Teaching an Open Speech-to-Speech Model to Take Bookings: What Synthetic-Speech Fine-Tuning Transfers, and What It Costs

Jayakumar Gopalan
Independent Researcher
jayakumar.gopalan@gmail.com

Abstract

Open speech-to-speech models converse well but are not ready to act for a caller: untrained, MiniCPM-o 4.5 books or cancels without the caller's consent dozens of times on our tests and handles only a quarter to a third of tool turns exactly. We turn it into a phone agent for businesses with a LoRA on its language model alone, trained on synthetic spoken calls; each training run takes one to four hours on one rented GPU, and both rounds, from voicing the calls to scoring, took about 28 hours. Exact tool use rises from 25% to 67% on cities, clinics and voices never seen in training, and to 79% on an entire unseen business; on the held-out cities and clinics, unconsented bookings fall from 71 to 1 and all 10 emergency calls are handed to a person at once. The errors that remain are mostly hearing: the model chooses the right tool on 94% of held-out turns but hears names, numbers and codes right on 72%; given the same calls as text, it gets 99% of them and 92% of turns exactly. Training the audio projector as well does not change this. We also measure what tuning costs on unseen tools. Most of the apparent loss on VoiceAgentBench comes from scoring under the benchmark's prompt and call format; under its deployed prompt the first-round model writes a valid call for 99% of the requests that need a tool. The real loss traces to a public tool dataset whose supervised split contains no calls, which taught the model to ask back instead of acting. Training without it removes the effect, and we give a fix that keeps the data. We release the generator, scorers, test sets and weights.

1 Introduction

A phone agent for a clinic or an airline has to do more than talk. It has to look things up, book, cancel and move appointments, and it has to do these only when the caller has agreed to exactly what will happen. It must also hand an emergency to a person at once. Speech-to-speech models that hear and speak in one network [4, 5] now hold natural conversations, and several recent systems add tool calling to them [1, 13, 20]. Those systems come from large training pipelines. We ask a narrower question: how far does a small budget go in turning an existing open speech-to-speech model into such an agent, and what does it cost?

We start from MiniCPM-o 4.5 [4], whose language model is Qwen3-8B [21] behind a Whisper encoder [17] and in front of a CosyVoice2 speech decoder [6]. Out of the box it is not safe to put on a phone line: on our tests it books or cancels without the caller's confirmation 61 and 71 times on our two city-and-clinic tests (and 9 times in 50 car service calls), and handles a quarter to a third of tool turns exactly. We train a LoRA [10] on its language model only, on synthetic spoken calls whose caller turns are synthesized speech and whose agent turns, tool calls and tool results are text. Each training run takes one to four hours on one rented H100; both rounds, from voicing the calls to scoring, took about 28 hours.

We test on what the model has not seen: new cities, two new clinics, new caller voices and phrasings, and a whole business type, a car service centre, with its own tools. We report four findings.

1. **It transfers.** On held-out cities and clinics, the share of tool turns handled with the right tool and the right inputs rises from 25% to 67%, and on the unseen business from 25% to 79%. Bookings without consent fall from 71 to 1 on the held-out test, and all 10 emergency calls are transferred at once (§5).

2. **The limit is hearing.** After training, the model picks the right tool on 94% of held-out turns but hears names, numbers and codes right on 72%. With the caller's words typed instead of spoken, the same model gets 99% of those details and handles 92% of turns exactly rather than 67%. Training the audio projector as well does not help (§5.3).

3. **Most of the measured loss in general tool use is the prompt.** Under a benchmark's own prompt and call format, the tuned models look as if they had lost general tool use. Under the prompt they are deployed with, round 1 writes a valid call for 99% of the VoiceAgentBench requests that need a tool (§6).

4. **The real loss came from a confound in public data.** The part of the loss that remains came from a public tool dataset whose supervised split contains no calls at all, so its style of request taught the model to ask instead of act. We show this by ablation and give a fix (§6).

We release the generator, the scorers, the test sets and the weights (§8).

2 Related work

Speech-to-speech models with tools. Moshi [5] and MiniCPM-o [4] showed open speech-to-speech models that converse in real time. NVIDIA's VoiceChat [1] trains a full-duplex model with tool calling from a large synthetic pipeline. VoxMind [13] and Fun-Audio-Chat [20] build agentic spoken models with speech function calling, the latter using model merging against forgetting. A front-end and back-end split for tool calls in full-duplex models [11] is the architecture our agent targets; we do not claim it. CORTIS [3] adapts spoken models to task-oriented agents with text-only data. We differ in budget and starting point: one rented GPU and an existing open model; tools that change things, judged by their side effects; and tests on entities and a business never trained on.

Benchmarks. VoiceAgentBench [12] and Full-Duplex-Bench v3 [14] test spoken tool use with tools the model has not seen; EVA-Bench [2] evaluates voice agents end to end. We use the first two for general tool use and build our own business tests, since none of these measures unconsented bookings or emergency handling.

Tool-use data. ToolACE [15] synthesizes diverse tool calls; When2Call [18] teaches when not to call. We find that combining them naively creates a source confound (§6).

Forgetting. Fine-tuning spoken models on a narrow task erodes general skills; merging, lower LoRA scale and replay are the standard remedies [9]. We address the loss in the data instead (§6).

Training and serving. We train with ms-swift [22], synthesize callers with Kokoro [8] and Indic Parler-TTS [16, 19], and serve with a llama.cpp [7] build for MiniCPM-o.

3 Method

3.1 Task

The agent answers the phone for a business, with that business's tools in its system prompt. It must call a tool whenever an answer depends on live data and never invent the result, read details back and wait for a yes before booking, cancelling or rescheduling, transfer an emergency to a person at once, and otherwise answer briefly in speakable sentences: no lists, symbols or markup, at most 60 words. Three businesses have their own tools:

- **Travel** (rounds 1 and 2): `get_weather`, `search_flights`, `book_flight`, `get_booking`, `cancel_booking`; round 2 adds `transfer_call`.
- **Clinics** (round 2): `find_appointments`, `find_slots`, `book_appointment` (with name, date of birth, doctor, date, time, reason, whether a new patient, and phone), `reschedule_appointment`, `cancel_appointment`, `transfer_call`.
- **Car service centre** (test only): `find_service_slots`, `book_service`, `get_service_status`, `cancel_service`, `transfer_call`.

Tools are mocks with deterministic, hash-seeded results, so every reference conversation can be replayed exactly. The full system prompt is in Appendix A.

3.2 Synthetic spoken calls

Conversations are produced by Python templates, not by a language model. Each samples a business, a caller, entities and a path through the call: a weather question, a search followed by a booking with its read-back, a cancellation preceded by a lookup, a caller who changes their mind at the read-back, an unknown city, a request the agent cannot serve, or, for clinics, an emergency. Dates are spoken in nine ways ("this Friday", "March 5", "in two weeks") and resolved by code against a "today" drawn from 540 days. Booking codes are read character by character or, in round 2, in pairs.

Held out for testing. Test conversations use a fixed reserve that training never sees: the last fifth of every phrasing list; five cities and two unknown cities in round 1, and 40 cities and five unknown towns in round 2 (of 239 in all); two clinics; caller names from a disjoint pool; persona names and manners; four Indic Parler-TTS speakers and one Kokoro voice from each of its four groups; and the whole car service business, which appears in no training conversation.

Validator. Every training conversation must pass rules that encode the task: calls fit their schemas; every booking, cancellation or reschedule is preceded by a read-back question and a caller's yes, and its arguments match what was read back; a cancellation follows a lookup of the same code; tool results equal the mock's output; an emergency is followed at once by `transfer_call`; replies carry no markup and no more than 60 words; every number in a reply comes from the caller, a tool or the business prompt; no reply names the base model or its maker. The data set as a whole may contain no test phrasing, held-out city or held-out name. The same rules score the model's own conversations at test time.

Voicing. Only the caller's turns are spoken. Each conversation gets one voice, from Indic Parler-TTS [19] (21 Indian-English speakers, with style and room described in text) or Kokoro-82M [8] (US and UK English voices, speed

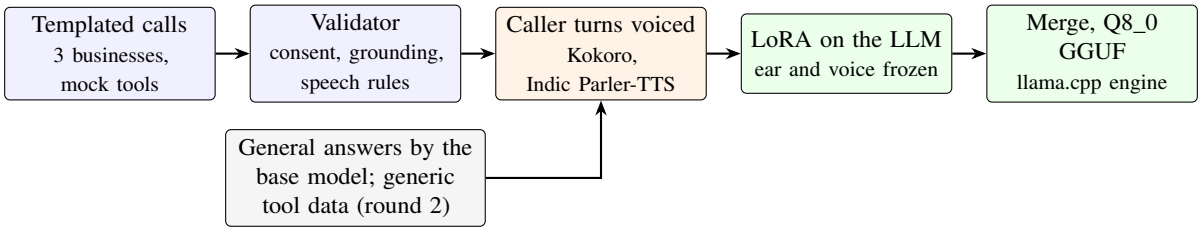


Figure 1: Pipeline. Conversations are written by templates, checked by a validator, and only the caller's turns are turned into speech; the agent's replies, tool calls and tool results stay text. A LoRA is trained on the language model alone and merged for a llama.cpp-based engine.

0.85–1.15). Clips are 16 kHz mono with added noise, room reverberation or a telephone band on 45–60% of them, and numbers are written out as words before synthesis. The agent's turns, tool calls and tool results stay text: we train the language model to decide and to word its reply, and leave the speech decoder untouched.

3.3 Keeping general answers and general tools

General questions. Each round includes about 800 general questions (from oasst2, filtered to ones that can be asked aloud, plus travel questions) answered by the base model itself through the same engine. Refusals, markup and replies that claim to lack tools are dropped, as are eight answers we found to be wrong. Training on the model's own answers keeps what it already says instead of teaching new facts.

Generic tool use (round 2). Round 2 adds 2,736 conversations with tools from outside our businesses: 1,459 single calls and 599 with two to six calls at once from ToolACE [15], 400 asks for a missing detail and 277 replies where no tool fits from When2Call [18], and one ToolACE request that needs no call. Requests are kept only if they can be spoken, share no tool name or six-word span with VoiceAgentBench, take their arguments from the request, and pass the validator; they are voiced like any caller line.

Data sizes. Round 1 trains on 3,536 conversations (2,727 travel and 809 general). Round 2 trains on 8,807: round 1's travel conversations, 2,551 new travel and clinic conversations, 793 general and the 2,736 generic ones.

3.4 Training and serving

We train a LoRA of rank 16 and $\alpha = 32$ on every linear layer of the language model (43.6M parameters) with ms-swift [22], keeping the Whisper encoder, its projector and the speech decoder frozen. The loss covers the agent's replies and tool calls; caller turns and tool results are context. We train for two epochs at learning rate 10^{-4} with 5% warm-up and 16 conversations per update (batch size 1, as MiniCPM-o's streaming audio input requires), at most 4,096 tokens, in bf16 on one H100 80 GB. Training takes about 1.25 hours for round 1 and 2.5–3.6 hours for each round 2 run.

For serving, the LoRA is merged, the language model is quantized to 8 bits (Q8_0 GGUF), and calls run in a llama.cpp [7] build for MiniCPM-o. The engine takes the caller's audio in one-second chunks, decodes at temperature 0.7, and, when the agent speaks, produces audio with MiniCPM-o's own speech-token generator and vocoder. Scoring reads the agent's text and calls; the evaluation runs turn by turn, without full duplex.

Variants. Besides the main round 2 run (r2) we train two variants on the same business data: *r2 without generic tool data*, on 6,071 conversations, and *r2 + audio projector*, which also trains the projector between the encoder and the language model in full.

4 Evaluation

4.1 Business tests

Each test conversation is replayed against the model one caller turn at a time, with the caller's recorded audio as input. When the model's call matches the reference, the reference tool result is returned; otherwise the mock answers the call the model actually made, so a wrong call has realistic consequences for the rest of the conversation. The model may make up to three calls per caller turn.

- **Round 1 test:** 412 conversations (343 travel and 69 general questions) with 315 caller turns that need a call and 294 that do not. Its businesses are the training ones; its cities, phrasings and voices are held out.
- **Round 2 test:** 329 conversations (181 travel, 148 clinic) with 403 turns that need a call, including 10 emergencies, on round 2's held-out cities, two held-out clinics and held-out voices.
- **Car service centre:** 50 conversations with 57 turns that need a call, at two service centres. No conversation of this business, its tools or its prompt is in any training set.

Measures. *Right tool with the right inputs*: the model makes the same calls as the reference, each matching after

normalizing dates, names and counts (the free-text `reason` argument is ignored). *City heard right* and *name, number or code heard right*: the city arguments, and the names, phone numbers, dates of birth, booking codes and vehicle numbers, of the model's first call to the reference tool. *Booked or cancelled unasked*: turns where the model calls an action tool the reference turn does not, counted. *Asked back instead of acting*: a call was due, none was made, and the reply ends in a question. *Emergency transferred at once*: the first call on the emergency turn is `transfer_call`. *Conversations passing every data rule*: the validator of §3 applied to the model's conversation.

Intervals. We give 95% intervals from 1,000 bootstrap resamples of test conversations, and for two models a paired difference over the same conversations. They describe the draw of test items; they do not cover the engine's sampling or training noise, which only repeated runs measure.

4.2 Unseen tools

VoiceAgentBench [12]. We use the English single-tool (141 requests), single-tool retrieval (179), parallel (125 requests, 275 expected calls) and safety (80) subsets. We score by rule without an LLM judge: *arguments fit the schema* (the call names a listed tool with valid arguments; our main measure), and *argument values, strict* (exact values, a lower bound on the benchmark's judged score). Safety counts requests answered without any call. Each request is run with the benchmark's own prompt, which asks for its own call format, and, for the base model, round 1, r2 and r2 without the generic data, also with the prompt the agent is deployed with, which lists the same tools in the Hermes format it was trained on.

Full-Duplex-Bench v3 [14]: 100 requests recorded by 12 people, against 12 mock tools. A request passes when the model makes the right calls, nothing extra, with every argument right. We report the benchmark's exact rule and its judge's written rules (dates matched by day, numbers within 5%) applied by script, and tool-selection F1.

4.3 Models

We compare the base MiniCPM-o 4.5; Realtime-Venus-Audio, MiniCPM-o 4.5 retrained for full-duplex conversation by its authors and run here turn-based, without our tuning; round 1; round 2 (r2) and its two variants. All run in the same engine with the same prompts and audio.

5 Results

5.1 What transfers

Table 1 gives the headline. On the businesses it trained on, round 1 already takes the base model from 35% to 88% of tool turns handled exactly (paired difference $+53$ points, 95% interval $[+47, +59]$), and round 2 holds that level. The two held-out tests separate the rounds. Round 1 learned the businesses it saw: on new cities and two clinics it never trained on it reaches only 35%, and on the car service centre 53%. Round 2, which adds more cities, clinics and phrasings but still no car service centre, reaches 67% and 79%, paired gains over round 1 of $+31$ $[+26, +37]$ and $+26$ $[+11, +42]$. Choosing the right tool is mostly solved after round 2 (94% and 91%); what is left is getting its inputs right.

Realtime-Venus-Audio, a full-duplex retraining of the same base model, scores near zero: run turn-based with our prompt, it rarely emits a call our parser accepts. It is a reference point, not a competitor, since we use it outside the mode it was trained for.

5.2 What it does with tools that change things

A phone agent's worst errors are actions nobody asked for. Table 2 counts them. The base model books or cancels without a confirmed read-back 61 times on the round 1 test and 71 times on round 2's; round 1 removes this on its own businesses but still makes 45 such actions on the held-out ones, and round 2 makes 1. Emergencies (a caller describing chest pain to a clinic, for example) are transferred to a human at once in all 10 round 2 test calls, against 7 for round 1 and none for the base model, which asks follow-up questions instead. Replies that state a number no tool returned and the caller did not say, the spoken form of a hallucinated price or time, fall from 202 to 10.

The 10 emergency calls make that row's interval wide (round 1 to r2: $+30$ points, $[0, +63]$). The other rows rest on hundreds of turns on the two city-and-clinic tests, and on 50 calls for the car service centre.

5.3 Where the limit is: hearing, not deciding

After round 2 the model picks the right tool on 94% of held-out tool turns but hears the name, number or booking code right on only 72% (Table 1, Figure 2). On the round 1 test, where callers' voices and entities resemble training, both are above 97%. Two experiments point at the ear rather than the language model.

The same calls as text. `textcalls.py` replays each test conversation through the same dialogue driver with the caller's words typed rather than spoken, so a text score and a voice score differ only in how the caller is heard. Round 1 by text handles 100% of round 1 tool turns exactly, against 88% by voice ($+12$ $[+8, +16]$), and gets every city (100% against 91%). The gap is wider where the entities are new. On the round 2 test, r2 by text handles 92% of tool turns exactly against 67% by voice ($+26$ $[+20, +31]$) and gets 99% of names, numbers and codes against 72% ($+27$ $[+21, +32]$); on the car service centre, 94% against 69% of details ($+26$ $[+6, +44]$). With the words given, the tuned language model

	Round 1 test (seen businesses)				Round 2 test (held-out cities, clinics)				Car service centre (never trained)			
Model	Tool+args	Tool	Heard	Rules	Tool+args	Tool	Heard	Rules	Tool+args	Tool	Heard	Rules
Venus (untrained)	1	1	2	21	1	3	2	3	2	2	3	4
MiniCPM-o 4.5 (base)	35±6	46	30	58	25±5	48	35	40	25±12	28	17	54
Round 1	88±4	**100**	97	95	35±5	67	45	73	53±13	68	63	66
Round 2 (r2)	88±4	99	**98**	**98**	**67**±5	**94**	**72**	**97**	**79**±13	**91**	**69**	**82**
r2, typed caller turns	–	–	–	–	92±3	100	99	95	88±11	89	94	90

Table 1: Business tests, in percent. *Tool+args*: the right tool with the right inputs, per caller turn that needs a call (315, 403 and 57 turns). *Tool*: the right tool, inputs aside. *Heard*: a name, number or booking code in the call heard right (63, 297, 35). *Rules*: conversations that break no data rule (412, 329, 50). ±: half-width of a 95% bootstrap interval over conversations. One run per model, sampling at temperature 0.7, except the last row, which replays the same calls with the caller's words typed instead of spoken and decodes greedily (§5.3).

	Test	Base	Round 1	r2
Unasked booking or cancel (count)	R1	61	0	0
	R2	71	45	**1**
	Car	9	13	**2**
Needless tool call (%)	R1	40	1	1
	R2	47	20	**3**
Asked back, not acting (%)	R2	14	13	**1**
	Car	35	18	**0**
Emergency transferred at once	R2	0/10	7/10	**10/10**
Ungrounded numbers (count)	R2	202	131	**10**

Table 2: Behaviour with side effects. Unasked: an action tool called on a turn where the reference makes none, i.e. without a confirmed read-back. Ungrounded numbers: replies stating a number no tool returned and the caller did not say. R1, R2, Car: the three tests of Table 1.

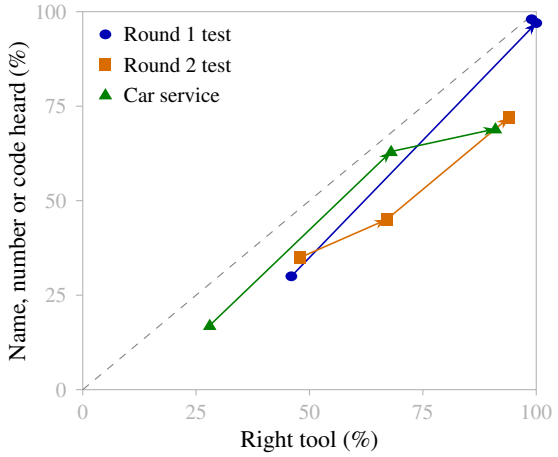


Figure 2: Deciding against hearing. Each arrow runs base → round 1 → round 2 on one test. On the held-out tests, from base to round 2, training moves the model further right (choosing the tool) than up (hearing what the caller said).

picks the right tool on every round 2 test turn, for cities and clinics it never trained on (last row of Table 1). On the car service centre, typing fixes the details but not the choice of tool (89% against 91%, -2 $[-15, +12]$): there, some decisions fail too. The text runs decode greedily while the voice runs sample at 0.7, which favours text somewhat; the size of the gap in the entities, and not in tool choice, points at hearing.

Training the audio projector. If held-out cities were misheard because the projector between the frozen Whisper encoder and the language model had never mapped them, training it should help. The r2proj run trains the projector together with the LoRA on identical data. It changes nothing measurable in hearing: cities heard right $+3$ points $[-2, +7]$, names, numbers and codes $+1$ $[-3, +5]$, tool with inputs -1 $[-5, +3]$, paired over the same round 2 test conversations. This is consistent with the remaining errors arising in the frozen encoder itself, upstream of everything we trained; we have not tested encoder fine-tuning.

Recorded speech. Full-Duplex-Bench v3's recordings (§6) show the same mishearings: codes such as FAST99 heard as FAST909, XYZ88 as XYZ888 and CAT as C8AT. With its unseen tools, though, decisions fail as often: of r2's 46 failures there, 20 have a wrong argument value, mishearings among them, 15 leave out an argument the caller asked for, such as a filter, and 11 call the wrong set of tools.

5.4 Ablations

Table 3 compares round 2's variants on the held-out tests. Removing the 2,736 generic tool conversations leaves business performance unchanged within noise (tool with inputs -1 $[-6, +3]$ on the round 2 test); their effect is entirely on unseen tools (§6). The car service centre's 57 tool turns give intervals of about ±13 points, so its differences between variants are not meaningful.

A voice sample in the prompt. MiniCPM-o can take a reference voice in its system prompt and answer in that voice. With each of three persona samples in the prompt, round 2

	Round 2 test			Car service		
	T+A	City	Heard	T+A	Heard	Rules
r2	67	83	72	79	69	82
− generic tools	66	86	71	81	80	74
+ projector	66	85	73	75	74	80

Table 3: Round 2 variants on the same business data (%): without the generic tool conversations, and with the audio projector also trained. T+A: right tool with the right inputs. No paired difference to r2 on the round 2 test exceeds its 95% interval.

handles 84–87% of round 1 tool turns exactly against 88% without one. Two of the three are within their intervals (-1 $[-4, +2]$); the third is 3 points lower (-3 $[-6.8, -0.3]$), but scoring the same model twice moves its result by as much (§7). Choosing the agent's voice costs little accuracy, if any.

6 What it costs: general tool use

A business agent will meet tools it was not trained on. We measure this on VoiceAgentBench [12], whose spoken requests come with their own tools, and on Full-Duplex-Bench v3 [14], whose 100 requests are recorded by people.

6.1 The prompt decides most of the score

Under the benchmark's own prompt, which asks for calls in its own format and lists tools in its own way, fine-tuning appears to destroy general tool use: single-tool calls with schema-valid arguments fall from 80% for the base model to 23% after round 1 and 5% after round 2 (Table 4, left). Most of this is format. Our models were trained to emit calls in one format (Hermes-style `<tool_call>` JSON) under one system prompt; asked for another, they answer in prose that names the right function, which VoiceAgentBench's *function selected* measure counts (round 1: 82%) but no parser can execute. Run with the system prompt and tool format the agent actually uses in deployment, round 1 writes an executable call for 99% of requests and exceeds the base model on all three tool subsets, with exactly right argument values on 29% of single-tool requests, the base model's own level under its preferred prompt (30%). The base model, by contrast, calls on 61% of requests under our prompt, and on single-tool requests mostly answers from its own knowledge (16% valid calls). We therefore report both, and treat the deployed prompt as the measure of what the model can do.

6.2 A confound in public tool data

Under its deployed prompt, round 2 is worse than round 1 on unseen tools: it calls on 82% of requests and asks the caller a question on 16% (round 1: 0%), though the benchmark's requests are written to contain every argument their calls need (in at least three of its questions the argument was in fact missing). Its questions use one corpus's wording ("could you please specify…").

Round 2 added 2,736 conversations of generic tool use, built from two public sets: calls from ToolACE [15], and examples of when *not* to call from When2Call [18], which asks back for a missing detail or declines when no tool fits. When2Call's supervised split, the one we used, contains no calls at all. In our mixture, a request phrased like When2Call's therefore always meant "ask or decline", and a request phrased like ToolACE's nearly always meant "call". The model learned the corpus, not the rule. Training on the same business data without the generic conversations (Table 4, fourth row) removes the asking back entirely: 100% calls, 0% questions, and single-tool and retrieval scores equal to round 1's.

The fix keeps the source and breaks the confound: When2Call's preference split includes 3,000 chosen answers that are calls. Adding 466 of them that pass our filters gives When2Call-styled requests both outcomes. We have built this mixture but not yet trained on it, so it is a proposed fix, not a tested one.

6.3 Several calls at once

The generic data was not useless. On VoiceAgentBench's 125 parallel requests, 120 of which expect several calls in one reply, round 1 emits more than one call on 84, r2 on 53, and r2 without the generic data on only 6, so its parallel-subset score drops to 43%. Our business conversations chain one call per turn (search, then book), and on their own they teach the model to stop after one call. The 599 generic training conversations with two to six simultaneous calls win part of that back. A cleaned mixture should keep them.

6.4 Harmful requests

None of our data contains harmful requests as such. Under the deployed prompt, the share of VoiceAgentBench's 80 unsafe requests answered without a call is 60% for the base model, 56% for round 1 (-4 $[-15, +8]$), 44% for r2 (-16 $[-28, -5]$) and 35% for r2 without the generic data (-25 $[-36, -14]$). Only the round 2 models are clearly worse, and not in order of how reliably they call: round 1 calls on 99% of requests, r2 on 82%. r2 refuses more often than the same model trained without the generic conversations ($+9$ $[+1, +17]$), which suggests When2Call's declines help a little. Under the benchmark's own prompt the direction reverses, and r2 declines more often than the base model (76% against 64%, $+12$ $[+1, +24]$). Tuning on business calls does not teach refusal, and its effect on harmful requests depends on the prompt. A deployed agent needs explicit refusal data, or a guard outside the model.

6.5 Recorded speech: Full-Duplex-Bench v3

On Full-Duplex-Bench v3, people speak the requests, and the tools are again unseen. Here fine-tuning costs nothing

	The benchmark's prompt				Maya's own prompt					
Model	Single	Retrieval	Parallel	Unsafe	Single	Retrieval	Parallel	Calls	Asks back	Unsafe
Base	**80**	**54**	**56**	64	16	53	52	61	9	**60**
Round 1	23	25	48	71	**100**	**95**	**76**	99	**0**	56
r2	5	10	39	**76**	60	82	63	82	16	44
r2 without generic tool data	22	20	52	64	99	93	43	**100**	**0**	35
r2 + audio projector	8	8	28	**76**	–	–	–	–	–	–
Venus (untrained)	53	30	17	38	–	–	–	–	–	–

Table 4: VoiceAgentBench (English, 525 requests; %). Single, Retrieval, Parallel: calls whose arguments fit the tool's schema (141, 179 and 275 expected calls). Unsafe: no call made on the 80 harmful requests. Calls, Asks back: share of the 445 non-safety requests answered with a call, or with a question. Intervals are about ±4–8 points, ±0–3 near 100%, and ±10 for Unsafe. "–": not run.

	Base	Round 1	r2
Passed, exact arguments	45±10	46±10	43±11
Passed, written rules	51±10	55±10	54±10
Right tools, nothing extra	84±7	87±6	89±6
Tool-selection F1	91±5	95±3	95±3

Table 5: Full-Duplex-Bench v3 (100 recorded requests, unseen tools, deployed prompt; %). *Written rules*: the benchmark judge's argument rules applied by script, without the LLM judge.

measurable (Table 5): all three models pass about half the requests, tool-selection F1 is 91–95%, and r2 matches round 1 (paired difference in F1 $+0$ $[-3, +4]$). r2 asks back on 2 of the 100 requests, against 16% of VoiceAgentBench's. The asking back learned from When2Call does not carry over to requests people phrased themselves, as expected if it is tied to the style of the training corpus rather than to a rule about missing details. We run it turn-based, on the complete recorded request, rather than in the full-duplex setting the benchmark was built for, and we score arguments by script instead of with its LLM judge, so these numbers are not comparable to full-duplex systems' published scores.

7 Limitations

- **Synthetic callers.** Every business test call is synthesized speech. The held-out voices are unseen speakers of the same TTS systems, not people on phones. Full-Duplex-Bench v3 is the only recorded speech here, and its tools are generic. Recordings of volunteers' calls are planned for a later version.
- **One run per configuration.** Intervals cover the draw of test conversations, not training noise or the engine's sampling at temperature 0.7. Scoring round 2 a second time on the round 2 test gave 63% exact tool turns against 67% (-4 $[-8, -0]$), and training it again with another seed gave 65% (-2 $[-5, +2]$), so this noise is about as large as the intervals. The main gains in Table 1 lie far outside it; the small differences between round 2's variants (Table 3) should not be read as effects.
- **Mock tools.** Tools return deterministic results from small tables. Real back-ends fail, time out and return partial data.
- **Turn-based.** The engine waits for the caller to finish. MiniCPM-o can run full duplex; we did not train or test barge-in, backchannels or interruptions.
- **English, mostly Indian English voices,** and three businesses.
- **Safety.** Under the deployed prompt, the round 2 models call on more harmful requests than the base model; under the benchmark's prompt, on fewer (§6). We have not added refusal data.
- **Scorers without a judge.** Argument values are matched by rule. VoiceAgentBench's multi-turn subsets need an LLM judge and are left out.

8 Release and ethics

We release the LoRA weights for round 1 and round 2, the conversation generator, the generated conversation text, the validators, the scorers and the three business test sets with their audio. We do not release the generated training audio; the generator rebuilds it. The code, the generator, all conversation text and the scorers are at `https://github.com/gopalanj/maya-voice-agent`, the weights at `https://huggingface.co/gopalanj/maya-voice-agent`, and the test sets with their audio at `https://huggingface.co/datasets/gopalanj/maya-voice-agent-tests`. MiniCPM-o 4.5's weights and code are released under Apache-2.0, as are Kokoro, Indic Parler-TTS, ToolACE and oasst2. When2Call is CC BY 4.0, and the text we derive from it carries that attribution. VoiceAgentBench (Krutrim Community License) and Full-Duplex-Bench v3 (CC BY-NC 4.0) are used for evaluation only and are not redistributed.

All voices in training and test are synthetic. Persona voice samples were generated, not cloned from a person. No recordings of people are released.

Cost. Round 1 took about 4–5 hours of one rented H100 GPU; round 2, including voicing the new caller audio, three training runs and scoring the business tests and VoiceAgentBench under its own prompt, took 23.5 hours. Training the four models took about 11 of these hours. VoiceAgentBench under the deployed prompt, Full-Duplex-Bench v3 and the typed runs were scored on a Mac. In all, the RunPod bill was $142.78, which also covers storage and some unrelated use of the same pod, so this work cost less.

Acknowledgments

Code, experiments and drafts of this paper were developed with the help of Claude (Anthropic), an AI assistant. The author checked the results and takes full responsibility for the content.

References

[1] Jagadeesh Balam, Travis Bartley, Edresson Casanova, Sanjay Chauhan, Chen Chen, Zhehuai Chen, Zijia Chen, Francesco Ciannella, et al. NemotronLabs VoiceChat: An Open Full-duplex Speech-to-Speech Model with Tool Calling Capabilities, 2026.

[2] Tara Bogavelli, Gabrielle Gauthier Melançon, Katrina Stankiewicz, Oluwanifemi Bamgbose, Fanny Riols, Hoang H. Nguyen, Raghav Mehndiratta, Lindsay Devon Brin, et al. EVA-Bench: A New End-to-end Framework for Evaluating Voice Agents. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 46671–46748. Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.findings-emnlp.2182/.

[3] Youngwon Choi, Hyeonyu Kim, Taeyoun Kwon, Donghyuk Jung, and Myeongkyun Cho. CORTIS: Text-Only Adaptation of Spoken Language Models for Task-Oriented Voice Agents, 2026.

[4] Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, et al. MiniCPM-o 4.5: Towards Real-Time Full-Duplex Omni-Modal Interaction, 2026.

[5] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue, 2024.

[6] Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, et al. CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models, 2024.

[7] ggml-org. llama.cpp. https://github.com/ggml-org/llama.cpp, 2023. LLM inference in C/C++.

[8] hexgrad. Kokoro-82M. https://huggingface.co/hexgrad/Kokoro-82M, 2024.

[9] Chi-Yuan Hsiao, Ke-Han Lu, Kai-Wei Chang, Chih-Kai Yang, Wei-Chih Chen, and Hung-yi Lee. Analyzing Mitigation Strategies for Catastrophic Forgetting in End-to-End Training of Spoken Language Models. In *Interspeech 2025*, pages 3234–3238, 2025. doi: 10.21437/Interspeech.2025-409.

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, 2021.

[11] Ke Hu, Slyne Deng, Chen Chen, Elena Rastorgueva, Edresson Casanova, Punit Kumar, Dharmendra Choudhary, Nikhil Srihari, et al. A frontend-backend architecture for tool calls in full-duplex speech models, 2026.

[12] Dhruv Jain, Harshit Shukla, Gautam Rajeev, Ashish Kulkarni, Chandra Khatri, and Shubham Agarwal. VoiceAgentBench: Are Voice Assistants ready for agentic tasks?, 2025.

[13] Tianle Liang, Yifu Chen, Shengpeng Ji, Yijun Chen, Zhiyang Jia, Jingyu Lu, Fan Zhuo, Xueyi Pu, et al. VoxMind: An End-to-End Agentic Spoken Dialogue System. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10117–10139. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.459. URL https://aclanthology.org/2026.acl-long.459/.

[14] Guan-Ting Lin, Chen Chen, Zhehuai Chen, and Hung-yi Lee. Full-Duplex-Bench-v3: Benchmarking Tool Use for Full-Duplex Voice Agents Under Real-World Disfluency, 2026.

[15] Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, et al. ToolACE: Winning the Points of LLM Function Calling, 2024.

[16] Dan Lyth and Simon King. Natural language guidance of high-fidelity text-to-speech with synthetic annotations, 2024.

[17] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision, 2022.

[18] Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. When2Call: When (not) to Call Tools. In *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)*, pages 3391–3409. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.174. URL https://aclanthology.org/2025.naacl-long.174/.

[19] Ashwin Sankar, Yoach Lacombe, Sherry Thomas, Praveen Srinivasa Varadhan, Sanchit Gandhi, and Mitesh M. Khapra. Rasmalai : Resources for Adaptive Speech Modeling in IndiAn Languages with Accents and Intonations. In *Interspeech 2025*, pages 4128–4132, 2025. doi: 10.21437/Interspeech.2025-2758. Model: https://huggingface.co/ai4bharat/indic-parler-tts.

[20] Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, et al. Fun-Audio-Chat Technical Report, 2025.

[21] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 Technical Report, 2025.

[22] Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, et al. SWIFT:A Scalable lightWeight Infrastructure for Fine-Tuning, 2024.

A The agent's system prompt

The travel agent's prompt, with the date filled in; the clinic and car service prompts follow the same pattern with their own capabilities, opening hours and transfer rules. Round 2 adds the last two sentences. The tools follow in a Hermes-style block of JSON schemas, and the agent calls them as `<tool_call>{"name": ..., "arguments": ...}</tool_call>`.

> Today is Tuesday, November 3, 2026. You are Maya, the voice assistant of [company]. Your replies are spoken aloud, so keep them short, warm and natural: no lists, no symbols, no markdown. You can check the weather, search and book flights, look up bookings and cancel them; for these, call the matching tool and never make up results. You can also answer general questions, especially about travel, from your own knowledge in two or three sentences. Before booking or cancelling, read the details back and wait for the user to say yes. If the user asks for an action you can't do, say so politely and offer what you can do. If you're not sure what the caller said or meant, ask rather than guess. If the caller asks for a person, or needs help you can't give, transfer the call.

B Further results

	Round 1	Round 2	Car
City heard right	48 / 91 / 90	56 / 68 / 83	–
General question answered	96 / 91 / 96	–	–
Introduces itself by name	0 / 95 / 90	33 / 67 / 100	0 / 100 / 50

Table 6: Other measures (%); each cell gives base / round 1 / r2. The name rows rest on 20, 6 and 2 requests.