WIGVO: Real-Time Bidirectional Speech Translation over Legacy PSTN Calls via Dual-Session Echo Gating

Anonymous ACL submission

Abstract

Real-time speech translation with large language models (LLMs) has become feasible in controlled settings such as mobile apps and web browsers, where developers assume wideband audio and client-side echo cancellation. However, deploying such systems over the Public Switched Telephone Network (PSTN) remains challenging due to narrowband G.711 audio, unpredictable round-trip delays, and absence of client-side signal processing. We present **WIGVO**, a server-side relay system that enables bidirectional LLM-based speech translation over ordinary telephone calls without requiring app installation or carrier integration. A central contribution is addressing what we term *echo-induced self-reinforcing translation loops*: synthesized speech echoing back through the PSTN gets re-ingested and repeatedly translated. WIGVO solves this through a dual-session architecture with deterministic silence injection and energy-based voice activity detection (VAD) gating. We evaluate WIGVO on 97 completed PSTN calls (38 fully instrumented) across three communication modes, observing 564 ms median Session A latency (N=148 turns), zero echo-induced translation loops, and USD 0.42 per minute cost. WIGVO additionally supports text-to-voice and voice-to-text relay modes for users with speech anxiety or hearing impairments. The system is deployed at `https://wigvo.run` with a live demo and video walkthrough.

1 Introduction

Streaming speech-to-speech translation has advanced rapidly with Realtime LLM APIs, but existing work focuses on WebRTC and VoIP settings where wideband audio and client-side acoustic echo cancellation (AEC) are available (Seamless Communication et al., 2023a; Radford et al., 2023). The Public Switched Telephone Network (PSTN), however, still handles the majority of calls to restaurants, hospitals, and government offices (International Telecommunication Union, 2020). PSTN operates at 8 kHz with μ-law companding, introduces 80–600 ms jitter, and provides no client-side AEC. For foreign residents, people with speech anxiety, or users with hearing impairments, making such calls remains a significant barrier.

We present **WIGVO**, a server-side relay that bridges a web client and a standard PSTN phone number through two concurrent LLM-backed streaming sessions. The caller speaks (or types) via a browser; the callee answers on an ordinary phone. WIGVO translates bidirectionally in real time, supporting voice-to-voice (V2V), text-to-voice (T2V), and voice-to-text (VTT) accessibility modes.

A key challenge is what we call an **echo-induced self-reinforcing translation loop**: TTS audio played to the callee echoes back through the PSTN, is re-recognized as new speech, and gets translated again—creating a runaway feedback cycle. This problem is specific to full-duplex telephony channels and cannot be solved by VAD threshold tuning alone. Our solution operates at the architectural level: (i) strict directional separation via two independent Realtime sessions, (ii) deterministic silence injection during TTS playback windows, and (iii) energy-based gating calibrated for μ-law dynamics.

We evaluate WIGVO on 97 completed PSTN calls and report latency, echo suppression, and cost metrics. Our contributions are:

- We formalize the *echo-induced translation loop* problem in streaming speech translation over telephony.

- We propose a *dual-session gated architecture* combining directional separation, silence injection, and energy-based gating for PSTN environments.

- We deploy and evaluate a working relay server across three communication modes, reporting system-level metrics from real PSTN calls.

2 Related Work

Simultaneous speech translation. Recent systems such as SeamlessM4T (Seamless Communication et al., 2023b) and Seamless Streaming (Seamless Communication et al., 2023a) achieve real-time speech translation with expressive and multilingual models. ESPnet-ST-v2 (Yan et al., 2023) provides a comprehensive toolkit for offline and simultaneous translation. However, all assume wideband audio inputs from controlled environments and do not address PSTN-specific challenges such as G.711 codec artifacts, telephony echo, or narrowband VAD.

Telephony AI agents. Google Duplex (Leviathan and Matias, 2018) demonstrated autonomous PSTN calls for restaurant reservations, but performs task completion in a single language rather than bidirectional translation. Commercial platforms such as Vapi and Bland.ai provide LLM-powered phone agents but focus on monolingual voice assistants.

Relay services for accessibility. Telecommunications Relay Services (TRS), mandated by the FCC in the United States, provide human operators who relay calls for deaf and hard-of-hearing users (Federal Communications Commission, 2024). These services are limited to specific countries, require human intermediaries, and do not support cross-lingual translation.

Accessibility-oriented translation. sign.mt (Moryossef, 2024) provides bidirectional sign language translation with an accessibility focus; our work targets telephony and addresses echo-induced loops. Table 1 positions WIGVO against these systems.

System	**PSTN**	**Bidir.**	**S2S**	**Echo**	**A11y**
SeamlessM4T	×	✓	✓	N/A	×
Google Duplex	✓	×	×	Unk.	×
FCC TRS	✓	✓	×	Human	✓
sign.mt	×	✓	✓	N/A	✓
Vapi / Bland	✓	×	×	Unk.	×
WIGVO	✓	✓	✓	✓	✓

Table 1: Comparison with existing systems. PSTN: works over telephone network; Bidir.: bidirectional translation; S2S: speech-to-speech; Echo: handles telephony echo; A11y: accessibility modes (T2V/VTT).

Figure 1: WIGVO architecture. The relay server manages Session A (caller→callee) and Session B (callee→caller) as independent Realtime API connections, with a telephony gateway for PSTN audio.

3 System Architecture

Figure 1 shows the high-level architecture. A browser-based client connects to the relay server via WebSocket, sending 16 kHz PCM audio and receiving translated audio, captions, and status events. The relay manages two independent Realtime LLM sessions and a telephony gateway connection.

AudioRouter and pipelines. An `AudioRouter` implements the Strategy pattern to dispatch events to one of four pipeline implementations: V2V, T2V, VTT, or Full-Agent. Each pipeline manages the lifecycle of two Realtime sessions and cross-cutting components (guardrails, context window, metrics).

Dual Realtime sessions. Session A receives browser audio (PCM16) and produces translated G.711 μ-law for the telephony gateway. Session B receives PSTN audio (G.711 μ-law, 8 kHz) and produces translated audio and captions for the browser. Each session has its own system prompt, sliding context window (6 turns), and event stream, ensuring strict directional separation.

Communication modes. **V2V**: both directions use streaming ASR+TTS. **T2V**: the caller types text, which is converted to TTS for the callee; the callee's speech is transcribed and displayed as captions. **VTT**: the caller speaks; the callee receives text captions only. **Full-Agent**: the LLM autonomously conducts the call based on collected user intent.

4 Echo Gating Mechanisms

4.1 The Echo Loop Problem

In a naive single-session design, both speakers' audio feeds into one streaming session. TTS output played to the callee echoes back through the PSTN after 80–600 ms, gets re-recognized, and triggers another translation cycle. We observed this reliably in early prototypes: the system would generate progressively distorted paraphrases of its own output until manually interrupted.

We initially attempted a Pearson-correlation echo detector comparing outgoing TTS buffers with incoming audio. While effective in controlled tests, it failed in production due to μ-law nonlinear quantization, variable delays, and background noise producing spurious correlations. This detector is now disabled.

A single-session baseline without gating produced loops in the majority of test calls; a Pearson-correlation detector reduced but did not eliminate loops while introducing frequent false positives that blocked legitimate callee speech. Echo Gate v2 eliminated loops entirely while preserving callee interruption capability.

4.2 Echo Gate v2: Silence Injection

Our production solution, **Echo Gate v2**, operates deterministically:

1. When Session A begins streaming TTS to the PSTN, an *echo window* is activated. Duration is estimated from byte length and sample rate, capped at 1.2 s.
2. During this window, incoming PSTN audio is replaced with μ-law silence (0xFF) before forwarding to Session B. This blocks echo at the server while allowing the Realtime API's VAD to naturally detect end-of-speech.
3. A dynamic cooldown of 0.3 s accounts for PSTN jitter before resuming normal forwarding.

While comfort noise generation (RFC 3389; Zopf 2002) addresses perceptual continuity for human listeners in VoIP, we identify a distinct failure mode: echo-suppression-induced audio gaps cause streaming VAD state machine deadlock in neural speech APIs. Specifically, if the relay simply *drops* audio during echo windows rather than injecting silence, the Realtime API's server-side VAD never observes a silence-to-speech transition and remains stuck in a "speaking" state indefinitely. To our knowledge, this failure mode and its mitigation via silence injection have not been reported in the context of neural streaming S2ST pipelines.

Across 97 completed calls, Echo Gate v2 produced zero echo-induced translation loops. Echo gate activated 136 times total (3.6/call on average), with 0 breakthrough incidents, confirming that silence injection effectively prevents self-reinforcing translation cycles.

4.3 Energy-Based Gating

RMS energy monitoring provides a secondary gate. During echo windows, only high-energy signals ($\geq$400 RMS) break through as genuine callee speech; typical echo energy (100–400 RMS) remains suppressed. Outside echo windows, a lower threshold (150 RMS) filters line noise without clipping normal speech. This two-tier scheme allows callee interruptions to override the echo gate immediately. A three-level interrupt priority (callee > caller > AI) ensures callee speech immediately cancels active TTS.

5 PSTN-Aware VAD and Robustness

5.1 Server VAD Failure Modes on PSTN

OpenAI's Realtime API provides a server-side VAD that works well for wideband WebRTC audio. On PSTN lines, however, we observed three failure modes:

1. **Background noise misclassification.** Constant low-level PSTN noise (line hum, codec quantization artifacts) is classified as speech, causing the VAD to enter a "speaking" state that never terminates. We observed stuck durations ranging from 15 s to 72 s before manual intervention.

2. **Delayed `speech_stopped` events.** Even after genuine speech ends, the noisy PSTN floor prevents the energy level from dropping below the server VAD's silence threshold, delaying end-of-utterance detection by 15–72 s.

3. **Audio discontinuity deadlock.** When the relay drops audio frames (e.g., during echo suppression), the server VAD never observes a clean silence interval and fails to emit a `speech_stopped` event entirely.

5.2 Two-Stage Local VAD

To address these failures, WIGVO implements a two-stage local VAD on the relay server, applied to incoming PSTN audio before it reaches the Realtime API:

Stage 1: RMS Energy Gate. A fast energy check filters sub-speech signals. We set the threshold at 150 RMS based on empirical μ-law noise distributions: background noise typically falls in the 50–200 RMS range, while speech aligns around 500–2,000+ RMS. Frames below 150 RMS are classified as noise and replaced with silence, preventing them from reaching Stage 2.

Stage 2: Silero VAD. Frames passing the energy gate are processed by Silero VAD (Silero Team, 2021), a lightweight neural voice activity detector. Since Silero expects 16 kHz input but PSTN audio arrives at 8 kHz, we apply zero-order hold (ZOH) upsampling before inference. Each frame is 32 ms (512 samples at 16 kHz).

Hysteresis state machine. To avoid rapid state oscillation, we apply asymmetric hysteresis thresholds:

- **SILENCE→SPEAKING**: probability ≥ 0.5 for 2 consecutive frames (64 ms). Fast onset preserves responsiveness.
- **SPEAKING→SILENCE**: probability < 0.35 for 15 consecutive frames (480 ms). Slow offset avoids premature cutoff during natural pauses.

This design reduces `speech_stopped` latency from 15–72 s (server VAD on PSTN) to a consistent 480 ms, a reduction of over 96%.

5.3 Silence Injection for VAD Continuity

As noted in Section 4.2, the echo gate replaces PSTN audio with μ-law silence (0xFF) rather than dropping frames. This design choice is critical for VAD continuity: if frames are dropped entirely, the Realtime API's streaming VAD loses its audio timeline and cannot detect speech boundaries. By injecting silence bytes, the relay maintains stream continuity while suppressing echo content. This is functionally distinct from comfort noise generation (RFC 3389; Zopf 2002), which synthesizes perceptually natural background noise for human listeners. Our silence injection instead targets the VAD state machine of neural speech APIs, preventing deadlock caused by audio gaps.

5.4 STT Hallucination Blocklist

A 15-pattern blocklist intercepts Korean broadcast-style hallucinations from Whisper on low-energy input; 5 outputs were blocked across 38 instrumented calls.

6 Evaluation

We evaluated WIGVO on 97 PSTN calls (38 fully instrumented) for Korean↔English translation.

6.1 Latency

Table 2 reports latency statistics across the 38 instrumented calls.

Session A achieves 564 ms median latency, within the range for interactive communication. Session B, which depends on PSTN audio quality and streaming ASR, shows 2,023 ms median. STT (Whisper) accounts for 74.7% of Session B mean latency across 103 paired turns, with translation contributing the remainder. Figure 2 shows the end-to-end latency distributions for Session A and Session B. Figure 3 shows the correlation between utterance length and Session B latency

Stage	Avg	P50	P95	Max
Session A (ms)	618	564	1,023	2,256
Session B E2E (ms)	2,270	2,023	5,378	10,503
STT (ms)	2,263	1,979	5,065	—
Translation (ms)	757	464	2,209	—
First message (ms)	2,585	1,550	7,460	11,308

Table 2: Translation latency across 38 instrumented calls. Session A: caller→callee (ASR+translate+TTS). Session B: callee→caller, decomposed into STT (Whisper) and translation (GPT-4o) components. N: number of measured turns.

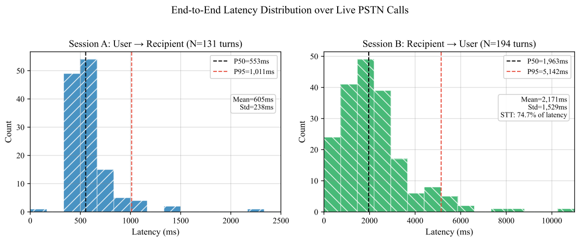

Figure 2: End-to-end latency distributions for Session A (caller→callee, N=148 turns) and Session B (callee→caller, N=214 turns) over live PSTN calls.

(r = 0.529, p < 0.001), STT accounts for 74.7% of Session B mean latency.

T2V mode shows lower Session A latency (no caller-side ASR), while Agent mode shows higher Session B latency due to function-calling overhead.

6.2 Echo and Safety

Table 3 reports echo suppression, VAD, and guardrail statistics across the 38 instrumented calls.

The echo gate activated 3.6 times per call on average, corresponding to TTS playback events. Critically, **zero echo-induced translation loops** were observed, whereas early prototypes without gating reliably produced such loops. VAD false triggers (1.6/call) were mitigated by a minimum speech duration filter (480 ms hysteresis). The guardrail module blocked 5 hallucinated outputs (broadcast-style phrases from Whisper on low-energy input).

6.3 Cost

Across the 38 instrumented calls (45.5 minutes total), WIGVO consumed 638,035 tokens at a total cost of USD 19.05, yielding **USD 0.42/min** (USD 0.50/call). These costs reflect dual-session Realtime API pricing and remain an order of magnitude cheaper than human relay services (USD 1–3/min).

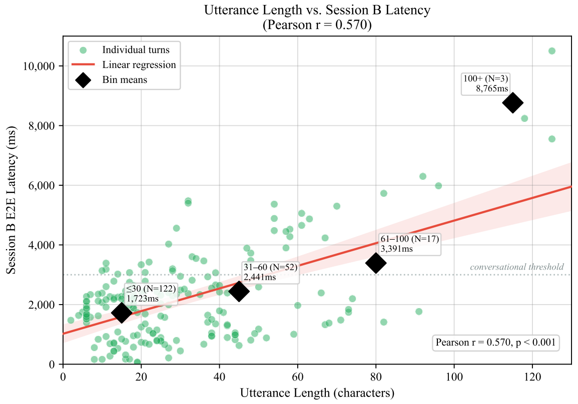

Figure 3: Utterance length (characters) vs. Session B E2E latency. Pearson $r = 0.529$ ($p < 0.001$), indicating that longer recipient utterances incur higher ASR-dominated latency.

Metric	Total	Per-call avg
Echo gate activations	136	3.6
Echo-induced loops	0	0.0
VAD false triggers	62	1.6
Hallucinations blocked	5	0.13

Table 3: Echo, VAD, and guardrail statistics over 38 instrumented calls. Zero echo loops were observed.

7 Demonstration

WIGVO is deployed at https://wigvo.run with a live demo accessible to reviewers.[1] The web interface supports scenario selection (restaurant, hospital, government office), real-time caption display, and mode switching. Figure 4 shows the interface during an active V2V call.

We demonstrate two scenarios:

Restaurant reservation (V2V). An English-speaking user calls a Korean restaurant. Session A translates caller utterances into Korean TTS for the callee; Session B returns Korean responses as English audio and captions. Across 38 instrumented calls, Session A achieved 564 ms median latency (P95: 1,023 ms) and Session B 2,023 ms median (P95: 5,378 ms; r=0.529 with utterance length, p<0.001).

[1] Demo video: https://youtu.be/PLACEHOLDER

Figure 4: WIGVO web interface during a V2V call. Left: chat and caption view with bidirectional translations. Right: call status, duration, and mode indicator.

Medical appointment (T2V). A Korean user with speech anxiety types messages that WIGVO speaks to a clinic via TTS; the clinic's spoken responses appear as real-time captions. A typing indicator is relayed to the callee during text composition.

8 Discussion and Limitations

Scope of contribution. WIGVO's contribution is architectural: demonstrating that existing Realtime LLM APIs can be made robust for PSTN environments through dual-session separation, deterministic echo gating, and energy-based filtering. Translation quality is not independently evaluated; WIGVO inherits the translation capabilities of GPT-4o Realtime (OpenAI, 2024), which achieves state-of-the-art performance on Korean↔English translation. Formal translation quality evaluation (e.g., BLEU, chrF) is left for future work.

Limitations. Session B latency (∼2 s median) introduces noticeable delay in the callee→caller direction, primarily due to streaming ASR. The evaluation covers Korean↔English only; multilingual generalization requires further study. We report system metrics but lack formal user studies; a System Usability Scale evaluation with target populations is planned. Echo gate parameters are hand-tuned; adaptive threshold learning could improve robustness.

Broader applications. The dual-session gated architecture is applicable beyond translation: multilingual call centers, voice inclusion services for users with disabilities, and telephony-native LLM agents that combine task completion with cross-lingual communication.

Ethics Statement

All calls were initiated by the authors and collaborators for system testing. Audio was logged for debugging and aggregate metrics only, with no personally identifiable information retained beyond temporary phone numbers. For deployment with real users, informed consent, data minimization, and compliance with privacy regulations (e.g., GDPR, PIPA) are required. The accessibility use cases (T2V for speech anxiety, VTT for hearing impairment) aim to reduce communication barriers but should be deployed with user agency and opt-in participation.

Acknowledgments

We thank the anonymous reviewers for their feedback. WIGVO uses OpenAI's Realtime API and Twilio Media Streams. The relay server is implemented in Python with FastAPI.

References

Federal Communications Commission. 2024. Telecommunications relay services (TRS). https://www.fcc.gov/consumers/guides/telecommunications-relay-service-trs.

International Telecommunication Union. 2020. Public switched telephone network (PSTN): Technical overview. ITU-T Recommendation Series Q and G. https://www.itu.int/rec/T-REC-G/en.

Yaniv Leviathan and Yossi Matias. 2018. Google Duplex: An AI system for accomplishing real-world tasks over the phone. Google AI Blog, May 2018. https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html.

Amit Moryossef. 2024. sign.mt: Real-time multilingual sign language translation application. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 182–190. Association for Computational Linguistics.

OpenAI. 2024. OpenAI realtime API documentation. https://platform.openai.com/docs/guides/realtime.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518. PMLR.

Seamless Communication et al. 2023a. Seamless: Multilingual expressive and streaming speech translation. In *arXiv preprint arXiv:2312.05187*. https://arxiv.org/abs/2312.05187.

Seamless Communication et al. 2023b. SeamlessM4T: Massively multilingual & multimodal machine translation. In *arXiv preprint arXiv:2308.11596*. https://arxiv.org/abs/2308.11596.

Silero Team. 2021. Silero VAD: Pre-trained enterprise-grade voice activity detector. https://github.com/snakers4/silero-vad.

Brian Yan, Jiatong Shi, Yun Tang, Hirofumi Inaguma, Yifan Peng, Siddhant Dalmia, Peter Polák, Patrick Fernandes, Dan Berrebbi, Tomoki Hayashi, Xiaohui Zhang, Zhaoheng Ni, Moto Hira, Soumi Maiti, Juan Pino, and Shinji Watanabe. 2023. ESPnet-ST-v2: Multipurpose spoken language translation toolkit. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 400–411. Association for Computational Linguistics.

R. Zopf. 2002. RTP payload for comfort noise (CN). Technical Report RFC 3389, IETF.