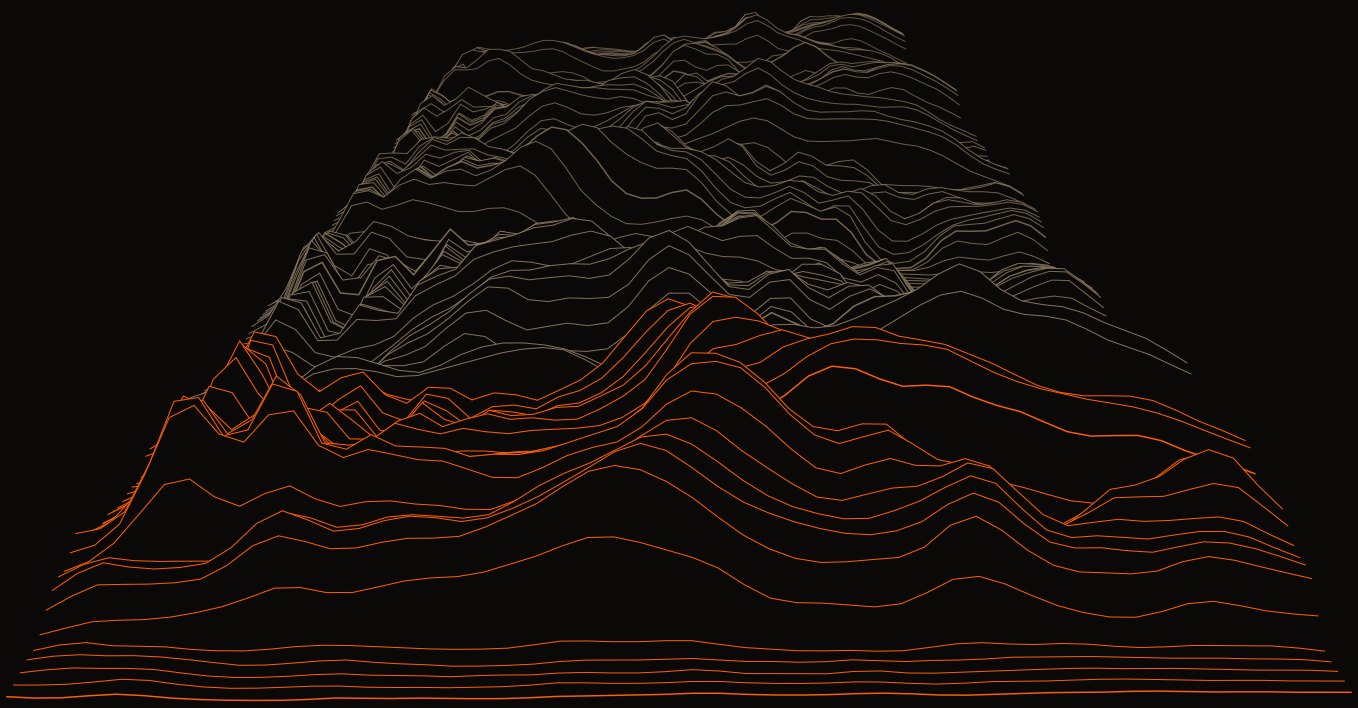


"COULD YOU TELL ME WHERE THE CHARGING STATION IS?" · ONE RIDGE PER FRAME

FIRST AUDIO · 17.6 MS

Real-time, In-Body Talk

A small streaming voice, trained in one tensor IR

CAM PEDERSEN · MUNICIPAL ROBOTICS

Abstract

In spoken dialogue, the latency that determines whether a reply feels conversational is the time from the end of the user's turn to the first sound of the response. For a text-to-speech system, the corresponding quantity is the time to first audio, not throughput. We present RIBT (Real-time, In-Body Talk), a 54M-parameter streaming English TTS system consisting of a Rust port of the misaki phonemizer, a FastSpeech2-style acoustic model, a conditional flow-matching decoder distilled to a single step, and a Vocos vocoder. All networks are expressed as programs in the tensor IR of tang, our compiler; all trained components were trained in that IR without PyTorch, and the same programs execute on Metal, CUDA and WebGPU.

On an M4 Max, RIBT produces its first audio a median of 17.6 ms after the text is available, at a real-time factor of 0.027. Kokoro-82M, which renders complete sentences, takes 349 ms on the same machine. The voice is distilled from Kokoro's af_heart voice; Kokoro's training data included output from closed commercial TTS systems, and no Kokoro weights are included in RIBT. Measured against af_heart, RIBT reaches a speaker similarity of 0.97–0.99, compared with Kokoro's own 0.98–0.99, a comparable word error rate on two of three test sets, and a lower predicted naturalness (UTMOS 4.39 against 4.52 on our main test set).

1 Motivation

Turn-taking in human conversation is fast: the typical gap between speakers is about 200 ms, and responses delayed beyond roughly 300–400 ms are perceived as hesitant (§0). We target a 99th-percentile latency below 250 ms from the end of the user's speech to the robot's first sound. That budget includes speech recognition, endpointing and response generation, so the share available to synthesis is small.

Our previous system used Kokoro-82M[1], a derivative of StyleTTS 2 with an iSTFTNet decoder[2]. Its output quality is high, but it synthesizes each utterance as a whole with bidirectional models, so no audio can be played until the entire utterance has been computed (§1). On an M4 Max, Kokoro's first audio arrived after 349 ms on the CPU and 815 ms on the GPU (§24.3); its real-time factor of 0.093 is adequate, so the limiting quantity is the time to first audio.

This gives three design requirements for RIBT: streaming synthesis with a low-cost first chunk, a model small enough to train and modify in-house, and a single implementation for a laptop, the robot and a web browser. The name also refers to a longer-term goal, a voice conditioned on the robot's physical state, which is outside the scope of this report.

> **ASIDE · SOURCES**
> Numbers marked §N refer to sections of the project's lab log. The evaluation files behind the tables, and the scripts that produced them, are published with this post in `data/`.

2 A single tensor IR

tang is our compiler and inference engine. Its tensor IR consists of affine views and a single operation, the comprehension

$$\text{out}[i] = \bigoplus_{k} \text{body}(x[\text{view}(i,k)], \ldots)$$

with a scalar body and a monoid. Matrix products, attention, convolutions, normalizations and the inverse STFT are instances of this form. Reverse-mode derivatives are again comprehensions, so backward passes are derived by the compiler rather than written by hand (§6). All networks trained for RIBT were trained in this representation. Python is used only for data preparation and for evaluation with external models (Whisper, UTMOS, WavLM).

Each training step compiles to one program containing the forward pass, the losses, the derived backward pass and the AdamW update. Optimizer state is double-buffered, so a step that produces non-finite values is discarded without a copy (§10). Before training, every graph is verified against tang's f64 interpreter, first by comparing derived gradients with finite differences and then by comparing one GPU step with the interpreter; worst-case relative errors were mostly between 10^{-7} and 10^{-5} (§13.3, §14.2, §15.2).

This approach has three practical consequences:

- **Portability.** The same graphs run on Metal and on CUDA (RTX 3090). The browser build executes tang's exported WGSL on WebGPU with no hand-written model code (Section 8).
- **Cost-driven operator choice.** Per-kernel costs determined several design rules: no recurrent layers, which are launch-bound at batch size one; polyphase convolutions with a reshape in place of transposed convolutions, which are 22 times faster and exact to 4×10^{-13} (§7); and fixed chunk shapes, so a single compiled program serves every chunk.
- **No separate inference stack.** The program that is trained is the program that is deployed.

The project also exposed more than two dozen limitations of the IR and its back ends, including Metal's 31-buffer argument limit, the absence of scan, gather and forward-mode differentiation, and CUDA-specific numerical errors. Verification against the f64 interpreter detected the numerical errors before they affected training. Several of these limitations have since been resolved upstream: guarded views, tanh, split reductions and shared parameter buffers.

3 Architecture

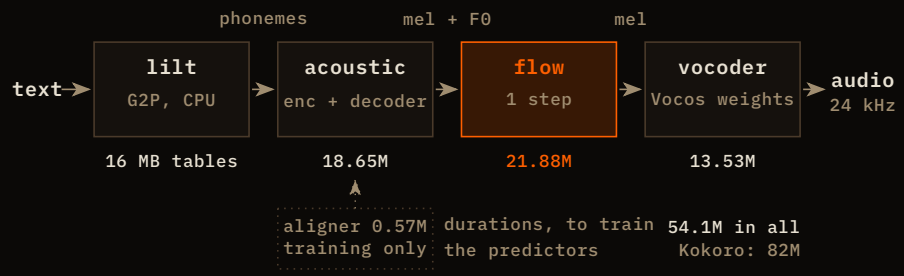


FIG 1 System overview. Text is converted to phonemes on the CPU; three networks on the GPU produce a 24 kHz waveform. The flow decoder (orange) is the only generative stage. The aligner is used only to produce duration targets for training.

3.1 Phonemization (lilt)

Kokoro uses misaki[3], a Python phonemizer built on spaCy. lilt is a stage-by-stage Rust port: spaCy's tokenization algorithm with its exported rules, an averaged-perceptron tagger distilled from spaCy's small English model, a line-by-line port of misaki's lexicon logic, and a number normalizer implemented independently (§12). It has no runtime dependencies and embeds about 16 MB of tables.

TABLE 1 Agreement of lilt with misaki 0.9.4 on in-vocabulary words. "Port alone" supplies spaCy's tokens and tags to lilt, so all remaining disagreement is due to the tagger. §12.3.

SET	SENTENCES	WORD PARITY	PORT ALONE
Held-out test	9,470	99.858%	100.000%
Edge cases	130	99.45%	100.000%

lilt processes 274,000 words per second on one M4 Max core. In streaming mode a word is emitted as soon as no later input can change its pronunciation, on average

1.28 words behind the input, at 67 µs per call. Streamed and batch output are identical on all 9,600 test and edge-case sentences (§12.1, §12.4).

misaki passes out-of-vocabulary words to espeak-ng, which is licensed under GPL-3.0; lilt instead resolves them in order through rules for times, addresses and handles, morphological decomposition and compounding from known stems, initialisms, a 2 MB letter-context decision tree fitted to its own lexicon, and letter-by-letter spelling (§27). On 9,599 test and edge-case sentences, one sentence still loses a word, a Chinese-script parenthetical. On 82 sentences built around names, neologisms and technical terms, Whisper recognizes 75 of the 82 target words. The decision tree reaches 58% exact accuracy (68% ignoring stress) on 8,806 held-out lexicon words.

3.2 Aligner

The duration predictor is trained on per-phoneme frame counts. We obtain them with an aligner following RAD-TTS[4] (0.57M parameters, forward-sum loss, monotonic alignment search; §13). Its word boundaries differ from those of a wav2vec2 forced aligner by a median of 18.4 ms, close to that aligner's 20 ms resolution.

3.3 Acoustic model

The acoustic model follows FastSpeech2[5] (18.65M parameters): a four-block transformer encoder, a reference encoder that derives the speaker from a speech clip, duration, pitch and energy predictors, a hard length regulator, and an eight-block ConvNeXt decoder that outputs 100 log-mel bands and frame-level F0 (§14). Trained with an L1 loss, it oversmooths: above 0.7 kHz its mel spectrograms retain only 21–42% of the harmonic and formant fine structure of the recordings (§15.4).

3.4 Flow decoder distilled to one step

A conditional flow-matching decoder[6] (21.88M parameters) conditioned on the acoustic model's mel and hidden state restores this detail (§15). Because a 16-step sampler is too slow for streaming, we distill it into a shortcut model[7]: the same network, conditioned on the step size and trained on input–output pairs from the teacher's ODE. The one-step student matches the 16-step teacher's mel error to three decimal places. A sampling temperature of $\tau = 0.3$ retains most of the restored structure without the roughness introduced at $\tau = 1$ (§15.5, §17.4). One step takes 9–12 ms for two seconds of speech (§15.7).

3.5 Vocoder

The vocoder has the Vocos architecture[8]: ConvNeXt blocks at frame rate followed by an inverse-STFT head (13.53M parameters). We first trained our own vocoder adversarially in the IR (§10). Swapping STFT components between our output and real audio showed that its magnitude spectra were as accurate as those of the public Vocos checkpoint while its phase was not: real phase combined with our magnitudes scored as well as Vocos, whereas our phase combined with real magnitudes scored below our own vocoder (§29.1). Because the architecture and mel front end are identical, RIBT uses the public Vocos weights (MIT), imported into our generator by renaming tensors. The imported model matches the PyTorch implementation to 1.5 ×

10^{-5} when both use the same GELU approximation, and it improved UTMOS by 0.17–0.30 on our test sets (§29.2).

4 Voice

An initial checkpoint trained only on LibriTTS-R[9] (CC BY 4.0) was intelligible but had low predicted naturalness (§18). Fine-tuning on individual readers, using LibriTTS-R's roughly 20 minutes per speaker and up to 48 hours per reader that we aligned from LibriVox, increased speaker similarity but not naturalness, because our vocoder did not reproduce restored MP3 recordings well (§20–§22).

The released voice is therefore distilled from a TTS teacher. We rendered 92,050 utterances (135 hours) with Kokoro's af_heart voice from 52,000 LibriTTS-R transcripts and 40,000 conversational and robot-specific lines, plus the 50 held-out texts, normalized to LibriTTS-R's loudness, with the 50 held-out texts and all evaluation texts excluded from training (§24.1). The phoneme sequences are those produced by Kokoro for each rendering. Starting from the LibriTTS-R checkpoint, we fine-tuned the acoustic model and the flow teacher, and re-distilled the one-step student (§24.2). The acoustic fine-tune increased UTMOS by 0.23–0.42 and speaker similarity to af_heart from 0.88–0.92 to 0.96–0.98 (§24.3).

> ASIDE · PROVENANCE
>
> The RIBT voice is **distilled from Kokoro-82M's af_heart voice** (kokoro 0.9.4, misaki 0.9.4). Kokoro's weights are released under Apache-2.0. Kokoro's training data included synthetic audio from closed commercial TTS providers that it does not name; the terms of those providers apply to whoever generated that data. The open questions are the likeness of the af_heart voice and Kokoro's data lineage. **No Kokoro weights are included in RIBT**: every RIBT network was trained by us, starting from a checkpoint trained only on LibriTTS-R, which remains available as a license-clean model. The vocoder is the public Vocos checkpoint (MIT), trained on LibriTTS.

5 Extensions beyond the teacher

Context-dependent prosody. Kokoro's prosody does not depend on preceding dialogue. RIBT includes a 0.25M-parameter module that encodes the previous turn and conditions the duration, pitch and energy predictors. It is trained on targets derived by rule from af_heart's own prosody, with magnitudes taken from studies of focus in English intonation[10]. Given the preceding questions "Who put it on the table?" and "Where did you put it?", the module places the accent of "I put it on the table." on the intended word in 23 of 24 pairs, raising it by about five semitones (§25.5). Without a preceding turn the output is identical to the base model. The module is included in the code; its weights are not part of this release.

Teacher selection. Rendering corpus lines several ways and keeping the best by intelligibility, UTMOS and speaker similarity reduced the teacher's word error rate from 3.41% to 2.91% (§26.2). Retraining the acoustic model on the selected renderings, plus 750 lines of robot-specific vocabulary, increased UTMOS by 0.02–0.05 on every test set (§26.4).

Regional accent. lilt implements an optional Upper Midwest accent: the vowel of “bag” is raised toward that of “bait” before /g/, and the vowel of “go” is lengthened. Interjections are adapted accordingly, e.g. “Ope!” (§26.7, §26.8). The release enables it by default.

6 Streaming

Streaming requires no retraining (§19). Words enter lilt one at a time. The encoder and predictors are re-run on the token prefix (2–5 ms), and all but the last two words are committed: their durations and pitch are frozen, so audio that has been played is never revised. Frames are rendered in chunks of 48 through the decoder, the one-step flow and the vocoder, using a fixed 128-frame window with left and right context and a short crossfade. All state is held on the host.

Measured on the LibriTTS-R checkpoint, streamed and non-streamed synthesis differ by less than 0.05 UTMOS, with no change in word error rate, and spectral flux at chunk boundaries is unchanged (§19.3). With the full text available, the released model produces first audio after a median of 17.6 ms (90th percentile 27.5 ms) at a real-time factor of 0.027, with no buffer underruns on ten test sentences (`data/latency`).

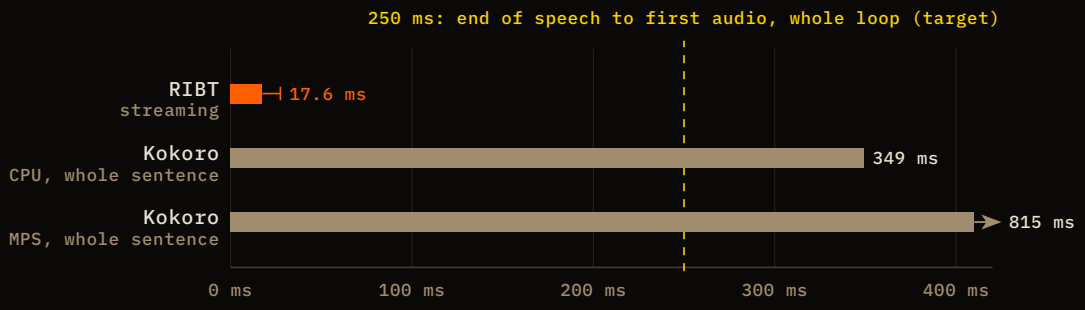


FIG 2 Time to first audio on an M4 Max, linear scale. RIBT streams (median 17.6 ms; whisker at the 90th percentile). Kokoro does not stream, so its first audio is the complete first sentence. The 250 ms line is the target for the entire conversational loop, not for synthesis alone.

When the text arrives incrementally from a language model at 15 words per second, first audio follows the first word by about 0.5 s, because the first chunk requires committed text worth roughly one second of speech (§19.4). In this setting latency is dominated by text arrival rather than computation.

7 Comparison with Kokoro

We evaluate on three sets: WER-50 (50 LibriTTS-R test sentences), listen-10 (ten conversational sentences) and held-50 (the 50 af_heart texts excluded from training). Naturalness is predicted with UTMOS22-strong[11], word error rate is computed from Whisper small.en transcripts[12], and speaker similarity is the cosine similarity between each utterance’s WavLM-base-plus speaker embedding[13] and the centroid of 121 real af_heart recordings.

TABLE 2 RIBT and its teacher: default settings, seed 0. Two of RIBT's three errors on listen-10 come from one sentence, where the spoken time "four fifteen" is transcribed as "4.15" (§29.2). Kokoro's similarity is not reported on held-50, whose reference clips are af_heart itself. `data/eval`.

MEASURE	WER-50	LISTEN-10	HELD-50
UTMOS, RIBT	4.394	4.336	4.304
UTMOS, Kokoro af_heart	4.521	4.497	4.464
Word error, RIBT	2.71%	2.44%	1.73%
Word error, Kokoro	2.61%	0.81%	1.73%
Similarity, RIBT	0.987	0.980	0.970
Similarity, Kokoro	0.990	0.982	–

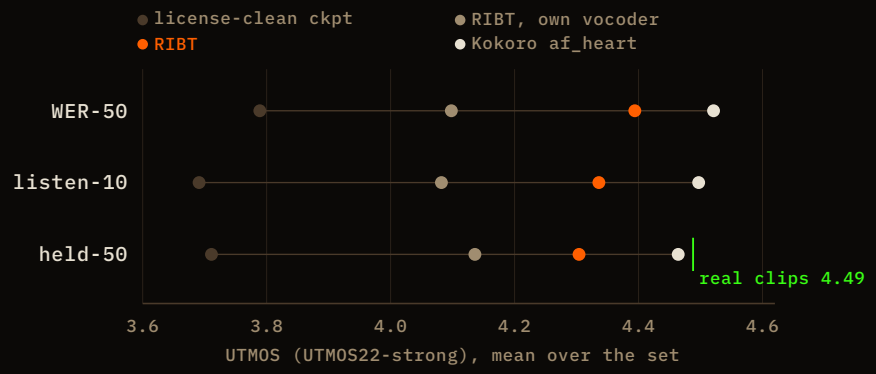


FIG 3 UTMOS on the three test sets for the LibriTTS-R checkpoint, the af_heart-distilled model with our own vocoder, RIBT with the Vocos weights, and Kokoro. The green mark is real af_heart recordings on a held-out set. `data/eval`.

Speaker similarity and word error rate are close to the teacher's; UTMOS is 0.13–0.16 lower on every set. For reference, 50 held-out real af_heart recordings score 4.488 (§29.1), within 0.06 of the teacher on every set; the teacher is therefore close to the metric's value for the voice itself, which bounds what distillation can recover on this metric.

8 Browser deployment

RIBT also runs entirely in a web browser. lilt is compiled unchanged to WebAssembly, and the acoustic model, flow decoder and vocoder run on WebGPU using tang's WGSL output: an exporter builds each graph at fixed bucket shapes, applies the same optimization passes as the native runtime, and serializes the kernels and their dispatch order. The JavaScript runtime only uploads buffers and dispatches kernels (`data/browser`).

TABLE 3 WebGPU deployment. Weights are stored in f16, except the pitch and energy heads (1.7M parameters), which are kept in f32; the 1–2% difference from the f32 model is due to this quantization. Five test sentences.

MEASURE (CHROME, M4 MAX)	VALUE
Audio vs. native, same weights	$\leq 6 \times 10^{-5}$ rel. RMS
Durations vs. native	identical
Audio vs. native, f32 weights	1–2% rel. RMS
Weights: acoustic / flow / vocoder	40.6 / 43.8 / 27.1 MB
Real-time factor (warm)	0.014
Page load to first audio (localhost)	about 1.05 s

The weights total about 112 MB (about 115 MB including lookup tables); over a network, download time rather than computation determines the latency of the first utterance.

9 Release

- **Code** (MIT): lilt, the four networks as tang programs, training and evaluation, the streaming engine and the browser runtime: muni-works/ribt.
- **Weights**: acoustic model, one-step flow student, imported Vocos vocoder and the af_heart reference clip, with SHA-256 checksums in the manifest (`data/release.toml`): municipal-robotics/ribt.
- **Default settings**: one flow step at $\tau = 0.3$, seed 0, a one-semitone pitch increase and the Upper Midwest accent; each is configurable.

10 Limitations and future work

- **Deployment conditions.** All measurements were taken on a laptop GPU. Measurements on the robot's onboard computer, within the full dialogue loop, are pending.
- **Latency with incremental text.** Computation takes 17.6 ms (median), but waiting for sufficient committed text takes about 0.5 s. A smaller first chunk combined with a playout policy driven by the text arrival rate has not yet been evaluated.
- **Voice provenance.** The released voice depends on a distillation teacher. The license-clean checkpoint and training pipeline can be used with new recordings; a vocoder that reproduces restored recordings faithfully would remove the need for a teacher.
- **Evaluation.** Naturalness is estimated with UTMOS; no listening test with multiple listeners has been run.
- **Body-state conditioning.** The prosody module already accepts dialogue context; conditioning on the robot's physical state is future work.

11 Conclusion

RIBT is a 54M-parameter streaming TTS system that produces first audio 17.6 ms after receiving text on an M4 Max, with no measurable loss of quality from streaming. All of its networks are tang programs that run unchanged on Metal, CUDA and WebGPU. Its voice, distilled from Kokoro's af_heart, matches the teacher in speaker similarity and word error rate and scores 0.13–0.16 lower on UTMOS. Code and weights are released with this report.

References

1. hexgrad. Kokoro-82M. https://huggingface.co/hexgrad/Kokoro-82M (2025).
2. Li, Y. A., Han, C., Raghavan, V. S., Mischler, G. & Mesgarani, N. StyleTTS 2: Towards Human-Level Text-to-Speech through Style Diffusion and Adversarial Training with Large Speech Language Models. https://arxiv.org/abs/2306.07691 (2023).
3. hexgrad. misaki: a G2P engine for TTS. https://github.com/hexgrad/misaki (2025).
4. Badlani, R. *et al.* One TTS Alignment To Rule Them All. https://arxiv.org/abs/2108.10447 (2021).
5. Ren, Y. *et al.* FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. https://arxiv.org/abs/2006.04558 (2020).
6. Mehta, S., Tu, R., Beskow, J., Székely, É. & Henter, G. E. Matcha-TTS: A fast TTS architecture with conditional flow matching. https://arxiv.org/abs/2309.03199 (2023).
7. Frans, K., Hafner, D., Levine, S. & Abbeel, P. One Step Diffusion via Shortcut Models. https://arxiv.org/abs/2410.12557 (2024).
8. Siuzdak, H. Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis. https://arxiv.org/abs/2306.00814 (2023).
9. Koizumi, Y. *et al.* LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus. https://arxiv.org/abs/2305.18802 (2023).
10. Xu, Y. & Xu, C. X. Phonetic realization of focus in English declarative intonation. *Journal of Phonetics* **33,** 159–197 (2005).
11. Saeki, T. *et al.* UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022. https://arxiv.org/abs/2204.02152 (2022).
12. Radford, A. *et al.* Robust Speech Recognition via Large-Scale Weak Supervision. https://arxiv.org/abs/2212.04356 (2022).
13. Chen, S. & others. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. https://arxiv.org/abs/2110.13900 (2021).