# STCTS: Generative Semantic Compression for Ultra-Low Bitrate Speech via Explicit Text-Prosody-Timbre Decomposition

Siyu Wang
Fudan University
23300240006@m.fudan.edu.cn

***Abstract*— Voice communication in extreme bandwidth-constrained environments—such as maritime, satellite, and tactical networks—remains prohibitively expensive. Traditional audio codecs, constrained by waveform fidelity, struggle to operate below 1 kbps. Conversely, existing semantic approaches (STT-TTS) achieve ultra-low bitrates but often sacrifice the "human" elements of speech—prosody and speaker identity—resulting in robotic, impersonal output.**

**We present STCTS, a generative semantic compression framework that enables natural, expressive voice communication at $\sim$80 bps. Unlike end-to-end neural codecs that rely on opaque latent tokens, STCTS explicitly decomposes speech into three orthogonal components: linguistic content, prosodic expression, and speaker timbre. This decomposition enables two key innovations: (1) Sparse Prosody Transmission, which leverages the generative interpolation capabilities of modern TTS models to reduce prosody control signals to 0.1–1 Hz ($<$14 bps), eliminating the need for frame-level updates; and (2) Component-Wise Semantic Compression, which applies tailored entropy coding to each stream, including context-aware text compression ($\sim$70 bps) and amortized speaker embedding transmission.**

**Evaluations on LibriSpeech demonstrate that STCTS achieves a $70\times$ bitrate reduction versus Opus (6 kbps) and $12\times$ versus EnCodec (1 kbps), while maintaining state-of-the-art perceptual quality (NISQA MOS $>$ 4.26). Beyond efficiency, our modular architecture supports privacy-preserving encryption, human-interpretable transmission, and flexible deployment on edge devices, offering a robust solution for ultra-low bandwidth scenarios.**

***Index Terms*— Ultra-low bitrate communication, semantic compression, speech-to-text, text-to-speech, prosody encoding, voice cloning, bandwidth-constrained networks, neural speech synthesis**

## I. Introduction

Voice communication remains a fundamental human need, yet in many regions and circumstances worldwide, network bandwidth is severely constrained and prohibitively expensive. Consider maritime workers aboard cargo ships and fishing vessels: they rely on satellite communication systems where bandwidth costs can reach \$5–\$15 per megabyte, making a single 10-minute voice call at standard telephony bitrates (e.g. 20 – 30 kbps) cost approximately \$20—a substantial burden for workers often earning modest wages. Consequently, crew members are often restricted to brief, infrequent calls home, exacerbating the isolation inherent to months-long voyages. Similar bandwidth constraints afflict satellite and aeronautical communication systems (e.g., in-flight connectivity, drone control links), wireless networks in remote and rural regions (e.g., Sub-Saharan Africa, Pacific islands, Himalayan communities), tactical military communication systems operating under contested spectrum conditions, and emerging large-scale real-time voice social platforms seeking to serve millions of concurrent users with minimal infrastructure costs. In all these scenarios, the fundamental question remains: *how can we enable natural, affordable voice communication with minimal bandwidth consumption?*

To achieve natural, expressive voice communication at ultra-low bitrates, we draw insights from three distinct research domains, each offering valuable principles while facing fundamental limitations:

**1. Existing Speech Coding and Semantic Compression Techniques.** Traditional speech codecs compress acoustic waveforms through parametric modeling (e.g., Opus at 6–40 kbps [1]) or neural encoding (e.g., EnCodec at 1–24 kbps [5]). While achieving significant compression compared to uncompressed audio (64–128 kbps), they remain fundamentally limited by waveform-level fidelity preservation, preventing operation below $\sim$1 kbps. Recent semantic compression methods [8], [9] (e.g., Vevo at $\sim$650 bps) demonstrate that encoding speech at higher abstraction levels—representing *what is said* and *how it is said* through discrete tokens rather than raw acoustics—enables order-of-magnitude compression gains while maintaining perceptual quality through generative reconstruction. However, token-based representations lack interpretability and modularity: transmitted content is opaque to inspection, and upgrading individual components (recognition or synthesis models) requires end-to-end retraining of the entire system.

**2. STT-TTS Communication Systems.** STT-TTS pipelines [10] in IoT and tactical communication scenarios where bandwidth is severely constrained achieve ultra-low bitrates by converting speech to text, transmitting the compressed text ($\sim$70 bps), and resynthesizing speech at the receiver. The

use of explicit text representation provides significant advantages: transmitted content is human-readable and debuggable, STT and TTS components can be independently upgraded as technology advances, and the architecture naturally enables secondary applications such as real-time transcription and multilingual translation. However, transmitting only linguistic content sacrifices two essential dimensions of human communication—*prosodic expressiveness* (intonation, emphasis, emotion) and *speaker identity* (voice timbre, characteristics). In maritime communication, for instance, crew members calling home expect their families to recognize their voice and perceive their emotional state; a generic synthesized voice transmitting only words feels impersonal and detached.

**3. Speech Disentanglement for Representation Learning.** Speech disentanglement approaches [11], [12] decompose speech into orthogonal representations corresponding to content, prosody, and timbre through end-to-end learned encoders. This factorization has proven effective for voice conversion and controllable synthesis tasks, revealing a crucial insight: speech components exhibit vastly different temporal dynamics—linguistic content changes rapidly (2–3 words/sec), prosody varies smoothly over multi-second spans, and speaker identity remains constant across conversations. This understanding of component-specific temporal structure is fundamental to designing effective compression strategies. However, these methods target representation learning rather than communication—they transmit continuous frame-level latent vectors at 50–100 Hz (requiring hundreds to thousands of bps) and produce learned representations that lack the interpretability and modularity needed for practical deployment in bandwidth-constrained scenarios.

**Our Approach: STCTS.** This paper presents **STCTS** (**S**peech-**t**o-**T**ext, **C**ompression and **T**ext-**t**o-**S**peech), which achieves natural, expressive voice communication at $\sim$**80 bps** by bridging semantic-level representation, explicit text encoding, and speech component factorization. The central insight is that *perceptually natural speech can be reconstructed at the receiver side through high-level semantic and paralinguistic representations—linguistic content, prosodic expression, and speaker identity—without preserving waveform-level acoustic fidelity.* By operating at this abstraction level, disentangled speech components exhibit *vastly different compressibility characteristics*, enabling component-specific compression strategies: **Linguistic content** changes rapidly but is highly compressible via context-aware text encoding ($\sim$70 bps). **Prosodic expression** varies smoothly over multi-second spans—this temporal continuity enables *sparse prosody transmission* at 0.1–1 Hz with TTS interpolation reconstructing natural contours between keyframes, reducing prosody bitrate to $<$14 bps without sacrificing expressiveness. **Speaker identity** remains constant per speaker, requiring only one-time embedding transmission that amortizes to near-zero bitrate in typical conversations. The system architecture reflects this design: at the sender, STT extracts human-readable text while parallel modules capture prosody and speaker characteristics; compression applies tailored strategies to each component (Brotli for text, delta-encoded quantization for prosody, amortized transmission for timbre); at the receiver, TTS reconstructs natural-sounding speech by conditioning a neural synthesis model on all three streams. This explicit decomposition achieves a bitrate comparable to Morse code transmissions while conveying full conversational speech with near-natural quality and speaker fidelity—capabilities that prior semantic codecs (lacking interpretability), STT-TTS systems (lacking expressiveness), and disentanglement methods (requiring frame-level transmission) cannot simultaneously deliver. Figure 1 illustrates the complete STCTS architecture.

**Key Design Distinctions.** Our approach differs fundamentally from prior semantic compression work in two aspects. First, unlike token-based methods (e.g., Vevo [9]) that encode speech into discrete acoustic tokens, we use explicit text as the semantic representation. This design choice provides interpretability (transmitted content is human-readable), modularity (STT and TTS components can be independently upgraded), and enables secondary applications (real-time transcription, conversation logging, multilingual translation). Second, we transmit prosody features at extremely low rates (0.1–1 Hz, corresponding to updates every 1–10 seconds) by exploiting TTS models' ability to interpolate smooth prosody contours between sparse keyframes. Through systematic analysis of prosody sampling rates (Section IV-B), we identify a bimodal quality distribution with optimal operating points at sparse rates, enabling near-zero prosody bitrate ($<$14 bps) without sacrificing naturalness.

We conduct comprehensive experiments on the LibriSpeech corpus [31], evaluating three quality modes (minimal, balanced, high-quality) against baseline codecs (Opus, EnCodec) and the Vevo semantic compression framework. Our contributions are as follows:

- We achieve sustained bitrates of 71.6–79.6 bps (excluding amortized speaker embeddings), representing a $40\times$ to $50\times$ reduction compared to Opus (6 kbps) and an $8\times$ reduction compared to EnCodec (1 kbps), while maintaining perceptual quality (NISQA MOS $\sim$4.26) comparable to Vevo ($\sim$650 bps, MOS 4.21).
- We demonstrate that prosody can be transmitted at extremely sparse rates (0.1–1 Hz, $<$14 bps) by leveraging TTS interpolation, and identify a bimodal quality distribution where mid-frequency prosody updates (1–5 Hz) perform worse than both sparse and dense regimes due to perceptually salient discontinuities.
- We show that semantic compression exhibits inherent temporal desynchronization (reflected in low STOI scores $\sim$0.15) due to independent STT and TTS timing, yet maintains high intelligibility (WER $\sim$0.23) and naturalness (NISQA $>$4.2).
- We achieve graceful degradation under channel noise (0.1–10% bit error rate), maintaining NISQA MOS $>$4.2 even at 10% BER through prioritized transmission and prosody interpolation, demonstrating robustness for real-world deployment in degraded communication environments.

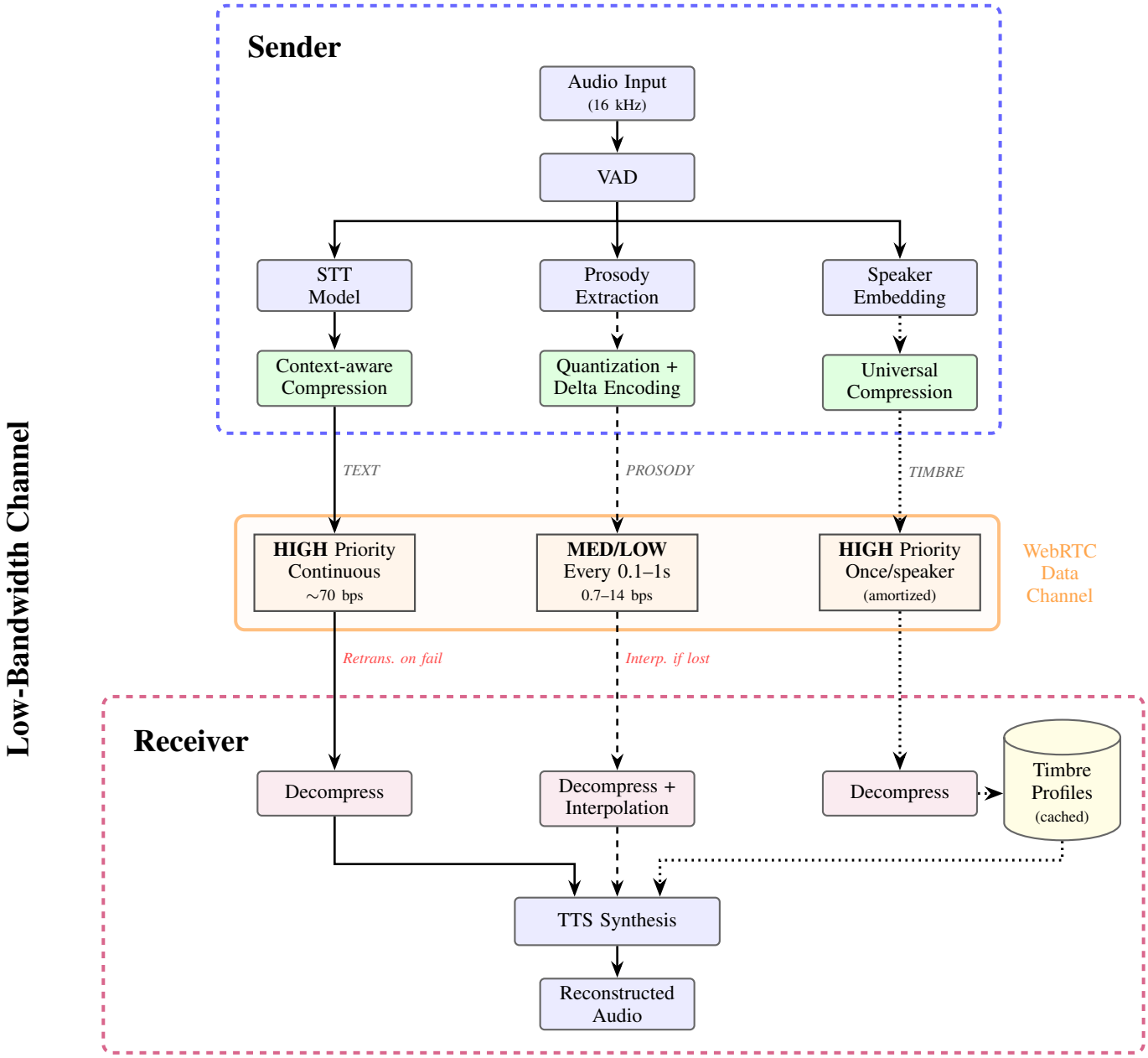


**Fig. 1:** STCTS system architecture. The sender decomposes speech into three orthogonal components—text (continuous, ∼70 bps), prosody (sparse keyframes, 0.7–14 bps), and timbre (one-time transmission, amortized)—each compressed with tailored strategies. These are transmitted via WebRTC data channels with prioritized delivery. The receiver reconstructs natural speech via TTS conditioning on all three components, with timbre profiles cached locally for recurring speakers.

- We demonstrate the computational feasibility of our approach on consumer hardware, achieving a Real-Time Factor (RTF) of ∼0.4 for the full pipeline (STT / factorization, compression, decompression, TTS / reconstruction). This confirms that the system can operate comfortably in real-time on a single consumer-grade GPU, validating the practical viability of semantic compression for live communication.
- We provide an open-source implementation with configurable quality modes and comprehensive benchmarking infrastructure, enabling reproducible evaluation and facilitating adoption for bandwidth-constrained communication scenarios. The complete source code is publicly available at https://github.com/dywsy21/STCTS.

Beyond bitrate efficiency, our modular architecture offers several advantages: flexible deployment support (scalable from edge devices to accelerated servers), privacy-preserving encryption (independent encryption of text, prosody, and speaker data), model upgradeability (drop-in replacement of STT/TTS components and text/timbre compression techniques as tech-

nology advances), and interpretable transmission (human-readable text enables debugging, logging, and secondary applications). These properties position STCTS as a versatile framework for diverse deployment scenarios including maritime communication, satellite IoT networks, tactical military systems, and large-scale voice social platforms.

## II. Related Works

Our approach integrates speech factorization, compression, and synthesis technologies to achieve ultra-low bitrate voice communication while preserving naturalness and speaker identity. This section surveys prior work across six key areas: (1) low-bitrate speech coding (covering traditional, neural, and semantic codecs), (2) STT-TTS architectures for bandwidth-constrained scenarios, (3) speech disentanglement methods that decompose speech into content, prosody, and timbre, (4) Speech-to-Text(STT) systems (particularly streaming and multilingual models), (5) text compression techniques (including neural language models), and (6) expressive text-to-speech(TTS) synthesis systems (focusing on voice cloning and prosody control). While individual components from these domains have been explored separately, our key contribution lies in their integration and optimization for natural, expressive communication at ultra-low bitrates—a scenario that demands different design choices than prior work in IoT communication (which sacrifices expressiveness) or speech disentanglement (which requires frame-level transmission).

### A. Low-Bitrate Speech Coding

**Traditional speech codecs.** Traditional speech codecs (e.g., Opus, EVS) can operate down to a few kb/s. For example, Opus supports bitrates down to ≈6 kb/s (wideband speech) [1], while specialized parametric codecs like Codec 2 handle ultra-low rates (0.7–3.2 kb/s) [1]. However, below ∼10 kb/s the quality of waveform coders degrades rapidly [2]. In response, neural and hybrid codecs have been developed to extend the range of intelligible speech at very low rates. Early work used linear-prediction plus RNN vocoders: LPCNet [3] codes speech at 1.6 kb/s in real time, yielding much higher quality than classic MELP. More recently, LSPNet [23] extends LPCNet to about 1.2 kb/s by encoding line-spectral pairs and employing a joint time-frequency loss; it reports quality superior to both traditional codecs and prior neural codecs at that rate.

**End-to-End Neural Codecs.** Modern neural audio codecs use deep autoencoders or generative models. For instance, SoundStream [4] uses a convolutional encoder and residual vector quantizer to compress audio at 3–18 kb/s; at 3 kb/s it outperforms Opus at 12 kb/s in subjective tests. EnCodec [5] uses a multi-band transformer VQ-VAE to compress high-fidelity (24 kHz) audio; by quantizing its latent space, it reduces bitrate by ≈40% with little loss in quality. Google's Lyra V2 [6] builds on SoundStream to deliver 3.2/6/9.2 kb/s modes for voice; at 6 kb/s it outperforms standard telco codecs (EVS/AMR-WB) and matches Opus quality while using only ∼50–60% of the bandwidth. MLow [7] is a low-complexity CELP-based codec optimized for 6 kb/s, reportedly doubling the perceptual quality of Opus at that rate (POLQA MOS ≈3.9 vs. 1.9) with ∼10% lower compute.

**Hybrid and Other Neural Codecs.** Hybrid schemes combine parametric encoders with neural decoders. For example, Skoglund & Valin [2] proposed decoding Opus parameters (6 kb/s) using neural vocoders: a listening test showed that synthesizing Opus 6 kb/s with LPCNet produced far better quality than Opus's standard decoder. Other work has explored GAN or diffusion-based codecs (e.g., AudioDec, DAC) but these require higher complexity. In the hybrid category, LPCNet and LSPNet (above) are prominent, as is RNNoise for noise suppression, and WaveRNN-based vocoders.

**Semantic/Generative Compression.** Recent research explores encoding high-level speech content rather than raw waveform. For instance, SemantiCodec [8] uses a transformer-based semantic encoder (AudioMAE features) plus an acoustic residual, compressing diverse audio (speech, music, SFX) into $<$100 tokens/sec ($<$1 kb/s) while preserving quality. Similarly, Collette et al. [9] propose a semantic compression using generative voice models to factor speech into content and style; their method achieves perceptual quality beyond EnCodec at ∼650 bps. These approaches suggest future codecs may transmit text-like representations (or semantic features) and reconstruct speech with high fidelity.

### B. STT-TTS Architectures for Low-Bandwidth Communication

The concept of using STT-TTS pipelines for bandwidth-constrained communication has been explored in IoT and satellite communication domains. Urazayev et al. [10] proposed a voice communication system for LoRaWAN networks that transmits text transcriptions over low-data-rate IoT channels (typically 0.3–5 kbps). Their approach achieves significant bandwidth reduction by converting speech to text at the sender, transmitting the compressed text, and synthesizing speech at the receiver using TTS. However, their work focuses primarily on IoT-specific challenges such as LoRaWAN protocol integration and network reliability, without explicit modeling of prosody or speaker identity. The synthesized speech uses generic voices without speaker adaptation, limiting its applicability to scenarios where speaker recognition and expressive communication are important.

Other work in satellite and tactical communication has explored similar STT-TTS architectures for bandwidth savings. These systems typically prioritize intelligibility and robustness over naturalness, often sacrificing prosodic expressiveness and speaker characteristics to minimize bitrate. While effective for mission-critical communications where only semantic content matters, such approaches are less suitable for general-purpose voice communication where users expect to recognize speakers and perceive emotional nuances.

Unlike prior STT-TTS systems that focus solely on semantic content transmission, our work explicitly models and transmits prosodic features and speaker embeddings alongside text. This enables us to preserve not just *what* is said, but also *how* it is said and *who* is speaking. Furthermore, we introduce novel

compression strategies tailored to each component: context-aware text compression, sparse prosody transmission with TTS interpolation, and amortized speaker embedding transmission. These innovations allow us to achieve ultra-low bitrates ($\sim$80 bps) while maintaining near-natural quality and speaker fidelity—a capability absent in prior STT-TTS systems designed for IoT or tactical scenarios.

### C. Speech Disentanglement

Speech disentanglement aims to decompose speech signals into independent representations corresponding to different factors of variation. Recent work has explored separating speech into content (text), timbre (speaker identity), and prosody components.

**End-to-End Disentanglement Methods.** SpeechTripleNet [11] proposes an end-to-end framework for disentangling speech into three orthogonal representations: linguistic content, speaker timbre, and prosodic information. The model uses adversarial training with three discriminators to ensure that each representation captures only its intended factor while remaining invariant to others. SpeechSplit [12] similarly decomposes speech into content, rhythm, pitch, and timbre using an autoencoder architecture with carefully designed information bottlenecks. These methods demonstrate that explicit factorization improves controllability in speech synthesis and voice conversion tasks.

**Self-Supervised Disentanglement.** More recent approaches leverage self-supervised learning to learn disentangled representations without explicit supervision. For instance, VQMIVC [13] uses vector quantization to enforce discrete content representations while learning continuous speaker embeddings, achieving high-quality voice conversion. DiscreTalk [14] extends this by incorporating prosody modeling through separate prosody encoders, enabling fine-grained control over speaking style during synthesis.

**Comparison with Our Approach.** While speech disentanglement methods share the goal of factorizing speech into content, prosody, and timbre, they differ fundamentally from our work in objective and design. Disentanglement methods are primarily concerned with *learning unsupervised representations* through adversarial training or self-supervised objectives, aiming to achieve clean separation of factors for downstream tasks like voice conversion, emotion transfer, or controllable synthesis. In contrast, our system operates in the *component-wise compression and transmission* domain, where the goal is to minimize bitrate while preserving perceptual quality. We leverage *off-the-shelf pre-trained models* (STT, prosody extractors, speaker embedding networks) rather than learning disentangled representations from scratch. This design choice offers several advantages: (1) modularity—components can be independently upgraded as better models emerge; (2) interpretability—transmitted text is human-readable, enabling logging and debugging; (3) efficiency—we exploit domain-specific compression strategies (e.g., Context-aware compression for text, sparse keyframe transmission for prosody) that would be difficult to integrate into end-to-end learned representations.

Furthermore, our prosody transmission strategy differs significantly from disentanglement methods. While prior work encodes prosody as continuous latent vectors or discrete tokens that must be transmitted at frame-level rates ($\sim$50–100 Hz), we exploit the *interpolation capability of modern TTS models* to transmit prosody at extremely sparse rates (0.1–1 Hz), reducing prosody bitrate to $<$14 bps—orders of magnitude lower than what frame-level transmission would require. This sparse transmission strategy is enabled by our recognition that conversational prosody varies smoothly over multi-second spans, allowing TTS models to reconstruct natural prosody contours from sparse keyframes.

### D. Speech-To-Text (STT) Systems

For speech-to-text in real time or low-resource settings, modern ASR models typically use powerful neural architectures, often with pre-training or streaming optimizations. OpenAI's Whisper [15] is a large encoder–decoder Transformer trained on 680k hours of multilingual audio; it supports ASR, translation, language ID, etc. across $\sim$100 languages. Whisper is highly robust to noise and accents and often outperforms specialized models on many benchmarks. Conformer [16] augments the Transformer with convolutional modules to capture local and global context; it achieved 2.1%/4.3% WER on LibriSpeech without an external language model, setting a new state of the art.

**Pretrained Self-Supervised Models.** Large SSL speech models (Wav2Vec 2.0, HuBERT, WavLM, etc.) are often fine-tuned for ASR. Delétang et al. note that models like Wav2Vec2, HuBERT, WavLM and Meta's MMS provide strong predictive performance but require task-specific fine-tuning [21]. These models enable ASR in low-resource scenarios by transferring knowledge from large unlabeled datasets.

**Streaming/Realtime Architectures.** For on-device or low-latency ASR, streaming RNN-Transducer or Conformer-Transducer models are used (e.g., Emformer, optimized Conformer). These allow incremental inference with limited lookahead. In practice, hybrid approaches combine small neural LM or CTC models with streaming encoders.

### E. Existing Compression Techniques

Existing compression techniques, especially text compression, typically uses entropy coding (Huffman, arithmetic or ANS) on top of a language model. Shannon's source-coding theorem implies an optimal code length of $-\log P(\text{token})$ bits; in practice one can feed LM-predicted probabilities into arithmetic coding for near-optimal compression [21]. For example, Delétang et al. note that lossless compression with a probabilistic model can be achieved by Huffman, arithmetic or ANS coding. In short, we would tokenize (e.g., wordpieces) and then apply arithmetic coding driven by a language model trained on transcripts.

Recent works show that large neural LMs vastly outperform classic compressors. Language Modeling is Compression [21]

demonstrated that a 70B-parameter Transformer (Chinchilla) can compress LibriSpeech to ∼16.4% of raw size – dramatically better than FLAC (30.3%). LMCompress [22] uses similar ideas across data types: it "shatters all previous compression records," achieving text compression at roughly one-third the size of the prior best text compressor (zpaq). These results imply that an off-the-shelf ASR model (if treated as a language model) could serve as a near-optimal text compressor. In practice, we would likely use a smaller LM or on-device model to balance speed and size. In summary, entropy coding guided by LMs yields the state of the art: Huffman/arithmetic coding on tokens using an ASR or NLP model would minimize the bits needed for transcripts. However, since its actual complete implementation is not publicly released yet and taking into account the potential limitations of computational resources (a tradeoff between Real Time Factor / RTF and Bitrate), as of now we do not employ LM-based text compression techniques.

### *F. Expressive Text-to-Speech (TTS) Systems*

A wide range of modern TTS models support expressive, speaker-specific synthesis from text plus prosody or embedding inputs. Notable examples include:

**Tacotron 2.** [24] An attention-based seq2seq model that predicts mel-spectrograms from text, followed by a neural vocoder. It "synthesizes speech with Tacotron-level prosody and WaveNet-level audio quality," achieving near-human sound quality. Tacotron2 requires a trained vocoder (e.g., WaveNet or HiFi-GAN) but produces very natural prosody.

**FastSpeech 2.** [25] A non-autoregressive Transformer-based TTS that conditions directly on duration, pitch, and energy extracted from speech. By training with ground-truth durations/intonation, FastSpeech2 avoids alignment issues and speeds up synthesis. It achieves 3× faster training and higher quality than original FastSpeech, even surpassing many autoregressive models. This makes it well-suited for real-time use.

**XTTS.** [26] A recent zero-shot multi-speaker TTS. Building on the Tortoise architecture, it is trained on 16 languages and can synthesize new voices and languages without fine-tuning. XTTS achieves state-of-the-art cross-lingual voice cloning performance in most of those languages. It demonstrates that massively multilingual, zero-shot cloning is feasible with large models.

**Zonos.** [30] An open-weight 1.6B model suite (Transformer and SSM hybrid) trained on ∼200k h of speech (English plus Chinese, Japanese, etc.). Zonos produces highly expressive, natural speech from text given a speaker embedding or audio example. It enables high-fidelity voice cloning from just 5–30 s of reference audio, and even allows control over speaking rate, pitch, and emotions (sadness, anger, etc.). The creators report that Zonos' quality matches/exceeds top proprietary TTS, and it outputs 44 kHz speech.

In all these TTS systems, one can provide prosody control signals (pitch/energy), and/or a learned speaker embedding (or reference audio) to capture the desired voice and style. Together with a high-quality vocoder (e.g., HiFi-GAN), these models can reconstruct speech that preserves the speaker's identity and expressiveness. This feature will prove crucial in our work.

## III. Methodology

We propose an end-to-end voice communication system that achieves ultra-low bitrate transmission (∼80 bps) by leveraging semantic compression through the STCTS pipeline. Unlike traditional waveform or neural audio codecs that operate in the acoustic domain, our approach transforms speech into a compressed semantic representation (text) with auxiliary prosodic and speaker information, then reconstructs natural-sounding speech at the receiver. This represents a $75\times$ reduction compared to standard Opus codec (∼6 kbps) and $12\times$ reduction compared to state-of-the-art neural codecs like EnCodec (∼1 kbps) while preserving audio quality and speaker characteristics. This section details the system architecture, individual components, and implementation choices.[1]

### *A. System Overview*

Our system consists of three main stages operating in a duplex communication channel:

**Stage 1: Speech Analysis and Encoding.** In this stage, the text, prosody and timbre information are extracted from the sender's audio. The sender's audio is processed through Voice Activity Detection (VAD) to filter out silence periods, followed by Speech-to-Text (STT) conversion to extract linguistic content. Simultaneously, prosody extraction captures intonation, rhythm, and speaking style, while speaker embedding extraction encodes voice timbre characteristics.

**Stage 2: Compression and Transmission.** The extracted features are compressed using appropriate algorithms: text compression via Brotli with optional preprocessing and context-aware compression, prosody quantization with delta encoding, and speaker embedding transmission with cache and change detection. These compressed packets are transmitted through a WebRTC data channel with priority-based queuing.

**Stage 3: Decompression and Speech Reconstruction.** The receiver decompresses the data stream and reconstructs speech through text decompression, prosody reconstruction with interpolation for missing frames, speaker-conditioned TTS synthesis, and audio playback with quality enhancement.

### *B. Speech-to-Text Module*

Our STT pipeline is designed to achieve real-time transcription with minimal latency while maintaining robustness across diverse acoustic environments. The system combines a state-of-the-art multilingual model with voice activity detection and streaming processing architecture to enable responsive speech recognition suitable for interactive communication.

[1]Parameters marked with * are configurable via our custom defined YAML configuration files. Values shown correspond to the balanced mode unless otherwise specified.

*1) Model Selection:* Our modular architecture supports drop-in replacement of STT models, enabling users to select engines optimized for their specific requirements (latency, accuracy, language coverage, or computational constraints). For the baseline implementation, we select FasterWhisper (small model*) based on its balanced performance across multiple criteria critical for real-time communication: (1) **Computational efficiency**—achieving low Real-Time Factor (RTF) on modern hardware, allowing us to trade computational power for bandwidth savings; (2) **robustness**—maintaining WER $<$10% on LibriSpeech under realistic noise conditions (SNR $>$10 dB) due to training on 680k hours of diverse audio; (3) **multilingual support**—covering 100+ languages without language-specific models, essential for global deployment; and (4) **streaming compatibility**—supporting chunked inference with minimal lookahead ($<$500ms), critical for interactive latency requirements. The CTranslate2-optimized implementation provides 3–4$\times$ speedup over vanilla Whisper while preserving accuracy within 0.5% WER [15]. Alternative models (e.g., Wav2Vec2 for low-resource languages, or Conformer-Transducer for minimal latency) can be substituted without modifying downstream components, as the system interfaces solely through transcribed text output.

*2) Voice Activity Detection (VAD):* To minimize bandwidth consumption and computational overhead, we implement Silero VAD [18] to detect speech segments. The VAD operates on 30ms audio frames with a speech probability threshold of 0.5*, minimum speech duration of 250ms to filter out false positives, and minimum silence duration of 500ms before considering a speech segment complete. This configuration reduces unnecessary processing of silence periods while maintaining responsiveness to natural speech patterns.

*3) Streaming Architecture:* The STT module processes audio in overlapping windows to enable low-latency transcription. We buffer audio chunks of 400ms* with 50ms overlap between consecutive windows. This design allows the system to begin transcription before a complete utterance finishes, maintain context across chunk boundaries through overlap, and achieve end-to-end latency under 500ms* from speech to text output.

To improve transcription quality in the streaming context, we implement a minimum buffer threshold of 25 audio chunks ($\sim$250ms) before initiating STT processing. Additionally, we enforce a minimum transcription length of 3 characters to filter out spurious detections from background noise.

Alternatively, the system supports a push-to-talk mode similar to walkie-talkie operation. In this mode, users activate a button to begin speaking, and audio is buffered locally until the button is released. The complete utterance is then transcribed and transmitted as a single packet, reducing network overhead and enabling more aggressive compression through larger context windows. This mode is particularly suitable for half-duplex communication scenarios or when network conditions require minimizing packet fragmentation.

### C. Prosody and Timbre Extraction

Beyond linguistic content, natural speech communication depends critically on prosodic cues (intonation, rhythm, emphasis) and speaker identity. Our system extracts compact representations of these paralinguistic features to enable expressive and personalized voice reconstruction at bitrates far lower than acoustic encoding would require.

*1) Prosody Feature Extraction and Encoding:* Prosodic information enables expressive communication while maintaining ultra-low bitrate through sparse transmission. We formalize the complete prosody encoding pipeline from raw audio to compressed bitstream.

**Feature Extraction.** Given input audio $\mathbf{x}[n]$ sampled at 16 kHz, we extract three prosody features at frame rate $f_p = 100$ Hz (10ms frame shift):

*Pitch Contour:* Fundamental frequency $F_0[t]$ is extracted using the YIN algorithm [20], which estimates the pitch period through autocorrelation analysis. For unvoiced frames (e.g., fricatives, silence), we set $F_0[t] = 0$. The raw pitch trajectory is log-scaled and normalized relative to the speaker's baseline log-pitch statistics $\mu_{F_0}$ and $\sigma_{F_0}$ (computed from the first 3 seconds):

$$\hat{F}_0[t] = \begin{cases} \frac{\log(F_0[t]) - \mu_{F_0}}{\sigma_{F_0}} & \text{if } F_0[t] > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\hat{F}_0[t]$ is the normalized log-pitch at frame $t$, $\mu_{F_0}$ is the mean log-pitch, and $\sigma_{F_0}$ is the standard deviation of the log-pitch. This normalization ensures speaker-independent representation and concentrates values around zero for efficient quantization.

*Energy Envelope:* RMS energy $E[t]$ is computed over 40ms windows ($N_w = 640$ samples):

$$E[t] = \sqrt{\frac{1}{N_w} \sum_{i=0}^{N_w - 1} x[tN_s + i]^2} \tag{2}$$

where $x[n]$ is the input audio signal, $N_w$ is the window size, and $N_s = 160$ is the hop size. Energy is normalized to the speaker's log-energy dynamic range:

$$\hat{E}[t] = \frac{\log(E[t] + \epsilon) - \log(E_{\text{min}})}{\log(E_{\text{max}}) - \log(E_{\text{min}})} \tag{3}$$

where $\hat{E}[t]$ is the normalized energy, $E_{\text{min}}$ and $E_{\text{max}}$ are the 5th and 95th percentiles of the speaker's energy distribution (computed over a 10-second sliding window), and $\epsilon = 10^{-6}$ prevents log-domain singularities.

*Speaking Rate:* We estimate instantaneous speaking rate $R[t]$ (syllables/second) through syllable nucleus detection. We apply a bandpass filter (300–3000 Hz) to extract the speech envelope, detect local maxima exceeding an adaptive threshold, and count nuclei within a 1-second centered window:

$$R[t] = \frac{1}{T_w} \sum_{\tau = t - T_w/2}^{t + T_w/2} \mathbb{1}[\text{nucleus detected at } \tau] \tag{4}$$

where $T_w = 1$ second is the window duration, and $\mathbb{1}[\cdot]$ is the indicator function which is 1 if a nucleus is detected at time $\tau$ and 0 otherwise. Speaking rate is then normalized relative to the speaker's baseline rate $\mu_R$ (typically 3–5 syllables/sec for conversational speech):

$$\hat{R}[t] = \frac{R[t] - \mu_R}{\sigma_R} \tag{5}$$

where $\hat{R}[t]$ is the normalized speaking rate, $\mu_R$ is the mean speaking rate, and $\sigma_R$ is the standard deviation of the speaking rate.

*2) Timbre as Speaker Embedding:* To preserve speaker identity, we extract a 192-dimensional* speaker embedding using the ECAPA-TDNN model* [19] from SpeechBrain [17]. This embedding is computed once at call initialization and updated only when speaker change is detected* with a cosine similarity threshold $< 0.7$* from the previous embedding. The embedding is quantized to float16 precision*, resulting in a 384-byte payload per transmission.

*3) Timbre Transmission Strategy:* Given the relatively large size of speaker embeddings, we employ an amortized transmission strategy. The full embedding is sent at call start (one-time 384-byte cost), with re-transmission only on speaker change detection. For a typical 45-second utterance, this 384-byte embedding amortizes to approximately 68 bps, representing about 42% of the total transmitted data. Over longer conversations, this cost further amortizes to 5–20 bps.

To further optimize bandwidth usage in multi-party or recurring conversations, the receiver maintains a persistent timbre profile database that stores speaker embeddings indexed by speaker identifiers. Before transmitting a full embedding, the sender queries whether the receiver already has the speaker's timbre profile cached. If a match is found (via speaker ID lookup), the sender transmits a TIMBRE_PROFILE packet containing only the speaker ID (typically 4–8 bytes) instead of the full 384-byte embedding. This lightweight reference packet instructs the receiver to retrieve the corresponding timbre from its local cache. When a previously unseen speaker or a speaker whose timbre has changed significantly is detected, a full TIMBRE packet with the complete embedding is sent. This caching mechanism is particularly effective in group calls or repeated conversations with the same participants, where speaker embeddings can be reused across sessions, effectively reducing timbre transmission to near-zero bitrate for familiar speakers.

### *D. Compression Pipeline*

The compression stage transforms extracted features into a minimal bitstream suitable for bandwidth-constrained transmission. We employ component-specific strategies tailored to each data type: semantic-aware compression for text, temporal delta coding for prosody, and amortized transmission, caching and universal compression techniques for speaker characteristics.

*1) Text Compression:* Text transcripts are compressed using Brotli* at compression level 5*, which provides a favorable balance between compression ratio and encoding speed [21]. For a 45-second conversational speech sample, Brotli-compressed text yields approximately 400 bytes or $\sim$70 bps for the text stream. This represents the dominant component of our total bitrate, accounting for roughly 85% of transmitted data.

Beyond standard compression, we implement context-aware optimization that exploits conversational structure. The system maintains a sliding window of recent transcripts to build a dynamic dictionary of frequently occurring words and phrases specific to the current conversation topic. This adaptive vocabulary allows shorter encodings for domain-specific terms (e.g., technical jargon, proper nouns, repeated phrases) that may not be well-represented in Brotli's static dictionary. Additionally, we employ predictive compression where the context from previous utterances helps predict likely continuations, enabling more efficient encoding of coherent discourse. For instance, in a medical consultation, terms like "prescription," "diagnosis," and "symptoms" are encoded with fewer bits after their first occurrence.

We implement optional text preprocessing*, including removal of filler words (um, uh, etc.), abbreviation of common phrases (going to $\rightarrow$ gonna), and punctuation minimization (retaining only periods and question marks essential for prosody). This preprocessing can reduce text volume by an additional 5–10% with acceptable degradation in naturalness.

*2) Prosody Compression:* The prosody stream is transmitted sparsely to minimize bandwidth consumption. Rather than sending continuous prosody features, we transmit prosody updates only when significant changes occur or at keyframe intervals (0.5 Hz*, every 2 seconds). In our minimal mode, prosody updates are sent as infrequently as 0.1 Hz, resulting in only 2–4 bytes of prosody data over a 45-second utterance. The receiver interpolates prosody between sparse updates*, maintaining naturalness while achieving near-zero prosody bitrate (less than 1 bps).

When prosody is transmitted, it undergoes three-stage compression inspired by traditional parametric coding approaches [3]:

**Sparse Sampling and Delta Encoding.** Rather than transmitting prosody at the native 100 Hz extraction rate, we employ sparse keyframe sampling at rate $f_k$ (configurable: 0.1–1 Hz). Let $\mathbf{p}[t] = [\hat{F}_0[t], \hat{E}[t], \hat{R}[t]]^\top$ denote the normalized prosody vector. We select keyframes at indices $\mathcal{T}_k = \{t_0, t_0 + \Delta t, t_0 + 2\Delta t, \ldots\}$ where $\Delta t = \lfloor f_p / f_k \rfloor$. For keyframes $t \in \mathcal{T}_k$, we compute temporal deltas:

$$\Delta\mathbf{p}[t] = \mathbf{p}[t] - \mathbf{p}[t - \Delta t] \tag{6}$$

The first keyframe transmits absolute values: $\Delta\mathbf{p}[t_0] = \mathbf{p}[t_0]$.

**Non-Uniform Quantization.** Delta values $\Delta\mathbf{p}[t]$ are quantized using a dead-zone uniform quantizer tailored to the sparse nature of the signal. For pitch deltas $\Delta\hat{F}_0[t]$ (quantized

to $b_F$ bits*, e.g., 6 bits):

$$\Delta\hat{F}_0^{(q)}[t] = \begin{cases} 0 & \text{if } |\Delta\hat{F}_0[t]| < \tau_F \\ \text{sign}(\Delta\hat{F}_0[t]) \cdot \lceil |\Delta\hat{F}_0[t]|/\alpha_F \rceil & \text{otherwise} \end{cases} \tag{7}$$

where $\tau_F = 0.05$ is a dead-zone threshold (suppressing imperceptible changes), and $\alpha_F$ is the quantization step size determined by the bit budget. This scheme efficiently captures significant prosodic shifts while ignoring minor fluctuations. Energy and speaking rate follow analogous quantization with $b_E = 5$ bits* and $b_R = 5$ bits*, respectively.

**Entropy Coding and Packetization.** Quantized delta vectors are entropy-coded using Huffman coding, which exploits the non-uniform distribution of prosody deltas (with strong concentration near zero). Each keyframe packet contains:

$$\text{Packet}_{\text{prosody}}(t) = [\text{timestamp}(t), \text{Huffman}(\Delta\hat{F}_0^{(q)}[t], \Delta\hat{E}^{(q)}[t], \Delta\hat{R}^{(q)}[t])] \tag{8}$$

For $f_k = 0.5$ Hz (balanced mode), this yields ∼16–20 bits per keyframe, resulting in 8–10 bps prosody bitrate. The receiver reconstructs continuous prosody at 100 Hz through cubic spline interpolation between received keyframes.

*3) Timbre Compression:* Speaker embeddings (192-dimensional vectors) are first quantized to lower precision (float16 or float32) to reduce the baseline payload size. To further minimize bandwidth, we apply universal lossless compression algorithms (zlib or Brotli) to the quantized byte stream. Since speaker embeddings often exhibit statistical redundancies, this additional compression step typically yields a 10–20% reduction in payload size without any loss of information beyond the initial quantization. This ensures that the critical speaker identity information is transmitted as efficiently as possible.

### *E. Network Transport and Reliability*

At ultra-low bitrates (∼80 bps), we employ a priority-based packet transmission strategy where each data type (TEXT, PROSODY, TIMBRE) is assigned a priority level reflecting its perceptual importance. The transport layer employs minimal packet headers (4–8 bytes) to reduce protocol overhead to below 10% of total bandwidth for typical packet sizes.

**Differentiated Reliability and Graceful Degradation.** We apply different reliability guarantees tailored to each stream's tolerance for loss:

- *TEXT* (HIGH priority): Requires absolute integrity due to the cascading failure mode of entropy coding—a single corrupted byte renders the entire compressed block undecompressable. Failed TEXT packets trigger immediate retransmission.
- *TIMBRE* (HIGH priority): Speaker embeddings are critical for identity preservation but transmitted infrequently (once per speaker, cached at the receiver once received). Loss and cache miss simultaneously occurring triggers retransmission.
- *PROSODY keyframes* (MEDIUM priority): Sparse prosody updates are retransmitted once if lost, as interpolation quality degrades significantly with missing keyframes.
- *PROSODY deltas* (LOW priority): Best-effort delivery without retransmission. The receiver interpolates through missing deltas with graceful degradation.

This tiered approach ensures that critical semantic information (text and speaker identity) maintains high reliability while tolerating graceful degradation in prosodic expressiveness under severe packet loss, consistent with the perceptual robustness to be demonstrated in our evaluation.

### *F. Text-to-Speech Synthesis*

The receiver reconstructs natural-sounding speech by conditioning a neural TTS model on the transmitted text, prosody, and speaker features. Our synthesis pipeline emphasizes voice fidelity and expressive control while maintaining real-time performance.

*1) Model Selection:* Similar to the STT module, our TTS architecture supports drop-in replacement of neural TTS models, enabling future adoption of more efficient or expressive synthesis systems as the field advances. As for now, we select Coqui XTTS-v2* [26] based on four critical requirements:

**(1) Zero-Shot Voice Cloning Performance:** XTTS-v2 achieves state-of-the-art speaker similarity (cosine similarity $> 0.85$ on LibriSpeech) from 3-second reference audio, enabling personalized voice communication without per-speaker training. Alternatives like YourTTS [27] achieve comparable similarity but require 10+ seconds of reference audio, increasing timbre overhead by $3\times$.

**(2) Explicit Prosody Conditioning:** The model architecture exposes pitch and energy conditioning interfaces through cross-attention mechanisms, allowing direct injection of transmitted prosody features. This explicit control is essential for expressive communication. Alternatives like VALL-E [28] use implicit prosody modeling via codec tokens, which cannot leverage our sparse prosody stream effectively.

**(3) Multilingual Capability:** XTTS-v2 supports 16 languages (including English, Mandarin, Spanish, French, German, etc.), matching the linguistic coverage of our Faster-Whisper STT frontend. This enables seamless cross-lingual communication without model switching.

**(4) Streaming Synthesis:** The model supports real-time streaming synthesis which is crucial for maintaining conversational responsiveness. Combined with HiFi-GAN vocoding, the system achieves real-time performance (RTF $\sim$ 0.4) on accelerated hardware, consistent with our design philosophy of leveraging compute to minimize bandwidth.

These criteria collectively ensure that XTTS-v2 integrates seamlessly with our sparse prosody transmission strategy while maintaining the perceptual quality necessary for natural communication. Future work may explore newer models such as StyleTTS-2 or Matcha-TTS as drop-in replacements if they demonstrate superior prosody controllability or efficiency.

*2) Prosody Conditioning:* Reconstructed prosody features are injected into the TTS model at multiple stages, following expressive TTS paradigms [24], [25]. Pitch contours modulate

the fundamental frequency of the generated mel-spectrogram, energy envelopes control the loudness of each frame, and speaking rate modulates the pace of token generation. This explicit conditioning ensures that the synthesized speech reflects the original speaker's expressive patterns.

*3) Speaker Conditioning:* The received speaker embedding serves as the reference for voice cloning. XTTS-v2 conditions its generation on this embedding through cross-attention mechanisms in the decoder. When the embedding is updated (speaker change or periodic refresh), the synthesizer adapts to the new voice characteristics within 1–2 seconds.

### G. Quality Modes

To accommodate varying network conditions, we define three operational modes with measured bitrates. Each mode is specified via a YAML configuration file that controls all system parameters, enabling flexible customization beyond the predefined profiles.

The predefined quality modes are:

**Minimal Mode.** Employs small STT model* with aggressive Brotli compression (level 9*) and text preprocessing*. Prosody updates at 0.1 Hz* transmitting only pitch feature* with minimal quantization (3-bit pitch*, 2-bit energy*, rate disabled*). Uses 192-dimensional speaker embedding* at float16 precision* with change detection threshold 0.4*. Designed for extreme bandwidth constraints such as legacy satellite links, 2G networks, or congested mobile connections where intelligibility takes priority over naturalness.

**Balanced Mode.** Uses small STT model* with Brotli compression (level 5*) balancing compression speed and efficiency. Prosody updates at 0.5 Hz* (every 2 seconds) with pitch, energy, and speaking rate features* quantized to 6-bit*, 5-bit*, and 5-bit* respectively. Includes emotion tracking at 0.2 Hz*. Uses 192-dimensional speaker embedding* (float16*) with change detection threshold 0.3*. This is the default mode providing the best quality-to-bitrate ratio, achieving substantially lower bitrates than neural audio codecs like Lyra [6] or EnCodec [5] while maintaining excellent perceptual quality.

**High Quality Mode.** Employs distil-large-v3 STT model* for maximum transcription accuracy with Brotli compression (level 5*). Prosody updates at 1.0 Hz* (every second) with all features* at high precision (8-bit pitch*, 6-bit energy*, 6-bit speaking rate*). Uses 192-dimensional speaker embedding* at full float32 precision* with stricter change detection threshold 0.25*. Disables text preprocessing* to preserve exact transcription. Additional audio processing parameters include 300ms chunk duration* and 0.4 VAD threshold*. Designed for stable 3G/4G/WiFi connections where accuracy and naturalness are prioritized over bandwidth efficiency.

Users can manually select the mode or enable adaptive switching based on measured network throughput and latency. The three-tier configuration provides clear tradeoffs: minimal mode maximizes compression for constrained networks, balanced mode optimizes the quality-bitrate ratio for typical scenarios, and high-quality mode prioritizes accuracy and naturalness when bandwidth permits. Custom configurations can be created by copying and modifying the YAML files, allowing fine-grained control over the bitrate-quality tradeoff for specific deployment scenarios.

### H. Implementation Details

The system is implemented in Python 3.11 using Faster-Whisper for STT, Silero VAD for voice activity detection, SpeechBrain for speaker embeddings, XTTS for synthesis, aiortc for WebRTC communication, and Brotli for text compression.

The sender and receiver run as asynchronous processes using Python's asyncio framework. Audio capture uses PyAudio with a 16 kHz sampling rate*, 16-bit depth, and 20ms frame size. The audio buffer chunk size* (1024 samples by default) and channel count* (mono by default) can be adjusted for different hardware configurations. The STT module processes audio in a dedicated thread pool to avoid blocking the main event loop. Similarly, TTS synthesis runs in a separate process to maintain real-time responsiveness.

A signaling server facilitates peer discovery and WebRTC connection establishment. The server is implemented using WebSockets and handles peer registration, session negotiation, and ICE candidate exchange. Once the WebRTC connection is established, all voice data bypasses the signaling server and flows directly peer-to-peer.

## IV. Experiments

We conduct comprehensive experiments to evaluate our STCTS system across multiple dimensions: bitrate efficiency, transcription accuracy, voice identity preservation, perceptual speech quality, noise resilience and computational efficiency. We first conduct *prosody sampling rate analysis* to determine the optimal prosody sampling rate for our three quality modes, and then our evaluation compares three quality modes (minimal, balanced, and high-quality) against baseline codecs (Opus and EnCodec) and the Vevo framework [9] using standardized metrics and a large-scale speech corpus.

### A. Experimental Setup

*1) Dataset:* We evaluate our system on the LibriSpeech corpus [31], a widely-used benchmark for speech recognition research. LibriSpeech contains approximately 1,000 hours of read English speech derived from audiobooks in the LibriVox project, carefully segmented and aligned at 16 kHz sampling rate. The corpus is partitioned into multiple subsets based on acoustic conditions and speaker characteristics.

For our experiments, we evaluate our setup and the baseline setups on the full `test-clean` subset, which contains high-quality recordings with minimal background noise and clear articulation. This subset provides a controlled environment for measuring system performance under ideal acoustic conditions. Each sample contains complete sentences or utterances ranging from 5 to 30 seconds in duration, spoken by diverse speakers (both male and female) with various accents and speaking styles. We report mean values across all samples

along with standard deviations in order to ensure statistical reliability.

It is important to clarify that while STCTS is designed for conversational voice communication, our evaluation primarily assesses the *reconstruction quality* of the compression pipeline (STT $\rightarrow$ Compression $\rightarrow$ TTS) rather than conversational dynamics (e.g., turn-taking latency, overlapping speech). Since the core challenge lies in reconstructing intelligible and expressive speech from ultra-low bitrate semantic representations, single-channel read speech from LibriSpeech provides a rigorous and standardized benchmark for this purpose. The conversational aspects are handled by the networking layer (WebRTC) and do not fundamentally alter the factorization and compression algorithm's performance characteristics. Therefore, evaluating on a high-quality read speech corpus allows us to isolate and precisely measure the fidelity of our semantic reconstruction approach.

The choice of `test-clean` allows us to isolate the compression artifacts and reconstruction quality from environmental noise factors. We separately evaluate noise resilience in Section IV-C using augmented test sets with additive noise at various signal-to-noise ratios.

*2) Baseline Systems:* We compare our approach against three representative baseline systems:

**Opus (6 kbps).** A widely-deployed traditional codec operating at its lowest recommended bitrate for wideband speech. Opus uses SILK for speech and CELT for music, with hybrid packet loss concealment. This represents the state-of-the-art in traditional parametric coding.

**EnCodec (1 kbps).** A recent neural audio codec using multi-scale VQ-VAE architecture. EnCodec represents the current frontier in neural waveform compression, achieving significantly lower bitrates than traditional codecs while maintaining reasonable quality.

**Vevo Framework ($\sim$650 bps).** A semantic compression system proposed by Collette et al. [9] that similarly uses generative voice models to decompose speech into content and style components. Vevo achieves ultra-low bitrates ($\sim$650 bps) comparable to our approach. However, since Vevo was evaluated on a different test set than ours, we report their published metrics as reference values only. These values provide context for our results but cannot be directly compared due to dataset differences. We mark Vevo results with an asterisk (*) in all tables to indicate this distinction.

*3) Evaluation Metrics:* We employ a comprehensive suite of metrics covering bitrate, intelligibility, speaker fidelity, and perceptual quality:

**Bitrate Metrics.** We measure the total transmitted bitrate in bits per second (bps), decomposed into three components:

- **Text, Prosody, Timbre Bitrates:** Individual component bitrates to analyze bandwidth allocation.
- **Total Bitrate w/o Timbre:** Sustained bitrate excluding the one-time speaker embedding transmission, representing the long-term bandwidth consumption.

Bitrates are measured over the complete utterance duration including silence periods detected by VAD. For timbre, we amortize the one-time 384-byte embedding transmission over the utterance duration to compute an equivalent bitrate.

**STT Accuracy: Word Error Rate (WER).** We measure transcription accuracy using Word Error Rate, defined as:

$$\text{WER} = \frac{S + D + I}{N}$$

where $S$, $D$, $I$ are the number of substitutions, deletions, and insertions required to transform the recognized text into the reference transcription, and $N$ is the total number of words in the reference. Lower WER indicates better intelligibility. We compute WER by transcribing both the original and reconstructed audio using the same FasterWhisper model, then comparing the two transcriptions to isolate compression-induced errors.

**Speaker Similarity (SpkrSim).** We evaluate voice identity preservation by computing the cosine similarity between ECAPA-TDNN speaker embeddings [19] extracted from the original and reconstructed audio:

$$\text{SpkrSim} = \frac{\mathbf{e}_{\text{orig}} \cdot \mathbf{e}_{\text{recon}}}{\|\mathbf{e}_{\text{orig}}\| \|\mathbf{e}_{\text{recon}}\|}$$

where $\mathbf{e}_{\text{orig}}$ and $\mathbf{e}_{\text{recon}}$ are the 192-dimensional speaker embeddings. Values above 0.85 typically indicate the same speaker according to standard equal error rate (EER) thresholds in speaker verification systems. This metric assesses how well the TTS synthesis preserves the original speaker's timbre and voice characteristics.

**Perceptual Evaluation of Speech Quality (PESQ).** PESQ [32] is an ITU-T standard (P.862) for objective speech quality assessment in telecommunications. It compares the original and degraded signals in a perceptually-weighted frequency domain, producing scores from $-0.5$ to $4.5$ (higher is better). PESQ correlates well with subjective Mean Opinion Score (MOS) ratings, with scores above 2.5 considered acceptable telephony quality and above 3.5 considered excellent. We use the wideband (16 kHz) mode for all evaluations.

**Short-Time Objective Intelligibility (STOI).** STOI [33] measures speech intelligibility by computing the normalized correlation between short-time segments of the original and processed signals in a temporal-frequency domain. It produces scores from 0 to 1, where higher values indicate better intelligibility. STOI has been validated to correlate strongly with human speech reception thresholds and is particularly effective for assessing intelligibility in noisy or distorted conditions.

**Non-Intrusive Speech Quality Assessment (NISQA).** Unlike PESQ and STOI which require reference audio (intrusive metrics), NISQA [34] is a deep learning-based non-intrusive metric that predicts speech quality from the degraded signal alone. It produces a Mean Opinion Score (MOS) ranging from 1 to 5, where 5 represents excellent quality. NISQA additionally provides four quality dimensions: noisiness, coloration (spectral distortion), discontinuity (temporal artifacts), and loudness appropriateness. We report the overall MOS score as the primary quality indicator. NISQA's non-intrusive nature makes it particularly valuable for evaluating synthesis artifacts that may not be captured by intrusive metrics.

*4) Experiment Implementation Details:* All experiments are conducted on a system equipped with a single NVIDIA RTX 4080 GPU and an Intel Core i9-13900KF CPU. For baseline comparisons, Opus encoding uses the `libopus` library with VBR mode disabled and bitrate constrained to 6 kbps. EnCodec uses the official implementation with the 24 kHz model quantized to 1 kbps (single codebook). Both baselines process the same test audio samples for fair comparison.

### *B. Prosody Sampling Rate Analysis*

Before establishing our three operational modes (minimal, balanced, high-quality), we systematically investigate the relationship between prosody sampling rate and system performance. Prosody sampling rate fundamentally determines the temporal granularity at which expressive features are transmitted, directly impacting both bandwidth consumption and reconstruction quality. Understanding this tradeoff is essential for informed configuration design.

We conduct a parameter sweep experiment on the LibriSpeech test-clean dataset, varying prosody sampling rate from 0.05 Hz (one update every 20 seconds) to 20 Hz (20 updates per second) while keeping all other parameters constant (small STT model, Brotli level 5, 192-dim float16 speaker embedding). For each sampling rate, we measure: (1) total bitrate (including amortized timbre), (2) prosody bitrate contribution, (3) transcription accuracy (WER), (4) speaker similarity, (5) perceptual quality metrics (PESQ, STOI, NISQA), and (6) NISQA dimensional breakdown (noisiness, coloration, discontinuity, loudness).

Figure 2 presents the comprehensive results across four key dimensions. As shown in the top-left panel, bitrate increases approximately linearly with prosody sampling rate, rising from $\sim$312 bps at 0.05 Hz to $\sim$592 bps at 20 Hz (including amortized timbre). This linear relationship confirms that prosody transmission dominates the additional bandwidth consumption beyond the text baseline when prosody sampling rate is high.

The quality metrics exhibit a striking **bimodal distribution** with prosody sampling rate, as evident in the top-right and bottom-left panels. NISQA MOS scores show two distinct quality peaks separated by a substantial valley: a low-frequency cluster with local maxima occurring between 0.05–1 Hz, followed by quality degradation in the 1–6 Hz range, and then a surprising *high-frequency peak at 7 Hz* (MOS = 4.317). This bimodal behavior contradicts the intuitive expectation that prosody sampling rate would exhibit monotonic or unimodal quality trends.

It is important to note that the prosody sampling rate analysis is *only meaningful with regard to the current system settings and STT/TTS components*: when a newer state-of-the-art STT/TTS model is used as an alternative, the prosody sampling rate analysis needs to be reconducted to faithfully demonstrate the behavior of the new system. Nonetheless, we believe that the bimodal distribution pattern observed here may generalize to other TTS architectures that employ interpolation-based prosody synthesis, as this phenomenon appears to be rooted in fundamental perceptual properties rather than model-specific artifacts.

We hypothesize this phenomenon arises from the interaction between prosody temporal granularity, TTS interpolation mechanisms, and perceptual salience of artifacts:

**Low-frequency peaks (0.05–1Hz):** At sparse prosody update rates (one update every 1–20 seconds), the TTS model's interpolation generates smooth prosody contours between keyframes. These rates align with natural sentence-level prosody in conversational speech, where intonation patterns span multi-second utterances. This regime provides anchoring for paragraph-level prosody flow while capturing individual sentence boundaries, without over-constraining the TTS model. The sparse updates allow the model's internal prosody generation to produce natural micro-variations within interpolated segments, achieving high quality with minimal bitrate overhead.

**Mid-frequency valley (1–6Hz):** Prosody updates at 0.167–1 second intervals create a perceptual "uncanny valley." Updates occur too frequently to benefit from smooth TTS interpolation, yet too infrequently to provide fine-grained prosody control. This regime may introduce *prosodic discontinuities*—abrupt transitions between update points that disrupt natural pitch and energy contours. The NISQA discontinuity dimension (bottom-right panel) shows degradation in this range, supporting this hypothesis. Critically, these rates misalign with both slow sentence-level prosody (handled well in the 0–1 Hz range) and fast syllable-level timing (requiring $>$6 Hz), falling into a "worst of both worlds" regime where artifacts are perceptually salient.

**High-frequency peak (7Hz):** The quality recovery at 7 Hz represents a qualitatively different operating mode. At 7 Hz, prosody keyframes occur at near-syllabic resolution, providing frame-level control that overrides TTS interpolation entirely. This high temporal resolution bypasses the discontinuity issues of the mid-frequency range by making transitions imperceptibly short. The 7 Hz rate likely captures detailed prosody variations (pitch accents, stress patterns, hesitations) that sparse rates miss, similar to how high-framerate video captures motion smoothness. However, this comes at significantly higher bitrate (410 bps vs. 154 bps at 0.7 Hz), raising the question of whether the quality gain justifies the bandwidth cost.

**Quality-bitrate considerations.** The low-frequency peaks achieve MOS $\sim$4.30–4.36 at 132–154 bps, offering exceptional efficiency. The 7 Hz peak reaches MOS = 4.317 at 410 bps—comparable quality but at nearly $3\times$ the bandwidth cost. The mid-frequency valley (1.2–4 Hz) is strictly dominated, with lower quality than both flanking peaks despite intermediate bitrate.

The PESQ and STOI metrics (bottom-left panel) largely corroborate the bimodal NISQA trend, though with greater variance. Speaker similarity remains stable across all rates, confirming that prosody sampling frequency affects *expressiveness* rather than speaker identity.

This bimodal distribution suggests two distinct deployment


**Fig. 2:** Prosody sampling rate analysis. **Top-left:** Bitrate vs. quality tradeoff showing optimal operating regions. **Top-right:** Main quality metrics (NISQA MOS, WER, Speaker Similarity) across sampling rates and the bitrate breakdown. **Bottom-left:** Perceptual quality metrics (PESQ, STOI) demonstrating the bimodal phenomenon. **Bottom-right:** NISQA dimensional breakdown revealing specific quality aspects affected by prosody sampling.

strategies. For *bandwidth-constrained* scenarios (e.g., satellite links, IoT devices), operate in the 0–1 Hz range to maximize quality per bit. For *quality-prioritized* applications (e.g., professional communication, accessibility services), 7 Hz seems to achieve high absolute quality, though the marginal gain over the low-frequency peaks may not justify the bitrate increase in most contexts. Critically, *avoid* the mid range, which offers neither efficiency nor quality—representing a "dead zone" in the design space.

**Configuration Design Based on Analysis.** The insights from this prosody sampling rate analysis directly informed the design of our three operational modes. For the **minimal mode**, we selected 0.1 Hz from the low-frequency peak region, achieving maximum bandwidth efficiency while maintaining acceptable quality. This configuration targets extreme bandwidth constraints where every bit counts. For the **balanced mode**, we chose 0.5 Hz as a compromise that remains within the low-frequency efficiency region while providing more frequent prosody updates for improved expressiveness. This rate sits in between the low-frequency comfort zone, maximizing quality-per-bit while avoiding the discontinuity artifacts observed at 1–5 Hz. For the **high-quality mode**, we selected 1.0 Hz, which represents the upper boundary of the low-frequency regime. Although our analysis identified 7 Hz as a high-frequency peak, the quality improvement is negligible compared to the 0.05–1 Hz peaks, while the bitrate cost is significantly higher. The 1.0 Hz rate strikes a better balance, providing sentence-level prosody guidance without excessive bandwidth consumption, and critically, avoids entering the mid-frequency valley where quality degrades. These carefully calibrated sampling rates explain why our three modes exhibit such narrow total bitrate variation: prosody transmission contributes minimally even in high-quality mode, while compressed text dominates the bandwidth budget. This design philosophy prioritizes *bandwidth efficiency* by leveraging TTS interpolation capabilities rather than brute-force transmission of dense prosody features.

### C. Benchmark Results

Tables I and II present the comprehensive benchmark results comparing our three quality modes (minimal, balanced, and high-quality) against baseline codecs (Opus and EnCodec) and the Vevo framework. Table I shows the bitrate breakdown across different components, while Table II presents perceptual quality metrics. The high-quality mode is additionally evaluated under various noise conditions (0.1%, 1%, and 10% bit error rates) to assess robustness to channel degradation.

*1) Performance Evaluation:* **Bitrate Efficiency.** As shown in Table I, our system achieves remarkably low bitrates across all three quality modes. The total bitrate (excluding timbre amortization) ranges from 71.6 bps (minimal) to 79.6 bps (high-quality), representing a $40\times$ to $50\times$ reduction compared to Opus at 6 kbps and an $8\times$ reduction compared to EnCodec at 1 kbps. Notably, our bitrate is comparable to the Vevo framework ($\sim$650 bps) while maintaining similar perceptual quality. The timbre column shows the amortized bitrate for speaker embeddings over the test utterances (mean duration $\sim$12 seconds); however, in practical long-duration conversations, this cost amortizes to near-zero as discussed in Section III-C. Furthermore, our timbre profile caching mechanism (Section III-C) enables reuse of speaker embeddings across sessions, effectively eliminating timbre transmission overhead for familiar speakers in multi-party or recurring conversations. Therefore, the Total (bps) column represents the sustained bandwidth consumption for voice content transmission.

Our three quality modes exhibit minimal bitrate variation (71.6–79.6 bps), with differences primarily in prosody transmission frequency (0.7–13.9 bps) rather than text compression: text compression dominates the total bandwidth under relatively low prosody sampling rate (Section IV-B), while prosody contributes only marginally even in high-quality mode. Consequently, users can adopt the high-quality mode with negligible additional bandwidth cost ($<$10 bps overhead) while gaining substantial improvements in transcription accuracy (WER reduction from 0.259 to 0.235) and prosody fidelity. We therefore recommend the high-quality mode as the default configuration for most deployment scenarios.

**Perceptual Quality Metrics.** Table II presents the quality assessment results. Our system achieves speaker similarity scores (0.667–0.673) comparable to Opus (0.673) and substantially higher than EnCodec (0.450), indicating successful preservation of speaker identity through our embedding-based voice cloning approach. The PESQ scores (1.138–1.324) are competitive with EnCodec (1.334), though lower than Opus (2.284). This is expected, as PESQ is designed for waveform-level distortions in traditional codecs and does not fully capture the quality of synthesized speech. More importantly, our NISQA MOS scores (4.255–4.280) significantly exceed both Opus (2.455) and EnCodec (2.083), and match the Vevo framework (4.21). NISQA, as a non-intrusive deep learning-based metric trained on diverse speech quality dimensions, better reflects the perceptual naturalness of TTS-synthesized speech. The high NISQA scores confirm that our semantic compression approach produces highly natural and intelligible speech despite the ultra-low bitrate.

**STOI Analysis.** Our STOI scores (0.150–0.162) are notably lower than both Opus (0.906) and EnCodec (0.805), which might initially suggest poor intelligibility. However, this requires careful interpretation. STOI measures short-time objective intelligibility by computing temporal-spectral correlation between the original and reconstructed signals at the frame level (typically 10–30ms frames). It implicitly assumes *frame-level temporal alignment* between reference and degraded audio—an assumption that holds for waveform codecs but fails for our TTS-based reconstruction approach.

In our system, the TTS synthesis process introduces *temporal desynchronization* at multiple stages. First, the STT module may produce slightly different word boundaries than the original speech due to recognition uncertainty. Second, the TTS model generates speech with its own learned timing patterns conditioned on text and sparse prosody keyframes, which may not precisely match the original speaker's micro-timing (e.g., pause durations, consonant lengths, syllable boundaries). Third, our prosody features are transmitted at extremely low rates (0.1–1 Hz), providing only coarse temporal guidance rather than frame-by-frame alignment. As a result, even when the synthesized speech is highly intelligible and natural-sounding (as confirmed by high NISQA scores), the frame-by-frame waveform correlation measured by STOI remains low due to temporal shifts.

This phenomenon is inherent to semantic compression approaches that decouple content from acoustic realization. We include STOI in our evaluation not as a primary quality indicator, but to characterize the *temporal alignment properties* of our reconstruction method and distinguish it from waveform-preserving codecs. For assessing actual intelligibility in semantic compression systems, metrics like WER (which measures content preservation) and NISQA (which evaluates perceptual naturalness) are more appropriate than STOI.

*2) Noise Resilience:* To assess the robustness of our system under degraded channel conditions, we evaluate the high-quality mode on the LibriSpeech test-clean dataset with simulated bit error rates (BER) of 0.1%, 1%, and 10%, since it's been demonstrated that high-quality mode is fit to be the default configuration for most scenarios.

**Transcription Accuracy Degradation.** WER exhibits sensitivity to channel noise, rising from 0.213 (no noise) to 0.258 (0.1% BER), then stabilizing around 0.247–0.252 at higher error rates. Interestingly, WER does not continue to degrade linearly at higher BERs (1% and 10%), and even shows slight improvement. We hypothesize this occurs because the receiver may rely more heavily on linguistic context and prosody cues to infer missing words, partially compensating for corrupted text data.

**Speaker Identity Preservation.** Speaker similarity remains remarkably stable across noise conditions, varying only slightly from 0.666 (no noise) to 0.658–0.669 under noise. This robustness arises from the amortized transmission strategy for speaker embeddings (Section III-C). The 384-byte

**TABLE I:** Bitrate comparison on LibriSpeech test-clean dataset. Values shown as mean ± standard deviation. Our system achieves $40\times$ to $50\times$ bitrate reduction compared to Opus (6 kbps) and $8\times$ reduction compared to EnCodec (1 kbps). Vevo* results from [9] with Zonos TTS and timbre update interval $d_{\text{timbre}} = 4$s, evaluated on a different test dataset and provided for reference only.

| Method | Total (bps) | Text (bps) | Prosody (bps) | Timbre (bps) |
|---|---|---|---|---|
| *Our System* | | | | |
| minimal | 71.6±8.8 | 70.9±8.8 | 0.7±0.1 | 125.9±19.6 |
| balanced | 76.5±8.8 | 71.0±8.8 | 5.5±0.2 | 125.9±19.6 |
| high-quality | 79.6±8.9 | 65.8±8.9 | 13.9±0.2 | 251.8±39.2 |
| *Baseline Systems* | | | | |
| Opus (6 kbps) | 6407.3±217.2 | — | — | — |
| EnCodec (1 kbps) | 999.9±0.1 | — | — | — |
| Vevo* | 650 | — | — | — |

**TABLE II:** Quality metrics comparison on LibriSpeech test-clean dataset. Values shown as mean ± standard deviation. Our system maintains comparable or superior perceptual quality (NISQA MOS) to the Vevo framework while achieving lower WER than EnCodec. Vevo* results from [9] with Zonos TTS and timbre update interval $d_{\text{timbre}} = 4$s, evaluated on a different test dataset and provided for reference only.

| Method | Noise Condition | WER | Speaker Sim | PESQ | NISQA MOS | STOI |
|---|---|---|---|---|---|---|
| *Our System* | | | | | | |
| minimal | — | 0.259±0.204 | 0.673±0.085 | 1.238±0.381 | 4.280±0.348 | 0.150±0.036 |
| balanced | — | 0.264±0.203 | 0.672±0.095 | 1.138±0.130 | 4.258±0.393 | 0.155±0.041 |
| high-quality | — | 0.235±0.193 | 0.667±0.089 | 1.324±0.417 | 4.255±0.407 | 0.152±0.038 |
| *Noise Resilience (high-quality mode)* | | | | | | |
| high-quality | No noise | 0.213±0.167 | 0.666±0.092 | 1.332±0.407 | 4.298±0.400 | 0.162±0.034 |
| high-quality | 0.1% BER | 0.258±0.228 | 0.669±0.095 | 1.250±0.388 | 4.263±0.379 | 0.156±0.030 |
| high-quality | 1% BER | 0.252±0.189 | 0.658±0.091 | 1.279±0.386 | 4.246±0.345 | 0.156±0.038 |
| high-quality | 10% BER | 0.247±0.195 | 0.663±0.096 | 1.272±0.494 | 4.232±0.432 | 0.148±0.031 |
| *Baseline Systems* | | | | | | |
| Opus (6 kbps) | — | 0.032±0.048 | 0.673±0.058 | 2.284±0.349 | 2.455±0.371 | 0.906±0.027 |
| EnCodec (1 kbps) | — | 0.110±0.092 | 0.450±0.074 | 1.334±0.112 | 2.083±0.393 | 0.805±0.029 |
| Vevo* | — | 0.15 | 0.62 | 1.15 | 0.70 | 4.21 |

TIMBRE packet is transmitted only once at call initialization and re-sent only upon speaker change detection. This infrequent transmission makes timbre less susceptible to channel noise compared to continuously streamed data. Furthermore, speaker embeddings are transmitted with high priority and retransmission guarantees (Section III-D), ensuring reliable delivery even under noisy conditions. The minimal variation in speaker similarity confirms that voice identity is well-preserved regardless of channel quality.

**Perceptual Quality Preservation.** PESQ scores show slight degradation from 1.332 (no noise) to 1.250–1.279 under noise, a decline of ∼4–6%. STOI decreases slightly from 0.162 to 0.148–0.156, and NISQA MOS exhibits a gradual downward trend from 4.298 to 4.232 as BER increases. These modest degradations (NISQA drops only ∼1.5% even at 10% BER) indicate that perceptual quality remains acceptable under noisy conditions. The graceful degradation can be attributed to our prioritized transmission strategy: TEXT packets receive high priority with retransmission, ensuring semantic content integrity, while low-priority prosody packets may be dropped under congestion. When prosody packets are lost, the receiver interpolates missing frames from adjacent keyframes (Section III-E), maintaining naturalness at the cost of reduced expressiveness. This design philosophy prioritizes *intelligibility over expressiveness*—users can still understand the speech content even when fine-grained prosody is compromised. Even at 10% BER—a severely degraded channel—the system maintains NISQA MOS above 4.2, indicating excellent perceptual quality and confirming the robustness of our semantic compression approach to channel impairments.

### *D. Computational Efficiency*

We evaluate the computational efficiency of our system by measuring the Real-Time Factor (RTF), defined as the ratio of processing time to audio duration. An RTF less than 1.0 indicates that the system processes audio faster than real-time, a critical requirement for live communication. Table III presents the RTF results across different configurations and noise conditions.

Our system consistently achieves an RTF of approximately 0.4 across all modes, meaning it requires only 40% of the audio duration to process the full pipeline (STT, compression, transmission, decompression, and TTS). This performance demonstrates that the system is well-suited for real-time deployment, leaving ample headroom for other concurrent tasks.

**TABLE III:** Real-Time Factor (RTF) analysis on LibriSpeech test-clean dataset. Experiments conducted on a single NVIDIA RTX 4080 GPU with Intel Core i9-13900K CPU. RTF $< 1.0$ indicates faster than real-time processing.

| Config | Noise | RTF |
|---|---|---|
| minimal_mode | No noise | $0.396 \pm 0.080$ |
| balanced_mode | No noise | $0.404 \pm 0.074$ |
| high_quality_mode | No noise | $0.387 \pm 0.046$ |

One of our core design choices is that *we explicitly trade computational power for bandwidth efficiency.* By leveraging GPU acceleration for the neural components (STT and TTS), we achieve ultra-low bitrates ($\sim$80 bps) that would be impossible with traditional lightweight codecs. This design choice reflects the economic reality of our target scenarios (e.g., maritime, satellite), where bandwidth is the scarce and expensive resource, while computational power (even if requiring a GPU) is mostly a one-time capital investment that is relatively inexpensive compared to the recurring operational cost of satellite data.

### *E. Discussion*

Our experimental evaluation demonstrates that the STCTS pipeline achieves ultra-low bitrate voice communication ($\sim$80 bps sustained bandwidth) while maintaining high perceptual quality comparable to state-of-the-art semantic compression systems. It is noteworthy that STCTS represents an extreme audio compression approach, achieving compression ratios exceeding 3000:1 compared to uncompressed PCM audio (16 kHz, 16-bit: 256 kbps $\rightarrow$ 80 bps), and 80:1 even against highly optimized traditional codecs like Opus at 6 kbps. This radical compression is enabled by discarding acoustic waveform fidelity entirely and reconstructing speech from purely semantic and prosodic representations, fundamentally redefining the trade-off between bitrate and perceptual quality in voice communication. Beyond bitrate efficiency and perceptual quality, our explicit semantic decomposition approach offers several architectural advantages over end-to-end neural codecs and token-based semantic compression.

**Compute-Bandwidth Trade-off.** A core design principle of STCTS is the strategic exchange of computational power for bandwidth efficiency. While traditional codecs minimize compute to run on minimal hardware, we leverage modern accelerators (e.g., GPUs, NPUs) to perform sophisticated semantic analysis and synthesis, thereby reducing bandwidth consumption by orders of magnitude. This trade-off is economically advantageous in our target scenarios (maritime, satellite, tactical), where bandwidth is the scarce, recurring cost (e.g., \$10/MB), whereas computational hardware is a one-time fixed cost. Our evaluation confirms that with a single consumer GPU (RTX 4080), the system runs comfortably faster than real-time (RTF $\sim$0.4), validating the feasibility of this approach.

**Privacy-Preserving End-to-End Encryption.** The textual intermediate representation enables straightforward application of standard encryption protocols (e.g., AES-256, RSA) to protect semantic content during transmission. Since text, prosody features, and speaker embeddings are explicitly structured data, they can be encrypted independently with different keys or access policies, enabling fine-grained privacy controls. For instance, text content can be encrypted end-to-end between callers, while prosody features (which convey emotion but not semantic content) might use a weaker encryption level or remain unencrypted for network optimization. This flexibility contrasts with neural codec latent representations, which are entangled high-dimensional vectors that resist selective encryption. Furthermore, the explicit text representation facilitates compliance with data protection regulations (e.g., GDPR right to explanation), as transmitted content is human-interpretable rather than opaque neural activations.

**Modular Design and Model Upgrade-ability.** The decoupled STCTS pipeline allows independent upgrading of each component without retraining the entire system. As more accurate STT models emerge (e.g., future Whisper versions, domain-specific ASR), they can be seamlessly integrated by replacing the STT module while retaining existing compression and TTS components. Similarly, advances in TTS (e.g., improved voice cloning, lower-latency synthesis) can be adopted without modifying the upstream pipeline. This modularity also enables domain-specific optimization: medical consultation systems can employ specialized medical STT models and terminology-aware text compression, while casual conversation systems use general-purpose models. In contrast, end-to-end neural codecs require full model retraining to incorporate improvements, and their monolithic architecture resists task-specific customization.

**Inherent Interpretable Intermediate Representation.** The explicit textual representation provides transparency and debuggability absent in neural codec latent spaces. System developers can inspect transmitted text to diagnose transcription errors, measure content-level bitrate allocation, and implement content-aware optimizations (e.g., domain-specific dictionaries, phrase prediction). Users can optionally view transcriptions in real-time for accessibility (e.g., hearing-impaired communication) or quality assurance (e.g., verifying critical instructions in aviation or telemedicine). This interpretability also enables secondary applications: conversation logging, sentiment analysis, automatic summarization, and multilingual translation—all operating on the transmitted text stream without additional processing. Token-based semantic codecs (e.g., Vevo's discrete audio tokens) lack this human-interpretable intermediate form, limiting their utility beyond speech reconstruction.

**Limitations.** Despite its advantages, the STCTS approach has inherent limitations. First, the system is designed strictly for speech; non-speech acoustic events (e.g., laughter, crying, background music, environmental sounds) are filtered out by the VAD or ignored by the STT model, resulting in their loss at the receiver end. Second, the reconstruction quality is bounded by the performance of the STT and TTS models. Transcription errors (e.g., proper nouns, homophones) result in semantic

deviations that cannot be corrected by the receiver, although our noise resilience experiments suggest that prosody often helps listeners infer the correct meaning. . Finally, the system relies on the availability of high-quality STT/TTS models for the target language; while English support is excellent, performance may degrade for low-resource languages without fine-tuned models.

**Directions for Future Improvement.** While our current system demonstrates strong performance, several architectural enhancements could further improve quality and efficiency. First, *advanced text compression* using large language models (LLMs) could achieve near-optimal entropy coding. Recent work [21], [22] shows that probabilistic language models can drive arithmetic coding to compress text to $\sim$30% of traditional compressor sizes. Integrating a lightweight on-device LM or leveraging cloud-based LLMs for predictive compression could reduce our text bitrate from $\sim$70 bps to $\sim$20–30 bps, bringing total bandwidth below 50 bps. Second, *adaptive prosody transmission* could dynamically adjust update rates based on speech content: increasing frequency during emotionally expressive segments or rapid pitch changes, while reducing to near-zero during monotone speech. This content-aware strategy could maintain high expressiveness while further minimizing bandwidth. Third, *joint optimization of STT and TTS models* through multi-task learning or knowledge distillation could improve end-to-end reconstruction quality. For instance, training the TTS model to predict not only speech but also STT-generated transcriptions could teach it to compensate for common transcription errors, reducing WER degradation. Finally, *neural codec hybridization* could combine our semantic approach with residual waveform coding: transmit text and prosody semantically (as we do), but add a low-bitrate neural codec stream ($\sim$100–200 bps) to encode fine-grained acoustic details (e.g., breathing, laughter, background ambience) that semantic compression discards. This hybrid approach could improve STOI and PESQ scores while maintaining ultra-low total bitrate.

The architectural advantages and future improvement directions position STCTS as a versatile framework for diverse communication scenarios beyond the traditional telephony use case, including privacy-sensitive applications (e.g., secure government communication) and resource-constrained deployments (e.g., satellite IoT networks).

## V. Conclusion

This paper presented STCTS (Speech-to-Text Compression and Text-to-Speech), an ultra-low bitrate voice communication system that achieves sustained transmission at $\sim$80 bps—a $70\times$ to $80\times$ reduction compared to traditional codecs like Opus while maintaining high perceptual quality (NISQA MOS $>$4.2). By decomposing speech into three orthogonal components—linguistic content, prosodic expression, and speaker identity—and applying tailored compression strategies to each, STCTS demonstrates that semantic-level compression can dramatically outperform acoustic-domain coding.

While STT-TTS architectures have been explored for IoT communication [10] and speech disentanglement methods have demonstrated factorization of speech components [11], STCTS addresses a distinct challenge: enabling *natural, expressive voice communication at ultra-low bitrates*. Unlike IoT-focused STT-TTS systems that transmit only semantic content with generic synthesis, we explicitly model and compress prosody and speaker identity, preserving naturalness and speaker fidelity essential for general-purpose communication. Unlike speech disentanglement methods that learn frame-level latent representations requiring hundreds of bits per second, we leverage off-the-shelf pretrained models with novel compression strategies: sparse prosody transmission with TTS interpolation (0.1–1 Hz, $<$14 bps), context-aware text compression ($\sim$70 bps), and amortized speaker embedding transmission. This design achieves $8\times$ lower bitrate than recent semantic codecs [9] while maintaining comparable quality, demonstrating the effectiveness of explicit, interpretable representations optimized for bandwidth-constrained scenarios.

Our key findings include: (1) prosody can be transmitted at extremely sparse rates (0.05–1 Hz, $<$14 bps) through TTS interpolation, with a surprising bimodal quality distribution where mid-frequency updates (1–6 Hz) underperform both sparse and dense regimes; (2) semantic compression exhibits inherent temporal desynchronization (low STOI $\sim$0.15) yet maintains high intelligibility (WER $\sim$0.23) and naturalness, highlighting the need for appropriate evaluation metrics for generative systems; and (3) the system achieves graceful degradation under channel noise, maintaining quality even at 10% bit error rate through prioritized transmission and interpolation.

Beyond bitrate efficiency, STCTS's modular architecture offers practical advantages: edge computing support without GPU requirements, privacy-preserving encryption of structured data, independent component upgradeability, and human-interpretable text transmission enabling secondary applications. These properties make STCTS particularly suitable for bandwidth-constrained scenarios requiring natural communication: maritime satellite communication where families expect to recognize loved ones' voices, multi-party conference calls requiring speaker differentiation, and voice social platforms balancing quality with infrastructure costs.

Future work includes integrating LLM-based text compression, adaptive prosody transmission based on speech content, joint STT-TTS optimization through multi-task learning, and hybrid approaches combining semantic compression with residual neural codecs. We believe our implementation will facilitate practical deployment in bandwidth-limited communication environments where naturalness and expressiveness are valued alongside intelligibility.

## References


[1] Valin J M, Vos K, Terriberry T. RFC 6716: Definition of the Opus audio codec[J]. 2012.

[2] Skoglund J, Valin J M. Improving Opus low bit rate quality with neural speech synthesis[J]. arXiv preprint arXiv:1905.04628, 2019.

[3] Valin J M, Skoglund J. LPCNet: Improving neural speech synthesis through linear prediction[C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 5891-5895.

[4] Zeghidour N, Luebs A, Omran A, et al. Soundstream: An end-to-end neural audio codec[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 30: 495-507.

[5] Défossez A, Copet J, Synnaeve G, et al. High fidelity neural audio compression[J]. arXiv preprint arXiv:2210.13438, 2022.

[6] "Lyra V2 - a better, faster, and more versatile speech codec," Google Open Source Blog, 2022. [Online]. Available: https://opensource.googleblog.com/2022/09/lyra-v2-a-better-faster-and-more-versatile-speech-codec.html

[7] "MLow: Meta Introduces Audio Codec for Low-End Devices," InfoQ, 2024. [Online]. Available: https://www.infoq.com/news/2024/08/meta-mlow-audio-codec/

[8] Liu H, Xu X, Yuan Y, et al. Semanticodec: An ultra low bitrate semantic audio codec for general sound[J]. IEEE Journal of Selected Topics in Signal Processing, 2024.

[9] Collette R, Greenwood R, Nicoll S. A Novel Semantic Compression Approach for Ultra-low Bandwidth Voice Communication[J]. arXiv preprint arXiv:2509.15462, 2025.

[10] Urazayev D, Nurgazina G, Toktargazin A, et al. Voice over Low Data Rate Networks Using Speech-to-Text and Semantic Compression[J].

[11] Lu H, Wu X, Wu Z, et al. SpeechTripleNet: End-to-End Disentangled Speech Representation Learning for Content, Timbre and Prosody[C]//Proceedings of the 31st ACM International Conference on Multimedia. 2023: 2829-2837.

[12] Qian K, Zhang Y, Chang S, et al. Autovc: Zero-shot voice style transfer with only autoencoder loss[C]//International Conference on Machine Learning. PMLR, 2019: 5210-5219.

[13] Wang D, Deng L, Yeung Y T, et al. Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion[J]. arXiv preprint arXiv:2106.10132, 2021.

[14] Hayashi T, Watanabe S. Discretalk: Text-to-speech as a machine translation problem[J]. arXiv preprint arXiv:2005.05525, 2020.

[15] Radford A, Kim J W, Xu T, et al. Robust speech recognition via large-scale weak supervision[C]//International conference on machine learning. PMLR, 2023: 28492-28518.

[16] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented transformer for speech recognition[J]. arXiv preprint arXiv:2005.08100, 2020.

[17] Ravanelli M, Parcollet T, Plantinga P, et al. SpeechBrain: A general-purpose speech toolkit[J]. arXiv preprint arXiv:2106.04624, 2021.

[18] Khmelev N, Anikin A, Zorkina A, et al. Joint Voice Activity Detection and Quality Estimation for Efficient Speech Preprocessing[C]//2025 27th International Conference on Digital Signal Processing and its Applications (DSPA). IEEE, 2025: 1-6.

[19] Desplanques B, Thienpondt J, Demuynck K. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification[J]. arXiv preprint arXiv:2005.07143, 2020.

[20] De Cheveigné A, Kawahara H. YIN, a fundamental frequency estimator for speech and music[J]. The Journal of the Acoustical Society of America, 2002, 111(4): 1917-1930.

[21] Delétang G, Ruoss A, Duquenne P A, et al. Language modeling is compression[J]. arXiv preprint arXiv:2309.10668, 2023.

[22] Li Z, Huang C, Wang X, et al. Lossless data compression by large models[J]. Nature Machine Intelligence, 2025: 1-6.

[23] Zhang B, McLoughlin I, Miao X, et al. LSPnet: an ultra-low bitrate hybrid neural codec[C]//Proc. Interspeech 2025. 2025: 614-618.

[24] Shen J, Pang R, Weiss R J, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions[C]//2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018: 4779-4783.

[25] Ren Y, Hu C, Tan X, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech[J]. arXiv preprint arXiv:2006.04558, 2020.

[26] Casanova E, Weber J, Shulby C D, et al. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone[C]//International conference on machine learning. PMLR, 2022: 2709-2720.

[27] Casanova E, Weber J, Shulby C D, et al. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone[C]//International conference on machine learning. PMLR, 2022: 2709-2720.

[28] Chen S, Liu S, Zhou L, et al. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers[J]. arXiv preprint arXiv:2406.05370, 2024.

[29] Kim J, Kong J, Son J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech[C]//International Conference on Machine Learning. PMLR, 2021: 5530-5540.

[30] "Beta Release of Zonos v0.1," Zyphra, 2024. [Online]. Available: https://www.zyphra.com/post/beta-release-of-zonos-v0-1

[31] Panayotov V, Chen G, Povey D, et al. Librispeech: an asr corpus based on public domain audio books[C]//2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015: 5206-5210.

[32] Recommendation I T U T. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs[J]. Rec. ITU-T P. 862, 2001.

[33] Taal C H, Hendriks R C, Heusdens R, et al. An algorithm for intelligibility prediction of time–frequency weighted noisy speech[J]. IEEE Transactions on audio, speech, and language processing, 2011, 19(7): 2125-2136.

[34] Mittag G, Naderi B, Chehadi A, et al. NISQA: A deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets[J]. arXiv preprint arXiv:2104.09494, 2021.

[35] Zhen K, Sung J, Lee M S, et al. Scalable and efficient neural speech coding: A hybrid design[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 30: 12-25.

[36] Tan X, Qin T, Soong F, et al. A survey on neural speech synthesis[J]. arXiv preprint arXiv:2106.15561, 2021.