of these models are slightly older than the current state of the art, their advantage is response speed. Newer models often produce reasoning tokens, and their responses are not as quick.

### 3.3.3 Latency Control

The **total latency** from the point when the user finishes their utterance to when the guide is available is composed of the following: First, there is the VAD silence threshold, i.e., how long the system waits before starting ASR, which we set to 300 ms. Then, the ASR itself usually takes about 100 ms. The rest is the LLM round-trip, which tends to be around 1 s. That makes the total around 1.5 s. The waiting could be slightly reduced if we incorporated streaming of the LLM outputs. However, this could theoretically split the guide and would require other technical complications while bringing presumably marginal gains. Other options include continuous ASR instead of the burst approach used now, which could also bring a potential speedup (and a possible quality reduction).

## 3.4 Evaluation Datasets

For internal evaluations we prepared several datasets based on the main categories (Section 3.1.3). Unless stated otherwise, we prepared 250 instances that were not present in the training set.

- `qa_uc_eval` includes the standard multi-turn conversations of questions and answers. Just like *QA-UC*, they were sampled from UltraChat (unused items of the main set) and TriviaQA (dev set).

- **`nat_fis_eval`** was made from a separate part of the Fisher data. The 250 elements comprise the same two sets of 125 samples, with the side voiced by the model changed. We ended up using `nat_fis_eval` only for loss calculations. Orchestrating evaluation on natural data without annotations would be difficult at this stage.

- **`tod_fc_eval`** includes conversations from FusedChat's dev set. The base 250 elements were made up of both samples where open-domain talk is prepended and appended to the base conversation. This made some of the metrics less precise or reliable—in the open-domain part, it does not make sense to run correctness or guide-related metrics (the metrics are explained further in Section 3.5). Therefore, we later extended it to `tod_fc_eval_prepended` and `tod_fc_eval_appended` (still drawing from the dev set), each comprising 250 elements of the respective version.

Guides are included in all datasets just like in their training counterparts (except for `nat_fis_eval` where no guide exists for either). Reference audio responses and matching text transcriptions are attached as well.

For evaluation, we used only the user voices that had not been used in training. Initially, we planned to create two versions of each evaluation set—*seen* and *unseen*—both containing new data, but with training voices reused for *seen*. Early tests showed correlated performance between the two conditions, with slightly lower scores on unseen voices. To avoid an excessive number of evaluation subsets, we proceeded with the unseen version only. Since real user voices will also differ from those the model has seen, this choice is more representative of real situations.

### 3.4.1 Scenarios

For testing different moments of the dialogue, we devised multiple *scenarios*. They are all turn-related, since the datasets used were synthesized with saved timestamps marking where the turn changes. This, of course, depends on the reference assistant audio length. An actual model output would almost certainly end its turn at a different point, which would be difficult to predict. Therefore, we always test only one turn [169], because feeding further ones would require more complicated synchronization. Since the model produces no token signifying an end of turn, we stop it once the length of the reference audio has elapsed, with a small margin (5 seconds) added on top. The agent is allowed to generate output already in the preceding user turn, so overlap is possible.

These are the scenarios we used:

- **`first_turn`** tests the agent's response after the very first user turn. The guide is inserted right after the user finishes speaking (same as in the base training sets).

- **`first_turn_delayed_guide`** is the same as `first_turn`, except the guide is always delayed by 2 seconds.

- **`first_turn_no_guide`** tests the first turn where no guide is inserted and the model has to answer without it.

- `last_turn` evaluates the response to the user's last request. The previous conversation (model text and audio, user audio) is force-fed up to the last turn, where the model is again allowed to generate its own output at any time from the moment the (last) user's speech starts.

The base evaluation datasets stay the same for these scenarios (`qa_uc_eval`, `tod_fc_eval_appended` and `tod_fc_eval_prepended`); for example, the same first turn *QA-UC* questions. Only the selected turn or guide placement changes.

Unlike in training, where all conversations were capped at 2 minutes, we did not want to impose an artificial, unnatural barrier during evaluation, so we set the maximum length to 4 minutes, which is merely a safety net that should be enough to contain all the generated conversations. This mostly mattered for the last turns, which could occur after the two-minute mark.

## 3.5 Metrics

Most of the following metrics are computed on the textual outputs of the model (the "inner monologue" mentioned in Section 1.8.2)—using text references for comparison. Textual metrics are more common and easier to debug and understand. This means that, at this stage, we mostly tested the linguistic and semantic quality of the responses.

We evaluated text quality against reference transcripts using two standard MT-style metrics. The first is **BLEU** [113]: an n-gram precision metric with a brevity penalty. It rewards surface-level overlap between the model output and the reference; higher BLEU indicates closer lexical match. The second metric is **BLEURT** [130]: a learned, reference-based metric built on a pretrained language model and fine-tuned on human ratings. It scores semantic adequacy and fluency beyond exact n-gram overlap; higher BLEURT indicates better perceived quality. We report both metrics because BLEU is sensitive to wording, while BLEURT better captures meaning preservation and paraphrasing.

Next, we incorporated an LLM (GPT-4.1-mini [109]) to score the answers based on various criteria in an LLM-as-a-judge style. Across all these metrics, per-sample scores are produced on a 1–5 ordinal scale and then averaged to obtain the aggregate score for a dataset. This methodology is consistent with the GPT-score evaluation used in TurnGuide [39], which validates the approach against human judgements and reports 83% agreement. The corresponding judge prompt templates are provided in Appendix A.3.3. The LLM metrics are:

- **Correctness**: this metric evaluates whether the model's response preserves the core factual content of the expected answer, given the preceding user turn—one turn should be enough for context, the main datapoint for comparison is the reference answer. A judge model receives the user turn, the reference answer, and the generated response, then assigns a score from 1 (completely incorrect) to 5 (fully correct), together with a short rationale. The metric is designed to tolerate natural conversational variation (for example, longer paraphrases) as long as key facts are preserved and not contradicted.

- **Guide Following**: this metric measures how well the response uses the provided guide. The judge model sees the previous user turn, the guide, and the generated response, and rates guide usage from 1 (ignored) to 5 (fully incorporated). The emphasis is on whether the guide's intent is reflected naturally in the response, rather than on exact wording or factual verification. That is, the goal is not to be objectively correct, as with the previous metric, but to use the guide precisely. Therefore, the reference answer is not even provided.

- **Naturalness**: this metric captures conversational quality: relevance to the user turn, coherence, and grammatical naturalness. The goal is to measure how well the model can maintain conversational flow. The judge model sees only the preceding user turn and the generated response, and scores from 1 (incoherent or unrelated) to 5 (natural and relevant). It explicitly does not evaluate factual correctness, and it does not penalize brief responses if they are plausible in dialogue or they could have a natural continuation. For example, the model may have begun a new sentence but was cut off due to the time limit.

We tuned the evaluation prompts iteratively to ensure sufficient judge performance: after each revision, we manually inspected judge outputs on a small set of samples to verify that the scores matched our expectations. In particular, adding an explicit reasoning field for the judge was highly beneficial, as it forces the model to justify its rating and made the resulting scores more stable across runs. We intentionally avoid including scored examples in the prompt: the goal is to leverage the model's reasoning capabilities, rather than to encourage it to mimic a fixed pattern.

While training, we noticed that a common cause of low scores was a completely empty response—the model simply did not react to the user at all. In such cases, the LLM-based metrics would naturally assign the lowest score (1 out of 5). Therefore, we decided to measure this aspect specifically. We added an **Emptiness** metric, which calculates the ratio of responses that are completely empty (no text). The lower the Emptiness, the better.

**Latency** is the only metric that takes the output audio into account. We estimate speech onset with a simple energy-based detector. The model-output waveform is split into short time windows (about 80 ms), and for each window we compute its root-mean-square (RMS) amplitude as a measure of signal energy. The onset is defined as the first window whose energy exceeds a fixed threshold, which suppresses low-level background noise and captures the start of clearly audible speech. The search starts from the point where the agent is allowed to respond, and latency is then computed as the time difference between this detected speech onset and the end of the preceding user turn. This yields an estimate of how quickly the system begins speaking, while still allowing overlap (negative numbers) when the model starts before the user has fully finished. If no speech is detected, latency is not computed. The latencies (from non-empty responses) are averaged to produce the final score.

We will generally focus on the LLM metrics, Emptiness, and Latency, because their outputs (including reasoning) are more easily interpretable by a human than the BLEU/BLEURT scores.

# 4 Evaluation

This chapter presents the main results and discusses their implications. The training configuration is described in Section 3.2, and the evaluation datasets, scenarios, and metrics are covered in Chapter 3. Building on that setup, Section 4.1 presents the internal ablation experiments that guided our dataset selection, including manual observations from live deployment. Finally, Section 4.2 reports performance on established benchmarks and Section 4.3 shows the findings of a 100-participant user study.

## 4.1 Internal Experiments

The following experiments guided our selection of the model used for the final benchmarks in Section 4.2.

Table 4.1 summarizes the experiment scenario codes used throughout this section.

| Code | Description |
|---|---|
| Q-FT | `first_turn` scenario for the `qa_uc_eval` dataset |
| Q-LT | `last_turn` scenario for the `qa_uc_eval` dataset |
| Q-FT-NG | `first_turn_no_guide` scenario for the `qa_uc_eval` dataset |
| Q-FT-DG | `first_turn_delayed_guide` scenario for the `qa_uc_eval` dataset |
| TA-FT | `first_turn` scenario for the `tod_fc_eval_appended` dataset |
| TA-LT | `last_turn` scenario for the `tod_fc_eval_appended` dataset |
| TA-FT-NG | `first_turn_no_guide` scenario for the `tod_fc_eval_appended` dataset |
| TA-FT-DG | `first_turn_delayed_guide` scenario for the `tod_fc_eval_appended` dataset |
| TP-FT | `first_turn` scenario for the `tod_fc_eval_prepended` dataset |
| TP-LT | `last_turn` scenario for the `tod_fc_eval_prepended` dataset |

**Table 4.1** Experiment scenario codes and their meaning.

### 4.1.1 Main Results

In Table 4.2, we present all measured results of the model trained on the "main dataset package". This package includes *QA-UC*, *Nat-Fis* and *TOD-FC* (see Table 3.2) and serves as the primary reference configuration for the internal experiments. While this section reports its performance directly, the following sections compare it against alternative training data selections to determine which parts of the dataset composition matter most. Specifically, Section 4.1.2 (*Subset Selection*) analyzes the effect of removing individual components, Section 4.1.3 (*Delay*) evaluates guide-delay variants, and Section 4.1.4 (*More Datasets*) examines extensions with additional data sources.

For completeness, one could ask whether the base Moshi model should be included as a direct baseline on these internal evaluation sets. In practice, this is not straightforward: such a comparison only meaningfully applies to no-guide conditions, while most scenarios here are explicitly guide-driven. In addition, base Moshi often starts speaking immediately, whereas our evaluation setup inserts the user request at the beginning of the sample. We intentionally avoided "cheating" by artificially keeping Moshi silent and then forcing generation via EPAD, because that would no longer reflect its natural runtime behavior.

| | BLEU ↑ | BLEURT ↑ | Corr. ↑ | Guide Following ↑ | Natur. ↑ | Empt. ↓ | Latency ↓ |
|---|---|---|---|---|---|---|---|
| Q-FT | 50.95 | 0.63 | 4.25 | 4.40 | 4.14 | 0.04 | 0.95 |
| Q-FT-DG | 44.10 | 0.62 | 3.86 | 3.96 | 4.09 | 0.05 | 1.90 |
| Q-FT-NG | 22.66 | 0.50 | 2.54 | — | 3.95 | 0.10 | 2.70 |
| Q-LT | 54.11 | 0.72 | 4.44 | 4.49 | 4.45 | 0.05 | 0.82 |
| TA-FT | 14.08 | 0.53 | 3.88 | 3.96 | 4.16 | 0.04 | 1.08 |
| TA-FT-DG | 10.84 | 0.49 | 3.36 | 3.40 | 4.12 | 0.05 | 1.76 |
| TA-FT-NG | 2.04 | 0.36 | 2.20 | — | 3.90 | 0.14 | 2.31 |
| TA-LT | 10.60 | 0.42 | — | — | 3.62 | 0.18 | 1.07 |
| TP-FT | 1.24 | 0.29 | — | — | 3.47 | 0.25 | 1.61 |
| TP-LT | 5.54 | 0.38 | 2.72 | 4.26 | 3.37 | 0.35 | 0.58 |

**Table 4.2** Results for the main dataset training package (*QA-UC*, *Nat-Fis*, *TOD-FC*). Metric definitions are given in Section 3.5; dataset/scenario construction is described in Section 3.4.

The datasets that are evaluated (`qa_uc_eval`, `tod_fc_eval_appended`, and `tod_fc_eval_prepended`) remain the same across the grouped rows of Table 4.2 (the same 250 elements); only the selected turn and guide placement change by scenario.

The first important observation is that on both evaluation datasets, at the first turn, the model always performs best with the immediate guide, while the delayed guide ranks second and no guide comes last, across all metrics. This may seem expected, but it signifies that the model actively tries to use the guide and performs best with it.

BLEU and BLEURT are somewhat correlated, but they diverge significantly depending on the evaluation dataset. This is likely because in the *QA* set, the guided answer is mostly predetermined, and even without a guide, the agent can deduce it from its own knowledge. In *TOD*, the reference answers can have very different wording because the important information might only be something like a phone number, the number of hotels, or the name of a restaurant. For the other metrics, the scores are more similar.

Overall, the model performs better on the *QA* evaluation than on *TOD*. This is probably because the *QA* dataset was bigger and closer to data the model had already been trained on. In Q&A, the assistant can try to answer the questions itself, and the guide only helps it achieve better correctness. With task-oriented dialogue, there is no way to guess the answer without external information. Thus,

when the agent makes an incorrect prediction or misunderstands the guide, it is more severely penalized.

Comparing the performance on the first turn against the last turn in *QA*, the last-turn score is slightly better on all metrics except Emptiness (worse by 0.01). This leads us to believe that the model takes some time to "warm up"—at the beginning, it is not sure whether it is its turn to speak yet and generally what is about to happen, and it may respond differently than intended. In the last turn, on the other hand, the whole conversation has been a stable stream of questions and answers, so the agent knows exactly what to do and reacts accordingly.

Comparing the first turn and the last turn in *TOD* is somewhat more complicated because of the two different versions (prepended and appended chitchat). The appended version (first turn) has the highest Correctness scores (if it has the guide) because the user request is usually direct at the start of that conversation—task-oriented dialogue, a concrete task to perform, a specific guide. The last turn, on the other hand, even when it is not open-domain dialogue, is often a point where the task-oriented conversation is finished and the user only says "thank you" or similar. At these points, the Guide Following metric can often give the highest score if the guide is actually empty (nothing to follow—no punishment).

When we look at Naturalness, it stays roughly around 4.1 for most scenarios. When it drops below 4, it is usually caused by the number of empty responses. We can see that the lowest Naturalness scores appear with the highest Emptiness.

The highest Emptiness scores can be seen with *TA-LT*, *TP-FT* and *TP-LT*. The first two of these involve open-domain dialogue. This often means the user shares some part of their life or remarks on something without actually providing a request or asking a question. That does not mean the assistant should not react (it does in the reference data), but it explains why it might hesitate to do so. Looking at it from the other perspective, failing to answer these utterances in only 1 out of 4 to 5 cases is a decent result. The other type of scenario with higher Emptiness is the last turn scenario. We remarked on this before: The conversation often ends with something like "goodbye", which does not necessitate an answer. Apart from this, an Emptiness of 0.1 or more appears only with the no-guide cases, where the agent tends to struggle more, as we explained above.

Lastly, looking at latencies, the quicker the guide appears, the faster the model answers. It is interesting that even though the delayed guide arrives 2 seconds later, the Latency is on average only about 1 second larger than the immediate one. Figures 4.1–4.3 present response latency histograms for the three main *QA* scenarios (bins of 0.1 s, middle 90% of observations shown). In the standard guided scenario (Q-FT, Figure 4.1), the distribution follows an approximately Poisson-like shape peaking around 0.7 seconds, confirming that the model responds quickly and consistently when a guide is immediately available.

After closely inspecting the data for the delayed-guide scenario (Q-FT-DG, Figure 4.2), we see that in many examples the model starts answering very quickly (within about 0.7 seconds), while at other times it takes about 2.5 seconds—producing a clear bimodal structure with one cluster under 1 second and a second just under 3 seconds, where the model waits for the guide. This hints at the possibility that the model makes an inner "decision" on whether to reply right away (it can still use the guide once it appears during its speech) or wait until the guide is present and only then begin the response. The fact that the average

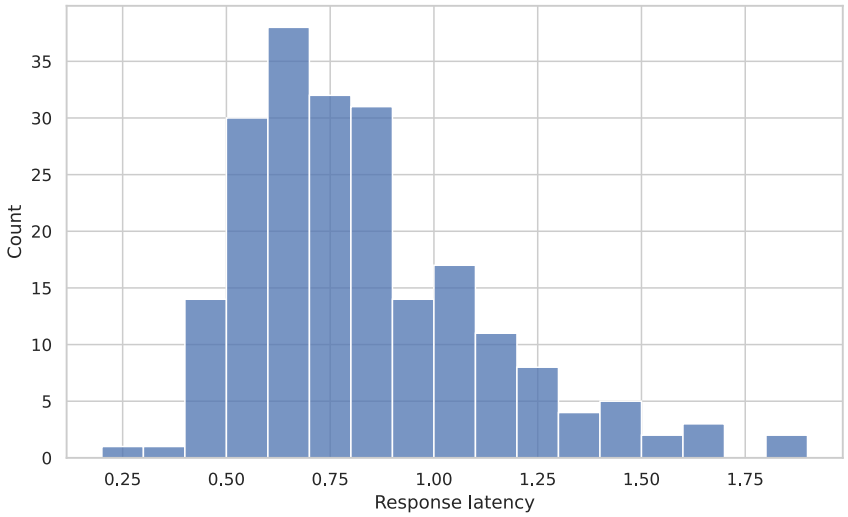


**Figure 4.1** Response latency histogram for Q-FT (immediate guide). Bins are 0.1 s wide; middle 90% shown. The distribution peaks around 0.7 s.

difference between the delayed and normal scenarios is only about 1 second can be explained by this mixture together with a general overhead that is always present.

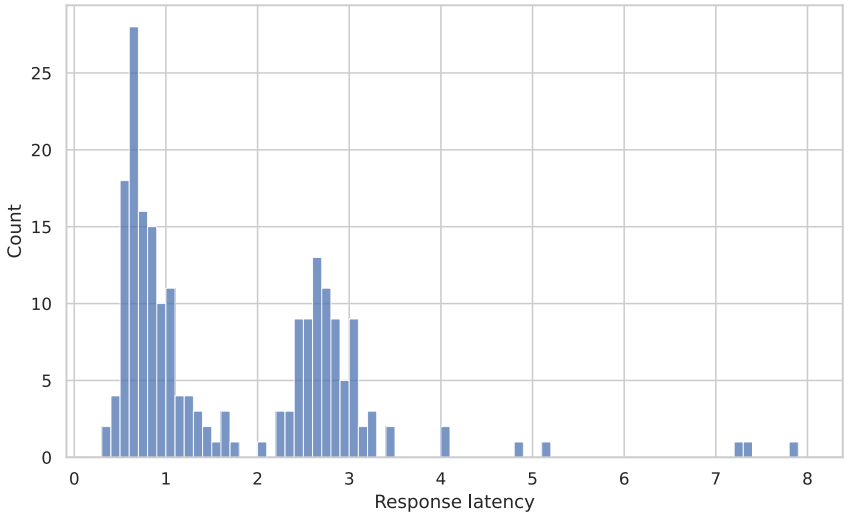


**Figure 4.2** Response latency histogram for Q-FT-DG (delayed guide, arriving at ∼2 s). Bins are 0.1 s wide; middle 90% shown. A bimodal structure is visible: early responses within ∼1 s and a second cluster just under 3 s.

In the no-guide scenario (Q-FT-NG, Figure 4.3), the early-response cluster within ∼1 second is roughly as large as in the delayed-guide case—these are instances where the model answers from its own knowledge. However, since no guide ever arrives to trigger a second cluster, the remaining responses are spread quite evenly across the following ∼10 seconds, which drives the noticeably higher average latency.

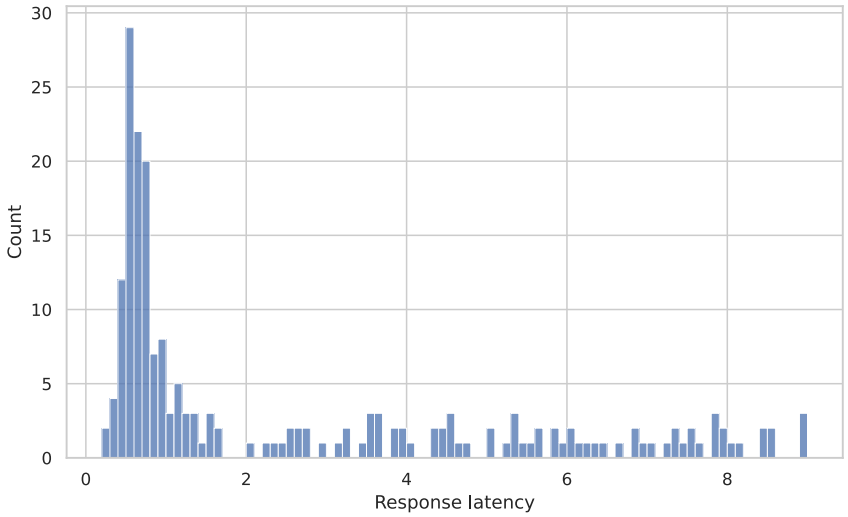


**Figure 4.3** Response latency histogram for Q-FT-NG (no guide). Bins are 0.1 s wide; middle 90% shown. An early-response cluster similar to Q-FT-DG is present, but without a delayed guide to trigger a second peak, the remaining responses spread across ∼10 s.

### 4.1.2 Subset Selection

In this section, we will look at the results on runs where some of the datasets were excluded from training. Since both evaluation sets work with (in principle) guided datasets, and the goal of this project is to test the guide, we always included either *QA-UC* or *TOD-FC*. We do not show all scenarios, only the most relevant ones for each metric.

Table 4.3 shows the Correctness results. We check the performance in the main guided situations and their unguided counterparts (if present). It is interesting to see the effect of including *Nat-Fis*. For *QA-UC* alone, it slightly degrades performance. For *TOD-FC* alone, the inclusion does not help on the guided case of the *QA* evaluation, but it does help achieve higher scores in the other cases. This could be because the *TOD-FC* dataset is the smallest, and more diversity is helpful to prevent overfitting. However, for the combination of *QA-UC* and *TOD-FC*, the additional inclusion of *Nat-Fis* helps substantially across all situations. When a given dataset is not present in training, the results on its relevant evaluation are degraded (both guided and unguided). The highest score in most cases is held by the full combination of datasets.

| *QA-UC* | *Nat-Fis* | *TOD-FC* | Q-FT | Q-FT-NG | TA-FT | TA-FT-NG |
|---|---|---|---|---|---|---|
| ✓ | | | 4.14 | 2.05 | 2.60 | 1.46 |
| ✓ | ✓ | | 4.02 | 2.01 | 2.51 | 1.38 |
| | | ✓ | 3.28 | 1.63 | 3.42 | 1.93 |
| | ✓ | ✓ | 2.98 | 1.67 | 3.70 | 2.39 |
| ✓ | | ✓ | 4.02 | 2.42 | 3.32 | 1.90 |
| ✓ | ✓ | ✓ | 4.25 | 2.54 | 3.88 | 2.20 |

**Table 4.3** Correctness (↑) for different training data subsets on selected scenarios.

Guide Following (results in Table 4.4) scores similarly to Correctness. Here we compare the first-turn-with-guide and delayed-guide situations. Inclusion of *Nat-Fis* surprisingly helps *QA-UC* on the delayed-guide cases, but for *TOD-FC* it varies. The full package still performs best or on par with the other variants.

| *QA-UC* | *Nat-Fis* | *TOD-FC* | Q-FT | Q-FT-DG | TA-FT | TA-FT-DG |
|---|---|---|---|---|---|---|
| ✓ | | | 4.30 | 3.65 | 3.01 | 2.53 |
| ✓ | ✓ | | 4.18 | 3.96 | 2.98 | 2.70 |
| | | ✓ | 3.50 | 2.87 | 3.52 | 3.22 |
| | ✓ | ✓ | 3.17 | 2.11 | 3.79 | 3.23 |
| ✓ | | ✓ | 4.15 | 3.97 | 3.42 | 3.02 |
| ✓ | ✓ | ✓ | 4.40 | 3.96 | 3.96 | 3.40 |

**Table 4.4** Guide Following (↑) for different training data subsets on selected scenarios

Table 4.5 shows Naturalness in scenarios without a guide.[1] Here, we would expect *Nat-Fis* to make the greatest difference. Its inclusion helps in almost all cases except for the *QA* evaluation of *QA-UC*, where training on the bare dataset seems to fare better. For the *TOD* evaluations, it depends again: for the *QA-UC* base set, the inclusion of *Nat-Fis* makes almost no difference. It does help *TOD-FC* in two of the scenarios, and the full package benefits from it every time. The highest score in this category, though, is not always held by the complete package. It is sometimes slightly beaten by the combination of *Nat-Fis* and *TOD-FC*. However, as we mentioned before, these *TOD* evaluations of Naturalness are tricky because the user utterances often do not include a direct request, and the agent gets punished if it stays silent.

| *QA-UC* | *Nat-Fis* | *TOD-FC* | Q-FT-NG | TA-FT-NG | TA-LT | TP-FT |
|---|---|---|---|---|---|---|
| ✓ | | | 3.12 | 2.40 | 2.06 | 1.65 |
| ✓ | ✓ | | 2.90 | 2.32 | 1.97 | 1.60 |
| | | ✓ | 3.16 | 3.36 | 3.70 | 3.17 |
| | ✓ | ✓ | 3.49 | 4.00 | 3.57 | 3.90 |
| ✓ | | ✓ | 3.59 | 3.59 | 3.47 | 2.98 |
| ✓ | ✓ | ✓ | 3.95 | 3.90 | 3.62 | 3.47 |

**Table 4.5** Naturalness (↑) for different training data subsets on selected scenarios

### 4.1.3 Delay

This set of experiments tests whether adding guide delay to the existing fine-tuning data improves performance. The base training dataset group (`main`) stays the same (*QA-UC*, *Nat-Fis*, *TOD-FC*). We tested two new combinations. In `main_delay`, we add a (uniformly) random delay of 0 to 2 seconds to the guide in the *QA-UC* and *TOD-FC* datasets. In `main_ext_delay`, the same is applied

[1] Out of the measured scenarios, we selected these because without a guide, the Naturalness difference is the most noticeable.

with the maximum delay reduced to only 1 second. However, a new dataset is added here—*Ext-UC-Delay* (see Table 3.2)—which includes Q&A data where the responses start with a delaying sentence and only afterward is the actual answer provided. The guide in these samples was delayed by 1 to 3 seconds. The idea is that, in reality, there will always be some delay; therefore, training the model with the delay present in the data could improve its performance or reduce the speech response latency. In this section, we primarily test the general effect of introducing guide delay (via artificial offsets) and also observe what changes when we additionally introduce a small amount of explicitly delay-shaped data; the next section then isolates the effect of adding extra datasets alone, without modifying the originals.

The outcomes presented in Table 4.6 are clearly disappointing. We chose only the *QA* category scenarios for evaluation because they are the most relevant. The strongest performer is still the base set without any artificial delays. This is interesting especially for the delayed-guide cases, where we would expect the extensions to work better. Some of the lower metric scores are caused by high Emptiness among the extensions. The main improvement can be seen on latency reduction on the delayed or no-guide cases. However, it also brings an increase in latency for the first and last turn scenarios where the guide is present right away. Overall, the high increase of Emptiness with these extensions is the biggest downside.

| **Correctness ↑** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT |
|---|---|---|---|---|
| `main` | 4.25 | 3.86 | 2.54 | 4.44 |
| `main_delay` | 3.84 | 3.69 | 2.57 | 3.60 |
| `main_ext_delay` | 3.78 | 3.62 | 2.33 | 3.68 |
| **Guide Following ↑** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT |
| `main` | 4.40 | 3.96 | — | 4.49 |
| `main_delay` | 3.91 | 3.77 | — | 3.71 |
| `main_ext_delay` | 3.90 | 3.68 | — | 3.76 |
| **Naturalness ↑** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT |
| `main` | 4.14 | 4.09 | 3.95 | 4.45 |
| `main_delay` | 3.94 | 3.99 | 3.70 | 3.72 |
| `main_ext_delay` | 3.81 | 3.88 | 3.47 | 3.71 |
| **Emptiness ↓** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT |
| `main` | 0.04 | 0.05 | 0.10 | 0.05 |
| `main_delay` | 0.13 | 0.14 | 0.15 | 0.23 |
| `main_ext_delay` | 0.14 | 0.14 | 0.22 | 0.22 |
| **Latency ↓** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT |
| `main` | 0.95 | 1.90 | 2.70 | 0.82 |
| `main_delay` | 1.82 | 1.82 | 2.60 | 1.27 |
| `main_ext_delay` | 1.15 | 1.72 | 2.27 | 1.31 |

**Table 4.6** Metrics with delayed guides

We can conclude that the artificial delay mostly degrades performance when it is applied to the main datasets.

### 4.1.4 More Datasets

In the following tests, we left the base fine-tuning package (*QA-UC*, *Nat-Fis*, *TOD-FC*) as it was but tried adding multiple new training datasets. *Ext-UC-Delay* was kept (as well as its 1 to 3 second delay parameter), with *Ext-UC-Miss* further added—it comprises unguided Q&A instances from UltraChat where the model responds with sentences like "That is an interesting question." and only comments on the topic without trying to provide an actual answer. We also extended the natural data with *Nat-Can*—CANDOR recordings, chunked into 2-minute segments. In `main_ext_1`, we added both the extension sets and half of *Nat-Can*—where only one side was replaced by the model's voice for each conversation. In `main_ext_2`, the full *Nat-Can* is included, but the weights—the likelihood of picking elements from a given dataset—are reduced to 0.5 for *Ext-UC-Delay* and *Ext-UC-Miss*.

Results are presented in Table 4.7. Let us consider Emptiness first because it can covertly influence the other metrics. Here, it looks very similar for all versions. `main_ext_2` performs slightly worse than the others.

With the LLM metrics, it appears that the `main` version still performs best on the delayed-guide and no-guide cases. This is interesting because it includes no preparation for these cases whatsoever, while the two extensions do. On the other hand, on the first and last turns with an immediate guide, `main_ext_1` achieves the highest scores. When we compare the extensions with each other, `main_ext_1` beats `main_ext_2` in almost all metrics (or is essentially on par). This would suggest that the full addition of *Ext-UC-Delay* and *Ext-UC-Miss* plays the most significant role. In comparison, *Nat-Can* seems to contribute little: the Fisher part alone yields roughly sixty thousand training samples, which is probably more than the model observes during training. Adding another very similar dataset probably has only marginal effects. We cannot determine whether the increased probability of sampling natural source data (which doubles when *Nat-Can* is added) meaningfully affects the results.

The last interesting metric is Latency, for which `main_ext_2` sometimes performs better and sometimes worse than the base package, but `main_ext_1` definitely beats both in all cases. Even though its semantic results (as measured by the LLM metrics) come out worse, it does respond faster in the delayed/missing-guide situations. The most probable explanation is that the agent starts answering very soon with its learned response: "Thank you for asking that", but is then not as good at using the guide it receives or maintaining a natural flow.

### 4.1.5 Preliminary Testing

We conducted several internal manual tests in which we used the full system (including the inference loop), noted its behavior, and made small tweaks to the prompt. The following observations stem from subjective monitoring and are not backed by a rigorous dataset. We believe that some aspects of natural dialogue can only be captured by manual use.

| **Correctness ↑** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT | TA-FT |
|---|---|---|---|---|---|
| `main` | 4.25 | 3.86 | 2.54 | 4.44 | 3.88 |
| `main_ext_1` | 4.38 | 3.68 | 1.58 | 4.58 | 4.09 |
| `main_ext_2` | 4.19 | 3.78 | 1.56 | 4.54 | 3.92 |
| **Guide Following ↑** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT | TA-FT |
| `main` | 4.40 | 3.96 | — | 4.49 | 3.96 |
| `main_ext_1` | 4.48 | 3.83 | — | 4.61 | 4.13 |
| `main_ext_2` | 4.21 | 3.89 | — | 4.60 | 4.09 |
| **Naturalness ↑** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT | TA-FT |
| `main` | 4.14 | 4.09 | 3.95 | 4.45 | 4.16 |
| `main_ext_1` | 4.28 | 4.02 | 3.46 | 4.52 | 4.26 |
| `main_ext_2` | 4.13 | 4.02 | 3.24 | 4.60 | 4.14 |
| **Emptiness ↓** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT | TA-FT |
| `main` | 0.04 | 0.05 | 0.10 | 0.05 | 0.04 |
| `main_ext_1` | 0.05 | 0.05 | 0.09 | 0.04 | 0.02 |
| `main_ext_2` | 0.06 | 0.05 | 0.16 | 0.03 | 0.07 |
| **Latency ↓** | Q-FT | Q-FT-DG | Q-FT-NG | Q-LT | TA-FT |
| `main` | 0.95 | 1.90 | 2.70 | 0.82 | 1.08 |
| `main_ext_1` | 0.82 | 1.29 | 2.35 | 0.67 | 0.92 |
| `main_ext_2` | 0.91 | 1.43 | 2.80 | 0.70 | 0.82 |

**Table 4.7** Metrics on extended training data

For most tests, we used the base training data configuration (*QA-UC*, *Nat-Fis*, *TOD-FC*). Observations from the alternative configurations are noted explicitly below.

### Conversation Starts and Flow

The agent generally stays on topic and produces coherent responses, but the conversation sometimes feels somewhat passive. The assistant does not drive the conversation in most cases. In some turns, the behavior looked strongly guide-coupled, that is, the model started speaking mainly after the guide arrived, instead of maintaining flow autonomously.

The starts of conversations varied considerably. Sometimes the model would start speaking very early and naturally. At other times, we needed to prod it to get it to answer. It also seemed as if the system was switching its operation based on distinct modes corresponding to its individual training datasets. This makes sense because we never combined the datasets within a single sample. When we started the dialogue with a question, the assistant would wait for the guide and provide an answer (just like in Q&A cases). If we began with more natural, laid-back chitchat, it would act more naturally too, but sometimes it would simply ramble about connection issues or "the question", and so on—this is a clear artifact of the Fisher data (3.1.1), where participants often begin with these topics (similar

cases would likely be learned even from CANDOR data, since their setups were similar). This is an undesirable leakage.

One solution we came up with was to use a *starting guide*. This is a simple prepared text that gets inserted into the model input right at the beginning. Recall that we did not have such examples in the training data. The idea is to put the model into the mode of conversations where a guide is present. We settled on a simple "greet" for this guide, and the leakage issue has not resurfaced since. However, the starting guide did not help the agent that much—it does not naturally start with a greeting. This is not necessarily a problem, since an immediate greeting can lead to awkward overlap if the user is about to speak, which we observed with the original Moshi. A more severe issue is when the agent stays silent even after the first user request, which does happen occasionally.

### Turn-Taking and Interruptions

On dynamic interactions, such as spontaneous turn-taking, the model behaved fairly well. When the user interrupts the agent mid-speech, it does not abruptly cut off. It tends to finish a sentence or a thought. It is not an immediate reaction, but in real life, when people are interrupted, they usually like to finish at least part of what they are saying, and cutting off mid-thought is generally uncommon. We would consider it undesirable behavior if the model kept talking or started a new sentence, but this almost never happened.

### Guide LLM Prompt Variants

When we tried a chitchat prompt for the external LLM, the agent often dominated the interaction and spoke significantly more than the user—the guide was being sent almost all the time. Guides were generated frequently, but not every guide was visibly reflected in the next model response, suggesting partial guide under-utilization. A Q&A-oriented prompt variant improved focus on factual turns, but spontaneous behavior outside guided moments felt weak. Without timely guide support, the conversation could lose direction and appear less confident.

One of the key modifications we made across multiple prompts was to forbid (or minimize) the use of long item listings. The agent would often simply repeat the list without providing any extra detail, which felt very robotic and unnatural.

### Periodic Guide Triggering

When we tested periodic guide triggering as opposed to end-of-turn triggers (2.2), our assumptions about its stability were confirmed. We used a 2-second period—we did not want to allow a larger gap than that, because it could cause too long a wait. This setup occasionally interrupted user flow, especially when a new guide arrived while the user was still talking—the model would start talking over them. Compared to event-driven guidance, this mode was typically worse for casual chitchat but sometimes better for Q&A, where early guidance could help structure the answer. The main drawback was the excessive guidance volume, including unnecessary LLM calls in the background (with empty responses) wasting resources and cluttering dialogue history.

**Training Artifacts**

Some unwanted artifacts appeared that were not present before the fine-tuning.

**Mispronunciation.** We discovered that some less common words are difficult for the model to pronounce. An example of this is "Blockchain". The agent received it in the guide and did attempt to use it, but instead of actually uttering it, it produced a different word, such as "Blackhawk" or "Blank chain"—both in its audio and text outputs. We have not uncovered the source of this issue, since the base Moshi model can pronounce these words correctly. This degrades performance because sometimes the guide is correct and the model wants to use it but pronounces something different. This can lead to miscommunication with the user and frustration.

**Noise.** Sometimes, random background noise would appear in the model outputs after approximately two to three minutes of continuous interaction. It did not influence the conversation much and stayed at low volume levels, but it was noticeable.

**Low response relevance.** In the extension configurations with additional training data—`main_ext_1` and `main_ext_2`, the behavior was quite similar, but we observed the patterns from the additional training sets. The response often started by hedging with sentences like "That's an interesting question to consider." It can become tedious when this pattern appears after every question. What was worse were situations where the agent answered a question, there was no follow-up from the user, and it would start answering a different, non-existent question (beginning the sentence with the aforementioned sequence). Moreover, when this happened, the segment was unskippable. While reactions to interruptions are generally quick, as described before, in this case the model tended to finish the whole segment.

## 4.2 Final Evaluations

Based on the results from Section 4.1, we chose the model trained on the base package (*QA-UC*, *Nat-Fis*, *TOD-FC*) as the one to use in all the final evaluations. Among the base sets, it performed best, and it was more stable during the manual observations than the extensions. When available, we will provide benchmark comparisons with Moshi, similar models, and the best performers in each category.

Moshi's evaluation by Défossez et al. [41] covered several areas. For **audio language modeling**, they relied on *textless NLP* metrics (e.g. sWUGGY, sBLIMP, and Spoken StoryCloze variants) and also reported **MMLU** [57] as a text-only sanity check. For **spoken question answering**, they evaluated on Spoken Web Questions and LlaMA-Questions [103] as well as an audio version of TriviaQA [68] (exact-match accuracy). For **dialogue quality**, they generated dialogues via self-play and reported **perplexity** (PPL) plus turn-taking statistics (e.g. IPU duration, pauses, gaps, and overlap).

We wanted to evaluate our model on more thorough benchmarks that would cover a wider range of capabilities and be more representative of real-life interactions. We consider external metrics, which measure full outputs of the model, more reliable than those using internal model scores. We chose not to leverage Moshi's capability to immediately respond after inserting the EPAD token (Section 1.8.2)—we wanted to leave it in full operational mode on its own.

We chose two existing benchmarks. For **general linguistic capabilities**, we selected **VoiceAssistant-Eval** [151] because of its extensive task coverage and its inclusion of audio-based instructions alongside text, providing a more robust assessment. For **interaction capabilities** (turn-taking, overlap handling, etc.), we selected **Full-Duplex-Bench** [86] because it has been a staple used by many similar models in the field and offers multiple versions targeting progressively more complex behaviors. Both are freely available and well documented, and offer plenty of results on existing models.

**VoiceAssistant-Eval [151]** VoiceAssistant-Eval is a comprehensive benchmark for evaluating AI voice assistants across listening, speaking, and viewing tasks. It consists of 10,497 curated examples spanning 13 task categories, including natural sounds, music, spoken dialogue, multi-turn dialogue, role-play, and heterogeneous image input. Evaluations of 21 open-source models and GPT-4o-Audio reveal that proprietary models do not universally outperform open-source ones, and that most models excel at speaking tasks but lag in audio understanding.

**Full-Duplex-Bench [86]** Full-Duplex-Bench is a family of benchmarks for systematically evaluating spoken dialogue models (SDMs) with a focus on full-duplex interactive behaviors. Version 1 and Version 1.5 form the core of the series: v1 targets key behaviors such as pause handling, backchanneling, turn-taking, and interruption management using automatic metrics; v1.5 extends this to specifically probe how models behave during speech overlap, simulating four representative scenarios (user interruption, user backchannel, talking to others, background speech) and distinguishing two behavioral strategies—responsive (prioritizing quick reaction) versus floor-holding (preserving conversational flow). Version 2 exchanges static datasets for a running agent that dynamically tests the model in real-time. It further evaluates multi-turn instruction following and task performance under two pacing setups across daily, correction, entity tracking, and safety task families. Version 3 expands on this by introducing naturalistic human audio with disfluency annotations and multi-step tool-use scenarios.

We also considered the following alternatives but did not select them:

- **VoiceBench** [28]: A widely used platform for evaluating LLM-based voice assistants under diverse speaker and environmental conditions; not selected because it is older and less sophisticated, relying solely on text-based evaluation metrics.
- **ADU-Bench** [50]: Evaluates audio dialogue understanding across multiple languages and ambiguity types including paralinguistic cues such as intonation and homophones; these aspects go beyond the scope of our model's capabilities.

- **VocalBench** [89]: Targets vocal communication capabilities across semantic, acoustic, conversational, and robustness dimensions; the acoustic and vocal-specific dimensions are less relevant given our fine-tuning focus.
- **MTR-DuplexBench** [177]: Evaluates FDSLMs in multi-round settings by segmenting continuous dialogues into turns; less established and not as widely adopted as Full-Duplex-Bench at the time of evaluation.
- **EVA-Bench** [15]: An end-to-end framework conducting bot-to-bot audio conversations across enterprise domains; its domain focus and evaluation setup are misaligned with our open-ended dialogue scenario.
- **URO-Bench** [163]: An S2S benchmark covering multilingualism, multi-round dialogues, and paralinguistics; our model was not trained for multilingual or paralinguistic tasks, making this benchmark a poor fit.

### 4.2.1 General Capabilities

VoiceAssistant-Eval [151] includes tasks across listening, speaking and viewing. Of these three, viewing is clearly intended for multimodal models, unlike ours, and listening comprises tasks where an audio recording is played (environmental sounds/music/someone else's speech) and then the user asks a question about that audio. These instances are also unsuitable for our model, because it was trained solely on dialogue. Therefore, we limit the benchmark only to the speaking part (excluding the role-play task, because our assistant was not trained for those scenarios either).

A strength of VoiceAssistant-Eval is that it does not rely solely on text instructions. Inspecting audio properties is essential from the viewpoint of robustness, because in certain use cases the textual output is not even available or useful to the user. Another key part of this benchmark is that it includes cases in varied audio contexts under realistic conditions (background noise, etc.). It also tests multi-round conversations, which only a few similar benchmarks do.

VoiceAssistant-Eval evaluates responses along three dimensions, and the final score per sample is their product converted to a percentage. **Content Score** is judged by an LLM—gpt-oss-20b [3] using task-specific prompts; for emotion-related tasks it also incorporates per-emotion probabilities from an emotion recognition model [93]. **Consistency Score** is computed as a modified Word Error Rate (WER) between a Whisper-Large-v3 [118] transcription of the audio output and the model's text response; a length threshold is applied to handle cases where the model produces only a very short answer rather than a full response. **Speech Score** is measured with UTMOS [125], a learned metric that reflects fluency and naturalness of synthesised speech.

The results can be found in Table 4.8. We report the average score across all speaking tasks as well as the scores for each category. For context, we share some of the results from the original study. `moshika-pytorch-bf16` and `moshiko-pytorch-bf16` are two Moshi variants (female and male voice respectively) from the original paper [41]. Freeze-Omni [154] is a cascaded full-duplex model. We also include the reported best open-source result from the original paper in each category—this may come from various models—and GPT-4o-Audio [111]

as a strong proprietary reference. The speaking categories used in the table are abbreviated as follows: AST (Assistant), EMO (Emotion), IF (Instruction Following), MR (Multi-Round), RSN (Reasoning), RBT (Robustness), and SFT (Safety). Ove. denotes the overall score across categories.

| **Metric / Model** | **AST** | **EMO** | **IF** | **MR** | **RSN** | **RBT** | **SFT** | **Ove.** |
|---|---|---|---|---|---|---|---|---|
| Content Score | 11.8 | 11.1 | 4.0 | 16.1 | 20.4 | 8.4 | 58.9 | |
| Consistency Score | 76.6 | 85.8 | 82.5 | 91.0 | 78.5 | 90.7 | 79.4 | |
| Speech Score | 63.9 | 69.7 | 69.1 | 78.7 | 66.2 | 76.8 | 67.9 | |
| Overall Score (ours) | 5.8 | 6.7 | 2.3 | 11.6 | 10.6 | 5.9 | 31.8 | *10.65* |
| moshika-pytorch-bf16 | 1.6 | 3.1 | 1.6 | 0.8 | 4.0 | 0.3 | 17.8 | 3.65 |
| moshiko-pytorch-bf16 | 1.6 | 3.4 | 1.3 | 2.1 | 4.7 | 0.4 | 23.7 | 4.66 |
| Freeze-Omni | 12.1 | 23.8 | 11.0 | 18.6 | 25.2 | 24.2 | 79.8 | 24.34 |
| Best open-source | 51.5 | 33.6 | 33.5 | 55.7 | 50.5 | 38.6 | 79.8 | 41.27 |
| GPT-4o-Audio | 62.7 | 32.5 | 44.3 | 64.0 | 63.8 | 54.7 | 74.5 | 51.26 |

**Table 4.8** VoiceAssistant-Eval speaking-part results by category. Each cell in the bottom part shows the Overall Score (the product of Content, Consistency, and Speech scores, expressed as a percentage).

Our overall speaking score is 10.65. This value is the average over the currently reported speaking metrics only. Because role-play is excluded, the overall score is not fully representative and should not be compared directly to fully reported category averages of other models. The best open-source overall score is the best score achieved by one model, not the average of the best scores in each category.

Overall, our model underperforms compared with stronger contemporary systems (which may not necessarily be full-duplex) on this benchmark. At the same time, it consistently outperforms both Moshi variants (`moshika-pytorch-bf16` and `moshiko-pytorch-bf16`) across all reported speaking categories. A key factor behind the weaker absolute scores is distribution mismatch: Warmblood is optimized for real-time dialogue, whereas many VoiceAssistant-Eval speaking instances are framed as instruction-like, single-shot requests that are closer to text-LLM usage patterns than to conversational turn-taking. This style caters more to turn-based audio models. In addition, we used the general prompt variant for guide generation. In instruction-heavy cases, this can confuse the guide generator, which may occasionally produce guidance that encourages follow-up questioning instead of direct answering. Two practical improvement paths follow from this observation: (1) prompt-level adaptation that explicitly biases the guide generator toward concise, instruction-following answers in this benchmark setting; and (2) data-level adaptation via additional training data that better reflects instruction-style speaking tasks. The second option is more costly but offers a more general and potentially more robust long-term solution.

We manually inspected some of the evaluation outputs. Some downsides of our approach show up in these results: in a few instances the VAD period appears too short—the user has not yet finished their speech but the system thinks it has, causing it to issue multiple guide requests for what is effectively a single user turn, and the model occasionally repeats the guide verbatim when it is instructed to “ask” or “acknowledge” something (i.e., it echoes those words rather than

performing the action). Addressing these issues likely requires additional tuning of guide-following behavior.

One limitation of this benchmark is that the content score is computed from binary *good*/*bad* judgments. This coarse scale can penalize near-correct answers the same as clearly wrong ones, reducing sensitivity to incremental improvements and masking partial correctness.

### 4.2.2 Interaction Capabilities

Full-Duplex-Bench (FDB) comprises four different versions. Of these, we chose only v1.0 [86] and v1.5 [85] because they are closely tied together. They both appear in many other research papers and offer various existing model comparisons themselves. v2 [84] is much newer (and not as widely used) and less straightforward—it requires running an additional real-time framework. v3 [82] requires tool-calling capabilities, which our model does not support.

#### Full-Duplex-Bench v1.0

For FDB v1.0, we follow the benchmark protocol based on a unified speech input stream and dimension-specific test scenarios. For each sample, the model produces speech output, which is transcribed with word-level timestamps (via ASR), and dedicated metrics are then computed per evaluation dimension.

The benchmark uses three central concepts. **Backchannel** denotes short listener feedback while the other side is speaking (e.g., "mm-hmm"), and in this benchmark is operationalized as utterances shorter than 1 second and containing fewer than two words. **Takeover (TO)** is a binary variable indicating whether the model attempts to assume the turn: silence or backchannel is treated as $TO = 0$, while any other non-silent response is $TO = 1$. **Takeover Rate (TOR)** is the dataset average of this binary variable.

FDB v1.0 evaluates four dimensions. In **Pause Handling**, lower TOR is better, because the model should not take over during natural user hesitations. In **Backchanneling**, three metrics are used to measure the model's backchanneling capacity: lower TOR (avoid dominating), backchannel frequency (events per second, interpreted jointly with timing), and lower Jensen-Shannon Divergence (JSD), which measures alignment between model and human backchannel timing distributions. In **Smooth Turn Taking**, the model should respond quickly after the user ends; thus lower response latency is better, and latency is measured only for takeover cases (when the model does react). In **User Interruption**, desirable behavior is taking the turn after interruption (higher TOR), producing a coherent, adaptive answer (higher GPT-4o judge score), and doing so quickly (lower latency after interruption).

The results are presented in Table 4.9. We offer a comparison with the models from the original paper [86], along with other similar works that report FDB results [124, 32]. The Gemini model the authors used was `gemini-2.0-flash-live-001`.

Our model performs very well on pause handling. It also scores the highest frequency and lowest Jensen-Shannon Divergence in backchanneling. Its biggest weakness is smooth turn-taking. We assume that the frequent silence we observed in 4.1.5 contributes here, too. On user interruption, Warmblood performs similarly

| Model | Pause (Synthetic) | Pause (Candor) | Backchannel | | |
|---|---|---|---|---|---|
| | TOR ↓ | TOR ↓ | TOR ↓ | Freq ↑ | JSD ↓ |
| Warmblood (ours) | 0.328 | 0.148 | 0.600 | 0.045 | 0.824 |
| PersonaPlex [124] | 0.584 | 0.662 | 0.327 | 0.025 | 0.649 |
| MoshiRAG [32] | 0.320 | 0.560 | 0.640 | 0.010 | 0.940 |
| Freeze-Omni [154] | 0.642 | 0.481 | 0.636 | 0.001 | 0.997 |
| Gemini [53] | 0.255 | 0.310 | 0.091 | 0.012 | 0.896 |
| Moshi [41] | 0.985 | 0.980 | 1.000 | 0.001 | 0.957 |
| dGSLM [104] | 0.934 | 0.935 | 0.691 | 0.015 | 0.934 |

| Model | Smooth Turn Taking | | User Interruption | | |
|---|---|---|---|---|---|
| | TOR ↑ | Latency ↓ | TOR ↑ | GPT-4o ↑ | Latency ↓ |
| Warmblood (ours) | 0.647 | 0.797 | 0.913 | 3.154 | 1.051 |
| PersonaPlex | 0.992 | 0.070 | 1.000 | 4.210 | 0.400 |
| MoshiRAG | 0.830 | 0.180 | 0.850 | 3.750 | 1.020 |
| Freeze-Omni | 0.336 | 0.953 | 0.867 | 3.615 | 1.409 |
| Gemini | 0.655 | 1.301 | 0.891 | 3.376 | 1.183 |
| Moshi | 0.941 | 0.265 | 1.000 | 0.765 | 0.257 |
| dGSLM | 0.975 | 0.352 | 0.917 | 0.201 | 2.531 |

**Table 4.9** FDB v1.0 results.

to some of the other models in TOR and GPT-4o score. Overall, it seems that after fine-tuning we lost the short latency Moshi originally had.

### Full-Duplex-Bench v1.5

FDB v1.5 shares the same streaming evaluation framework as v1.0 but shifts focus from general turn-taking behavior to how models react during *controlled overlap events*—moments when user audio arrives while the model is already speaking. The core evaluation primitive is a *trial*: pre-recorded user audio is injected mid-model-speech, and the model output is segmented into pre-overlap and post-overlap regions so that reactions can be measured precisely and in isolation.

Four scenarios probe distinct conversational capabilities. In **User Interruption**, the user barges in with a new request; the model should rapidly yield the floor and address the new query. In **User Backchannel**, the user produces a short affirmation such as "uh-huh" or "mm-hmm"—a non-floor-taking cue that the model should filter out and continue speaking through. In **User Talking to Others**, the user addresses a third party, not the system; the model should recognize it is not the intended recipient and gracefully hold. In **Background Speech**, an unrelated voice is present in the environment; the model should remain focused and not be derailed.

The post-overlap response is classified into one of four semantic categories using a GPT-4o judge operating on ASR transcripts: **Respond** (addresses the

overlapping content), **Resume** (continues prior turn, ignoring the overlap), **Uncertain** (asks for clarification), or **Unknown** (irrelevant or silent). Two timing measures are also computed. **Stop latency** ($t_{\text{stop}} = t_{\text{model\_stop}} - t_{\text{user\_start}}$) is the interval from the onset of the overlapping speech to when the model stops—a short value indicates rapid yielding, while a high value indicates floor-holding. **Response latency** ($t_{\text{resp}} = t_{\text{model\_start}} - t_{\text{user\_end}}$) is the gap from the end of the overlap to the model's next utterance; lower is better across all scenarios.

The expected outcomes differ by scenario: for User Interruption, a high Respond rate with low stop and response latency; for the other three scenarios, a high Resume rate with high stop latency (the model should not easily yield) and low response latency once it does resume.

Results are shown in Table 4.10. We compare with the models that were evaluated in the original paper: Freeze-Omni [154], Moshi [41], Gemini [37] (`gemini-2.0-flash-live-001`), Amazon Nova Sonic [64] (`amazon.nova-sonic-v1:0`) and GPT-4o Realtime [111] (`gpt-4o-realtime-preview-2024-12-17`).

| Scenario | Metric | Freeze-Omni | Moshi | Gemini | Sonic | GPT-4o | Warmbl. (ours) |
|---|---|---|---|---|---|---|---|
| User Inter. | RESPOND ↑ | 0.72 | 0.50 | 0.33 | 0.24 | 0.78 | 0.34 |
| | RESUME ↓ | 0.12 | 0.26 | 0.55 | 0.71 | 0.10 | 0.29 |
| | UNCERTAIN ↓ | 0.03 | 0.00 | 0.01 | 0.01 | 0.02 | 0.05 |
| | UNKNOWN ↓ | 0.13 | 0.25 | 0.10 | 0.04 | 0.12 | 0.32 |
| | STOP (s) ↓ | 1.42 | 1.16 | 2.20 | 2.25 | 0.23 | 1.43 |
| | RESP (s) ↓ | 1.35 | 1.47 | 2.62 | 2.75 | 1.50 | 1.31 |
| User Backch. | RESPOND ↓ | 0.07 | 0.02 | 0.01 | 0.00 | 0.03 | 0.04 |
| | RESUME ↑ | 0.80 | 0.06 | 0.93 | 0.98 | 0.70 | 0.24 |
| | UNCERTAIN ↓ | 0.02 | 0.00 | 0.02 | 0.00 | 0.01 | 0.04 |
| | UNKNOWN ↓ | 0.11 | 0.92 | 0.04 | 0.02 | 0.25 | 0.67 |
| | STOP (s) ↑ | 0.66 | 0.42 | 0.66 | 0.64 | 0.21 | 0.61 |
| | RESP (s) ↓ | 2.16 | 3.00 | 2.45 | 1.45 | 1.32 | 1.89 |
| Talking to Oth. | RESPOND ↓ | 0.58 | 0.20 | 0.00 | 0.10 | 0.91 | 0.09 |
| | RESUME ↑ | 0.25 | 0.19 | 0.99 | 0.90 | 0.02 | 0.17 |
| | UNCERTAIN ↑ | 0.00 | 0.02 | 0.00 | 0.00 | 0.01 | 0.04 |
| | UNKNOWN ↓ | 0.15 | 0.59 | 0.01 | 0.00 | 0.06 | 0.70 |
| | STOP (s) ↑ | 1.39 | 0.87 | 1.69 | 1.77 | 0.18 | 0.91 |
| | RESP (s) ↓ | 1.32 | 2.38 | 1.78 | 2.04 | 1.16 | 1.45 |
| Backg. Speech | RESPOND ↓ | 0.63 | 0.21 | 0.70 | 0.01 | 0.93 | 0.10 |
| | RESUME ↑ | 0.25 | 0.07 | 0.30 | 0.98 | 0.04 | 0.14 |
| | UNCERTAIN ↑ | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.04 |
| | UNKNOWN ↓ | 0.11 | 0.71 | 0.00 | 0.01 | 0.03 | 0.72 |
| | STOP (s) ↑ | 0.98 | 0.54 | 0.95 | 1.05 | 0.18 | 0.76 |
| | RESP (s) ↓ | 1.60 | 1.62 | 2.38 | 2.76 | 1.26 | 1.48 |

**Table 4.10** FDB v1.5 results by scenario. STOP and RESP denote stop latency and response latency in seconds.

Looking at the correct determination of behavior, our model performs poorly

in the user interruption scenario. With backchanneling, it produces many *unknown* outputs but still outperforms Moshi. In the talking-to-others scenario, a similar outcome appears, with results comparable to Moshi on resumption but fewer (incorrect) responses. Background speech presents almost the same unknown score as Moshi but better correct classification in the other cases.

In terms of latency, Warmblood performs acceptably—its scores are broadly in line with the other models. This does not contradict the poor smooth turn-taking latency seen in FDB v1.0: that benchmark measures how quickly the model responds after the user finishes speaking, whereas here latency is measured relative to the onset or end of an overlap event mid-model-speech, a fundamentally different scenario. Notably, Moshi scores poorly on most latency metrics here, and Warmblood outperforms it in every case except STOP during user interruptions.

## 4.3 User Testing

The goal was to evaluate the system in realistic dialogue conditions. Unstructured conversations with no explicit goal would be difficult to assess and would not reflect typical use cases; therefore, we designed two primary interaction scenarios. Half (50) of the participants were given a chitchat-style goal, in which they were simply asked to casually talk with the voicebot. The other half was tasked with an information retrieval (Q&A) goal: to ask questions about a certain topic and try to find out more about it.

Participants received instructions on how to use the system along with an LLM-generated contextual hint—a short text based on a randomly selected topic from 50 possibilities. These instructions were generated in the style of one of the two task types (chitchat or Q&A). Participants were asked to conduct a conversation lasting 3–5 minutes. Conversations shorter than 2 minutes were not allowed, though no upper time limit was imposed. The instruction-generation templates and topic lists are provided in Appendix A.3.5.

After reading their instructions, participants initiated the online conversation. The interface displayed a visual indicator showing both the participant and the system's speech. Real-time transcriptions showed both the model's output (its own generated text) and what the participant said (ASR transcriptions produced by the guide loop's ASR, described in Section 2.2). The guide remained hidden from participants. A text box displayed their assigned task instructions throughout the interaction. The interface is shown in Figure 4.4.

Following each dialogue, participants completed a brief questionnaire about their experience. Both scenarios used identical questions, each rated on a 0–10 scale. The metrics and corresponding survey prompts were as follows:

- **General:** "What is your overall feeling about the conversation?"
  0 = Horrible, 10 = Amazing
- **Coherence:** "How coherent was the conversation? Did you manage to fulfill your task? Did the bot understand the topics and responded accordingly?"
  0 = Zero understanding, 10 = Perfect understanding
- **Truthfulness:** "How truthful do you think the bot's responses were? (You don't need to fact-check, assess this based on your own knowledge.)"


**Figure 4.4** User testing interface used during the study, including live speech indicator, dialogue transcript, and displayed task prompt.

0 = Always lying, 10 = Absolutely truthful

- **Naturalness:** "How natural did the conversation feel, compared to your every-day interactions? Was the bot able to react appropriately to your queries?"
  0 = Extremely robotic, 10 = Absolutely natural

- **Promptness:** "Did the bot answer promptly (as a human would)? Was it able to react quickly in case you interrupted its talk?"
  0 = Super slow, 10 = On time/very fast

We conducted the tests with 100 human participants. We tested only with participants located in Europe due to server location constraints. The study was conducted entirely in English, though participants did not need to be native English speakers; self-reported English fluency was sufficient. Participants were recruited via Prolific and were paid £2.50 for a 10-minute task (equivalent to £15/hour, in line with Prolific recommendations at the time). The study protocol was approved by an internal ethical advisor of the supervisor.

### 4.3.1 Quantitative Results

We report the averaged results in Table 4.11 for both scenarios. Mean ratings generally clustered around 5 with substantial variance. Notably, all categories showed higher ratings for the Q&A scenario, which can be attributed to the fact that these situations were closer to the data which the model was trained on.

| | General | Coherence | Truthful. | Naturalness | Promptness |
|---|---|---|---|---|---|
| Chitchat | $4.52 \pm 2.96$ | $5.36 \pm 3.35$ | $5.84 \pm 3.31$ | $3.82 \pm 3.25$ | $5.00 \pm 3.26$ |
| Q&A | $5.12 \pm 2.81$ | $5.96 \pm 2.94$ | $6.54 \pm 2.57$ | $4.12 \pm 2.79$ | $5.30 \pm 2.94$ |

**Table 4.11** Average user ratings (0–10 scale) and standard deviation across all participants and both task scenarios.

The ranking of metrics remained consistent across both scenarios, ordered from highest to lowest: Truthfulness, Coherence, Promptness, General, and Naturalness. This pattern suggests that the system performs fairly well at providing factual information (leveraging the guide) and maintaining conversational coherence, yet struggles with maintaining a natural flow and creating an overall positive user impression.

Figures 4.5 and 4.6 display correlation matrices for both scenarios. We can see that most of the metrics are strongly correlated, especially in the chitchat scenario. It probably means that the users' overall experience strongly influenced all of their scores.

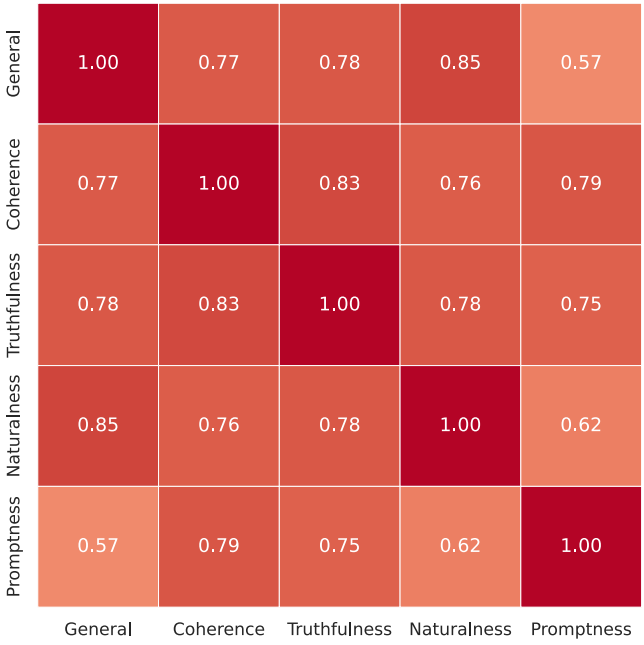


**Figure 4.5** Correlation matrix of user ratings for the chitchat scenario, showing pairwise associations between evaluation metrics.

Observed conversation lengths are summarized in Table 4.12. The averages in both scenarios are close to 5 minutes, while the maxima indicate that some participants continued substantially longer.

| Scenario | Average | Std. dev. | Min | Max |
|---|---|---|---|---|
| Chitchat | 4.88 min | 2.28 min | 2.23 min | 15.22 min |
| Q&A | 4.95 min | 2.09 min | 2.47 min | 11.49 min |

**Table 4.12** Conversation length statistics in the user study.

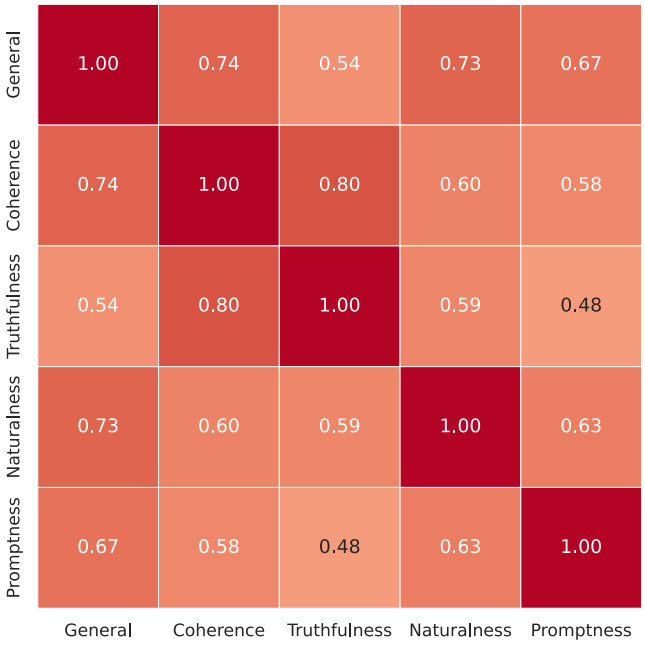


**Figure 4.6** Correlation matrix of user ratings for the Q&A scenario, showing pairwise associations between evaluation metrics.

### 4.3.2 Technical Issues and Filtered Analysis

Participants were asked to report technical difficulties, and we reviewed the corresponding audio recordings. We found no major system malfunctions. Occasionally, we experienced lower audio quality on the participant input or a higher latency. Most perceived technical problems were instances where the desired model output failed—specifically, cases where the system did not respond or produced unintelligible speech, stuttering, or background noise. When these prevailed, it could cause the interaction to be completely unsatisfactory. These observations parallel issues we observed during initial Moshi testing. Participants additionally reported minor problems such as incorrect transcriptions from either ASR or the model's text output or the model interrupting them too quickly.

Since we only had textual feedback regarding technical difficulties, we manually categorized reported issues. We distinguished two groups: those experiencing *severe problems* (e.g., minimal assistant speech output) and those reporting *minor difficulties*—stuttering, uncomfortable silences, frequent interruptions, incorrect transcriptions, and similar issues. The former group usually provided very low ratings across all metrics, while the latter gave varied ratings that were possibly reduced by the issues but not uniformly poor. This factor may have contributed to the strong correlations we saw earlier.

For the chitchat scenario, we identified **10** participants in the severe problems group and **23** in the minor difficulties group. In the Q&A scenario, **3** participants experienced severe problems and **29** reported minor difficulties.

We excluded participants with severe technical issues and analyzed ratings from the remaining users (40 for chitchat, 47 for Q&A), presented in Table 4.13. Interestingly, the relative metric rankings between the two scenarios changed. In the full chitchat results, participants showed a somewhat bimodal distribution—either experiencing extremely poor interactions or reasonably successful ones.

Conversely, Q&A participants rarely encountered major difficulties, yet their overall conversational quality was modestly lower.

| | **General** | **Coherence** | **Truthful.** | **Naturalness** | **Promptness** |
|---|---|---|---|---|---|
| Chitchat | $5.45 \pm 2.44$ | $6.70 \pm 2.22$ | $7.13 \pm 2.17$ | $4.78 \pm 2.93$ | $6.25 \pm 2.32$ |
| Q&A | $5.38 \pm 2.68$ | $6.30 \pm 2.69$ | $6.79 \pm 2.40$ | $4.34 \pm 2.72$ | $5.55 \pm 2.83$ |

**Table 4.13** User ratings excluding participants with severe technical issues (40 chitchat, 47 Q&A participants).

### 4.3.3 Qualitative Feedback

Participants were also invited to provide verbal comments about their experience and highlight significant moments. Responses varied widely—some reported positive interactions while others expressed dissatisfaction. A recurring theme was that conversations felt robotic. We present detailed findings from both scenarios below.

#### Chitchat Scenario Observations

Many participants reported that the assistant ignored certain questions or discussion topics. To make individual viewpoints explicit, we summarize representative participant statements below:

- One participant described initially positive rapport that deteriorated over time, noting the agent "focused mainly on delivering information rather than actively engaging."
- Another participant observed that "there was no give and take between normal human interactions in speech, i.e. waiting for every person to reply, there were many interruptions and not stopping naturally." The same participant also noted that responses seemed generic, "as if pulled from Wikipedia," suggesting correct guide usage but unnaturally formal presentation.
- One user commented, "By the end he wasn't talking as much. Seems like it wanted to get out whatever was programmed in the beginning," which implies guide-driven behavior but highlights the negative effects of training on short conversations only.
- A generally satisfied participant praised the assistant's comprehension and above-average audio quality but still observed: "It did not feel like a natural conversation and it more felt like it was talking *at* me rather than *with* me."

Several participants additionally mentioned being interrupted before they could finish their thoughts, reinforcing turn-taking as a central weakness in the chitchat setting.

### Question-Answering Scenario Observations

Interruptions were a significantly more prevalent complaint in this scenario. Representative participant-level observations are listed below:

- Multiple participants reported that the assistant began responding before they finished speaking, leading to fragmented exchanges. Several described the system as "too fast," with insufficient time to complete thoughts or formulate follow-up questions.
- Some participants nevertheless expressed satisfaction, reporting that the assistant engaged appropriately with their topics and provided substantive answers in standard question-answering turns.
- One participant reported factually incorrect information. This can occur when the model attempts to discuss specific names or numerical values: even when guides contain correct information, the model may occasionally misinterpret and communicate incorrect content. For example, a name of a camera model or its price.
- Another participant summarized the interaction as follows: "Like talking to an uninformed teenager. Only talked in generalities and platitudes. Also extremely robotic."

Taken together, these comments suggest that while the assistant can perform adequately in standard turn-taking exchanges, it still struggles with interruption timing and with extended, nuanced discussions requiring deeper reasoning.

### Dataset Selection Behavior

During review of the recordings, we observed instances in which the assistant appeared to implicitly *select* which training dataset distribution to follow. For instance, conversations beginning with casual small talk received prompt, natural responses. However, when participants transitioned to factual questions, the system seemed disoriented—despite guides being generated (and regenerated following user requests), the model would fail to respond and exhibited highly unnatural behavior. This behavior may indicate that the model struggles to adapt between dialogue styles, particularly when transitioning from open-domain to task-oriented interaction.

### Conversation Initialization

Many participants began by remaining silent. While original Moshi typically initiates with a greeting such as "What can I help you with?", we considered such starts unnecessary and potentially disruptive if participants intended to speak immediately. However, the frequent initial silence appears to have confused the system. On several occasions, after prolonged user silence (exceeding 10 seconds), the assistant spontaneously began speaking about unrelated topics. Participants did not always successfully redirect the conversation, resulting in awkward exchanges some of which never recovered.

# Conclusion

In this work, we explored the current state of full-duplex spoken dialogue systems and their extension with external text guidance. We present Warmblood, a modified and fine-tuned version of Moshi [41]. The core architectural contribution is a new token channel carrying guide text, added to the input, and an inference loop with an external LLM which generates the guide. The system was trained on conversational data in which the assistant side is paired with externally supplied guide annotations. During inference, the guide is provided by an LLM that receives a transcript of the conversation and generates concise steering text.

Our internal evaluation results indicate that the model learns to use the guide, significantly improving its question-answering capabilities. Task type had a notable impact on performance: when training data for a given category was excluded, results on the corresponding evaluation tasks degraded. We confirmed that including general conversations—for example, from the Fisher [34] dataset—also improved overall scores. Overall, the experiments confirmed that combining all training categories yields the best results, and that providing the guide at inference time meaningfully raises answer quality relative to operating without one.

We then evaluated the system on two established external benchmarks. First, VoiceAssistant-Eval [151] measures content-oriented capabilities such as question answering, multi-round memory, and reasoning; Warmblood outperforms Moshi on all of its categories across speech consistency, robustness, and safety. Full-Duplex-Bench [86] assesses interaction-level behavior such as turn-taking, response latency, and handling of overlapping speech; here the results were mixed. In some respects (pause handling and backchanneling), Warmblood performed better, but in others it lost some capability for smooth turn-taking and low latency.

We also conducted a user study in which participants interacted with the system and rated their experience. The results were moderate, with stronger scores on coherence and intelligence and lower scores on naturalness and overall impression. Although these results are not fully satisfactory, we encourage future researchers to conduct similar evaluations. User studies are an indispensable assessment step for SDSs, and evaluation should not rely solely on artificial benchmarks.

There is a notable gap between our internal evaluations and the user study results, with external benchmark scores falling somewhere in between. Our internal naturalness scores are high—LLM-as-judge metrics typically placed the model at around 4 out of 5—whereas user ratings averaged around 5 out of 10. This discrepancy is likely because our evaluation datasets closely mirror the fine-tuning scenarios, while real user interactions are far less predictable. Furthermore, our evaluations were done on text outputs, while the users communicated with the whole system. Nevertheless, the model did retain meaningful conversational capabilities even under those conditions.

The greatest limitation of Warmblood is that it can still be unpredictable: it may fail to respond to the user, ignore the guide, or misunderstand it. It performs best in scenarios closely resembling those seen during training, but it cannot always maintain full naturalness in open-ended conversations. Although

its theoretical response latency is low, we were not able to train the model to react as quickly as Moshi typically does. While Warmblood outperforms Moshi on speech intelligence tasks, it does not yet match the quality of other state-of-the-art systems.

We note that MoshiRAG [32] achieves a very similar effect to Warmblood. The principal difference is that MoshiRAG focuses specifically on knowledge retrieval, whereas our system is designed to accept guide information of any kind. In MoshiRAG, the core model decides when to issue a retrieval request; in our approach, that decision is handled externally. Both architectures incur a delay between the moment a request is made and the moment the external response becomes available.

# Future Work

We consider our main contribution to be a prototype demonstrating that an online SDS can use a textual guide to produce more intelligent answers. The system retains certain limitations, however, which we address in the following discussion.

A primary direction for improvement lies in the training data. Greater diversity of sources and more natural interaction patterns would likely yield better performance. Most of our fine-tuning used regular question-answer turns with no spontaneous activity. Modern TTS systems [80, 122] could be leveraged to introduce greater prosodic variability and emotional expressiveness into the synthesized training data.

Another area that warrants further exploration is the inference and runtime architecture. We adopted a straightforward approach involving a quick LLM, a short prompt, VAD, and burst ASR. Possible extensions include the following:

- Alternative ASR systems, including continuously streaming ones that do not require the user to finish speaking before transcription begins [14].
- A different LLM, or a combination of LLMs in which one model responds quickly while another provides more reliable answers with function calling, reasoning, and similar capabilities.
- Prompt engineering—systematic testing across diverse situations to design prompts that yield concise, precise answers with high coherence and minimal repetition.

Ideally, training would incorporate scenarios that reflect the full inference framework, so that the model learns to operate with the live guide system in mind.

One aspect that was not explored is the use of negative (incorrect) guide signals. The training data could include examples in which the guide contradicts the factual answer, forcing the model to learn to follow the guide regardless of its own prior knowledge. We focused instead on promoting overall correctness and relied on the model to make use of the guide when it was provided. Incorporating negative guides could improve the reliability of guide adherence and, combined with a richer training regime, could yield a more controllable system.

Warmblood could be further strengthened by pairing it with techniques such as the *system prompt* mechanism introduced by PersonaPlex [124]. In that approach, the model is informed of its role from the outset, which constrains overall variability while still leaving room for external control via the guide mechanism. This architecture closely resembles how contemporary LLMs are used in production settings: a system prompt establishes the context, and tool calls supply dynamic information at runtime.

## Safety

Deploying a spoken dialogue system raises safety considerations that, while not the primary focus of this work, merit brief discussion. Warmblood inherits its safety behavior from two components. The base model, Moshi, carries its safety alignment that was instilled during pretraining [41]; our fine-tuning did not target safety explicitly, nor did we introduce any harmful content into the training data. The guide-generating LLM contributes its own safety mechanisms: a typical model will refuse to produce harmful or inappropriate content, providing a degree of implicit filtering for the guide channel.

One specific risk is impersonation: a voice-based AI assistant could be used to deceive users into believing they are speaking with a real person, raising both ethical and legal concerns. Any deployment of such a system should ensure that users are clearly informed of its AI nature [128, 155].

A further concern is adversarial manipulation. Because the guide LLM is exposed to conversation transcripts, a user could attempt to steer its output through carefully crafted utterances—a form of prompt injection. Mitigating this risk would require input validation or sandboxed prompting strategies on the guide-generation side [25, 137].

In summary, no targeted safety training beyond what is present in the pre-trained models was applied. For any production deployment, dedicated measures—such as content filtering, output monitoring, and explicit disclosure of the system's AI nature—would be necessary.

# Bibliography


1. Addlesee, Angus; Cherakara, Neeraj; Nelson, Nivan; Hernandez Garcia, Daniel; Gunson, Nancie; Sieińska, Weronika; Dondrup, Christian; Lemon, Oliver. Multi-party Multimodal Conversations Between Patients, Their Companions, and a Social Robot in a Hospital Memory Clinic. In: Aletras, Nikolaos; De Clercq, Orphee (eds.). *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. St. Julians, Malta: Association for Computational Linguistics, 2024, pp. 62–70. Available also from: `https://aclanthology.org/2024.eacl-demo.8`.

2. Addlesee, Angus; Eshghi, Arash; Konstas, Ioannis. *Current Challenges in Spoken Dialogue Systems and Why They Are Critical for Those Living with Dementia*. arXiv, 2019. No. arXiv:1909.06644. Available from doi: `10.48550/arXiv.1909.06644`.

3. Agarwal, Sandhini; Ahmad, Lama; Ai, Jason; Altman, Sam; Applebaum, Andy; Arbus, Edwin; Arora, Rahul K; Bai, Yu; Baker, Bowen; Bao, Haiming, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*. 2025.

4. Al Moubayed, Samer; Beskow, Jonas; Skantze, Gabriel; Granström, Björn. Furhat: a back-projected human-like robot head for multiparty human-machine interaction. In: *Cognitive behavioural systems: COST 2102 international training school, dresden, Germany, february 21-26, 2011, revised selected papers*. Springer, 2012, pp. 114–130.

5. Aliannejadi, Mohammad; Kiseleva, Julia; Chuklin, Aleksandr; Dalton, Jeff; Burtsev, Mikhail. *ConvAI3: Generating Clarifying Questions for Open-Domain Dialogue Systems (ClariQ)* [arXiv.org]. 2020-09-23. Available also from: `https://arxiv.org/abs/2009.11352v1`.

6. Allen, James; Campana, Ellen; Gomez Gallo, Carlos; Stoness, Scott; Swift, Mary; Tanenhaus, Micheal K. Incremental dialogue system faster than and preferred to its nonincremental counterpart. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. 2007, vol. 29. No. 29. Available also from: `https://escholarship.org/content/qt0f5051fq/qt0f5051fq.pdf`.

7. Ao, Junyi; Wang, Yuancheng; Tian, Xiaohai; Chen, Dekun; Zhang, Jun; Lu, Lu; Wang, Yuxuan; Li, Haizhou; Wu, Zhizheng. SD-Eval: A Benchmark Dataset for Spoken Dialogue Understanding Beyond Words. *Advances in Neural Information Processing Systems*. 2024, vol. 37, pp. 56898–56918. Available from doi: `10.52202/079017-1813`.

8. Araki, Masahiro; Tomimasu, Sayaka; Nakano, Mikio; Komatani, Kazunori; Okada, Shogo; Fujie, Shinya; Sugiyama, Hiroaki. Collection of Multimodal Dialog Data and Analysis of the Result of Annotation of Users' Interest Level. In: Calzolari, Nicoletta; Choukri, Khalid; Cieri, Christopher; Declerck, Thierry; Goggi, Sara; Hasida, Koiti; Isahara, Hitoshi; Maegaard, Bente; Mariani, Joseph; Mazo, Hélène; Moreno,

Asuncion; Odijk, Jan; Piperidis, Stelios; Tokunaga, Takenobu (eds.). *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018. Available also from: https://aclanthology.org/L18-1250.

9. Ardila, Rosana; Branson, Megan; Davis, Kelly; Kohler, Michael; Meyer, Josh; Henretty, Michael; Morais, Reuben; Saunders, Lindsay; Tyers, Francis; Weber, Gregor. Common voice: A massively-multilingual speech corpus. In: *Proceedings of the twelfth language resources and evaluation conference*. 2020, pp. 4218–4222.

10. Arora, Siddhant; Chang, Kai-Wei; Chien, Chung-Ming; Peng, Yifan; Wu, Haibin; Adi, Yossi; Dupoux, Emmanuel; Lee, Hung-Yi; Livescu, Karen; Watanabe, Shinji. *On The Landscape of Spoken Language Models: A Comprehensive Survey*. arXiv, 2026. No. arXiv:2504.08528. Available from doi: 10.48550/arXiv.2504.08528.

11. Arora, Siddhant; Tian, Jinchuan; Futami, Hayato; Shi, Jiatong; Kashiwagi, Yosuke; Tsunoo, Emiru; Watanabe, Shinji. *Chain-of-Thought Reasoning in Streaming Full-Duplex End-to-End Spoken Dialogue Systems*. arXiv, 2025. No. arXiv:2510.02066. Available from doi: 10.48550/arXiv.2510.02066.

12. Baevski, Alexei; Zhou, Yuhao; Mohamed, Abdelrahman; Auli, Michael. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*. 2020, vol. 33, pp. 12449–12460.

13. Banerjee, Satanjeev; Lavie, Alon. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 2005, pp. 65–72.

14. Banfic, Nenad; Fan, David; Vaishnavi, Kunal; Kemp, Sam; Choi, Sunghoon; Ren, Rui; Shaw, Sayan; Tang, Meng. Pushing the Limits of On-Device Streaming ASR: A Compact, High-Accuracy English Model for Low-Latency Inference. *arXiv preprint arXiv:2604.14493*. 2026.

15. Bogavelli, Tara; Melançon, Gabrielle Gauthier; Stankiewicz, Katrina; Bamgbose, Oluwanifemi; Riols, Fanny; Nguyen, Hoang H.; Mehndiratta, Raghav; Brin, Lindsay Devon; Marinier, Joseph; Subramani, Hari; Madamala, Anil; Nemala, Sridhar Krishna; Sunkara, Srinivas. *EVA-Bench: A New End-to-end Framework for Evaluating Voice Agents*. arXiv, 2026. No. arXiv:2605.13841. Available from doi: 10.48550/arXiv.2605.13841.

16. Boland, Julie E.; Fonseca, Pedro; Mermelstein, Ilana; Williamson, Myles. Zoom disrupts the rhythm of conversation. *Journal of Experimental Psychology: General*. 2022, vol. 151, no. 6, pp. 1272–1282. issn 1939-2222. Available from doi: 10.1037/xge0001150. Place: US.

17. Borsos, Zalán; Marinier, Raphaël; Vincent, Damien; Kharitonov, Eugene; Pietquin, Olivier; Sharifi, Matt; Roblek, Dominik; Teboul, Olivier; Grangier, David; Tagliasacchi, Marco; Zeghidour, Neil. AudioLM: A Language Modeling Approach to Audio Generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2023, vol. 31, pp. 2523–2533. ISSN 2329-9304. Available from DOI: 10.1109/TASLP.2023.3288409. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.

18. Branigan, Holly P.; Pickering, Martin J.; Cleland, Alexandra A. Syntactic co-ordination in dialogue. *Cognition*. 2000, vol. 75, no. 2, B13–B25. ISSN 0010-0277. Available from DOI: 10.1016/S0010-0277(99)00081-5.

19. Budzianowski, Paweł; Wen, Tsung-Hsien; Tseng, Bo-Hsiang; Casanueva, Iñigo; Ultes, Stefan; Ramadan, Osman; Gašić, Milica. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In: Riloff, Ellen; Chiang, David; Hockenmaier, Julia; Tsujii, Jun'ichi (eds.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 5016–5026. Available from DOI: 10.18653/v1/D18-1547.

20. Çetin, Özgür; Shriberg, Elizabeth. Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition. In: *Ninth international conference on spoken language processing*. 2006. Available also from: https://www.isca-archive.org/interspeech_2006/cetin06_interspeech.pdf.

21. Chen, Derek; Chen, Howard; Yang, Yi; Lin, Alexander; Yu, Zhou. Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems. In: Toutanova, Kristina; Rumshisky, Anna; Zettlemoyer, Luke; Hakkani-Tur, Dilek; Beltagy, Iz; Bethard, Steven; Cotterell, Ryan; Chakraborty, Tanmoy; Zhou, Yichao (eds.). *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 3002–3017. Available from DOI: 10.18653/v1/2021.naacl-main.239.

22. Chen, Guoguo; Chai, Shuzhou; Wang, Guanbo; Du, Jiayu; Zhang, Wei-Qiang; Weng, Chao; Su, Dan; Povey, Daniel; Trmal, Jan; Zhang, Junbo, et al. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*. 2021.

23. Chen, Junjie; Hu, Yao; Li, Junjie; Li, Kangyue; Liu, Kun; Li, Wenpeng; Li, Xu; Li, Ziyuan; Shen, Feiyu; Tang, Xu; Wei, Manzhen; Wu, Yichen; Xie, Fenglong; Xu, Kaituo; Xie, Kun. *FireRedChat: A Pluggable, Full-Duplex Voice Interaction System with Cascaded and Semi-Cascaded Implementations*. arXiv, 2025. No. arXiv:2509.06502. Available from DOI: 10.48550/arXiv.2509.06502.

24. Chen, Kai; Gou, Yunhao; Huang, Runhui; Liu, Zhili; Tan, Daxin; Xu, Jing; Wang, Chunwei; Zhu, Yi; Zeng, Yihan; Yang, Kuo; Wang, Dingdong; Xiang, Kun; Li, Haoyuan; Bai, Haoli; Han, Jianhua; Li, Xiaohui; Jin, Weike; Xie, Nian; Zhang, Yu; Kwok, James T.; Zhao, Hengshuang; Liang, Xiaodan; Yeung, Dit-Yan; Chen, Xiao; Li, Zhenguo; Zhang, Wei; Liu, Qun; Yao, Jun; Hong, Lanqing; Hou, Lu; Xu, Hang. *EMOVA: Empowering Language Models to See, Hear and Speak with Vivid Emotions*. arXiv, 2024. No. arXiv:2409.18042. Available from DOI: 10.48550/arXiv.2409.18042.

25. Chen, Meng; Wang, Kun; Lu, Li; Zhang, Jiaheng; Zhang, Tianwei. Hijacking Large Audio-Language Models via Context-Agnostic and Imperceptible Auditory Prompt Injection. In: *2026 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2026, pp. 310–328.

26. Chen, Qian; Chen, Yafeng; Chen, Yanni; Chen, Mengzhe; Chen, Yingda; Deng, Chong; Du, Zhihao; Gao, Ruize; Gao, Changfeng; Gao, Zhifu; Li, Yabin; Lv, Xiang; Liu, Jiaqing; Luo, Haoneng; Ma, Bin; Ni, Chongjia; Shi, Xian; Tang, Jialong; Wang, Hui; Wang, Hao; Wang, Wen; Wang, Yuxuan; Xu, Yunlan; Yu, Fan; Yan, Zhijie; Yang, Yexin; Yang, Baosong; Yang, Xian; Yang, Guanrou; Zhao, Tianyu; Zhang, Qinglin; Zhang, Shiliang; Zhao, Nan; Zhang, Pei; Zhang, Chong; Zhou, Jinren. *MinMo: A Multimodal Large Language Model for Seamless Voice Interaction*. arXiv, 2025. No. arXiv:2501.06282. Available from DOI: 10.48550/arXiv.2501.06282.

27. Chen, Sanyuan; Wang, Chengyi; Chen, Zhengyang; Wu, Yu; Liu, Shujie; Chen, Zhuo; Li, Jinyu; Kanda, Naoyuki; Yoshioka, Takuya; Xiao, Xiong, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*. 2022, vol. 16, no. 6, pp. 1505–1518.

28. Chen, Yiming; Yue, Xianghu; Zhang, Chen; Gao, Xiaoxue; Tan, Robby T.; Li, Haizhou. *VoiceBench: Benchmarking LLM-Based Voice Assistants*. arXiv, 2024. No. arXiv:2410.17196. Available from DOI: 10.48550/arXiv.2410.17196.

29. Chen, Yuxuan; Yu, Haoyuan. *From Turn-Taking to Synchronous Dialogue: A Survey of Full-Duplex Spoken Language Models* [arXiv.org]. 2025-09-18. Available also from: https://arxiv.org/abs/2509.14515v1.

30. Cherakara, Neeraj; Varghese, Finny; Shabana, Sheena; Nelson, Nivan; Karukayil, Abhiram; Kulothungan, Rohith; Afil Farhan, Mohammed; Nesset, Birthe; Moujahid, Meriam; Dinkar, Tanvi; Rieser, Verena; Lemon, Oliver. FurChat: An Embodied Conversational Agent using LLMs, Combining Open and Closed-Domain Dialogue with Facial Expressions. In: Stoyanchev, Svetlana; Joty, Shafiq; Schlangen, David; Dusek, Ondrej; Kennington, Casey; Alikhani, Malihe (eds.). *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Prague, Czechia: Association for Computational Linguistics, 2023, pp. 588–592. Available from DOI: 10.18653/v1/2023.sigdial-1.55.

31. Chiang, Cheng-Han; Wang, Xiaofei; Li, Linjie; Lin, Chung-Ching; Lin, Kevin; Liu, Shujie; Wang, Zhendong; Yang, Zhengyuan; Lee, Hung-yi; Wang, Lijuan. *STITCH: Simultaneous Thinking and Talking with Chunked Reasoning for Spoken Language Models*. arXiv, 2026. No. arXiv:2507.15375. Available from doi: 10.48550/arXiv.2507.15375.

32. Chien, Chung-Ming; Orsini, Manu; Kharitonov, Eugene; Zeghidour, Neil; Livescu, Karen; Défossez, Alexandre. *MoshiRAG: Asynchronous Knowledge Retrieval for Full-Duplex Speech Language Models*. arXiv, 2026. No. arXiv:2604.12928. Available from doi: 10.48550/arXiv.2604.12928.

33. Chu, Yunfei; Xu, Jin; Yang, Qian; Wei, Haojie; Wei, Xipin; Guo, Zhifang; Leng, Yichong; Lv, Yuanjun; He, Jinzheng; Lin, Junyang. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*. 2024. Available also from: https://qwen2.org/wp-content/uploads/2024/09/2407.10759v1.pdf.

34. Cieri, Christopher; Miller, David; Walker, Kevin. The Fisher corpus: A resource for the next generations of speech-to-text. In: *LREC*. 2004, vol. 4, pp. 69–71.

35. Cobbe, Karl; Kosaraju, Vineet; Bavarian, Mohammad; Chen, Mark; Jun, Heewoo; Kaiser, Lukasz; Plappert, Matthias; Tworek, Jerry; Hilton, Jacob; Nakano, Reiichiro; Hesse, Christopher; Schulman, John. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*. 2021.

36. Coman, Andrei C.; Yoshino, Koichiro; Murase, Yukitoshi; Nakamura, Satoshi; Riccardi, Giuseppe. *An Incremental Turn-Taking Model For Task-Oriented Dialog Systems* [arXiv.org]. 2019-05-28. Available also from: https://arxiv.org/abs/1905.11806v3.

37. Comanici, Gheorghe; Bieber, Eric; Schaekermann, Mike; Pasupat, Ice; Sachdeva, Noveen; Dhillon, Inderjit; Blistein, Marcel; Ram, Ori; Zhang, Dan; Rosen, Evan, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*. 2025.

38. Cui, Wenqian; Yu, Dianzhi; Jiao, Xiaoqi; Meng, Ziqiao; Zhang, Guangyan; Wang, Qichao; Guo, Yiwen; King, Irwin. *Recent Advances in Speech Language Models: A Survey*. arXiv, 2024. No. arXiv:2410.03751. Available from doi: 10.48550/arXiv.2410.03751.

39. Cui, Wenqian; Zhu, Lei; Li, Xiaohui; Guo, Zhihan; Bai, Haoli; Hou, Lu; King, Irwin. *TurnGuide: Enhancing Meaningful Full Duplex Spoken Interactions via Dynamic Turn-Level Text-Speech Interleaving*. arXiv, 2026. No. arXiv:2508.07375. Available from doi: 10.48550/arXiv.2508.07375.

40. Défossez, Alexandre; Copet, Jade; Synnaeve, Gabriel; Adi, Yossi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*. 2022.

41. Défossez, Alexandre; Mazaré, Laurent; Orsini, Manu; Royer, Amélie; Pérez, Patrick; Jégou, Hervé; Grave, Edouard; Zeghidour, Neil. *Moshi: a speech-text foundation model for real-time dialogue*. arXiv, 2024. No. arXiv:2410.00037. Available from doi: 10.48550/arXiv.2410.00037.

42. DeVault, David; Artstein, Ron; Benn, Grace; Dey, Teresa; Fast, Ed; Gainer, Alesia; Georgila, Kallirroi; Gratch, Jon; Hartholt, Arno; Lhommet, Margaux; Lucas, Gale; Marsella, Stacy; Morbini, Fabrizio; Nazarian, Angela; Scherer, Stefan; Stratou, Giota; Suri, Apar; Traum, David; Wood, Rachel; Xu, Yuyu; Rizzo, Albert; Morency, Louis-Philippe. SimSensei kiosk: a virtual human interviewer for healthcare decision support. In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1061–1068. AAMAS '14. ISBN 978-1-4503-2738-1.

43. Ding, Ning; Chen, Yulin; Xu, Bokai; Qin, Yujia; Hu, Shengding; Liu, Zhiyuan; Sun, Maosong; Zhou, Bowen. Enhancing Chat Language Models by Scaling High-quality Instructional Conversations. In: Bouamor, Houda; Pino, Juan; Bali, Kalika (eds.). *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, 2023, pp. 3029–3051. Available from DOI: 10.18653/v1/2023.emnlp-main.183.

44. Du, Zhihao; Wang, Yuxuan; Chen, Qian; Shi, Xian; Lv, Xiang; Zhao, Tianyu; Gao, Zhifu; Yang, Yexin; Gao, Changfeng; Wang, Hui; Yu, Fan; Liu, Huadai; Sheng, Zhengyan; Gu, Yue; Deng, Chong; Wang, Wen; Zhang, Shiliang; Yan, Zhijie; Zhou, Jingren. *CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models*. arXiv, 2024. No. arXiv:2412.10117. Available from DOI: 10.48550/arXiv.2412.10117.

45. Dunbar, Ewan; Bernard, Mathieu; Hamilakis, Nicolas; Nguyen, Tu Anh; Seyssel, Maureen de; Rozé, Patricia; Rivière, Morgane; Kharitonov, Eugene; Dupoux, Emmanuel. *The Zero Resource Speech Challenge 2021: Spoken language modelling*. arXiv, 2021. No. arXiv:2104.14700. Available from DOI: 10.48550/arXiv.2104.14700.

46. Dziri, Nouha; Kamalloo, Ehsan; Milton, Sivan; Zaiane, Osmar; Yu, Mo; Ponti, Edoardo M.; Reddy, Siva. FaithDial: A Faithful Benchmark for Information-Seeking Dialogue. *Transactions of the Association for Computational Linguistics*. 2022, vol. 10, pp. 1473–1490. Available from DOI: 10.1162/tacl_a_00529.

47. Elgohary, Ahmed; Peskov, Denis; Boyd-Graber, Jordan. Can You Unpack That? Learning to Rewrite Questions-in-Context. In: *Empirical Methods in Natural Language Processing*. 2019.

48. Fang, Qingkai; Guo, Shoutao; Zhou, Yan; Ma, Zhengrui; Zhang, Shaolei; Feng, Yang. *LLaMA-Omni: Seamless Speech Interaction with Large Language Models*. arXiv, 2024. No. arXiv:2409.06666. Available from DOI: 10.48550/arXiv.2409.06666.

49. Gao, Heting; Shao, Hang; Wang, Xiong; Qiu, Chaofan; Shen, Yunhang; Cai, Siqi; Shi, Yuchen; Xu, Zihan; Long, Zuwei; Zhang, Yike; Dong, Shaoqi; Fu, Chaoyou; Li, Ke; Ma, Long; Sun, Xing. *LUCY: Linguistic Understanding and Control Yielding Early Stage of Her*. arXiv, 2025. No. arXiv:2501.16327. Available from DOI: 10.48550/arXiv.2501.16327.

50. Gao, Kuofeng; Xia, Shu-Tao; Xu, Ke; Torr, Philip; Gu, Jindong. *Benchmarking Open-ended Audio Dialogue Understanding for Large Audio-Language Models*. arXiv, 2024. No. arXiv:2412.05167. Available from doi: 10.48550/arXiv.2412.05167.

51. Gervits, Felix; Scheutz, Matthias. Pardon the Interruption: Managing Turn-Taking through Overlap Resolution in Embodied Artificial Agents. In: Komatani, Kazunori; Litman, Diane; Yu, Kai; Papangelis, Alex; Cavedon, Lawrence; Nakano, Mikio (eds.). *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 99–109. Available from doi: 10.18653/v1/W18-5011.

52. Godfrey, J.J.; Holliman, E.C.; McDaniel, J. SWITCHBOARD: telephone speech corpus for research and development. In: *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. 1992, vol. 1, 517–520 vol.1. issn 1520-6149. Available from doi: 10.1109/ICASSP.1992.225858. ISSN: 1520-6149.

53. Google. *Gemini 2 Family Expands*. 2025. Available also from: https://developers.googleblog.com/en/gemini-2-family-expands/.

54. Gunson, Nancie; Hernandez Garcia, Daniel; Sieińska, Weronika; Addlesee, Angus; Dondrup, Christian; Lemon, Oliver; Part, Jose L.; Yu, Yanchao. A Visually-Aware Conversational Robot Receptionist. In: Lemon, Oliver; Hakkani-Tur, Dilek; Li, Junyi Jessy; Ashrafzadeh, Arash; Garcia, Daniel Hernández; Alikhani, Malihe; Vandyke, David; Dušek, Ondřej (eds.). *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Edinburgh, UK: Association for Computational Linguistics, 2022, pp. 645–648. Available from doi: 10.18653/v1/2022.sigdial-1.61.

55. Hassid, Michael; Remez, Tal; Nguyen, Tu Anh; Gat, Itai; Conneau, Alexis; Kreuk, Felix; Copet, Jade; Defossez, Alexandre; Synnaeve, Gabriel; Dupoux, Emmanuel; Schwartz, Roy; Adi, Yossi. Textually Pretrained Speech Language Models. *Advances in Neural Information Processing Systems*. 2023, vol. 36, pp. 63483–63501. Available also from: https://proceedings.neurips.cc/paper_files/paper/2023/hash/c859b99b5d717c9035e79d43dfd69435-Abstract-Conference.html.

56. Hendrycks, Dan; Burns, Collin; Basart, Steven; Zou, Andy; Mazeika, Mantas; Song, Dawn; Steinhardt, Jacob. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*. 2021.

57. Hendrycks, Dan; Burns, Collin; Basart, Steven; Zou, Andy; Mazeika, Mantas; Song, Dawn; Steinhardt, Jacob. *Measuring Massive Multitask Language Understanding*. arXiv, 2021. No. arXiv:2009.03300. Available from doi: 10.48550/arXiv.2009.03300.

58. Hochreiter, Sepp; Schmidhuber, Jürgen. Long short-term memory. *Neural computation*. 1997, vol. 9, no. 8, pp. 1735–1780.

59. Hsu, Wei-Ning; Bolte, Benjamin; Tsai, Yao-Hung Hubert; Lakhotia, Kushal; Salakhutdinov, Ruslan; Mohamed, Abdelrahman. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*. 2021, vol. 29, pp. 3451–3460.

60. Hu, Edward J; Shen, Yelong; Wallis, Phillip; Allen-Zhu, Zeyuan; Li, Yuanzhi; Wang, Shean; Wang, Liang; Chen, Weizhu, et al. Lora: Low-rank adaptation of large language models. *Iclr*. 2022, vol. 1, no. 2, p. 3.

61. Hu, Ke; Hosseini-Asl, Ehsan; Chen, Chen; Casanova, Edresson; Ghosh, Subhankar; Żelasko, Piotr; Chen, Zhehuai; Li, Jason; Balam, Jagadeesh; Ginsburg, Boris. *SALM-Duplex: Efficient and Direct Duplex Modeling for Speech-to-Speech Language Model*. arXiv, 2025. No. arXiv:2505.15670. Available from DOI: 10.48550/arXiv.2505.15670.

62. Huang, Ailin et al. *Step-Audio: Unified Understanding and Generation in Intelligent Speech Interaction*. arXiv, 2025. No. arXiv:2502.11946. Available from DOI: 10.48550/arXiv.2502.11946.

63. Huang, Rongjie; Li, Mingze; Yang, Dongchao; Shi, Jiatong; Chang, Xuankai; Ye, Zhenhui; Wu, Yuning; Hong, Zhiqing; Huang, Jiawei; Liu, Jinglin; Ren, Yi; Zou, Yuexian; Zhao, Zhou; Watanabe, Shinji. AudioGPT: Understanding and Generating Speech, Music, Sound, and Talking Head. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024, vol. 38, no. 21, pp. 23802–23804. ISSN 2374-3468. Available from DOI: 10.1609/aaai.v38i21.30570. Number: 21.

64. Intelligence, Amazon Artificial General. Amazon Nova Sonic: Technical report and model card. *Amazon Technical Reports*. 2025. Available also from: https://www.amazon.science/publications/amazon-nova-sonic-technical-report-and-model-card.

65. Ji, Shengpeng; Chen, Yifu; Fang, Minghui; Zuo, Jialong; Lu, Jingyu; Wang, Hanting; Jiang, Ziyue; Zhou, Long; Liu, Shujie; Cheng, Xize; Yang, Xiaoda; Wang, Zehan; Yang, Qian; Li, Jian; Jiang, Yidi; He, Jingzhen; Chu, Yunfei; Xu, Jin; Zhao, Zhou. *WavChat: A Survey of Spoken Dialogue Models*. arXiv, 2024. No. arXiv:2411.13577. Available from DOI: 10.48550/arXiv.2411.13577.

66. Ji, Shengpeng; Jiang, Ziyue; Wang, Wen; Chen, Yifu; Fang, Minghui; Zuo, Jialong; Yang, Qian; Cheng, Xize; Li, Ruiqi; Zhang, Ziang, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In: *International Conference on Learning Representations*. 2025, vol. 2025, pp. 93809–93826.

67. Jiang, Xilin; Han, Cong; Mesgarani, Nima. *Dual-path Mamba: Short and Long-term Bidirectional Selective Structured State Space Models for Speech Separation* [arXiv.org]. 2024-03-27. Available also from: https://arxiv.org/abs/2403.18257v2.

68. Joshi, Mandar; Choi, Eunsol; Weld, Daniel; Zettlemoyer, Luke. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In: Barzilay, Regina; Kan, Min-Yen (eds.). *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1601–1611. Available from doi: 10.18653/v1/P17-1147.

69. Kahn, Jacob; Riviere, Morgane; Zheng, Weiyi; Kharitonov, Evgeny; Xu, Qiantong; Mazaré, Pierre-Emmanuel; Karadayi, Julien; Liptchinsky, Vitaliy; Collobert, Ronan; Fuegen, Christian, et al. Libri-light: A benchmark for asr with limited or no supervision. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7669–7673.

70. Kalatzis, Dimitrios; Eshghi, Arash; Lemon, Oliver. *Bootstrapping incremental dialogue systems: using linguistic knowledge to learn from minimal data*. arXiv, 2016. No. arXiv:1612.00347. Available from doi: 10.48550/arXiv.1612.00347.

71. Kennington, Casey; Lison, Pierre; Schlangen, David. *Incremental Dialogue Management: Survey, Discussion, and Implications for HRI*. arXiv, 2025. No. arXiv:2501.00953. Available from doi: 10.48550/arXiv.2501.00953.

72. Kim, Heeseung; Seo, Soonshin; Jeong, Kyeongseok; Kwon, Ohsung; Kim, Jungwhan; Lee, Jaehong; Song, Eunwoo; Oh, Myungwoo; Yoon, Sungroh; Yoo, Kang Min. Unified speech-text pretraining for spoken dialog modeling. *arXiv preprint arXiv:2402.05706*. 2024, p. 136.

73. Kong, Jungil; Kim, Jaehyeon; Bae, Jaekyoung. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*. 2020, vol. 33, pp. 17022–17033.

74. Labiausse, Tom; Mazaré, Laurent; Grave, Edouard; Pérez, Patrick; Défossez, Alexandre; Zeghidour, Neil. *High-Fidelity Simultaneous Speech-To-Speech Translation*. arXiv, 2025. No. arXiv:2502.03382. Available from doi: 10.48550/arXiv.2502.03382.

75. Lakhotia, Kushal; Kharitonov, Eugene; Hsu, Wei-Ning; Adi, Yossi; Polyak, Adam; Bolte, Benjamin; Nguyen, Tu-Anh; Copet, Jade; Baevski, Alexei; Mohamed, Abdelrahman; Dupoux, Emmanuel. On Generative Spoken Language Modeling from Raw Audio. *Transactions of the Association for Computational Linguistics*. 2021, vol. 9, pp. 1336–1354. issn 2307-387X. Available from doi: 10.1162/tacl_a_00430.

76. Lee, S; Schulz, H; Atkinson, A; Gao, J; Suleman, K; El Asri, L; Adada, M; Huang, M; Sharma, S; Tay, W, et al. Multi-domain task-completion dialog challenge. *Dialog system technology challenges*. 2019, vol. 8, no. 9.

77. Lee, Sehun; Kim, Kang-wook; Kim, Gunhee. Behavior-SD: Behaviorally Aware Spoken Dialogue Generation with Large Language Models. In: Chiruzzo, Luis; Ritter, Alan; Wang, Lu (eds.). *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: Association for Computational Linguistics, 2025, pp. 9574–9593. ISBN 979-8-89176-189-6. Available from DOI: 10.18653/v1/2025.naacl-long.484.

78. Lian, Wing; Goodson, Bleys; Pentland, Eugene; Cook, Austin; Vong, Chanvichet; "Teknium". *OpenOrca: An Open Dataset of GPT Augmented FLAN Reasoning Traces* [https://https://huggingface.co/datasets/Open-Orca/OpenOrca]. HuggingFace, 2023.

79. Liao, Borui; Xu, Yulong; Ou, Jiao; Yang, Kaiyuan; Jian, Weihua; Wan, Pengfei; Zhang, Di. *FlexDuo: A Pluggable System for Enabling Full-Duplex Capabilities in Speech Dialogue Systems*. arXiv, 2025. No. arXiv:2502.13472. Available from DOI: 10.48550/arXiv.2502.13472.

80. Liao, Shijia; Wang, Yuxuan; Liu, Songting; Cheng, Yifan; Zhang, Ruoyi; Li, Tianyu; Li, Shidong; Zheng, Yisheng; Liu, Xingwei; Wang, Qingzheng, et al. Fish audio s2 technical report. *arXiv preprint arXiv:2603.08823*. 2026.

81. Lin, Chin-Yew. Rouge: A package for automatic evaluation of summaries. In: *Text summarization branches out*. 2004, pp. 74–81.

82. Lin, Guan-Ting; Chen, Chen; Chen, Zhehuai; Lee, Hung-yi. *Full-Duplex-Bench-v3: Benchmarking Tool Use for Full-Duplex Voice Agents Under Real-World Disfluency* [arXiv.org]. 2026-04-06. Available also from: https://arxiv.org/abs/2604.04847v1.

83. Lin, Guan-Ting; Chiang, Cheng-Han; Lee, Hung-yi. *Advancing Large Language Models to Capture Varied Speaking Styles and Respond Properly in Spoken Conversations*. arXiv, 2024. No. arXiv:2402.12786. Available from DOI: 10.48550/arXiv.2402.12786.

84. Lin, Guan-Ting; Kuan, Shih-Yun Shan; Shi, Jiatong; Chang, Kai-Wei; Arora, Siddhant; Watanabe, Shinji; Lee, Hung-yi. *Full-Duplex-Bench-v2: A Multi-Turn Evaluation Framework for Duplex Dialogue Systems with an Automated Examiner* [arXiv.org]. 2025-10-09. Available also from: https://arxiv.org/abs/2510.07838v2.

85. Lin, Guan-Ting; Kuan, Shih-Yun Shan; Wang, Qirui; Lian, Jiachen; Li, Tingle; Watanabe, Shinji; Lee, Hung-yi. *Full-Duplex-Bench v1.5: Evaluating Overlap Handling for Full-Duplex Speech Models* [arXiv.org]. 2025-07-30. Available also from: https://arxiv.org/abs/2507.23159v4.

86. Lin, Guan-Ting; Lian, Jiachen; Li, Tingle; Wang, Qirui; Anumanchipalli, Gopala; Liu, Alexander H.; Lee, Hung-yi. *Full-Duplex-Bench: A Benchmark to Evaluate Full-duplex Spoken Dialogue Models on Turn-taking Capabilities*. arXiv, 2025. No. arXiv:2503.04721. Available from DOI: 10.48550/arXiv.2503.04721.

87. Lin, Guan-Ting; Shivakumar, Prashanth Gurunath; Gandhe, Ankur; Yang, Chao-Han Huck; Gu, Yile; Ghosh, Shalini; Stolcke, Andreas; Lee, Hung-Yi; Bulyko, Ivan. Paralinguistics-Enhanced Large Language Modeling of Spoken Dialogue. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024, pp. 10316–10320. ISSN 2379-190X. Available from DOI: 10.1109/ICASSP48485.2024.10446933. ISSN: 2379-190X.

88. Lin, Guan-Ting; Shivakumar, Prashanth Gurunath; Gourav, Aditya; Gu, Yile; Gandhe, Ankur; Lee, Hung-yi; Bulyko, Ivan. *Align-SLM: Textless Spoken Language Models with Reinforcement Learning from AI Feedback*. arXiv, 2024. No. arXiv:2411.01834. Available from DOI: 10.48550/arXiv.2411.01834.

89. Liu, Heyang; Wang, Yuhao; Cheng, Ziyang; Wu, Ronghua; Gu, Qunshan; Wang, Yanfeng; Wang, Yu. *VocalBench: Benchmarking the Vocal Conversational Abilities for Speech Interaction Models*. arXiv, 2025. No. arXiv:2505.15727. Available from DOI: 10.48550/arXiv.2505.15727.

90. Lu, Haitian; Cheng, Gaofeng; Luo, Liuping; Zhang, Leying; Qian, Yanmin; Zhang, Pengyuan. *SLIDE: Integrating Speech Language Model with LLM for Spontaneous Spoken Dialogue Generation*. arXiv, 2025. No. arXiv:2501.00805. Available from DOI: 10.48550/arXiv.2501.00805.

91. Lu, Hui; Chen, Xueyuan; Wang, Huimeng; Peng, Shuhai; Kang, Shiyin; Wu, Xixin; Wu, Zhiyong. *How Should LLMs Listen While Speaking? A Study of User-Stream Routing in Full-Duplex Spoken Dialogue*. arXiv, 2026. No. arXiv:2605.10199. Available from DOI: 10.48550/arXiv.2605.10199.

92. Ma, Ziyang; Song, Yakun; Du, Chenpeng; Cong, Jian; Chen, Zhuo; Wang, Yuping; Wang, Yuxuan; Chen, Xie. *Language Model Can Listen While Speaking*. arXiv, 2024. No. arXiv:2408.02622. Available from DOI: 10.48550/arXiv.2408.02622.

93. Ma, Ziyang; Zheng, Zhisheng; Ye, Jiaxin; Li, Jinchao; Gao, Zhifu; Zhang, ShiLiang; Chen, Xie. emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. In: Ku, Lun-Wei; Martins, Andre; Srikumar, Vivek (eds.). *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 15747–15760. Available from DOI: 10.18653/v1/2024.findings-acl.931.

94. Maben, Leander Melroy; Lakshmy, Gayathri Ganesh; Radhakrishnan, Srijith; Arora, Siddhant; Watanabe, Shinji. *AURA: Agent for Understanding, Reasoning, and Automated Tool Use in Voice-Driven Tasks*. arXiv, 2025. No. arXiv:2506.23049. Available from DOI: 10.48550/arXiv.2506.23049.

95. Maeng, Kiwan; Colin, Alexei; Lucia, Brandon. Alpaca: Intermittent execution without checkpoints. *Proceedings of the ACM on Programming Languages*. 2017, vol. 1, no. OOPSLA, pp. 1–30.

96. Maier, Angelika; Hough, Julian; Schlangen, David. Towards Deep End-of-Turn Prediction for Situated Spoken Dialogue Systems. In: 2017, pp. 1676–1680. Available from DOI: 10.21437/Interspeech.2017-1593.

97. Maimon, Gallil; Elmakies, Avishai; Adi, Yossi. *Slamming: Training a Speech Language Model on One GPU in a Day*. arXiv, 2025. No. arXiv:2502.15814. Available from doi: 10.48550/arXiv.2502.15814.

98. Maimon, Gallil; Roth, Amit; Adi, Yossi. Salmon: A suite for acoustic language model evaluation. In: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

99. Meng, Ziqiao; Wang, Qichao; Cui, Wenqian; Zhang, Yifei; Wu, Bingzhe; King, Irwin; Chen, Liang; Zhao, Peilin. Parrot: Autoregressive Spoken Dialogue Language Modeling with Decoder-only Transformers. In: 2024. Available also from: https://openreview.net/forum?id=Ttndg2Jl5F.

100. Meta AI. *Introducing Llama 3.1: Our Most Capable Models to Date*. 2024-07. Available also from: https://ai.meta.com/blog/meta-llama-3-1/.

101. Mitsui, Kentaro; Mitsuda, Koh; Wakatsuki, Toshiaki; Hono, Yukiya; Sawada, Kei. Pslm: Parallel generation of text and speech with llms for low-latency spoken dialogue systems. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024, pp. 2692–2700.

102. Mizumoto, Tomoya; Kojima, Atsushi; Fujita, Yusuke; Liu, Lianbo; Sudo, Yui. Is Synthetic Data Truly Effective for Training Speech Language Models? In: *Proc. Interspeech 2025*. 2025, pp. 1808–1812. Available also from: https://www.isca-archive.org/interspeech_2025/mizumoto25_interspeech.pdf.

103. Nachmani, Eliya; Levkovitch, Alon; Hirsch, Roy; Salazar, Julian; Asawaroengchai, Chulayuth; Mariooryad, Soroosh; Rivlin, Ehud; Skerry-Ryan, R. J.; Tadmor Ramanovich, Michele. Spoken Question Answering and Speech Continuation Using Spectrogram-Powered LLM. *International Conference on Learning Representations*. 2024, vol. 2024, pp. 51883–51898. Available also from: https://proceedings.iclr.cc/paper_files/paper/2024/hash/e393677793767624f2821cec8bdd02f1-Abstract-Conference.html.

104. Nguyen, Tu Anh; Kharitonov, Eugene; Copet, Jade; Adi, Yossi; Hsu, Wei-Ning; Elkahky, Ali; Tomasello, Paden; Algayres, Robin; Sagot, Benoît; Mohamed, Abdelrahman; Dupoux, Emmanuel. Generative Spoken Dialogue Language Modeling. *Transactions of the Association for Computational Linguistics*. 2023, vol. 11, pp. 250–266. issn 2307-387X. Available from doi: 10.1162/tacl_a_00545.

105. Nguyen, Tu Anh; Muller, Benjamin; Yu, Bokai; Costa-jussa, Marta R.; Elbayad, Maha; Popuri, Sravya; Ropers, Christophe; Duquenne, Paul-Ambroise; Algayres, Robin; Mavlyutov, Ruslan; Gat, Itai; Williamson, Mary; Synnaeve, Gabriel; Pino, Juan; Sagot, Benoit; Dupoux, Emmanuel. *Spirit LM: Interleaved Spoken and Written Language Model*. arXiv, 2024. No. arXiv:2402.05755. Available from doi: 10.48550/arXiv.2402.05755.

106. Novikova, Jekaterina; Dušek, Ondřej; Curry, Amanda Cercas; Rieser, Verena. Why We Need New Evaluation Metrics for NLG. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 2231–2240. Available from DOI: 10.18653/v1/D17-1237.

107. Ohashi, Atsumoto; Iizuka, Shinya; Jiang, Jingjing; Higashinaka, Ryuichiro. *Towards a Japanese Full-duplex Spoken Dialogue System*. arXiv, 2025. No. arXiv:2506.02979. Available from DOI: 10.48550/arXiv.2506.02979.

108. Oord, Aaron van den; Li, Yazhe; Vinyals, Oriol. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*. 2018.

109. OpenAI. *GPT-4.1*. 2025. Large language model.

110. OpenAI. *gpt-realtime-2.1 Model | OpenAI API*. [N.d.]. Available also from: https://developers.openai.com/api/docs/models/gpt-realtime-2.1.

111. OpenAI et al. *GPT-4o System Card*. 2024. Available from arXiv: 2410.21276 [cs.CL].

112. Panayotov, Vassil; Chen, Guoguo; Povey, Daniel; Khudanpur, Sanjeev. Librispeech: an asr corpus based on public domain audio books. In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

113. Papineni, Kishore; Roukos, Salim; Ward, Todd; Zhu, Wei-Jing. Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

114. Park, Se; Kim, Chae; Rha, Hyeongseop; Kim, Minsu; Hong, Joanna; Yeo, Jeonghun; Ro, Yong. Let's Go Real Talk: Spoken Dialogue Model for Face-to-Face Conversation. In: Ku, Lun-Wei; Martins, Andre; Srikumar, Vivek (eds.). *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 16334–16348. Available from DOI: 10.18653/v1/2024.acl-long.860.

115. Park, Se Jin; Salazar, Julian; Jansen, Aren; Kinoshita, Keisuke; Ro, Yong Man; Skerry-Ryan, R. J. *Long-Form Speech Generation with Spoken Language Models*. arXiv, 2024. No. arXiv:2412.18603. Available from DOI: 10.48550/arXiv.2412.18603.

116. Peng, Jing; Wang, Yucheng; Xi, Yu; Li, Xu; Zhang, Xizhuo; Yu, Kai. *A Survey on Speech Large Language Models*. arXiv, 2024. No. arXiv:2410.18908. Available from DOI: 10.48550/arXiv.2410.18908.

117. Pratap, Vineel; Xu, Qiantong; Sriram, Anuroop; Synnaeve, Gabriel; Collobert, Ronan. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*. 2020.

118. Radford, Alec; Kim, Jong Wook; Xu, Tao; Brockman, Greg; McLeavey, Christine; Sutskever, Ilya. *Robust Speech Recognition via Large-Scale Weak Supervision*. arXiv, 2022. No. arXiv:2212.04356. Available from DOI: 10.48550/arXiv.2212.04356.

119. Rafailov, Rafael; Sharma, Archit; Mitchell, Eric; Manning, Christopher D; Ermon, Stefano; Finn, Chelsea. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*. 2023, vol. 36, pp. 53728–53741.

120. Reddy, Chandan KA; Gopal, Vishak; Cutler, Ross; Beyrami, Ebrahim; Cheng, Roger; Dubey, Harishchandra; Matusevych, Sergiy; Aichner, Robert; Aazami, Ashkan; Braun, Sebastian, et al. The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. *arXiv preprint arXiv:2005.13981*. 2020.

121. Reece, Andrew; Cooney, Gus; Bull, Peter; Chung, Christine; Dawson, Bryn; Fitzpatrick, Casey; Glazer, Tamara; Knox, Dean; Liebscher, Alex; Marin, Sebastian. The CANDOR corpus: Insights from a large multimodal dataset of naturalistic conversation. *Science Advances*. 2023, vol. 9, no. 13, eadf3197. Available from DOI: 10.1126/sciadv.adf3197.

122. Resemble AI. *Chatterbox-TTS* [https://github.com/resemble-ai/chatterbox]. 2025. GitHub repository.

123. Roddy, Matthew; Skantze, Gabriel; Harte, Naomi. *Investigating Speech Features for Continuous Turn-Taking Prediction Using LSTMs*. arXiv, 2018. No. arXiv:1806.11461. Available from DOI: 10.48550/arXiv.1806.11461.

124. Roy, Rajarshi; Raiman, Jonathan; Lee, Sang-gil; Ene, Teodor-Dumitru; Kirby, Robert; Kim, Sungwon; Kim, Jaehyeon; Catanzaro, Bryan. *PersonaPlex: Voice and Role Control for Full Duplex Conversational Speech Models*. arXiv, 2026. No. arXiv:2602.06053. Available from DOI: 10.48550/arXiv.2602.06053.

125. Saeki, Takaaki; Xin, Detai; Nakata, Wataru; Koriyama, Tomoki; Takamichi, Shinnosuke; Saruwatari, Hiroshi. *UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022*. arXiv, 2022. No. arXiv:2204.02152. Available from DOI: 10.48550/arXiv.2204.02152.

126. Sakshi, Sakshi; Tyagi, Utkarsh; Kumar, Sonal; Seth, Ashish; Selvakumar, Ramaneswaran; Nieto, Oriol; Duraiswami, Ramani; Ghosh, Sreyan; Manocha, Dinesh. MMAU: A Massive Multi-Task Audio Understanding and Reasoning Benchmark. *International Conference on Learning Representations*. 2025, vol. 2025, pp. 84929–84964. Available also from: https://proceedings.iclr.cc/paper_files/paper/2025/hash/d36f208919582785db965fe648b9fe59-Abstract-Conference.html.

127. Schalkwyk, Johan; Kumar, Ankit; Lyth, Dan; Eskimez, Sefik; Hodari, Zack; Resnick, Cinjon; Sanabria, Ramon; Jiang, Raven. *Crossing the uncanny valley of conversational voice | Sesame*. Available also from: https://www.sesame.com/blog/crossing-the-uncanny-valley-of-voice.

128. Schanke, Scott; Burtch, Gordon; Ray, Gautam. Digital lyrebirds: Experimental evidence that voice-based deep fakes influence trust. *Management Science*. 2026, vol. 72, no. 1, pp. 386–405.

129. Schick, Timo; Dwivedi-Yu, Jane; Dessì, Roberto; Raileanu, Roberta; Lomeli, Maria; Zettlemoyer, Luke; Cancedda, Nicola; Scialom, Thomas. *Toolformer: Language Models Can Teach Themselves to Use Tools* [arXiv.org]. 2023-02-09. Available also from: https://arxiv.org/abs/2302.04761v1.

130. Sellam, Thibault; Das, Dipanjan; Parikh, Ankur. BLEURT: Learning robust metrics for text generation. In: *Proceedings of the 58th annual meeting of the association for computational linguistics*. 2020, pp. 7881–7892.

131. Seyssel, Maureen de; Dhekane, Eeshan Gunesh. *Which Evaluation for Which Model? A Taxonomy for Speech Model Assessment*. arXiv, 2025. No. arXiv:2510.19509. Available from doi: 10.48550/arXiv.2510.19509.

132. Si, Shuzheng; Ma, Wentao; Gao, Haoyu; Wu, Yuchuan; Lin, Ting-En; Dai, Yinpei; Li, Hangyu; Yan, Rui; Huang, Fei; Li, Yongbin. Spokenwoz: A large-scale speech-text benchmark for spoken task-oriented dialogue agents. *Advances in Neural Information Processing Systems*. 2023, vol. 36, pp. 39088–39118.

133. Siuzdak, Hubert; Grötschla, Florian; Lanzendörfer, Luca A. Snac: Multi-scale neural audio codec. *arXiv preprint arXiv:2410.14411*. 2024.

134. Skantze, Gabriel. Turn-taking in Conversational Systems and Human-Robot Interaction: A Review. *Computer Speech & Language*. 2021, vol. 67, p. 101178. issn 0885-2308. Available from doi: 10.1016/j.csl.2020.101178.

135. Skantze, Gabriel; Hjalmarsson, Anna. Towards Incremental Speech Generation in Dialogue Systems. In: 2010, pp. 1–8.

136. Snyder, David; Chen, Guoguo; Povey, Daniel. *MUSAN: A Music, Speech, and Noise Corpus* [arXiv.org]. 2015-10-28. Available also from: https://arxiv.org/abs/1510.08484v1.

137. Song, Zirui; Jiang, Qian; Cui, Mingxuan; Li, Mingzhe; Gao, Lang; Zhang, Zeyu; Xu, Zixiang; Wang, Yanbo; Ouyang, Guangxian; Chen, Zhenhao, et al. Audio jailbreak: An open comprehensive benchmark for jailbreaking large audio-language models. In: *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2026, pp. 27294–27308.

138. Stephens, Jane; Beattie, Geoffrey. On Judging the Ends of Speaker Turns in Conversation. *Journal of Language and Social Psychology*. 1986, vol. 5, no. 2, pp. 119–134. issn 0261-927X. Available from doi: 10.1177/0261927X8652003.

139. Stivers, Tanya; Enfield, N. J.; Brown, Penelope; Englert, Christina; Hayashi, Makoto; Heinemann, Trine; Hoymann, Gertie; Rossano, Federico; Ruiter, Jan Peter de; Yoon, Kyung-Eun; Levinson, Stephen C. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*. 2009, vol. 106, no. 26, pp. 10587–10592. Available from doi: 10.1073/pnas.0903616106.

140. Subakan, Cem; Ravanelli, Mirco; Cornell, Samuele; Bronzi, Mirko; Zhong, Jianyuan. Attention Is All You Need In Speech Separation. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 21–25. ISSN 2379-190X. Available from DOI: 10.1109/ICASSP39728.2021.9413901. ISSN: 2379-190X.

141. Suglia, Alessandro; Hemanthage, Bhathiya; Nikandrou, Malvina; Pantazopoulos, Georgios; Parekh, Amit; Eshghi, Arash; Greco, Claudio; Konstas, Ioannis; Lemon, Oliver; Rieser, Verena. Demonstrating EMMA: Embodied MultiModal Agent for Language-guided Action Execution in 3D Simulated Environments. In: Lemon, Oliver; Hakkani-Tur, Dilek; Li, Junyi Jessy; Ashrafzadeh, Arash; Garcia, Daniel Hernández; Alikhani, Malihe; Vandyke, David; Dušek, Ondřej (eds.). *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Edinburgh, UK: Association for Computational Linguistics, 2022, pp. 649–653. Available from DOI: 10.18653/v1/2022.sigdial-1.62.

142. SYSTRAN. *faster-whisper*. GitHub, 2023. Available also from: https://github.com/SYSTRAN/faster-whisper.

143. Tang, Changli; Yu, Wenyi; Sun, Guangzhi; Chen, Xianzhao; Tan, Tian; Li, Wei; Lu, Lu; Ma, Zejun; Zhang, Chao. *SALMONN: Towards Generic Hearing Abilities for Large Language Models*. arXiv, 2024. No. arXiv:2310.13289. Available from DOI: 10.48550/arXiv.2310.13289.

144. Team, Silero. *Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier* [https://github.com/snakers4/silero-vad]. GitHub, 2024.

145. Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017. Available also from: https://user.phil.hhu.de/~cwurm/wp-content/uploads/2020/01/7181-attention-is-all-you-need.pdf.

146. Veluri, Bandhav; Peloquin, Benjamin N.; Yu, Bokai; Gong, Hongyu; Gollakota, Shyamnath. *Beyond Turn-Based Interfaces: Synchronous LLMs as Full-Duplex Dialogue Agents*. arXiv, 2024. No. arXiv:2409.15594. Available from DOI: 10.48550/arXiv.2409.15594.

147. Verma, Prateek. *Whisper-GPT: A Hybrid Representation Audio Large Language Model*. arXiv, 2024. No. arXiv:2412.11449. Available from DOI: 10.48550/arXiv.2412.11449.

148. Wang, Bin; Zou, Xunlong; Lin, Geyu; Sun, Shuo; Liu, Zhuohan; Zhang, Wenyu; Liu, Zhengyuan; Aw, AiTi; Chen, Nancy F. AudioBench: A Universal Benchmark for Audio Large Language Models. In: Chiruzzo, Luis; Ritter, Alan; Wang, Lu (eds.). *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: Association for Computational Linguistics, 2025, pp. 4297–4316. ISBN 979-8-89176-189-6. Available from DOI: 10.18653/v1/2025.naacl-long.218.

149. Wang, Changhan; Riviere, Morgane; Lee, Ann; Wu, Anne; Talnikar, Chaitanya; Haziza, Daniel; Williamson, Mary; Pino, Juan; Dupoux, Emmanuel. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In: *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)*. 2021, pp. 993–1003.

150. Wang, Hankun; Wang, Haoran; Guo, Yiwei; Li, Zhihan; Du, Chenpeng; Chen, Xie; Yu, Kai. *Why Do Speech Language Models Fail to Generate Semantically Coherent Outputs? A Modality Evolving Perspective*. arXiv, 2024. No. arXiv:2412.17048. Available from DOI: 10.48550/arXiv.2412.17048.

151. Wang, Ke; Ren, Houxing; Lu, Zimu; Zhan, Mingjie; Li, Hongsheng. *VoiceAssistant-Eval: Benchmarking AI Assistants across Listening, Speaking, and Viewing*. arXiv, 2025. No. arXiv:2509.22651. Available from DOI: 10.48550/arXiv.2509.22651.

152. Wang, Peng; Lu, Songshuo; Tang, Yaohua; Yan, Sijie; Xiong, Yuanjun; Xia, Wei. *A Full-duplex Speech Dialogue Scheme Based On Large Language Models* [arXiv.org]. 2024-05-29. Available also from: https://arxiv.org/abs/2405.19487v1.

153. Wang, Qichao; Meng, Ziqiao; Cui, Wenqian; Zhang, Yifei; Wu, Pengcheng; Wu, Bingzhe; King, Irwin; Chen, Liang; Zhao, Peilin. *NTPP: Generative Speech Language Modeling for Dual-Channel Spoken Dialogue via Next-Token-Pair Prediction*. arXiv, 2025. No. arXiv:2506.00975. Available from DOI: 10.48550/arXiv.2506.00975.

154. Wang, Xiong; Li, Yangze; Fu, Chaoyou; Shen, Yunhang; Xie, Lei; Li, Ke; Sun, Xing; Ma, Long. *Freeze-Omni: A Smart and Low Latency Speech-to-speech Dialogue Model with Frozen LLM*. arXiv, 2024. No. arXiv:2411.00774. Available from DOI: 10.48550/arXiv.2411.00774.

155. Wenger, Emily; Bronckers, Max; Cianfarani, Christian; Cryan, Jenna; Sha, Angela; Zheng, Haitao; Zhao, Ben Y. " Hello, It's Me": Deep Learning-based Speech Synthesis Attacks in the Real World. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, pp. 235–251.

156. Wu, Anne; Mazaré, Laurent; Zeghidour, Neil; Défossez, Alexandre. *Aligning Spoken Dialogue Models from User Interactions*. arXiv, 2025. No. arXiv:2506.21463. Available from DOI: 10.48550/arXiv.2506.21463.

157. Wu, Donghang; Zhang, Haoyang; Chen, Chen; Zhang, Tianyu; Tian, Fei; Yang, Xuerui; Yu, Gang; Liu, Hexin; Hou, Nana; Hu, Yuchen; Chng, Eng Siong. *Chronological Thinking in Full-Duplex Spoken Dialogue Language Models*. arXiv, 2025. No. arXiv:2510.05150. Available from DOI: 10.48550/arXiv.2510.05150.

158. Xie, Zhifei; Wu, Changqiao. *Mini-Omni: Language Models Can Hear, Talk While Thinking in Streaming* [arXiv.org]. 2024-08-29. Available also from: https://arxiv.org/abs/2408.16725v3.

159. Xie, Zhifei; Wu, Changqiao. *Mini-Omni2: Towards Open-source GPT-4o with Vision, Speech and Duplex Capabilities*. arXiv, 2024. No. arXiv:2410.11190. Available from DOI: 10.48550/arXiv.2410.11190.

160. Xu, Jin; Guo, Zhifang; He, Jinzheng; Hu, Hangrui; He, Ting; Bai, Shuai; Chen, Keqin; Wang, Jialin; Fan, Yang; Dang, Kai; Zhang, Bin; Wang, Xiong; Chu, Yunfei; Lin, Junyang. *Qwen2.5-Omni Technical Report*. arXiv, 2025. No. arXiv:2503.20215. Available from DOI: 10.48550/arXiv.2503.20215.

161. Xue, Hongfei; Liang, Yuhao; Mu, Bingshen; Zhang, Shiliang; Chen, Mengzhe; Chen, Qian; Xie, Lei. E-Chat: Emotion-Sensitive Spoken Dialogue System with Large Language Models. In: *2024 IEEE 14th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. 2024, pp. 586–590. Available from DOI: 10.1109/ISCSLP63861.2024.10800447.

162. Yamazaki, Takato; Mizumoto, Tomoya; Yoshikawa, Katsumasa; Ohagi, Masaya; Kawamoto, Toshiki; Sato, Toshinori. An Open-Domain Avatar Chatbot by Exploiting a Large Language Model. In: Stoyanchev, Svetlana; Joty, Shafiq; Schlangen, David; Dusek, Ondrej; Kennington, Casey; Alikhani, Malihe (eds.). *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Prague, Czechia: Association for Computational Linguistics, 2023, pp. 428–432. Available from DOI: 10.18653/v1/2023.sigdial-1.40.

163. Yan, Ruiqi; Li, Xiquan; Chen, Wenxi; Niu, Zhikang; Yang, Chen; Ma, Ziyang; Yu, Kai; Chen, Xie. *URO-Bench: A Comprehensive Benchmark for End-to-End Spoken Dialogue Models*. arXiv, 2025. No. arXiv:2502.17810. Available from DOI: 10.48550/arXiv.2502.17810.

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

165. Yang, Jianing; Fujita, Yusuke; Sudo, Yui. *DuplexCascade: Full-Duplex Speech-to-Speech Dialogue with VAD-Free Cascaded ASR-LLM-TTS Pipeline and Micro-Turn Optimization*. arXiv, 2026. No. arXiv:2603.09180. Available from DOI: 10.48550/arXiv.2603.09180.

166. Yang, Qian; Xu, Jin; Liu, Wenrui; Chu, Yunfei; Jiang, Ziyue; Zhou, Xiaohuan; Leng, Yichong; Lv, Yuanjun; Zhao, Zhou; Zhou, Chang; Zhou, Jingren. AIR-Bench: Benchmarking Large Audio-Language Models via Generative Comprehension. In: Ku, Lun-Wei; Martins, Andre; Srikumar, Vivek (eds.). *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 1979–1998. Available from DOI: 10.18653/v1/2024.acl-long.109.

167. Yang, Shu-wen; Chi, Po-Han; Chuang, Yung-Sung; Lai, Cheng-I Jeff; Lakhotia, Kushal; Lin, Yist Y; Liu, Andy T; Shi, Jiatong; Chang, Xuankai; Lin, Guan-Ting, et al. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*. 2021.

168. Yao, Shunyu; Zhao, Jeffrey; Yu, Dian; Du, Nan; Shafran, Izhak; Narasimhan, Karthik; Cao, Yuan. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*. 2022.

169. Yi, Sanghyun; Goel, Rahul; Khatri, Chandra; Cervone, Alessandra; Chung, Tagyoung; Hedayatnia, Behnam; Venkatesh, Anu; Gabriel, Raefer; Hakkani-Tur, Dilek. Towards Coherent and Engaging Spoken Dialog Response Generation Using Automatic Conversation Evaluators. In: Deemter, Kees van; Lin, Chenghua; Takamura, Hiroya (eds.). *Proceedings of the 12th International Conference on Natural Language Generation*. Tokyo, Japan: Association for Computational Linguistics, 2019, pp. 65–75. Available from DOI: 10.18653/v1/W19-8608.

170. Young, Steve; Gašić, Milica; Thomson, Blaise; Williams, Jason D. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*. 2013, vol. 101, no. 5, pp. 1160–1179.

171. Young, Tom; Xing, Frank; Pandelea, Vlad; Ni, Jinjie; Cambria, Erik. Fusing Task-Oriented and Open-Domain Dialogues in Conversational Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022, vol. 36, no. 10, pp. 11622–11629. ISSN 2374-3468. Available from DOI: 10.1609/aaai.v36i10.21416. Number: 10.

172. Yu, Wenyi; Wang, Siyin; Yang, Xiaoyu; Chen, Xianzhao; Tian, Xiaohai; Zhang, Jun; Sun, Guangzhi; Lu, Lu; Wang, Yuxuan; Zhang, Chao. *SALMONN-omni: A Codec-free LLM for Full-duplex Speech Understanding and Generation*. arXiv, 2024. No. arXiv:2411.18138. Available from DOI: 10.48550/arXiv.2411.18138.

173. Zen, Heiga; Dang, Viet; Clark, Rob; Zhang, Yu; Weiss, Ron J.; Jia, Ye; Chen, Zhifeng; Wu, Yonghui. *LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech*. arXiv, 2019. No. arXiv:1904.02882. Available from DOI: 10.48550/arXiv.1904.02882.

174. Zeng, Aohan; Du, Zhengxiao; Liu, Mingdao; Wang, Kedong; Jiang, Shengmin; Zhao, Lei; Dong, Yuxiao; Tang, Jie. *GLM-4-Voice: Towards Intelligent and Human-Like End-to-End Spoken Chatbot*. arXiv, 2024. No. arXiv:2412.02612. Available from DOI: 10.48550/arXiv.2412.02612.

175. Zhang, Dong; Li, Shimin; Zhang, Xin; Zhan, Jun; Wang, Pengyu; Zhou, Yaqian; Qiu, Xipeng. *SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities*. arXiv, 2023. No. arXiv:2305.11000. Available from DOI: 10.48550/arXiv.2305.11000.

176. Zhang, Hao; Li, Weiwei; Chen, Rilin; Kothapally, Vinay; Yu, Meng; Yu, Dong. *LLM-Enhanced Dialogue Management for Full-Duplex Spoken Dialogue Systems*. arXiv, 2025. No. arXiv:2502.14145. Available from DOI: 10.48550/arXiv.2502.14145.

177. Zhang, He; Cui, Wenqian; Xu, Haoning; Li, Xiaohui; Zhu, Lei; Bai, Haoli; Ma, Shaohua; King, Irwin. *MTR-DuplexBench: Towards a Comprehensive Evaluation of Multi-Round Conversations for Full-Duplex Speech Language Models*. arXiv, 2026. No. arXiv:2511.10262. Available from DOI: 10.48550/arXiv.2511.10262.

178. Zhang, Qinglin; Cheng, Luyao; Deng, Chong; Chen, Qian; Wang, Wen; Zheng, Siqi; Liu, Jiaqing; Yu, Hai; Tan, Chaohong; Du, Zhihao; Zhang, Shiliang. *OmniFlatten: An End-to-end GPT Model for Seamless Voice Conversation*. arXiv, 2025. No. arXiv:2410.17799. Available from doi: 10.48550/arXiv.2410.17799.

179. Zhang, Xin; Lyu, Xiang; Du, Zhihao; Chen, Qian; Zhang, Dong; Hu, Hangrui; Tan, Chaohong; Zhao, Tianyu; Wang, Yuxuan; Zhang, Bin; Lu, Heng; Zhou, Yaqian; Qiu, Xipeng. *IntrinsicVoice: Empowering LLMs with Intrinsic Real-time Voice Interaction Abilities*. arXiv, 2024. No. arXiv:2410.08035. Available from doi: 10.48550/arXiv.2410.08035.

180. Zhang, Xin; Zhang, Dong; Li, Shimin; Zhou, Yaqian; Qiu, Xipeng. Speechtokenizer: Unified speech tokenizer for speech language models. In: *International Conference on Learning Representations*. 2024, vol. 2024, pp. 31798–31818.

181. Zheng, Lianmin; Chiang, Wei-Lin; Sheng, Ying; Zhuang, Siyuan; Wu, Zhanghao; Zhuang, Yonghao; Lin, Zi; Li, Zhuohan; Li, Dacheng; Xing, Eric, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*. 2023, vol. 36, pp. 46595–46623.

182. Zhou, Jiawei; Eisner, Jason; Newman, Michael; Platanios, Emmanouil Antonios; Thomson, Sam. Online Semantic Parsing for Latency Reduction in Task-Oriented Dialogue. In: Muresan, Smaranda; Nakov, Preslav; Villavicencio, Aline (eds.). *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 1554–1576. Available from doi: 10.18653/v1/2022.acl-long.110.

183. Zilka, Lukas; Jurcicek, Filip. *Incremental LSTM-based Dialog State Tracker*. arXiv, 2015. No. arXiv:1507.03471. Available from doi: 10.48550/arXiv.1507.03471.

184. Züfle, Maike; Klejch, Ondrej; Sanders, Nicholas; Niehues, Jan; Birch, Alexandra; Lam, Tsz Kin. *F-Actor: Controllable Conversational Behaviour in Full-Duplex Models*. arXiv, 2026. No. arXiv:2601.11329. Available from doi: 10.48550/arXiv.2601.11329.

# A Technical Documentation

This appendix provides an implementation-oriented overview of the Warmblood system and includes selected prompt artifacts in code-style form. Section A.1 outlines the project structure, Section A.2 summarizes the end-to-end workflow, and Section A.3 introduces the included prompt families; Sections A.3.1–A.3.5 then provide the individual prompt artifacts.

## A.1 System Overview

The project codebase consists of five main parts:

- **Synthetic data pipeline** (`synthetic_data/`): converts source dialogue data, generates guide-conditioned text, synthesizes or converts audio, and assembles final training/evaluation samples.
- **Runtime inference stack** (`warmblood/moshi/`): modified Moshi server with real-time full-duplex inference, guide generation, ASR integration, and benchmark adapters.
- **Finetuning stack** (`warmblood/moshi-finetune/`): guide-aware data handling, chain training, and heavy internal evaluation.
- **Web clients** (`warmblood/client/`, `warmblood/client_survey/`): baseline interactive client and survey-specific client variant.
- **External benchmark integrations** (`warmblood/evaluation/`): Full-Duplex-Bench and VoiceAssistant-Eval setup and execution.

## A.2 Operational Flow

The practical end-to-end workflow is:

1. Prepare or generate data in `synthetic_data/`.
2. Train or finetune in `warmblood/moshi-finetune/`.
3. Run internal heavy evaluations of checkpoints (`heavy_eval` / evaluation-only flow).
4. Run real-time inference in `warmblood/moshi/`.
5. Interact through `warmblood/client/` or `warmblood/client_survey/`.
6. Run external benchmark evaluations in `warmblood/evaluation/`.

## A.3 Prompt Artifacts Included in this Appendix

The following prompt families are embedded directly in wrapped code-style boxes:

- Guide-generation prompts used during synthetic data preparation.
- Legacy prompts used in the older UltraChat/TriviaQA preparation path.
- Internal LLM-as-judge metric prompts for correctness, dialogue naturalness, and guide-following.
- Main runtime inference guide prompt.
- Survey instruction-generation prompts and topic lists.

For the runtime inference category, this appendix includes `inference_main.md` in full. Other test prompts are available in `warmblood/moshi/moshi/prompts/` and are referenced in the main text.

### A.3.1 Guide Generation Prompts (Synthetic Data)

**`synthetic_data/guide_generation/prompts/fused_chat_batch.md`**

Purpose: generates per-turn guide annotations for FusedChat dialogues while distinguishing chitchat turns from task-oriented turns. Used in text: synthetic guide generation in Section 3.1, primarily for category *TOD* data discussed in Section 3.1.3.

```
You generate short keyword guides for an AI assistant's responses in
task-oriented dialogues that also contain casual chitchat.

You will receive a full dialogue with numbered turns and a dialogue type:
- "prepended": chitchat turns appear at the beginning, task-oriented turns
follow.
- "appended": task-oriented turns come first, chitchat turns are at the end.

Your task:
1. Identify which agent turns are chitchat and which are task-oriented.
2. For chitchat agent turns, set the guide to null.
3. For task-oriented agent turns, produce a short keyword guide.

The idea is that a model will be trained on this data to learn to generate
the responses while using the guide. During inference, the guide will be
provided and the model should use it to generate the response. Therefore,
the guide should be constructed so that the model should be able to
produce the original agent response if it receives the guide.

Guide rules:
- Use only keywords and key phrases, separated by commas. Do not write
full sentences.
```

```
- The guide must ONLY contain information that appears in the agent's
own message. NEVER include details from earlier turns or the user's
message that the agent does not repeat or mention.
- Include specific details present in the agent's message: names, addresses,
phone numbers, postcodes, reference numbers, prices, star ratings, dates,
times, car types, counts.
- Accurately describe what the agent is doing -- distinguish between
confirming a completed action (e.g. "booking successful") vs. offering to
perform one (e.g. "offer to book"). Use the verb that matches the agent's
actual phrasing.
- When the agent asks for missing information, lead with "ask" (e.g. "ask
check-in day", "ask number of people").
- When the agent simply provides requested facts, just list the facts without
adding category labels.
- Be maximally concise: omit context that is obvious from the user's
preceding question. For example, if the user asks "What area is the hotel
located in?" and the agent replies "The hotel is in the south area", the
guide should be just "south" -- not "hotel location south".
- Do not include filler words, politeness phrases, or category/topic labels
that the agent does not say.

Output a list of guides, one per agent turn only, with "turn" (the 0-based
turn index from the input) and "guide" (string or null for chitchat).
```

**synthetic_data/guide_generation/prompts/ultrachat_late_guide.md**

Purpose: rewrites Q&A answers to simulate delayed guide arrival by adding a short natural preface before the factual part. Used in text: delayed-guide data construction in Section 3.1.3 (*Ext-UC-Delay*) and analyzed in Section 4.1.3.

```
You are rewriting answers for a spoken dialogue model that receives external
factual guidance (the "guide") with a slight delay. Your task is to prepend
a short natural opening to the answer -- simulating how the model would
begin responding before the guide arrives.

The model is like a knowledgeable conversationalist: it understands topics
broadly, can engage naturally with any subject, and can start talking about
it -- but it doesn't have the specific facts, numbers, names, or details
until the guide arrives.

Your job:
- Add 1--2 short spoken sentences BEFORE the original answer.
- These sentences should naturally lead into the topic of the answer. They
can acknowledge the question, reframe it, or begin discussing the general
area -- but must NOT reveal any of the specific factual content that comes
from the guide.
- The opening should feel like a real person starting to speak: varied,
natural, context-dependent. Avoid formulaic or repetitive patterns.
- After your opening sentences, include the original answer verbatim and
unchanged.
```

```
Use the "Guide" field to understand which parts of the answer contain
external knowledge that the model wouldn't have on its own. Everything in
the guide represents information the model needs to receive externally --
your opening must not leak any of it.

Output ONLY the full modified answer (your opening + the original
answer). Do not include labels, explanations, or formatting.
```

**synthetic_data/guide_generation/prompts/ultrachat_no_guide.md**

Purpose: rewrites Q&A answers for the no-guide setting so responses stay conversational but avoid unsupported factual claims. Used in text: no-guide extension data in Section 3.1.3 (*Ext-UC-Miss*) and additional-dataset analysis in Section 4.1.4.

```
You are rewriting answers for a spoken dialogue model that never receives
its external factual guidance (the "guide"). The model must respond
entirely on its own.

The model can talk about any topic but does NOT know any of the specific
information listed in the "Guide" field. Treat EVERYTHING in the guide as
information the model simply does not have - no exceptions. Do not include
any of it in your output, not even paraphrased or hinted at.

Your job:
- Start with 1-2 short spoken sentences that naturally lead into the topic -
acknowledging the question, reframing it, or beginning to discuss the general
area.
- Then, admit that the model doesn't know the answer or doesn't
remember the details. It can talk around the topic in general terms but
must not provide any factual claims from the guide or the original answer.
- The response should sound natural and conversational - like a real person
who simply doesn't know. Not robotic or overly apologetic.
- The total response should be shorter - similar length to the original
answer. Do not copy the content from it, though.

Use the "Answer" field only to understand the general topic area - do NOT
reproduce any part of it.

Output ONLY the new answer text. Do not include labels, explanations, or
formatting.
```

### A.3.2 Legacy Prompt Set (Older Preparation Path)

**synthetic_data/_older_prep/prompts/main_question_batch.md**

Purpose: normalizes raw questions into TTS-friendly text for the legacy synthetic-data preparation flow. Used in text: referenced as part of the legacy preparation path in Section 3.1.

```
I need your help with generating a dataset. You will receive multiple
questions. Your task is to write each question as you received it but so that
it is ready to be read by a TTS system. This mostly means changing
characters where the pronunciation is not clear to their word variant. Also
fix any small typos that you encounter. Do not change the questions
themselves, only the problematic parts to their readable counterparts.
The input will be a json list of strings where each string is one question.
The output should be a json list of strings where each element is only one
line with no extra comments. Make sure to match the number of elements
of the input. Separate the strings by commas. Don't add extra comments
to the list.
```

**synthetic_data/_older_prep/prompts/main_guide_batch.md**

Purpose: generates compact keyword notes (guides) from questions in the legacy Q&A pipeline. Used in text: referenced as part of the legacy preparation path in Section 3.1.

```
I need your help with answering a set of questions. I don't know the
answers but I will need to say them. Your goal is to provide basic notes for
me that I will use to answer each question. The notes for each question
should be as brief as possible. If the main answer is just one to three
words, only output those. I know how to make sentences. If the question
requires a longer answer, make the notes as brief as possible. You don't
need to build whole sentences.
The input will be a json list of strings where each string is one question.
The output should be a json list of strings where each element is only one
line - one note/set of notes with no extra comments. Each string should
contain all the notes for the given question. There cannot be more strings
on the output than questions on the input. Separate the strings by
commas. Don't add extra comments to the list.
```

**synthetic_data/_older_prep/prompts/main_answer_batch.md**

Purpose: generates spoken-style answers conditioned on question plus guide in the legacy preparation path. Used in text: referenced as part of the legacy preparation path in Section 3.1.

```
I need your help with generating a dataset. You will receive a question and
a guidance. The guidance includes the information which is needed to
answer the question. Your task is to take the guidance and generate an
answer that will be later converted to speech. Only use information provided
in the guidance and don't invent anything new unless it is very common
knowledge or you need it to construct the sentences. Generate the answer so
that it is ready to be read by a TTS system - change characters where the
pronunciation is not clear to the word variant.
The input will be a json list of objects where each object has a key
"question" whose value is the question and a key "guide" whose value is the
guide.
The output should be a json list of strings where each element is only one
line (the answer for a given question with the given guide) with no extra
```

comments. Each element of the list should correspond to one element of the input. Make sure to match the number of elements of the input. Separate the strings by commas. Don't add extra comments to the list.

### A.3.3 Internal Evaluation Prompt Templates

**`LLM_Correctness` (from `llm_correctness_metric.py`)**

Purpose: defines the LLM-judge rubric and prompt template for correctness scoring. Used in text: metric definition in Section 3.5 and reported experiment results in Section 4.1.

```
You are a speech quality assessor for a voice assistant research project. You
will receive a transcribed user turn, an expected answer, and a transcribed
assistant response. Your task is to judge the general correctness of the
actual answer based on the expected answer, taking the context of the
user's last turn into account. The transcripts may contain disfluencies,
partial sentences, or unusual content because they come from real recorded
speech.

Rate the actual answer on a scale of **1 to 5**:

1 -- **Completely incorrect**: The actual answer is entirely wrong,
irrelevant, or contradicts the expected answer.
2 -- **Mostly incorrect**: The actual answer contains some relevant words
but misses or distorts the key information from the expected answer.
3 -- **Partially correct**: The actual answer captures some of the main
points but omits or alters important details.
4 -- **Mostly correct**: The actual answer conveys nearly all key
information from the expected answer. If the expected answer is a short fact
and the actual answer embeds it in a longer correct sentence, it should
score highly.
5 -- **Fully correct**: The actual answer captures the expected answer
completely. It is perfectly acceptable if the actual answer is longer or adds
conversational filler, as long as the core facts remain correct and it logically
fits the user's turn.

Important guidelines:
- Focus on **factuality and correctness** given the context of the user turn.
- If the expected answer is short (e.g., a single word or number), the
actual answer should contain that information. It is fine if the actual answer
is a full sentence containing the expected short answer.
- Additional information provided in the actual answer is acceptable as long
as it doesn't contradict the expected answer.
- First provide your reasoning, then assign a score from 1 to 5.
```

```
Below are transcribed speech segments to evaluate for correctness.

<user_turn>
{user_turn}
</user_turn>
```

<expected_answer>
{expected}
</expected_answer>

<assistant_response>
{actual}
</assistant_response>

Rate the assistant response for correctness against the expected answer.

### `LLM_Dialogue_Naturalness` (from `llm_dialogue_naturalness_metric.py`)

Purpose: defines the LLM-judge rubric for conversational naturalness independent of factual correctness. Used in text: metric definition in Section 3.5 and reported experiment results in Section 4.1.

You are a speech quality assessor for a voice assistant research project. Your ONLY task is to evaluate if the assistant's response is a natural, coherent, and grammatically correct reaction to the user's turn. The transcripts may contain disfluencies or partial sentences because they come from real recorded speech.

CRITICAL RULES:
1. **Relevance is key**: The response MUST be relevant to the user's turn. If it empty or addresses a completely different topic, it should be rated very low.
2. **Brevity is perfectly fine**: Do NOT penalize short answers. A totally natural human response is often just a few words. Do NOT deduct points because an answer "could go into more detail" or "is too short".
3. **Factual accuracy is irrelevant**: The answer can be completely factually wrong. If it sounds like a natural, grammatically valid conversational response to the prompt, it gets full marks.
4. **Grammar and coherence over perfection**: The main factors are whether the response makes grammatical sense as a reply to the user.

Rate the actual answer on a scale of **1 to 5**:

1 -- **Incoherent / No response**: Empty, completely broken grammar, or entirely unrelated.
2 -- **Poor**: Barely relates to the user's turn, extremely unnatural, or mostly incoherent.
3 -- **Acceptable but awkward**: Relates to the user's turn but is highly unnatural, very awkward, or noticeably disjointed.
4 -- **Almost perfect**: Relevant and natural, but has some awkward phrasing, minor grammatical errors, or a genuinely jarring cutoff.
5 -- **Natural & Relevant**: The response directly addresses the user's turn in a grammatically coherent, natural way. **Even if it is extremely short (e.g., "Yes", "I don't know") or factually incorrect, it MUST receive a 5 if it is a coherent conversational reply. Do NOT require deep engagement or lengthy elaboration.**

First provide your reasoning, then assign a score from 1 to 5.

Below are two transcribed speech segments to evaluate.

<user_turn>
{user_turn}
</user_turn>

<assistant_response>
{actual}
</assistant_response>

Rate the assistant response for conversational naturalness.

### LLM_Guide_Following (from llm_guide_following_metric.py)

Purpose: defines the LLM-judge rubric for how well model responses use the provided guide. Used in text: metric definition in Section 3.5 and reported experiment results in Section 4.1.

You are a speech quality assessor for a voice assistant research project. Your task is to judge how well a voice assistant used a "guide" (short instruction, keywords, or talking points) when responding to a user. The transcripts may contain disfluencies, partial sentences, or unusual content because they come from real recorded speech.

You will be given:
1. The **previous user turn** (what the user just said).
2. The **guide** (the information, words, or instruction the assistant was supposed to use).
3. The **actual answer** produced by the assistant.

Think of the guide as a hint or cue card a human speaker would glance at before answering. The assistant should not blindly copy the guide verbatim -- it should **think about** the guide and weave it into a response that makes sense given what the user said. Ask yourself: "If a human had this guide and this user turn, would they produce a similar answer?"

Rate how well the actual answer used the guide on a scale of **1 to 5**:

1 -- **Completely ignored**: The actual answer shows no trace of the guide -- it talks about something entirely different or is empty.
2 -- **Barely used**: The actual answer only tangentially relates to the guide. The core intent or keywords of the guide are missing.
3 -- **Partially used**: The actual answer picks up some elements of the guide but misses other important parts, or uses them in a forced / awkward way.
4 -- **Well used**: The actual answer clearly draws from the guide and addresses the user's turn. It adapts or rephrases the guide so it fits the

```
conversation naturally. Minor deviations or rewording are acceptable -- a
human would likely do the same.
5 -- **Fully used**: The actual answer naturally incorporates the guide's
core content, adapts it to the conversational context, and the guide's intent
is fully realized.

Important guidelines:
- Do NOT judge factual correctness. Focus on whether the guide was used.
- Guides can be vague or incomplete. If the guide is ambiguous and the
assistant made a reasonable interpretation of it, do not penalize.
- The guide may contain extra words that are unnecessary for the answer
(e.g., repeating something the user already said). It is fine to omit such
redundancies in the answer.
- A human given the same vague guide might produce a range of valid
answers -- be lenient as long as the assistant's answer falls within that
range.
- First provide your reasoning, then assign a score from 1 to 5.
```

```
Below are transcribed speech segments and a guide to evaluate against.

<user_turn>
{user_turn}
</user_turn>

<guide>
{guide}
</guide>

<assistant_response>
{actual}
</assistant_response>

Rate how well the assistant response used the guide.
```

### A.3.4 Runtime Inference Prompt

**`warmblood/moshi/moshi/prompts/inference_main.md`**

Purpose: main runtime prompt for the external guide-generating LLM in real-time inference. Used in text: inference design and prompting strategy in Section 2.2 and Section 2.2.1, with runtime setup context discussed in Chapter 3.

```
You generate concise keyword guides for a spoken dialogue AI assistant in
real time.

You receive new conversation turns -- user transcriptions from ASR and
agent responses from the model. Your previous responses (guides you
already sent) are visible in this chat history; do not repeat information you
already provided unless the user asks the same thing again in a new way or
the agent hasn't mentioned it yet and it makes sense to do so.
```

Your task: produce a short keyword guide containing useful content the agent should say next -- facts, suggestions, follow-up questions, or relevant details. The agent is trained to incorporate whatever you provide, so always aim to give something helpful. Only respond with "null" when there is genuinely nothing to add (e.g. the user said something short, or the turn is an incomplete fragment).

The agent is trained to speak on its own but when it receives a guide, it will very likely use it. Make sure it's short (not a long list of things) and understandable.

When the agent starts speaking about something irrelevant, lead it back to the topic.

Guide rules:
- Use short, informative phrases.
- Provide actual content: concrete facts, numbers, names, dates, suggestions, follow-up points.
- Drop every word that is obvious from context. If the user asked about something, do not write it in the guide -- just give the answer details.
- When the agent should ask for missing info, lead with "ask" (e.g., "ask about plans", "ask number of people").
- Never use third-person pronouns (he/him/she/her/they/them) when referring to the user or other nouns ("user") because the agent might repeat what you say. Use neutral phrasing.
- Distinguish confirming a completed action ("booking confirmed") from offering one ("offer to book").
- Do not use special symbols (no degree signs, slashes, parentheses for unit conversions). Write units as plain words or abbreviations.
- Do not include filler words, politeness phrases, or hedges.

Respond with ONLY the guide text or "null". No explanation, no JSON, no formatting.

### A.3.5 Survey Instruction Generation Prompts

**warmblood/moshi/moshi/prompts/instructions_chitchat.md**

Purpose: generates participant instructions for open-ended chitchat survey conversations. Used in text: prompt-family discussion in Section 4.1.5 and survey-evaluation context in Section 4.2.

You are writing short instructions for a participant in a spoken-dialogue survey.

The participant will have a live voice conversation with an experimental spoken dialogue system. Your job is to give them a concrete, engaging task so they know what to talk about. The conversation should last 3 to 5 minutes.

```
You will be given a TOPIC. Write a brief instruction (2 to 4 sentences)
that:
- Frames a natural, conversational situation around that topic.
- Suggests what the participant could talk about, while leaving them
freedom to keep the conversation open-ended.
- Encourages them to keep talking for a few minutes and to speak naturally.

Style guidelines:
- Address the participant directly using "you".
- Be friendly and concise. Do not use bullet points or headings.
- Do not mention that you are an AI or that these instructions were
generated.
- Do not include a greeting or sign-off. Output only the instruction text.
```

**warmblood/moshi/moshi/prompts/topics_chitchat.txt**

Purpose: topic bank used together with `instructions_chitchat.md` to sample chitchat survey tasks. Used in text: instruction-generation artifacts referenced in Section 4.1.5 and Section 4.2.

```
Planning a weekend trip to a city you have never visited
Your favourite book, film, or TV series and why you love it
Cooking a meal and asking for recipe ideas
A hobby you would like to pick up and how to get started
Discussing the pros and cons of remote work
Talking about your favourite music and discovering new artists
Sharing a goal you hope to achieve within the next year
Planning a small celebration or party for a friend
Debating whether cats or dogs make better pets
Reminiscing about a memorable holiday or childhood memory
Brainstorming gift ideas for a family member
Discussing how cities could be made more livable
Sharing opinions on a recent change in your daily routine
Talking about the best way to relax after a busy day
Describing a place you would like to visit someday
Your funniest or most awkward travel mishap
A time you learned something surprising about yourself
Planning a weekend of outdoor activities
Talking about a favorite local restaurant and why you like it
Discussing a recent news story and your reaction to it
Sharing tips for saving money or budgeting
Describing your ideal morning routine
Talking about a creative project you'd like to start
Recommending podcasts or shows you enjoy
Discussing how you keep in touch with old friends
Sharing a tradition your family celebrates
Talking about your favorite season and activities you enjoy
Debating whether paper books or e-readers are better
Describing a skill you were proud to master
Planning a themed movie night with friends
Talking about an inspiring person in your life
```

```
Sharing tips for improving sleep quality
Discussing sustainable habits you'd like to try
Describing a memorable meal you cooked or ate
Brainstorming weekend activities on a tight budget
Talking about balancing work and personal life
Sharing your experience learning a new language
Planning a DIY home improvement project
Talking about volunteering or community involvement
Discussing the role of social media in daily life
Sharing your favorite apps that make life easier
Describing how you celebrate small wins
Talking about a pet peeve and why it bothers you
Brainstorming ways to de-stress during busy weeks
Discussing how you'd plan a surprise for someone
Sharing a memorable encounter with a stranger
Describing a childhood game you loved
Talking about how you choose gifts for friends
Discussing your favorite ways to spend a rainy afternoon
Sharing a simple habit that improved your life
```

**warmblood/moshi/moshi/prompts/instructions_qa.md**

Purpose: generates participant instructions for information-seeking Q&A survey conversations. Used in text: prompt-family discussion in Section 4.1.5 and survey-evaluation context in Section 4.2.

```
You are writing short instructions for a participant in a spoken-dialogue
survey.

The participant will have a live voice conversation with an experimental
spoken dialogue system. Your job is to give them a concrete task so they
know what to ask about. The conversation should last 3 to 5 minutes.

You will be given a TOPIC. Write a brief instruction (2 to 4 sentences)
that:
- Frames a natural information-seeking situation around that topic.
- Tells the participant to ask a couple of questions and learn something
specific about the topic.
- Encourages them to keep the conversation going naturally for a few
minutes.

Style guidelines:
- Address the participant directly using "you".
- Be friendly and concise. Do not use bullet points or headings.
- Do not mention that you are an AI or that these instructions were
generated.
- Do not include a greeting or sign-off. Output only the instruction text.
```

**warmblood/moshi/moshi/prompts/topics_qa.txt**

Purpose: topic bank used together with `instructions_qa.md` to sample Q&A-style survey tasks. Used in text: instruction-generation artifacts referenced in Section 4.1.5 and Section 4.2.

```
How electric cars work and what affects their range
What makes a good home coffee setup
The basics of investing and saving money
How to start learning a new language
What to look for when choosing a laptop
How public transportation systems help cities
How solar panels turn sunlight into electricity
What makes a healthy daily routine
How to plan a safe trip to a new city
How podcasts are made and distributed
The main causes of climate change
What to know before starting a podcast
How to prepare for a job interview
What makes a good camera for beginners
How weather forecasting works
How to set up a basic home network securely
What to consider when adopting a pet
Basics of first aid and when to seek help
How to choose the right smartphone for you
How to start a small vegetable garden
Tips for effective remote work and productivity
How streaming services recommend shows
What to look for in a running shoe
What to pack for an international flight
How online privacy and tracking work
Tips for buying a used car safely
How to plan a monthly personal budget
How vaccines are developed and approved
What to know about recycling and waste sorting
How to choose a good pair of headphones
How to prepare taxes as a freelancer
How to maintain a bicycle at home
What to consider when choosing a mortgage
How to start a blog and attract readers
Basics of ethical eating and sustainable diets
How to set fitness goals and track progress
What to know before buying travel insurance
How to read and negotiate apartment rental agreements
Tips for reducing energy use at home
How public libraries provide community services
What makes a great science museum visit
How to choose a microphone for podcasting
Tips for introducing children to books
How to build a simple personal website
What to know about emergency preparedness kits
How to evaluate online course quality
```

```
How to choose a commuter bike
Basics of soil care for urban gardening
How to prepare for a driver's license test
How to compare health insurance plans
```