CHARDUPLEX: BUILDING CHARACTER-CONSISTENT FULL-DUPLEX SPOKEN DIALOGUE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full-duplex speech models are moving voice interaction beyond conventional turn-taking, yet natural conversation is shaped not only by when an agent speaks, but also by how it behaves as a conversational character. We present CharDuplex, a character-driven full-duplex speech model that combines real-time spoken interaction with persona-conditioned behavior. We first adapt GLM-4-Voice to an always-on dual-stream architecture and train the model for full-duplex conversation. Then a fully automated pipeline constructs character-conditioned dialogue data from open-source character descriptions for character-conditioned supervised fine-tuning. The model is further refined with the proposed FDGym, where an LLM-simulated user dynamically interacts with the model, enabling reinforcement learning over evolving multi-turn interactions. On SpeechRole-Eval, CharDuplex achieves the highest average score among the evaluated open-source models, while remaining competitive with closed-source systems. It also demonstrates competitive general speech intelligence and strong full-duplex interaction capabilities. CharDuplex demonstrates a practical training recipe for building full-duplex voice assistants that are not only interactive, but also character-consistent.

1 INTRODUCTION

Recent advances in Full-Duplex Spoken Dialogue Language Models (SDLMs) have brought spoken dialogue closer to continuous, real-time conversation. By processing incoming speech alongside response generation, full-duplex models allow listening and speaking to occur concurrently rather than exclusively in alternating turns (Défossez et al., 2024; Wu et al., 2025; Zhang et al., 2026; Yu et al., 2025; Wu et al., 2026). However, a natural conversation also depends on responses that reflect the specific character. A speech model should convey that character's personality and perspective throughout the conversation, rather than respond uniformly as a general-purpose assistant.

Character-consistent dialogue requires responses to address the user's input while remaining compatible with the prescribed character's knowledge, personality, and background. Knowledge constrains what the model can plausibly claim, while personality and perspective shape how it interprets and answers questions (Jiang et al., 2025; Zhang et al., 2025). In multi-turn interaction, these constraints must persist across follow-up questions, corrections, and topic shifts without contradicting earlier replies. Existing full-duplex work has begun to incorporate such role conditioning. PersonaPlex, for example, uses text-based role prompts together with speech-based voice prompts (Roy et al., 2026). Our focus is on strengthening character consistency in the response content of SDLMs across evolving multi-turn full-duplex interactions, where responses must remain grounded in the prescribed character and coherent with the dialogue history.

Building such a full-duplex model requires training data that connect character descriptions to multi-turn spoken conversations. The supervision must capture how a character responds to different questions and dialogue histories, with speech inputs and target responses suitable for full-duplex training. Supervised training data, however, represent conversations whose continuations have already been determined. During live conversation, a user may question a claim, request an explanation, or ask a new follow-up question based on what the model has actually said. The model must address these reactions while remaining consistent with the same character. In full-duplex interaction, the user can also interrupt a response while the model is speaking, rather than waiting for it to finish. Therefore,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

character-consistent post-training in realistic multi-turn interaction requires response-dependent rollouts, where subsequent user behavior is conditioned on what the model actually generates.

In this paper, we present **CharDuplex**, a character-driven full-duplex speech model built on GLM-4-Voice (Zeng et al., 2024). We first establish its full-duplex capabilities through architectural adaptation and training on duplex conversational data, then construct synthetic multi-turn conversations from open-source character descriptions and convert them into speech data for character-conditioned Supervised Fine-Tuning (SFT). Finally, we develop **Full-Duplex Gym (FDGym)**, a real-time interactive training framework that connects the SDLM to a user simulator powered by a large language model (LLM). The simulator reacts to the model's incrementally available outputs and generates user speech online, forming a closed-loop, streaming conversation. Reinforcement Learning (RL) over these dynamically generated conversations further refines character-related response quality. CharDuplex substantially outperforms full-duplex SDLMs, especially PersonaPlex, across all four evaluated dimensions of SpeechRole-Eval (Jiang et al., 2025) while maintaining strong performance on Full-Duplex-Bench (Lin et al., 2025) and VoiceBench (Chen et al., 2024), demonstrating broad competence in both full-duplex interaction and general speech intelligence.

Our contributions are threefold:

- We introduce CharDuplex, a full-duplex SDLM that combines character-consistent response generation with strong general speech intelligence and full-duplex interaction capabilities.
- We develop a fully automated pipeline for constructing character-conditioned full-duplex dialogue data, with dialogue generation and verification, speech-quality control, and synchronized interaction-event construction.
- We introduce FDGym, a response-dependent interactive RL framework in which subsequent simulated user speech is generated online from the model's progressively available responses, enabling closed-loop multi-turn optimization of character consistency while preserving general conversational capabilities.

2 Related Work

2.1 Full-Duplex Spoken Dialogue Models

Full-duplex SDLMs enable concurrent listening and speaking rather than alternating between fixed conversational turns (Xie et al., 2026; Wu et al., 2026; Yan et al., 2026). Moshi models user and system speech in parallel streams (Défossez et al., 2024), while subsequent systems explore different architectural designs, including the frozen-LLM formulation of Freeze-Omni (Wang et al., 2024) and the codec-free SALMONN-omni (Yu et al., 2025). Recent work further improves the intelligence of full-duplex models through temporally structured or latent reasoning (Wu et al., 2025; 2026). These studies primarily target general interaction and reasoning capabilities; CharDuplex instead focuses on maintaining character consistency while preserving these general capabilities.

2.2 Character-Conditioned Spoken Interaction

Character modeling has recently been extended from text dialogue to spoken interaction. OmniCharacter models linguistic personality and vocal traits (Zhang et al., 2025), while SpeechRole provides large-scale training data and evaluation for spoken role-playing (Jiang et al., 2025). Most closely related to our work, PersonaPlex conditions full-duplex conversation on textual role descriptions and voice prompts (Roy et al., 2026), while SteerDuplex considers broader steerability over persona, tone, and speaking style (Tyagi et al., 2026). CharDuplex focuses on character-consistent response generation across multi-turn full-duplex interaction, supported by automated data construction and interactive post-training.

2.3 Reinforcement Learning for Full-Duplex SDLM

Reinforcement learning has increasingly been used to address interaction-level behaviors that are difficult to optimize through token-level supervised learning. SALMONN-omni applies preference optimization after supervised training to improve context-dependent barge-in and backchannel handling (Yu et al., 2025). ORISE further introduces online policy optimization with automatically

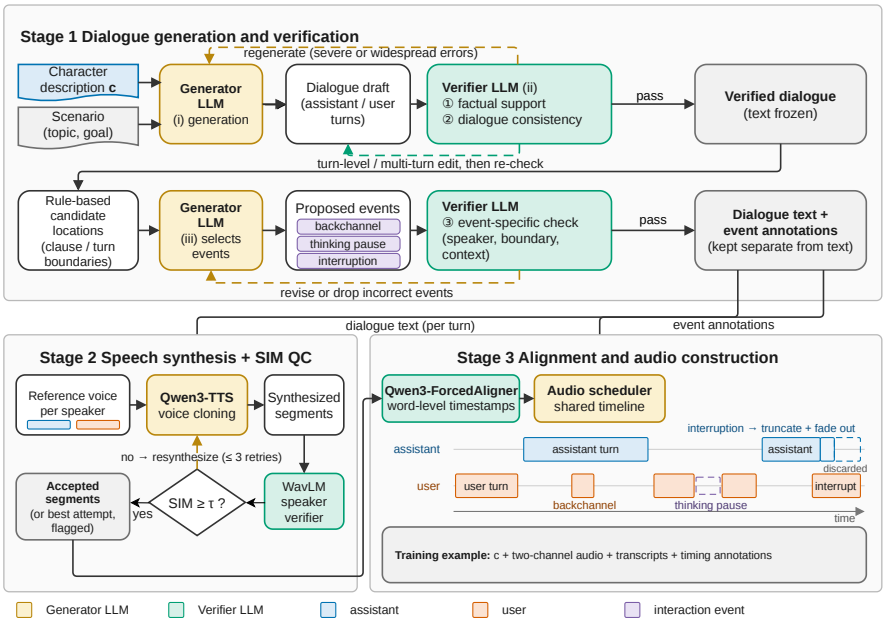

Figure 1: Overview of the automated character-conditioned data construction pipeline. Stage 1 alternates a generator and verifier to generate, validate, repair, and annotate multi-turn character dialogues. Stage 2 synthesizes speaker-conditioned speech with speaker-similarity-based quality control. Stage 3 uses forced alignment to realize pauses, backchannels, and interruptions on synchronized user and assistant audio channels.

derived speech-based rewards (Chen et al., 2025). DuplexPO decouples *when to speak* from *what to say*, optimizing selected interaction windows with factorized rewards for turn initiation, backchanneling, yielding, and participation (Li et al., 2026). These approaches substantially advance RL-based optimization of full-duplex interaction, but their rollouts are primarily constructed from prerecorded speech, or predetermined conversational contexts. However, different responses produced by the model should induce different subsequent user utterances. This distinction becomes important for multi-turn character interaction. FDGym therefore introduces an LLM-based user simulator inside the rollout loop, allowing character consistency to be optimized under conversational contexts that emerge from the model's own behavior.

3 METHOD

CharDuplex is a character-driven full-duplex SDLM built on GLM-4-Voice (Zeng et al., 2024). Its construction comprises full-duplex adaptation and supervised training, character-conditioned SFT, and interactive RL with FDGym. We first describe the automated character-conditioned data construction pipeline, then introduce the model architecture, followed by the SFT and RL objectives.

3.1 CHARACTER-CONDITIONED DATA CONSTRUCTION

Character-conditioned speech data are constructed using character descriptions from the training splits of open-sourced datasets (Zhang et al., 2025; Jiang et al., 2025). The pipeline integrates an automated generator–verifier process for character-conditioned text dialogue generation, speech synthesis with speaker-similarity-based quality control, and forced alignment for constructing synchronized two-channel dialogue audio. The whole pipeline is shown in Figure 1.

Stage 1: Automated character-conditioned dialogue generation and verification. Stage 1 follows an automated generator–verifier loop that alternates generation, verification, and targeted correction. Given a character description and a sampled conversational scenario, the generator first

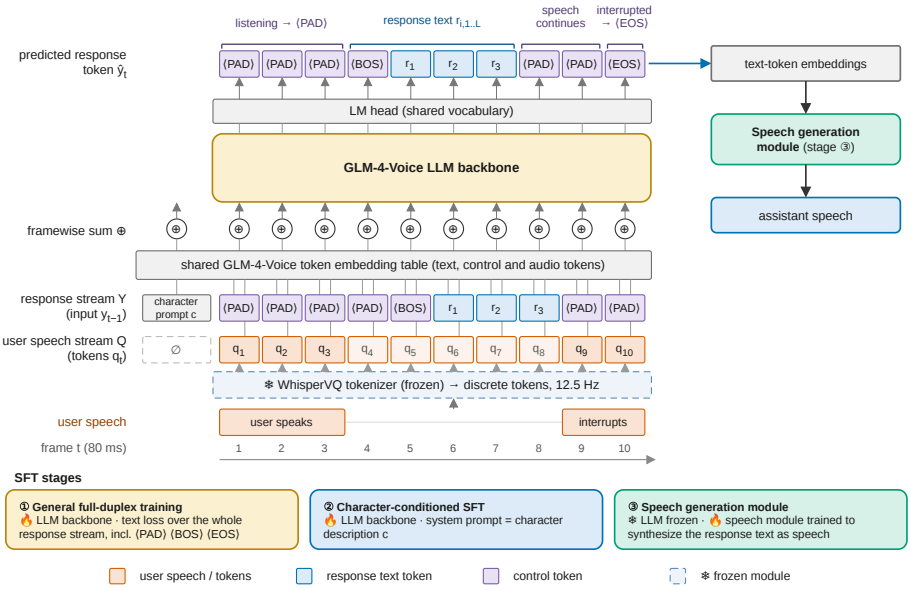

Figure 2: Overview of the CharDuplex architecture and supervised training. GLM-4-Voice is reformulated into synchronous, always-on user-speech and response-text streams that share the token embedding and causal backbone. General full-duplex training is followed by character-conditioned SFT and speech-generation training.

produces a multi-turn dialogue conditioned on the prescribed character. The verifier then checks the dialogue with task-specific prompts for factual grounding with respect to the character description and dialogue history, as well as logical consistency across turns. Local defects trigger turn- or span-level correction followed by re-verification, whereas broader failures trigger dialogue regeneration. Once the dialogue content passes verification, the generator then identifies the locations for events including backchannels, user thinking pauses, and interruptions, and the verifier checks the resulting annotations before they are passed to the subsequent speech synthesis stages. Detailed generation, verification, repair, and data construction prompts are provided in Appendix A.

Stage 2: Speech synthesis. We use Qwen3-TTS to synthesize speech from each speaker's dialogue text (Hu et al., 2026), using a distinct reference recording for each speaker throughout the conversation. After that, a WavLM-based speaker verifier (Chen et al., 2022) from SEED-TTS-Eval (Anastassiou et al., 2024) measures similarity between synthesized and reference audio. Segments below a threshold of 0.5 are resynthesized within a fixed retry budget; if none passes, the highest-scoring attempt is retained and marked.

Stage 3: Speech alignment and audio construction. We use Qwen3-ForcedAligner to align each synthesized segment with its text to obtain word-level timestamps (Shi et al., 2026). An audio scheduler uses these timestamps to place speech in separate user and assistant channels on a shared timeline. It inserts silence for user thinking pauses and overlays backchannels on ongoing speech. User interruptions start at annotated word boundaries; after a short overlap, the assistant audio is truncated and faded out, and the discarded continuation is removed from its transcript. Each training example retains the character description, two-channel audio, transcripts, and temporal annotations. For SFT, the description supplies the system prompt, the user audio supplies the speech input, and the assistant transcript and speech boundaries determine the target response stream.

3.2 Full-Duplex Model Architecture

Concurrent speech and response streams. We reformulate GLM-4-Voice's serialized conversation stream into two synchronous, always-on streams: a user speech stream $Q = (q_1, \ldots, q_T)$ represented by frozen WhisperVQ tokens at 12.5 Hz, and a response stream $Y = (y_1, \ldots, y_T)$ containing text

and control tokens on the same frame timeline. Their embeddings are combined framewise and fed to the causal backbone. The model architecture is shown in Figure 2. Given the character description c as the system prompt, response generation is formulated as

$$p_\theta(Y \mid Q, c) = \prod_{t=1}^{T} p_\theta(y_t \mid c, q_{<t}, y_{<t}). \tag{1}$$

The response stream is autoregressive, while the user stream is continuously supplied by incoming audio.

Turn-level text–speech alignment. Response text and speech are aligned at the turn level rather than requiring individual text tokens to coincide with spoken words. The response stream emits <PAD> tokens while listening, followed by a <BOS> token and the response text. After text generation finishes, the model continues to predict <PAD> tokens until the corresponding speech ends, followed by an <EOS> token. For a response containing L_i text tokens, the stream around turn i takes the form

$$\begin{aligned} Y^{(i)} = \big[& \underbrace{\langle \mathrm{PAD} \rangle, \ldots, \langle \mathrm{PAD} \rangle}_{\text{before the response}}, \langle \mathrm{BOS} \rangle, r_{i,1}, \ldots, r_{i,L_i}, \\ & \underbrace{\langle \mathrm{PAD} \rangle, \ldots, \langle \mathrm{PAD} \rangle}_{\text{speech continues}}, \langle \mathrm{EOS} \rangle \big], \end{aligned} \tag{2}$$

where $r_{i,j}$ denotes the j-th text token of response i.

When the user interrupts an ongoing response, the model emits $\langle \mathrm{EOS} \rangle$ to terminate the current turn and stop its speech output, without waiting for the remaining text or speech to finish. The response stream then returns to PAD prediction, while the user speech stream remains active. This construction maintains continuous dual-stream processing while allowing a spoken turn to end either through normal completion or interruption.

3.3 Supervised Fine-Tuning

Supervised training comprises three stages: general full-duplex training, character-conditioned SFT, and speech-generation-module training. In the first two stages, the LLM backbone is optimized over the complete response stream using next-token prediction (Brown et al., 2020):

$$\mathcal{L}_{\text{text}} = -\sum_{t=1}^{T} \log p_\theta \left(y_t \mid c, q_{\leq t}, y_{<t} \right), \tag{3}$$

where c denotes the system-prompt context when present. The WhisperVQ tokenizer remains frozen.

The first stage trains the adapted GLM-4-Voice on general full-duplex conversations to establish response generation under the dual-stream representation and turn-level alignment. Data statistics, the generation toolkit, and configurations are provided in Section 4 and Appendix B. Starting from this model, the second stage continues training on the character-conditioned conversations constructed in Section 3.1, where the character description is provided as the system prompt c. The same objective is retained, so the response is conditioned jointly on the character description, incoming user speech, and preceding response tokens.

Finally, the LLM backbone is frozen and the speech-generation module is trained separately. An autoregressive Transformer predicts speech codec tokens conditioned on the response text, and the predicted tokens are converted to waveforms using the streaming flow-matching decoder of CosyVoice2 (Du et al., 2024).

3.4 Real-Time Interactive Reinforcement Learning

RL rollouts with prerecorded user continuations cannot capture how a model's replies affect subsequent user input (Li et al., 2026; Chen et al., 2025). We develop FDGym, a streaming training environment that couples CharDuplex with an LLM-based user simulator and a dialogue evaluator. The pipeline is shown in Figure 3. The simulator generates user speech in response to the model's unfolding replies, while the evaluator provides character and interaction feedback. Starting from the SFT model, we alternate dynamic dialogue sampling and policy updates.

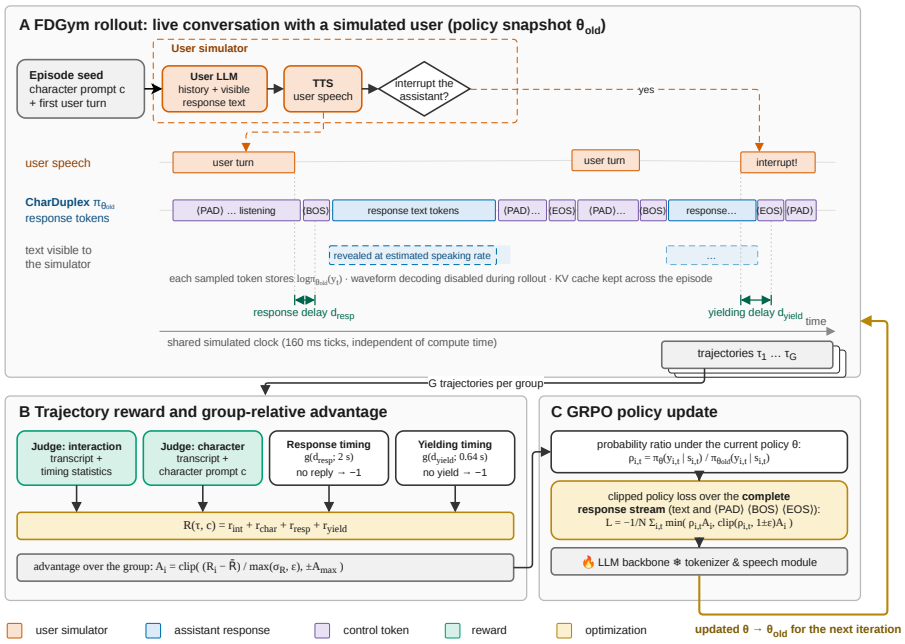

Figure 3: Overview of FDGym for real-time interactive RL. (A) CharDuplex interacts with an LLM-based simulated user whose subsequent speech depends on the model's progressively available responses. (B) Each multi-turn trajectory receives character, interaction, and timing rewards, from which a group-relative advantage is computed. (C) GRPO updates the response policy using the sampled trajectory, and the updated model is used for the next rollout iteration.

Full-duplex rollout generation. Each rollout starts with a character prompt and an initial user turn. FDGym represents dialogue time using fixed-duration simulation ticks, with each tick corresponding to 160 ms (Ray et al., 2026). A tick serves as the basic synchronization step of the full-duplex interaction: the user side supplies the speech available within the current interval, while CharDuplex processes the accumulated incoming audio and advances its response stream. The resulting partial interaction state is then carried into the next tick, allowing user speech and model responses to evolve incrementally on the same simulated timeline.

The user simulator generates subsequent utterances from the dialogue history and the model's progressively available response, with Qwen3-TTS synthesizing the corresponding user speech. During an ongoing model response, only the portion that would have become audible by the current tick is exposed to the simulator, according to a speaking rate estimated from the SFT data. The simulator can therefore react to the response as it unfolds, including deciding to interrupt it. During rollout, model-side waveform decoding is disabled and the KV cache is retained throughout the rollout. The resulting trajectory records user audio and text, model responses, interaction timestamps, and token log probabilities; CharDuplex is conditioned on user audio, whereas the LLM judge evaluates the textual dialogue.

Reward computation. Each completed trajectory under character condition c receives two LLM-judged rewards and two deterministic timing rewards. The interaction reward $r_{\text{int}} \in [0, 1]$ evaluates conversational coherence, responsiveness, and relevance, while the character reward $r_{\text{char}} \in [0, 1]$ evaluates adherence to the prescribed identity, style, and role constraints. Both are computed from the textual dialogue; the character judge additionally receives the character description, while the interaction judge receives timing statistics. The corresponding judge prompts are provided in Appendix D.

The timing rewards measure response initiation and yielding under interruption. Specifically, d_{resp} denotes the delay from the end of a normal user turn to the onset of the model response, and d_{yield} denotes the delay from a user interruption to the model becoming silent. For successful events, both

Table 1: Results on SpeechRole-Eval. IA: Instruction Adherence; CC: Conversational Coherence; PeC: Personality Consistency; KC: Knowledge Consistency. Average is computed over the four metrics. Bold values indicate the best result within each model group.

Model	IA ↑	CC ↑	PeC ↑	KC ↑	Avg. ↑
Closed-source models					
Grok Voice Think Fast 2.0 [1]	0.727	0.587	0.711	0.792	0.704
Gemini 3.1 Flash Live Preview [2]	**0.798**	**0.700**	**0.819**	**0.853**	**0.793**
Open-source models					
Moshi (Défossez et al., 2024)	0.456	0.445	0.233	0.261	0.349
PersonaPlex (Roy et al., 2026)	0.750	0.725	0.552	0.578	0.651
Freeze-Omni (Wang et al., 2024)	0.781	0.798	0.595	0.623	0.699
MiniCPM-o4.5 (Cui et al., 2026)	**0.815**	0.747	0.655	0.727	0.736
CharDuplex-SFT	0.807	0.800	0.729	0.770	0.777
CharDuplex	**0.815**	**0.805**	**0.738**	**0.781**	**0.785**

[1] https://ai.google.dev/gemini-api/docs/models/gemini-3.1-flash-live-preview

[2] https://x.ai/news/grok-voice-think-fast-2

rewards use

$$g(d; D) = \max\left(0, 1 - \frac{d}{D}\right), \quad D_{\text{resp}} = 2\,\text{s}, \quad D_{\text{yield}} = 0.64\,\text{s}. \tag{4}$$

Failure to respond before the next user turn or the end of the rollout receives -1, as does failure to yield during an interrupting user turn. Each timing reward is averaged over its eligible events and is zero when no eligible event occurs. The final trajectory reward is

$$R(\tau, c) = r_{\text{int}} + r_{\text{char}} + r_{\text{resp}} + r_{\text{yield}}. \tag{5}$$

Policy optimization. After full-duplex rollout and reward calculation, we use group relative policy optimization (GRPO) to train CharDuplex (Shao et al., 2024). The detailed algorithm of GRPO is shown in Appendix E.

4 Experimental Setup

4.1 Data and Training Details

The pipeline in Section 3.1 is applied to character descriptions from OmniCharacter and the SpeechRole training set (Zhang et al., 2025; Jiang et al., 2025). Qwen3.5-397B-A17B[1] serves as the generator, while GPT-5.6-Luna[2] serves as the verifier. Qwen3-TTS[3] and Qwen3-ForcedAligner[4] are used for speech synthesis and word-level alignment, respectively. The resulting corpus contains 70 characters and 40,608 dialogues, totaling 756.04 hours of speech. Character descriptions used for training are disjoint from those used for evaluation.

CharDuplex is initialized from GLM-4-Voice (Zeng et al., 2024) and trained with general full-duplex data before character-conditioned SFT. The two stages use learning rates of 5×10^{-4} and 5×10^{-6}, respectively. RL starts from CharDuplex-SFT. Qwen-235B[5] is used for both user simulation and LLM-based reward computation, and Qwen3-TTS synthesizes user speech. Each policy iteration samples eight trajectories for each of four groups, yielding 32 trajectories in total. GRPO uses a learning rate of 2×10^{-6}. Detailed rollout, sampling, and optimization configurations are provided in Appendix C.

4.2 Benchmarks and Evaluation

Character-related capabilities. SpeechRole-Eval (Jiang et al., 2025) assesses interaction quality and role-playing fidelity. Four metrics are reported: Instruction Adherence (IA), Conversational

[1] https://huggingface.co/Qwen/Qwen3.5-397B-A17B

[2] https://developers.openai.com/api/docs/models/gpt-5.6-luna

[3] https://huggingface.co/Qwen/Qwen3-TTS-12Hz-1.7B-CustomVoice

[4] https://huggingface.co/Qwen/Qwen3-ForcedAligner-0.6B

[5] https://huggingface.co/Qwen/Qwen3-235B-A22B

Table 2: General speech intelligence results. FD indicates full-duplex support. LlamaQ, WebQ, OBQA, AdvB., AlpacaE, ComE, and WildV denote LLaMA-Questions, WebQuestions, OpenBookQA, AdvBench, AlpacaEval, CommonEval, and WildVoice, respectively. Published baseline results are collected from prior works(Wu et al., 2026; Tyagi et al., 2026). "–" denotes an unreported result. Bold marks the best score separately among full-duplex or non-full-duplex models.

		Accuracy (%) ↑						Refusal (%)↑	GPT-Score (1–5) ↑		
Model	FD	LlamaQ	WebQ	OBQA	MMSU	SD-QA	BBH	AdvB.	AlpacaE	ComE	WildV
GLM-4-Voice	✗	65.7	37.0	53.4	39.8	37.0	52.8	88.08	3.97	3.42	3.18
Qwen2-Audio	✗	69.7	**45.2**	49.5	35.7	35.7	54.7	96.73	3.74	3.43	3.01
Kimi-Audio	✗	68.3	37.3	**83.5**	**62.2**	**63.1**	**69.7**	**100.00**	**4.46**	3.97	**4.20**
Baichuan-Audio	✗	**74.0**	40.7	71.7	53.2	45.8	54.8	99.42	4.41	**4.08**	3.92
Moshi	✓	54.5	22.1	25.9	24.0	15.6	47.4	44.23	2.01	1.60	1.30
Freeze-Omni	✓	56.2	27.9	31.0	28.1	**53.5**	50.7	**97.30**	**4.03**	**3.46**	**3.15**
SALMONN-omni	✓	73.6	**43.7**	–	30.0	–	–	–	3.22	–	–
PersonaPlex	✓	46.2	21.5	27.2	27.8	23.4	47.9	12.50	2.92	2.01	2.05
SteerDuplex	✓	-	-	-	-	26.88	49.9	99.3	2.28	2.21	1.74
CharDuplex-SFT	✓	74.3	40.8	**63.1**	**44.2**	47.4	50.0	95.8	3.63	3.09	2.30
CharDuplex	✓	**74.7**	40.5	62.4	44.0	49.2	**51.2**	95.6	3.63	3.09	2.29

Coherence (CC), Personality Consistency (PeC), and Knowledge Consistency (KC). All four metrics are evaluated in the prescribed role-playing setting. IA assesses adherence to the role-playing instructions, including remaining in character and avoiding out-of-role explanations. CC measures logical consistency with the preceding dialogue, while PeC and KC assess consistency with the character's personality traits and established knowledge, respectively. Each evaluation dialogue is conditioned on its associated character description.

General speech intelligence and conversational behavior. General speech intelligence is evaluated on AlpacaEval, CommonEval, WildVoice, OpenBookQA, MMSU, SD-QA, BBH, and AdvBench from VoiceBench (Chen et al., 2024), as well as LLaMA-Questions (Nachmani et al., 2023) and WebQuestions (Berant et al., 2013). These datasets cover spoken question answering, reasoning, open-ended response quality, and safety. Full-Duplex-Bench (Lin et al., 2025) further evaluates turn-taking and interruption handling in full-duplex interaction. Throughout the experiments, **CharDuplex-SFT** refers to the character-conditioned SFT model, and **CharDuplex** refers to the final model after FDGym-based RL.

5 Results

5.1 Character-Related Capabilities

Table 1 compares CharDuplex with representative open- and closed-source full-duplex SDLMs on SpeechRole-Eval. CharDuplex reaches an average score of 0.785, improving over PersonaPlex by 0.134 absolute and outperforming it across all four dimensions. The largest gains occur in Personality Consistency (+0.186) and Knowledge Consistency (+0.203), indicating that the improvement extends beyond instruction following to the two dimensions most directly associated with maintaining character identity and background. Among the evaluated open-source models, CharDuplex achieves the highest average score, leads on Conversational Coherence, Personality Consistency, and Knowledge Consistency, and ties for the best Instruction Adherence. These results indicate that the improvement is not limited to following role-playing instructions, but extends to maintaining the character's personality and knowledge across the conversation.

Compared with the closed-source systems, CharDuplex remains competitive overall. It exceeds Grok on four dimensions and Gemini on Instruction Adherence and Conversational Coherence, while Gemini retains stronger Personality Consistency and Knowledge Consistency. More importantly, CharDuplex-SFT already exhibits strong character-consistent behavior before RL, supporting the effectiveness of the character-conditioned data and SFT stage. FDGym subsequently improves all four SpeechRole-Eval dimensions, showing that interactive post-training further refines character consistency on top of an already strong supervised model.

Table 3: Results on Full-Duplex-Bench (Lin et al., 2025). TOR denotes turn-over rate, ICC denotes interjectional conversational cue, and JSD denotes Jensen–Shannon divergence. ↑ and ↓ indicate that higher and lower values are better, respectively. Bold values indicate the best result among open-source models, including CharDuplex.

Model	Pause Handling		Backchannel			Smooth Turn Taking		User Interruption		
	Synthetic TOR ↓	Candor TOR ↓	TOR ↓	ICC Freq ↑	JSD ↓	TOR ↑	Candor Latency ↓	TOR ↑	Synthetic GPT-4o ↑	Latency ↓
dGSLM	0.934	0.935	0.691	0.015	0.934	**0.975**	0.352	0.917	0.201	2.531
Moshi	0.985	0.980	1.000	0.001	0.957	0.941	0.265	**1.000**	0.765	**0.257**
Freeze-Omni	**0.642**	**0.481**	**0.636**	0.001	0.997	0.336	0.953	0.867	3.615	1.409
Gemini Live[1]	0.255	0.310	0.091	0.012	0.896	0.655	1.301	0.891	3.376	1.183
CharDuplex-SFT	0.934	0.815	**0.636**	0.054	0.890	0.840	**0.133**	0.990	**4.081**	0.558
CharDuplex	0.927	0.815	0.709	**0.055**	**0.889**	0.857	0.136	0.990	4.071	0.565

[1] https://ai.google.dev/gemini-api/docs/live

5.2 Speech Intelligence

We further evaluate whether the improved character consistency of CharDuplex comes at the expense of general speech intelligence. As shown in Table 2, CharDuplex compares favorably with established full-duplex models across the evaluated benchmarks and exceeds its GLM-4-Voice foundation on all five factual and multiple-choice QA tasks. These tasks do not rely on character descriptions, indicating that the resulting model retains general knowledge and response capability beyond role-playing.

Comparing CharDuplex-SFT with CharDuplex, FDGym improves all four SpeechRole-Eval dimensions while the general benchmark results remain broadly stable. Across the ten general speech-intelligence metrics, eight change by no more than 1.11% relative after FDGym, while the two larger changes are improvements on SD-QA (+3.80%) and BBH (+2.40%). This pattern indicates that the character-consistency gains from interactive post-training are obtained without a systematic loss of general speech capability.

5.3 Full-duplex interaction capability

We evaluate the interaction performance of CharDuplex on Full-Duplex-Bench (Lin et al., 2025). As shown in Table 3, CharDuplex exhibits strong full-duplex interaction capabilities across multiple dimensions. In particular, it performs strongly on backchannel behavior and user interruption handling, achieving competitive backchannel statistics and a high interruption-quality score while maintaining a high interruption TOR. The results indicate that the character-oriented training pipeline does not come at the cost of the interaction behaviors required for full-duplex conversation.

The comparison between CharDuplex-SFT and CharDuplex further shows that FDGym-based RL largely preserves the full-duplex interaction profile established during supervised training. Nine of the ten Full-Duplex-Bench metrics change by at most 2.26% relative after FDGym. Together, these results suggest that interactive post-training can strengthen character-consistent response generation without degrading the model's general intelligence and full-duplex interaction capabilities. A consolidated comparison of CharDuplex-SFT and CharDuplex across character consistency, general speech intelligence, and full-duplex interaction is provided in Appendix F. Further ablation studies and analysis for FDGym and the data-construction pipeline are provided in Appendix G.

6 Conclusion

This work presents CharDuplex, a character-driven full-duplex SDLM that combines character-consistent response generation with general speech intelligence and full-duplex interaction capabilities. CharDuplex integrates an always-on dual-stream architecture, automated character-conditioned data construction, and FDGym, which enables response-dependent multi-turn RL with a simulated user. CharDuplex achieves the highest average SpeechRole-Eval score among the evaluated open-source models while remaining competitive with closed-source systems. FDGym further improves all four character-related metrics, while general speech intelligence and full-duplex interaction performance remain broadly stable. These results show that character consistency can be strengthened without sacrificing the general capabilities required for full-duplex spoken interaction.

AI USAGE STATEMENT

Generative AI tools were used to assist with language polishing, grammatical correction, and improving the clarity of the manuscript. AI models also served as components of the research methodology, including character-conditioned dialogue generation and verification of generated data, user simulation and reward computation during reinforcement learning, and automated evaluation and metric computation where applicable. The specific models, prompts, and procedures used for these purposes are described in the corresponding sections of the paper. All research ideas, methodological design, experiments, analysis, and conclusions were developed and verified by the authors. The authors reviewed all AI-assisted content and take full responsibility for the manuscript and reported results.

REFERENCES

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

Artificial Analysis. Nvidia has released nemotron-3 voicechat. `https://www.linkedin.com/posts/artificial-analysis_nvidia-has-released-nemotron-3-voicechat-activity-7439407814835593216-4X08/`, 2026.

Evelina Bakhturina, Vitaly Lavrukhin, Boris Ginsburg, and Yang Zhang. Hi-fi multi-speaker english tts dataset. *arXiv preprint arXiv:2104.01497*, 2021.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1533–1544, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chen Chen, Ke Hu, Chao-Han Huck Yang, Ankita Pasad, Edresson Casanova, Weiqing Wang, Szu-Wei Fu, Jason Li, Zhehuai Chen, Jagadeesh Balam, et al. Reinforcement learning enhanced full-duplex spoken dialogue language models for conversational interactions. In *Second Conference on Language Modeling*, 2025.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505–1518, 2022.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T Tan, and Haizhou Li. Voicebench: Benchmarking llm-based voice assistants. *arXiv preprint arXiv:2410.17196*, 2024.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, et al. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*, 2024.

Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *Proceedings of the 21st ACM international conference on Multimedia*, pp. 411–412, 2013.

Hangrui Hu, Xinfa Zhu, Ting He, Dake Guo, Bin Zhang, Xiong Wang, Zhifang Guo, Ziyue Jiang, Hongkun Hao, Zishan Guo, et al. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, 2026.

Changhao Jiang, Jiajun Sun, Yifei Cao, Jiabao Zhuang, Xinmeng Che, Hui Li, Xiaoran Fan, Ming Zhang, Junjie Ye, Shihan Dou, et al. SpeechRole: A large-scale dataset and benchmark for evaluating speech role-playing agents. *arXiv preprint arXiv:2508.02013*, 2025.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*, 2019.

Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe. Yodas: Youtube-oriented dataset for audio and speech. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1–8. IEEE, 2023.

Yuxin Li, Donghang Wu, Guan-Ting Lin, Hung-yi Lee, Chengwei Qin, Zhehuai Chen, and Chen Chen. Decoupling conversational dynamics in full-duplex spoken models through reinforcement learning. *arXiv preprint arXiv:2607.07148*, 2026.

Guan-Ting Lin, Jiachen Lian, Tingle Li, Qirui Wang, Gopala Anumanchipalli, Alexander H Liu, and Hung-yi Lee. Full-duplex-bench: A benchmark to evaluate full-duplex spoken dialogue models on turn-taking capabilities. *arXiv preprint arXiv:2503.04721*, 2025.

Eliya Nachmani, Alon Levkovitch, Roy Hirsch, Julian Salazar, Chulayuth Asawaroengchai, Soroosh Mariooryad, Ehud Rivlin, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich. Spoken question answering and speech continuation using spectrogram-powered llm. *arXiv preprint arXiv:2305.15255*, 2023.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. Speechbrain: A general-purpose speech toolkit, 2021. URL `https://arxiv.org/abs/2106.04624`.

Soham Ray, Keshav Dhandhania, Victor Barres, and Karthik Narasimhan. τ-voice: Benchmarking full-duplex voice agents on real-world domains. *arXiv preprint arXiv:2603.13686*, 2026.

Rajarshi Roy, Jonathan Raiman, Sang-gil Lee, Teodor-Dumitru Ene, Robert Kirby, Sungwon Kim, Jaehyeon Kim, and Bryan Catanzaro. Personaplex: Voice and role control for full duplex conversational speech models. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16137–16141. IEEE, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

Utkarsh Tyagi, Ramaneswaran Selvakumar, Advait Gosai, Sonal Kumar, Nikhil Barhate, Isabell Sagar, Steven Li, Miheer Bavare, Daniel Quigley, Fabiola Tapia Carrillo, et al. Steerduplex: Steerable duplex speech dialogue models. *arXiv preprint arXiv:2609.12623*, 2026.

Xiong Wang, Yangze Li, Chaoyou Fu, Yunhang Shen, Lei Xie, Ke Li, Xing Sun, and Long Ma. Freeze-omni: A smart and low latency speech-to-speech dialogue model with frozen llm. *arXiv preprint arXiv:2411.00774*, 2024.

Donghang Wu, Haoyang Zhang, Chen Chen, Tianyu Zhang, Fei Tian, Xuerui Yang, Gang Yu, Hexin Liu, Nana Hou, Yuchen Hu, et al. Chronological thinking in full-duplex spoken dialogue language models. *arXiv preprint arXiv:2510.05150*, 2025.

Donghang Wu, Tianyu Zhang, Yuxin Li, Hexin Liu, Chen Chen, Eng Siong Chng, and Yoshua Bengio. The silent thought: Modeling internal cognition in full-duplex spoken dialogue models via latent reasoning. *arXiv preprint arXiv:2603.17837*, 2026.

Zhifei Xie, Jiaqi Lang, Ze An, Yifan Zhao, Dongchao Yang, Kai Li, Ziyang Ma, Mingbao Lin, Chunyan Miao, and Shuicheng Yan. Voicemem: Streaming dual-brain memory for real-time interaction. *arXiv preprint arXiv:2608.26005*, 2026.

Ruiqi Yan, Wenxi Chen, Zhanxun Liu, Ziyang Ma, Haopeng Lin, Hanlin Wen, Hanke Xie, Jun Wu, Yuzhe Liang, Yuxiang Zhao, et al. Soulx-duplug: Plug-and-play streaming state prediction module for realtime full-duplex speech conversation. *arXiv preprint arXiv:2603.14877*, 2026.

Wenyi Yu, Siyin Wang, Xiaoyu Yang, Xianzhao Chen, Xiaohai Tian, Jun Zhang, Guangzhi Sun, Lu Lu, Yuxuan Wang, and Chao Zhang. Salmonn-omni: A standalone speech llm without codec injection for full-duplex conversation. *arXiv preprint arXiv:2505.17060*, 2025.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*, 2024.

Haonan Zhang, Run Luo, Xiong Liu, Yuchuan Wu, Ting-En Lin, Pengpeng Zeng, Qiang Qu, Feiteng Fang, Min Yang, Lianli Gao, et al. Omnicharacter: Towards immersive role-playing agents with seamless speech-language personality interaction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26318–26331, 2025.

Haoyang Zhang, Jun Chen, Donghang Wu, Yuxin Li, Yuxin Zhang, Xiangyu Tony Zhang, Che Liu, Qingjian Lin, Yizhou Peng, Hexin Liu, et al. Duplexsla: A full-duplex spoken language model with synchronized speech, language, and action. *arXiv preprint arXiv:2605.20755*, 2026.

Prompt Template for Character-Conditioned Dialogue Generation

[Task]
Create one realistic and useful multi-turn spoken conversation for full-duplex dialogue training. Maintain one causally connected situation and one primary conversational goal. Every turn should respond naturally to the preceding dialogue and provide a plausible reason for the next turn.

[Character and Scenario]
Character / role description: {Character_Description}
Scenario: {Scenario}
Conversational goal: {Goal}

[Character Grounding]
Construct a private `system_instruction` containing the character's relevant personality, background, and role-specific knowledge. Any private, current, or scenario-specific fact later stated by the character must already be supported by this instruction or by information previously provided by the user. User-specific facts must originate from the user.

[Dialogue Requirements]
- Keep the conversation internally coherent and grounded in the available information.
- Do not invent unsupported prices, schedules, policies, actions, observations, or other private/current facts.
- If required information is unavailable, ask for it or express uncertainty.
- Conclusions and calculations must follow from information already established in the dialogue.
- Do not include interaction-event annotations, timing markers, or stage directions in the dialogue text.

[Output Format]
Return only one JSON object:

```
{"system_instruction": "...",
 "turns": [
   {"speaker": "s1", "text": "..."},
   {"speaker": "s0", "text": "..."}, ... ]}
```

Use only `s0` for the character and `s1` for the user. Keep speaker turns alternating and end with `s0`.

Figure 4: Prompt template used by the generator to construct character-conditioned multi-turn dialogues. The character description and sampled scenario define the conversational context, while explicit grounding constraints require role-specific claims to be supported by the character background or preceding dialogue.

A CHARACTER-CONDITIONED DATA CONSTRUCTION

We propose an *automated generator–verifier pipeline* for textual dialogue generation that alternates between generation and verification. A *generator* produces dialogue text and event annotations, while a *verifier* checks and corrects these outputs using task-specific prompts. The pipeline comprises the following three steps.

(i) Dialogue generation. The generator receives a character description and a sampled conversational scenario. The description specifies the character's personality and background; the scenario supplies a topic and conversational goal. The generation prompt requires replies to reflect the prescribed character and respond to the preceding dialogue, which is shown in Figure 4. This conditioning associates character traits with responses in specific conversational situations. The character description is retained as the system prompt and excluded from the text passed to TTS.

(ii) Content verification and correction. The verifier reviews the generated dialogue using two separate prompts. The first checks factual support: role-specific claims must be supported by the character description or preceding dialogue, and facts about the user must not be attributed to the character. The corresponding prompt is shown in Figure 5. The second checks dialogue consistency: conclusions must follow from information already available, and subsequent replies must respect earlier statements and corrections, shown in Figure 6. The verifier identifies erroneous turns and specifies corrections. An isolated error receives a turn-level edit; if the edit changes information used in subsequent replies, those replies are revised together. The verifier then checks the corrected dialogue again for unresolved

Prompt Template for Factual-Grounding Verification

[System Instruction]
You are the factual-grounding verifier for a synthetic spoken conversation. Audit the complete dialogue for truth, grounding, fact ownership, authority, workflow state, factual consistency, and safety.

[Inputs]
Character system instruction: {System_Instruction}
Dialogue: {Dialogue}

[Verification Criteria]
- Trace every private or current claim made by the character to either the system instruction or an earlier user turn.
- User-specific facts must originate from the user.
- Later dialogue cannot retroactively justify an earlier unsupported claim.
- Reject unsupported observations, completed actions, account states, policies, schedules, or other specific facts.
- Check factual contradictions, authority, safety, and dialogue-state consistency.
- Verify exact numerical and temporal claims when applicable.

[Repair Policy]
Choose the narrowest valid repair: `EDIT_SYSTEM`, `ADD_SYSTEM_FACT`, `EDIT_TURN`, `EDIT_SPAN`, or `REGENERATE`. Use regeneration only when the defect is non-local and cannot be safely corrected with a bounded edit.

[Output]
Return only a JSON object containing the audited turns and all detected issues, together with the required repair action and correction.

Figure 5: Prompt template used by the factual-grounding verifier. The verifier checks whether character-specific claims are supported by the prescribed background or previously established dialogue context and returns localized repairs when possible.

errors or contradictions introduced by the edits. Further errors that invalidate the scenario or affect much of the conversation trigger regeneration of the whole conversation instead of local editing.

(iii) Interaction-event annotation. Once the dialogue passes verification, a rule-based procedure identifies candidate locations for backchannels, user thinking pauses, and interruptions. The generator LLM now acts as the event annotator, selecting suitable locations from these candidates according to the dialogue context. The prompt for annotation is shown in Figure 7. The verifier checks the proposed events with an event-specific prompt, and incorrect annotations are revised. The prompt for the annotation verifier is shown in Figure 8. Each annotation records the speakers and text boundaries needed for Stage 3. Event annotations remain separate from the dialogue text, so the audio construction procedure controls pauses and overlaps without passing timing instructions to TTS.

B Data details

B.1 General Full-Duplex SFT Data

For general full-duplex training, we use the same corpus and data construction setup as (Wu et al., 2026; Artificial Analysis, 2026). The corpus contains three components: approximately 530K hours of speech-continuation data, 70K hours of instruction-following QA data, and 20K hours of ASR-QA data.

The speech-continuation data are constructed by converting continuous text passages into alternating user–assistant turns and synthesizing the two speakers separately. The instruction-following QA data contain both single-turn and multi-turn conversations generated from diverse textual contexts, including Wikipedia passages[6], while the ASR-QA data are built from real speech in NeMo ASRSET by generating questions grounded in the corresponding transcriptions. Across the synthetic subsets,

[6]https://huggingface.co/datasets/wikimedia/wikipedia

Prompt Template for Dialogue-Consistency and Reasoning Verification

[System Instruction]
You are the reasoning and dialogue-consistency verifier for a synthetic spoken conversation. Read the dialogue strictly from left to right. Later information must never be used to justify an earlier response.

[Inputs]
Character system instruction: {System_Instruction}
Dialogue: {Dialogue}
Scenario and task specification: {Scenario_Contract}

[Verification Criteria]
Capability: Verify that the intended conversational capability is actually realized and contributes to the active goal.
Quantitative reasoning: Check numbers, dates, times, rates, percentages, units, and comparisons against their actual operands.
Logic and causality: Conclusions must follow from information already available in the dialogue; reject future-turn leakage and hidden premises.
Task completion: Distinguish proposed, pending, confirmed, and completed actions and verify that the conversational goal is actually resolved.
Progression: Reject repeated spans that merely restate previous content without adding a new constraint, correction, decision, or consequence.

[Repair Policy]
Apply the narrowest safe correction using `EDIT_TURN`, `EDIT_SPAN`, `ADD_SYSTEM_FACT`, `EDIT_SYSTEM`, or `REGENERATE`.

[Output]
Return only a compact JSON object containing the verification result and all material issues.

Figure 6: Prompt template used by the second semantic verifier to assess cross-turn reasoning, chronology, task completion, and dialogue progression. It complements the factual-grounding verifier by focusing on dependencies and consistency across the multi-turn interaction.

text generation uses GPT-OSS-120B[7], Qwen2.5-72B-Instruct[8], and Llama-3.1-70B-Instruct[9], and speech synthesis uses Chatterbox[10], Magpie-TTS[11], and MoonCast[12]. Voice references are sampled from LibriTTS (Zen et al., 2019), YODAS (Li et al., 2023), and HiFi-TTS (Bakhturina et al., 2021), yielding a prompt pool with over 100K segments from more than 20K speakers.

During training, SpecAugment is applied to user speech features, and background noise from Freesound (Font et al., 2013) and MUSAN (Snyder et al., 2015) is mixed into the user stream for additional acoustic diversity. Further details of the original construction procedure are provided in Wu et al. (2026).

Character-conditioned data. The pipeline in Section 3.1 is applied to character descriptions from OmniCharacter and the SpeechRole training set (Zhang et al., 2025; Jiang et al., 2025).Qwen3.5-397B-A17B[13] serves as the generator for dialogue generation and event annotation, while GPT-5.6-Luna[14] serves as the verifier for content verification, correction, and event validation. Qwen3-TTS[15] and Qwen3-ForcedAligner[16] are used for speech synthesis and word-level alignment, respectively. The resulting corpus contains 70 characters and 40,608 dialogues, totaling 756.043 hours of speech. We ensure that the character descriptions used for training are disjoint from those used for evaluation.

[7] https://huggingface.co/openai/gpt-oss-120b
[8] https://huggingface.co/Qwen/Qwen2.5-72B-Instruct
[9] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
[10] https://github.com/resemble-ai/chatterbox
[11] https://docs.nvidia.com/nemo-framework/user-guide/latest/speech_ai/magpietts.html
[12] https://github.com/jzq2000/MoonCast
[13] https://huggingface.co/Qwen/Qwen3.5-397B-A17B
[14] https://developers.openai.com/api/docs/models/gpt-5.6-luna
[15] https://huggingface.co/Qwen/Qwen3-TTS-12Hz-1.7B-CustomVoice
[16] https://huggingface.co/Qwen/Qwen3-ForcedAligner-0.6B

Prompt Template for Interaction-Event Annotation

[System Instruction]
Choose natural advanced interaction behaviors for a validated spoken dialogue. Candidate locations, quotas, word boundaries, and collision constraints have already been generated deterministically. Select only from the provided candidates; do not rewrite the dialogue.

[Dialogue Context]
System instruction: {System_Instruction}
Local dialogue contexts: {Contexts}
Candidate slots: {Candidate_Slots}

[Backchannel]
Select a location where a brief listener acknowledgement is pragmatically natural. The backchannel must not anticipate or answer a question, contradict the audible clause, endorse unsafe content, or interrupt an unfinished semantic unit. Select an appropriate form such as "Uh-huh.", "Yeah.", "Right.", or "Okay."

[User Thinking Pause]
Select a candidate pause location and an appropriate semantic cue family. The pause should naturally indicate thinking, remembering, checking, calculating, searching, or deciding while preserving continuation of the same conversational task.

[User Interruption]
Select a candidate interruption location. When required, produce a short, on-topic disposable continuation for the character so that the interruption can occur naturally. The continuation must not introduce unsupported facts, new commitments, questions, or information used later in the dialogue.

[Output]
For every provided slot, return only the selected candidate option and its event-specific attributes in JSON format.

Figure 7: Prompt template used to annotate full-duplex interaction events. Candidate locations are generated deterministically, while the generator selects contextually appropriate backchannels, user thinking pauses, and interruptions without modifying the verified dialogue.

C Training Details

Supervised training. CharDuplex is initialized from GLM-4-Voice (Zeng et al., 2024) and trained using NeMo Toolkit (Kuchaiev et al., 2019). The general training corpus comprises 530K hours of speech continuation data, 70K hours of instruction-following QA data, and 20K hours of ASR-QA data (Wu et al., 2026). Character-conditioned SFT uses the additional corpus described above. Both supervised stages use AdamW with $(\beta_1, \beta_2) = (0.9, 0.98)$ and zero weight decay. An inverse-square-root learning-rate schedule is applied, with a learning rate of 5×10^{-4} and 2,500 warm-up steps for general full-duplex training, and 5×10^{-6} and $1{,}000$ warm-up stepsfor character-conditioned SFT. The model is trained on 32 A100 GPUs. SpecAugment is applied to user speech features (Ravanelli et al., 2021), and background noise from Freesound (Font et al., 2013) and MUSAN (Snyder et al., 2015) is added to user audio with probability 0.5, with the signal-to-noise ratio sampled uniformly between 0 and 60 dB.

Interactive RL. RL starts from the character-conditioned SFT checkpoint. Qwen-235B is used for both user simulation and LLM-based reward computation, and Qwen3-TTS synthesizes user speech. Each policy iteration samples eight trajectories for each of four initial dialogue conditions, yielding 32 trajectories. Trajectories are limited to 260 ticks of 160 ms, corresponding to 41.6 s of simulated dialogue. Response sampling uses a temperature of 1.0 and top-p of 0.9. GRPO uses a learning rate of 2×10^{-6} and a clipping threshold of 0.2. Each collected batch is processed once with one trajectory per GPU on eight A100 GPUs, giving four optimizer steps per policy iteration. The RL budget is set to 500 policy iterations. During the rollout, only the portion that would have become audible by the current tick is exposed to the simulator, using a speaking-rate estimate of 3.4 words per second derived from the SFT datasets.

Prompt Template for Interaction-Event Verification

[System Instruction]
You are the advanced-conversation behavior verifier. Structural properties such as speaker identities, candidate anchors, and event order have already been checked deterministically. Audit every proposed event for semantic and pragmatic validity.

[Backchannel Verification]
Determine whether the completed audible clause is a natural point for a short listener acknowledgement and whether the selected form is appropriate. Reject anticipation, contradiction, substantive answers, unsafe endorsement, or acknowledgements inserted inside an unfinished semantic dependency.

[User-Pause Verification]
Check whether the selected pause cue is compatible with the user's subsequent speech and whether the pause can be inserted without breaking the local utterance. The continuation after the pause must naturally resume the same conversational task.

[User-Interruption Verification]
Evaluate the interaction using only information audible before the interruption. Mentally remove the hidden continuation of the interrupted character response. The user's interruption and all subsequent dialogue must remain coherent without relying on any fact, question, explanation, or conclusion contained only in the discarded speech. Also reject repetitive, off-topic, unsupported, or role-inconsistent disposable continuations.

[Decision]
Audit every supplied event. Return `PASS` for valid interruptions and report only material failures for other events. Invalid events are assigned a bounded `DROP` or `REFINE` action with the corresponding failure code and correction.

[Output]
Return only one JSON object containing the audit count, interruption verdicts, backchannel replacements when needed, and detected event issues.

Figure 8: Prompt template used by the event-specific verifier. Backchannels and pauses are checked for local pragmatic compatibility, while interruptions are additionally validated under the information actually audible before the interruption, preventing subsequent dialogue from relying on discarded speech.

D PROMPTS USED IN FDGYM

FDGym uses an LLM-based user simulator to generate response-dependent user speech during each rollout. The simulator generates subsequent user utterances conditioned on the dialogue history and the progressively available CharDuplex response. Its prompt template is shown in Figure 9.

FDGym uses two independent LLM-based judges to compute the interaction and character reward components described in Section 3.4. The interaction judge receives timing statistics together with the textual dialogue, whereas the character judge additionally conditions on the character system prompt. Figures 10 and 11 show the corresponding prompt templates used during RL. Dynamic rollout-dependent inputs are represented by placeholders.

E GRPO FOR CHARDUPLEX

We optimize CharDuplex using group relative policy optimization (GRPO) (Shao et al., 2024). Let $\pi_\theta(y \mid s)$ denote the response-token distribution of CharDuplex with trainable parameters θ, conditioned on dialogue context s. At each iteration, we collect G trajectories $\{\tau_i\}_{i=1}^{G}$ under the same character prompt and initial user utterance using a parameter snapshot θ_{old}. This rollout policy remains fixed while the collected trajectories are used to update θ.

For trajectory reward $R_i = R(\tau_i, c)$, we compute the group-relative advantage

$$A_i = \text{clip}\left(\frac{R_i - \bar{R}}{\max(\sigma_R, \epsilon_{\text{num}})}, -A_{\max}, A_{\max}\right), \tag{6}$$

Prompt Template for the User Simulator

[System Instruction]
You are simulating a CUSTOMER making a phone call to customer service.

[Voice Call Rules]
- This is a VOICE conversation—you must speak like a real person on the phone.
- Generate ONE short, natural spoken sentence at a time, usually no more than 20 words.
- Use casual spoken language with fillers (e.g., "um", "uh", "like") and self-corrections.
- Spell out emails letter by letter (e.g., "j, o, h, n, at, gmail, dot, com").
- Do NOT use Markdown, bullets, or formatted text.
- Be concise—this is a phone call, not an essay.
- Only reveal information gradually, as a real person would.

[Special Tokens]
- Say `###STOP###` when the conversation is naturally complete.
- Say `###TRANSFER###` if the agent offers to transfer you.

[Patience]
- If the agent is silent for too long, ask "Hello? Are you still there?"
- If the agent is unhelpful after multiple attempts, express frustration.

[Persona]
{Persona_Guidelines}

[Scenario]
{Scenario_Instructions}

Figure 9: Prompt template used by the LLM-based user simulator in FDGym. The simulator produces short, natural spoken user turns conditioned on the provided persona and dialogue scenario, revealing information progressively as the interaction evolves.

where $\bar{R} = \frac{1}{G}\sum_{j=1}^{G} R_j$ and $\sigma_R = \left[\frac{1}{G}\sum_{j=1}^{G}(R_j - \bar{R})^2\right]^{1/2}$ are the group reward mean and population standard deviation. The positive constant ϵ_{num} prevents division by zero, and $A_{\max}$ bounds the advantage magnitude.

Let $y_{i,t}$ be the sampled response token at position t in trajectory i. Its context $s_{i,t}$ contains the character prompt, available user speech, and preceding model outputs. We recompute its probability under the current policy and form the ratio

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid s_{i,t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid s_{i,t})}. \tag{7}$$

The clipped policy loss is computed over the complete response stream of each sampled trajectory:

$$\mathcal{L}_{\text{policy}} = -\frac{1}{N}\sum_{i=1}^{G}\sum_{t=1}^{T_i} \min\Big(\rho_{i,t}(\theta)A_i, \; \text{clip}(\rho_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\, A_i\Big), \tag{8}$$

where T_i is the number of response tokens generated in trajectory τ_i, $N = \sum_{i=1}^{G} T_i$ is the total number of generated response tokens, and $\varepsilon > 0$ controls probability-ratio clipping. After one optimization pass over the collected trajectories, the updated model is used for the next rollout iteration.

F Comparison Before and After Interactive Post-Training

To further examine the effect of FDGym, Table 4 consolidates the results of CharDuplex-SFT and CharDuplex across character consistency, general speech intelligence, and full-duplex interaction capabilities. Because the metrics have different scales and optimization directions, we report a

Prompt Template for Interaction Reward

[System Instruction]
You are an expert evaluator for spoken conversations.

You will be given:
- Timing statistics.
- A transcript of the CUSTOMER and AI AGENT.

[Inputs]
Timing Statistics: {Timing_Statistics}
Conversation Transcript: {Conversation_Transcript}

[Evaluation]
Evaluate only the AGENT.

interaction_quality: Was the agent's side of the conversation natural, coherent, responsive, and appropriate to the preceding user turns? Do not evaluate character imitation, persona, role adherence, style requirements, brevity, or verbosity.

[Output Format]
Return ONLY the following JSON object, with no Markdown or extra text:

```
{
  "interaction_quality": {
    "score": <float 0-1>,
    "reason": "<brief explanation>"
  }
}
```

Figure 10: Prompt template used to compute the interaction reward r_{int}. The judge evaluates the model's conversational coherence, responsiveness, and appropriateness from the rollout transcript and timing statistics, independently of character consistency.

direction-aware relative change:

$$\Delta_{\text{rel}} = \begin{cases} \dfrac{x_{\text{RL}} - x_{\text{SFT}}}{x_{\text{SFT}}} \times 100\%, & \text{for } \uparrow \text{ metrics}, \\ \dfrac{x_{\text{SFT}} - x_{\text{RL}}}{x_{\text{SFT}}} \times 100\%, & \text{for } \downarrow \text{ metrics}. \end{cases} \tag{9}$$

Thus, a positive value consistently denotes improvement and a negative value denotes degradation.

FDGym improves every SpeechRole-Eval metric, with consistent gains in instruction adherence, conversational coherence, personality consistency, and knowledge consistency. In contrast, the general speech-intelligence benchmarks show small bidirectional fluctuations: improvements on some QA and reasoning tasks are accompanied by similarly limited decreases on others, while the open-ended response and safety metrics remain close to the SFT model. Full-duplex interaction shows a similar overall pattern, with most metrics remaining stable or changing only modestly, although Backchannel TOR exhibits a larger relative decrease. Overall, the consolidated results indicate that the primary effect of interactive post-training is stronger character consistency, while the general speech intelligence and full-duplex interaction capabilities established during supervised training are largely preserved.

G Ablation Studies and analysis

Effect of Timing Rewards We examine the contribution of the two deterministic timing rewards, r_{resp} and r_{yield}, used during FDGym-based RL. We train an ablated variant by removing both timing terms while keeping the LLM-judged rewards unchanged. Removing the timing rewards does not produce a meaningful improvement on SpeechRole-Eval, while the full-duplex turn-taking performance decreases slightly. This indicates that the timing rewards help preserve the full-duplex interaction behavior established during SFT. Since both rewards are computed deterministically from the rollout timestamps and incur negligible additional computational cost, we retain them in the final training objective.

Prompt Template for Character Reward

[System Instruction]
You are an expert evaluator for character conversations.

You will be given:
- The AGENT system prompt, including role, character, persona, and style requirements.
- A transcript of the CUSTOMER and AI AGENT.

[Inputs]
Agent System Prompt: {Character_System_Prompt}
Conversation Transcript: {Conversation_Transcript}

[Evaluation]
Evaluate only the AGENT.

character_style_consistency: Did the agent faithfully follow the persona, role identity, tone, and behavioral constraints explicitly specified in the AGENT system prompt? Judge only persona and role adherence. Penalize customer-role leakage, violating the requested speaking style, and behavior or claims inconsistent with the assigned role. Do not score generic response quality, helpfulness, factual correctness, answer completeness, brevity, verbosity, timing, or user satisfaction.

[Output Format]
Return ONLY the following JSON object, with no Markdown or extra text:

```
{
  "character_style_consistency": {
    "score": <float 0-1>,
    "reason": "<brief explanation>"
  }
}
```

Figure 11: Prompt template used to compute the character reward r_{char}. The judge evaluates whether the model's responses remain consistent with the prescribed identity, persona, tone, and behavioral constraints, independently of generic response quality and interaction timing.

Effect of Dialogue Verification We evaluate whether the verifier in the automated data-construction pipeline materially improves dialogue quality. We compare the complete generator–verifier pipeline against an ablated variant that directly accepts the generator output without verification or repair. For evaluation, we randomly sample generated dialogues and use GPT-5.6 Sol[17] with maximum reasoning effort as an independent judge. Without verification, the pass rate is 65.6%. In the full generator–verifier pipeline, 68.8% of generated dialogues pass the initial verification directly, and the final pass rate, as judged by gpt-5.6-sol, increased to 93.7%. Although removing verification reduces data-construction cost, it substantially lowers the quality of the resulting corpus and would require additional downstream filtering. We therefore retain the generator–verifier process in the final data-construction pipeline.

[17] https://developers.openai.com/api/docs/models/gpt-5.6-sol

Table 4: Consolidated comparison between CharDuplex-SFT and CharDuplex after FDGym-based interactive post-training. Relative changes are direction-aware according to Eq. 9; positive values indicate improvement and negative values indicate degradation. Percentages are computed from the reported scores and may be affected slightly by rounding.

Capability	Metric	CharDuplex-SFT	CharDuplex	Relative Change (%)
Character consistency	IA ↑	0.807	0.815	+0.99
	CC ↑	0.800	0.805	+0.63
	PeC ↑	0.729	0.738	+1.23
	KC ↑	0.770	0.781	+1.43
	Avg. ↑	0.777	0.785	+1.03
General speech intelligence	LlamaQ ↑	74.3	74.7	+0.54
	WebQ ↑	40.8	40.5	-0.74
	OBQA ↑	63.1	62.4	-1.11
	MMSU ↑	44.2	44.0	-0.45
	SD-QA ↑	47.4	49.2	+3.80
	BBH ↑	50.0	51.2	+2.40
	AdvBench ↑	95.8	95.6	-0.21
	AlpacaEval ↑	3.63	3.63	0.00
	CommonEval ↑	3.09	3.09	0.00
	WildVoice ↑	2.30	2.29	-0.43
Full-duplex interaction	Pause Synthetic TOR ↓	0.934	0.927	+0.75
	Pause Candor TOR ↓	0.815	0.815	0.00
	Backchannel TOR ↓	0.636	0.709	-11.48
	Backchannel ICC Freq. ↑	0.054	0.055	+1.85
	Backchannel JSD ↓	0.890	0.889	+0.11
	Smooth Turn Taking TOR ↑	0.840	0.857	+2.02
	Smooth Turn Taking Latency ↓	0.133	0.136	-2.26
	User Interruption TOR ↑	0.990	0.990	0.00
	User Interruption GPT-4o ↑	4.081	4.071	-0.25
	User Interruption Latency ↓	0.558	0.565	-1.25