VoiceCraft: Zero-Shot Speech Editing and Text-to-Speech in the Wild

Puyuan Peng[1] **Po-Yao Huang**[2] **Shang-Wen Li**[2]
Abdelrahman Mohamed[3] **David Harwath**[1]
[1]The University of Texas at Austin [2]FAIR, Meta [3]Rembrand
pyp@utexas.edu

Abstract

We introduce VoiceCraft, a token infilling neural codec language model, that achieves state-of-the-art performance on both speech editing and zero-shot text-to-speech (TTS) on audiobooks, internet videos, and podcasts[1]. VoiceCraft employs a Transformer decoder architecture and introduces a token rearrangement procedure that combines causal masking and delayed stacking to enable generation within an existing sequence. On speech editing tasks, VoiceCraft produces edited speech that is nearly indistinguishable from unedited recordings in terms of naturalness, as evaluated by humans; for zero-shot TTS, our model outperforms prior SotA models including VALL-E and the popular commercial model XTTS v2. Crucially, the models are evaluated on challenging and realistic datasets, that consist of diverse accents, speaking styles, recording conditions, and background noise and music, and our model performs consistently well compared to other models and real recordings. In particular, for speech editing evaluation, we introduce a high quality, challenging, and realistic dataset named RealEdit. We encourage readers to listen to the demos at https://jasonppy.github.io/VoiceCraft_web.

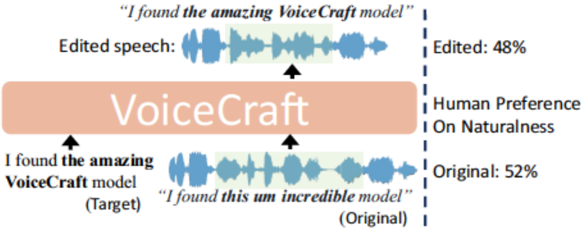

"I found the amazing VoiceCraft model"
Edited speech:
Edited: 48%
VoiceCraft
Human Preference
On Naturalness
I found the amazing VoiceCraft model
(Target)
Original: 52%
"I found this um incredible model"
(Original)

Figure 1: Speech editing with VoiceCraft. Human listeners prefer VoiceCraft edited speech over the original real recording 48% of the time in side-by-side naturalness comparison (details in §5.3)

1 Introduction

We introduce VoiceCraft, a Transformer-based neural codec language model (NCLM) that performs infilling generation of neural speech codec tokens autoregressively conditioned on bidirectional context. VoiceCraft achieves state-of-the-art (SotA) performance on both speech editing (shown in Fig. 1) and zero-shot TTS. Our method is based on a two-step token rearrangement procedure that consists of a *causal masking* step and *delayed stacking* step. The causal masking technique is inspired by the success of causal masked multimodal model in joint text-image modeling (Aghajanyan et al., 2022), and our proposed technique works for speech codec sequences, which enables autoregressive generation with bidirectional context. In addition, we further integrate causal masking with delayed stacking (Kharitonov et al., 2021a; Copet et al., 2023) as our proposed token rearrangement procedure, to ensure efficient multi-codebook modeling.

To evaluate speech editing, we manually crafted a first-of-its-kind, realistic, and challenging dataset named RealEdit. RealEdit consists of 310 real world speech editing examples, with waveforms sourced from audiobooks (Zen et al., 2019), YouTube videos (Chen et al., 2021a), and Spotify podcasts (Clifton et al., 2020), and duration ranging from 5 seconds to 12 seconds. To create the target transcripts, the transcripts of the source speech are edited in such a way that the edited transcripts remain grammatically correct and are semantically coherent. The dataset is designed to cover a wide range of editing scenarios, including insertion, deletion, substitution, and multi-span editing, with the length of the edited text ranging from 1 word to 16 words. Compared to commonly used speech synthesis evaluation datasets that only contain audiobooks such as VCTK (Yamagishi et al., 2019), LJSpeech (Ito and Johnson, 2017), and LibriTTS (Zen et al., 2019), RealEdit is

[1] Data, code, and model weights are available at https://github.com/jasonppy/VoiceCraft.

more challenging in that the recordings have diverse content, accents, speaking styles, recording conditions, and background sounds. We believe that the realism and diversity of REALEDIT makes it a reliable indicator of the practicality of speech editing models in the real world.

In the subjective human listening tests, VOICECRAFT significantly outperforms prior SotA speech editing model on REALEDIT. Importantly, the edited speech produced by VOICECRAFT is nearly indistinguishable from the original unedited recording in terms of naturalness. We found that VOICECRAFT generalizes well to zero-shot TTS without any finetuning, achieving SotA performance on a dataset comprised of audiobooks and YouTube videos, outperforming strong baselines including reproduced VALL-E (Wang et al., 2023a) and the popular commercial model XTTS v2 (COQUI, 2023). In summary, our contributions are:

1. We introduce VOICECRAFT, a neural codec language model for speech editing that generates synthesized speech that is nearly indistinguishable from in-the-wild recordings according to human listeners. We also release the code and model weights for VOICECRAFT.
2. We show that VOICECRAFT generalizes well to zero-shot TTS without finetuning.
3. We release a high quality, challenging, and realistic speech editing evaluation dataset REALEDIT.

2 Related Work

Neural codec langauge models (NCLM) and zero-shot TTS. Tokenizing speech signals into sequences of learnable, discrete units and then training a language model on the resulting unit sequences was initially proposed in the context of textless NLP (Hsu et al., 2021; Lakhotia et al., 2021; Kharitonov et al., 2021b; Nguyen et al., 2022), where the goal is to perform NLP tasks directly on spoken utterances without the need to first transcribe the speech into text. Recently, NCLMs that operates on tokens from Residual vector quantization (RVQ)-based models (Zeghidour et al., 2021; Defossez et al., 2022) attract increased attention due to its high quality generation. For example, AudioLM (Borsos et al., 2022a) exhibits strong performance on long-term coherent speech continuation. Zero-shot TTS is a task where a model needs to synthesize speech in a target voice which was unseen during training, given only the target transcript and a short reference recording of the target voice. Framing zero-shot TTS as transcript-conditioned speech continuation, VALL-E (Wang et al., 2023a) and Spear-TTS (Kharitonov et al., 2023) are the first applications of NCLMs on this task, significantly outperforming non-NCLM approaches. Zhang et al. (2023) extends VALL-E to cross-lingual TTS. Guo et al. (2022); Yang et al. (2023); Liu et al. (2023); Ji et al. (2023); Lyth and King (2024) adapt NCLMs style-controlled speech synthesis. Song et al. (2024); Du et al. (2024b) enhance phoneme alignment in NCLMs to reduce error. Wang et al. (2023b) proposes a unified NCLM for both speech generation and recognition tasks. Borsos et al. (2023) proposes an efficient parallel decoding method. Jiang et al. (2024) proposes disentangled timbre and prosody modeling, where the latter is modeled with a NCLM. NCLMs have also been successfully applied to other audio domains. Kreuk et al. (2022) applies NCLM to sound effects generation, and Agostinelli et al. (2023); Donahue et al. (2023); Garcia et al. (2023); Copet et al. (2023) use NCLMs for music generation.

Speech editing. This task requires a model to alter words or phrases within an utterance to match a target transcript, but the regions of the original speech not targeted for editing must remain unchanged (see Fig. 1 for an example). Early methods achieve text-guided speech insertion and substitution by combining a single speaker TTS model and a voice conversion model to generate desired speech segment, which is then concatenated with unedited part (Jin et al., 2017). Since the generation is not conditioned on the unedited part of the speech, the result sounds unnatural due to prosody mismatch and boundary artifacts (Morrison et al., 2021). More recent speech editing models have attempted to condition their generation on surrounding speech context. Tan et al. (2021) uses two unidirectional LSTM models with bidirectional fusion. Wang et al. (2022); Bai et al. (2022); Borsos et al. (2022b) uses the masked reconstruction objective with Convolutional or Transformer models to further improve contextualization. FluentSpeech (Jiang et al., 2023b) is a diffusion-based speech editing model that achieves SotA performance on speech editing on LibriTTS and VCTK.

The research community starts to investigate the possibility of having a unified model for both zero-shot TTS and speech editing. Yin et al. (2022); Jiang et al. (2023a) propose modular models for the two tasks, while our model is end-to-end. Concurrent work SpeechX (Wang et al., 2023c) adapt VALL-E by prompt tuning for a range of tasks including speech editing and zero-shot TTS, but no human evaluation is conducted in their paper. Concurrent work UniCATS (Du et al., 2024a) is a diffusion-based modular model for the two tasks. However their model is only evaluated on masked speech reconstruction of span length less than 2 seconds, while our model is evaluated on as much as 16 words editing. Voicebox (Le et al., 2023) is a recent flow matching based model capable of a wide range of tasks including speech editing and zero-shot TTS. However the speech editing capability is not evaluated in their paper, and only shown in their demo page. We therefore compare our model's editing results with Voicebox's on our demo page using on the same examples from their demo page.

3 Method

VOICECRAFT casts both sequence infilling (for speech editing) and continuation (for zero-shot TTS) as a simple left-to-right language modeling by rearranging neural codec's output tokens. The rearrangement involves two steps: (1) causal masking (§3.1) to enable autoregressive continuation/infilling with bidirectional context and (2) delayed stacking (§3.2) to ensure efficient multi-codebook modeling. VOICECRAFT employs decoder-only Transformers and is trained with an autoregressive sequence prediction (§3.3). We introduce the inference setup for speech editing and zero-shot TTS in §3.4.

3.1 Rearrangement Step 1: Causal Masking

As shown on the left hand side of Fig. 2, given a continuous speech waveform as input, we first use Encodec (Defossez et al., 2022) to quantize it into a T by K codec matrix X, where T is the number of temporal frames, and K is the number of RVQ codebooks. X can be written as $(X_1, \cdots, X_T)$, where X_t is a vector of length K representing the codes from different codebooks at time step t, and we assume that code from codebook k models the residual from codebook $k-1$. During training, our goal is to randomly mask some span of tokens $(X_{t_0}, \ldots, X_{t_1})$, and then autoregressively predict these masked tokens conditioned on all of the unmasked tokens. This is a problem when $t_1 < T$, because we cannot condition on future outputs when performing autoregressive generation. We need to modify the masking on X so that it is *causal*, by moving the span to be masked to the end of the sequence, so that when infilling these tokens the model can condition on both past and future unmasked tokens (Aghajanyan et al., 2022; Donahue et al., 2020; Bavarian et al., 2022).

The procedure outlined above can be trivially extended to multiple masked spans by simply moving *all* masked spans to the end of the sequence. The number of spans to be masked n is sampled from $\text{Poison}(\lambda)$, and then for each span, we sample a span length $l \sim \text{Uniform}(1, L)$. Finally, we randomly select the locations of the spans within X under the constraint that they do not overlap with each other. The selected n spans are then replaced with mask tokens $\langle \text{M}_1 \rangle, \cdots, \langle \text{M}_n \rangle$. The original tokens within these masked spans are moved to the end of the sequence X, with each span preceded by its corresponding mask token.

Consider this example: let $X = (X_1, \ldots, X_6)$ and imagine we wish to mask a single span from X_2 to X_4. The original sequence X is rearranged into $Y = (Y_1; \langle \text{M}_1 \rangle; Y_2; \langle \text{M}_1 \rangle; Y_3;)$, where $Y_1 = (X_1)$, $Y_2 = (X_5, X_6)$, and $Y_3 = (X_2, X_3, X_4)$. We call Y_1 and Y_2 the unmasked spans, and Y_3 the masked span. An *end of span* or EOS token is added to the end of each masked span (in this example at the end of Y_3), and an *end of utterance* or EOU token is added to the end of the utterance (i.e. Y_2). For simplicity, we do not explicitly denote these special tokens and assume they are part of the spans.

3.2 Rearrangement Step 2: Delayed Stacking

After the causal masking token rearrangement, each timestep of the rearranged matrix Y is vector of K tokens. Copet et al. (2023) observed that when performing autoregressive generation over stacked RVQ tokens, it is advantageous to apply a *delay pattern* so that the prediction of codebook k at time t can be conditioned on the prediction of codebook $k-1$ from the same timestep. We take a similar approach which we describe here. Assume a span Y_s is of shape $L_s \times K$. Applying the delay pattern rearranges it into $Z_s = (Z_{s,0}, Z_{s,1}, \cdots, Z_{s,L_s+K-1})$, where $Z_{s,t}, t \in [L_s + K - 1]$ is defined as[2]:

$$Z_{s,t} = (Y_{s,t,1}, Y_{s,t+1,2}, \cdots, Y_{s,t-K+1,K}) \quad (1)$$

where $Y_{s,t-k+1,k}$ denotes the token located at coordinate $(t-k+1, k)$ in matrix Y_s, i.e. the kth codebook entry at the $(t-k+1)$th timestep. To make

[2] $[N]$ represents integer set $\{0, 1, \cdots, N\}$

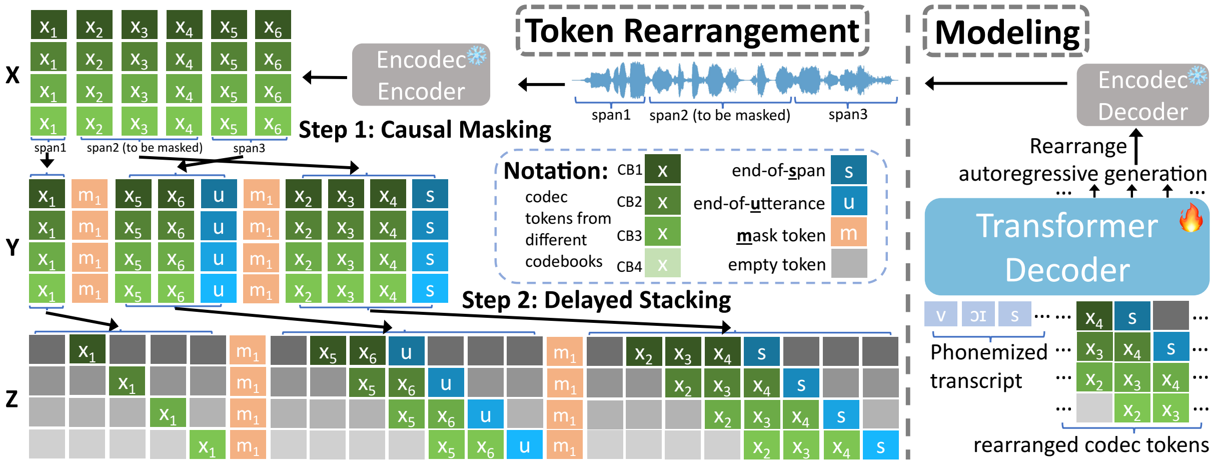


Figure 2: An example of the token rearrangement procedure and modeling framework. The rearrangement procedure involves two steps: (1) *Causal masking*, where masked spans are replaced with mask tokens and moved to the end, and (2) *Delayed stacking*, where tokens are shifted in the time dimension based on their codebook index.

sure that $\forall t \in [L_s + K - 1]$, $Z_{s,t}$ contains K valid tokens, we introduce a special learnable `[empty]` token and define $Y_{s,t-k+1,k} \triangleq$ `[empty]`, $\forall t \in \{s : s < k \cup s - k + 1 > L_s\}$. Note that the mask tokens are not part of any span and are not changed during delayed stacking. We define the resulting matrix of delayed stacking $Z = (Z_1, \langle \mathrm{M}_1 \rangle, Z_2, \langle \mathrm{M}_1 \rangle, \cdots, \langle \mathrm{M}_{\frac{S-1}{2}} \rangle, Z_S)$ (assuming Y consists of S spans). See the diagram for Z in Fig. 2 for an illustration.

3.3 Modeling

As shown in the right hand side of Fig. 2, we use a Transformer decoder to model Z autoregressively, conditioned on transcript of the speech W. Therefore, the input to the decoder is $[W; Z]$, where ";" denotes concatenation. At timestep t of span s in codec matrix Z, the model predicts all K tokens of $Z_{s,t}$ simultaneously, by using K MLP heads to project the transformer's final hidden state to K sets of logits, one for each of the K codebooks. Note that the prediction is conditioned on transcript W, and all tokens in Z before $Z_{s,t}$, denoted as $H_{s,t}$. Mathematically, the Transformer decoder models the factorized conditional distribution of Z:

$$\mathbb{P}_\theta(Z|W) = \prod_s \prod_t \mathbb{P}_\theta(Z_{s,t}|W, H_{s,t}) \tag{2}$$

$$= \prod_s \prod_t \prod_{k=1}^{K} \mathbb{P}_\theta(Z_{s,t,k}|W, H_{s,t}) \tag{3}$$

Where θ represent the parameters of the model. Equation 2 is the autoregressive factorization across time, while Equation 3 is the factorization across codebooks given an independence assumption - given W and $H_{s,t}$, the K RVQ codes in $Z_{s,t}$ are assumed to be independent of each other. We argue in appendix D that this assumption is mild.

With the token level probability formulation in Equation 3, we derive the training loss as the negative log likelihood $\mathcal{L}(\theta) = -\log \mathbb{P}_\theta(Z|W) = -\sum_{k=1}^{K} \mathcal{L}_k(\theta)$. Empirically, we found that weighting the first residual codebooks more than the latter codebooks leads to better performance, and therefore our final loss is $\mathcal{L}(\theta) = \sum_{k=1}^{K} \alpha_k \mathcal{L}_k(\theta)$, where $(\alpha_k)_{k=1}^{K}$ are tunable hyperparameters. Note that we follow Aghajanyan et al. (2022) and calculate the prediction loss on all tokens (not just the tokens in the masked spans), except for mask tokens and `[empty]` tokens.

3.4 Inference

Speech Editing. The setting for speech editing is the following: we have a speech recording R and its transcript W, and we want the model to modify only the relevant spans of R so that it matches the target transcript W'. We assume that W' is an edited version of W, where some words have been inserted, substituted, or deleted. This task is almost exactly the same as the training task, with two differences: 1) during training, the input transcript is simply the transcript of the original recording W, while during inference it is a modified transcript W' 2) during training, the spans to be masked (i.e. edited) are chosen randomly. During inference, we select them by comparing the original transcript and the target transcript to identify the words that should be masked out, and then use the word level forced alignment of the original transcript to identify the codec token spans that correspond to these

Table 1: Examples of the speech editing dataset REALEDIT. More examples are shown in table 8.

Edit Types	Original	Edited
deletion	I wrote the title **of the course many years ago, ah,** when I created this course.	I wrote the **title when** I created this course.
insertion	And **we're at** this point.	And we're **all extremely excited** at this point.
substitution, substitution	See why it's extremely **valuable to it's kind of like** it's kind of like having a **wall hack** to watch a demo.	See why it's extremely **important right?** it's kind of like having a **rough time** to watch a demo.

words to be masked. To ensure a smooth transition between the edited speech and the unedited speech, the neighboring words surrounding the span to be edited also need to be slightly modified in order to model co-articulation effects. Therefore, we specify a small margin hyperparameter ϵ, and extend the mask span length by ϵ on both the left and right sides[3]. During autoregressive generation, we feed the model the target transcript with all unmasked spans, with mask tokens inserted in the locations where the edits should take place. We then have the model autoregressively continue this sequence, whereby it fills in the masked spans. The generated codec tokens are then spliced back into their correct location in the utterance, and we map the complete codec token sequence back to a waveform using the Encodec decoder network.

Zero-shot TTS. As we previously noted, zero-shot TTS for our model is straightforward because it simply corresponds to performing an insertion edit at the end of the original utterance. In this case, the model is provided a voice prompt with its transcription, as well as the target transcript of the speech to be generated. The three inputs are concatenated together and fed to the model, after which it generates the codec sequence of the target transcript autoregressively.

4 REALEDIT: a realistic and challenging speech editing dataset

To support as realistic an evaluation as possible, we constructed a **first-of-its-kind** dataset of 310 manually-crafted speech editing examples. Each example consists of a tuple: (original audio, original transcript, edited transcript). The dataset contains 100 utterances from LibriTTS (dev-clean and dev-other) (Zen et al., 2019), 100 utterances from YouTube (from Gigaspeech testset) (Chen et al., 2021a) and 110 utterances from the Spotify Podcast dataset (Clifton et al., 2020). We manually checked the utterances for accuracy, then had native English speakers revise them to create edited transcripts. For each utterance, we determine the type of modification using predefined probability distributions of editing type, number of disjoint spans to be edited, and editing span length. Specifically, we study the following categories: 1) number of edited spans: 1 or 2; 2) type of edits: *insertion*, *deletion* and *substitution*; 3) editing span length: short (1-2 words), medium (3-6 words), long (7-12 words). Crucially, a edited transcript must be **grammatically correct and semantically coherent**. Examples of the dataset are shown in table 1 and 8, and statistics are shown in table 2,

Table 2: Dataset statistics for speech editing evaluation. Note that for 2-span editing, each example is edited using 2 of the 3 edit types.

type / length	Insert.	Delet.	Substi.	Total
1-2 words (1 span)	8	17	38	63
3-6 words (1 span)	22	24	79	125
7-12 words (1 span)	15	11	56	82
1 span total	45	52	173	270
2 spans total	13	13	54	40

5 Experiments

5.1 Setup

Data. Gigaspeech training set (Chen et al., 2021a) is used as the training data, which contains 9k hours of audiobooks, podcasts, and YouTube videos at 16kHz audio sampling rate. Audio files that shorter than 2 seconds are dropped. For ablation studies, we use the masked reconstruction task, and a 1000-utterance random subset of Gigaspeech validation set as the testing utterances (detailed in §C). For speech editing evaluation, we use the proposed REALEDIT dataset. For zero-shot TTS evaluation, we constructed a 250 prompt-transcript paired dataset from LibriTTS (Zen et al., 2019) and the YouTube portion of the Gigaspeech test set, with half of the examples drawn from each dataset. The length of each voice prompt is kept as close as pos-

[3] for substitution and deletion, the spans that are to be masked are just those words that are different from the target plus the margin; for insertion, the spans are just left and right margin spanning from the middle of the two words where the insertion happens

sible to 3 seconds long, with the constraint applied that we only cut the audio between complete words. The transcript is a concatenation of the transcript of the voice prompt and the target transcript. The target transcripts are chosen from different utterances spoken by the same speaker as the prompt, and range from 8 to 40 words in length. We only select utterances with a WER lower than 15% by Whisper medium.en (Radford et al., 2022).

Model. Encodec (Defossez et al., 2022) is used as the speech tokenizer, which has 4 RVQ codebooks each with vocabulary size of 2048, and a codec framerate of 50Hz on 16kHz recordings. (see §C for detailed config). To choose the number of spans to mask in training, we use a Poison(1) distribution truncated to a minimum of 1 and maximum of 3. Span lengths are sampled from Uniform(1, 600) i.e. the masked speech can be as long as 12 seconds. At each time step, the embeddings of codes from different codebooks are summed (Wang et al., 2023a), then added by sinusoidal positional encoding (Vaswani et al., 2017), before being fed to the transformer. Text transcripts are phonemized based on the IPA phoneset using the toolkit provided by Bernard and Titeux (2021). Our main VOICECRAFT model has 16 transformers layer with hidden/FFN dimensions of 2048/8192, and 12 attention heads. The output of the last layers are fed to four separate 2-layer MLP modules to get prediction logits. Our Main model has 830M parameters and codebook weight hyperparameters α is set to be $(5, 1, 0.5, 0.1)$. Ablations on model sizes and codebook weights are shown in §5.2.

Training and inference. The training of the Encodec model largely follows the setting in Copet et al. (2023), detailed in §C. To train VOICECRAFT, we used the ScaledAdam optimizer and Eden Scheduler proposed in (Yao et al., 2024) with a base learning rate of 0.05, batch size of 400k frames (i.e. 133.2 minutes), and total training step of 50k with gradient accumulation. The training of the 830M VOICECRAFT model took about 2 weeks on 4 NVIDIA A40 GPUs. More details can be found in §C. We compare the performance of ScaledAdam and AdamW in §A.1. For inference, we use Nucleus sampling (Holtzman et al., 2020) with $p = 0.8$ and a temperature of 1 for all experiments. Due to the stochasticity of autoregressive generation, via manual inspection we found that while most of the time the model produces natural sounding speech, it sometimes produces excessively long silence or drags out certain sounds. We found that happens when the codec token generation gets stuck in a repeating loop. To resolve it, we use a simple heuristic: for each input utterance we generate several different output utterances and throw away the longest outputs. Specifically for speech editing, we run inference 10 times with different margin parameters, stepping ϵ up from 0.05 to 0.14 in 0.01 increments. The 4 longest outputs are discarded, and then we randomly select one sample from the remaining 6 outputs. For zero-shot TTS, we reduce the probability of generating the same token in consecutive timesteps in proportion to how many times that token was consecutively generated in the immediately preceding timesteps. In addition, we generate 5 samples with different random seeds, and select the shortest for TTS evaluation. The sample selection process is completely automatic and unsupervised (i.e. no human intervention or ASR scoring).

Table 3: Effect of scaling model sizes and codebook re-weighting. Lower is better for all metrics.

Params	Weights	WER	MCD	F0	Energy
120M	(1,1,1,1)	10.18	8.75	78.49	3.22
120M	(5,1,0.5,0.1)	7.75	8.31	87.74	3.54
430M	(1,1,1,1)	7.87	8.22	70.05	3.17
430M	(5,1,0.5,0.1)	7.30	8.13	73.41	3.19
830M	(5,1,0.5,0.1)	6.68	8.05	67.81	3.12

Baselines. For speech editing, we compare VOICECRAFT with the diffusion-based model FluentSpeech (Jiang et al., 2023b) which is the current open-source SotA model for speech editing. Since the original FluentSpeech model is trained on LibriTTS, for a fair comparison, we took the official GitHub repo and trained the model on Gigaspeech. Please find more details in §C. For zero-shot TTS, we compare our VOICECRAFT with VALL-E (Wang et al., 2023a), XTTS v2 (COQUI, 2023), YourTTS (Casanova et al., 2021), and FluentSpeech. Since the original VALL-E is not open-sourced, we use the code from the popular open-source implementation by Li (2023), and also trained the model on Gigaspeech. XTTS v2 is a popular commercial zero-shot TTS model[4] trained on a mixture of publicly available data and web-crawled data, although the exact data sources are unknown. YourTTS is trained on VCTK, LibriTTS, and also French and Portugese corpora.

Metrics. For ablation studies, since ground truth waveform is avaliable, in addition to WER (using Whisper medium.en as the ASR model), we

[4] The GitHub repo hosting XTTS v2 has 26k stars by Jan 2024.

Table 4: Performance comparison on speech editing.

Model	WER	Intelligibility MOS				Naturalness MOS			
		LibriTTS	YouTube	Spotify	Total	LibriTTS	YouTube	Spotify	Total
FluentSpeech	**4.5**	$3.89_{\pm 0.09}$	$4.08_{\pm 0.08}$	$3.95_{\pm 0.08}$	$3.97_{\pm 0.05}$	$3.42_{\pm 0.10}$	$4.07_{\pm 0.10}$	$3.93_{\pm 0.10}$	$3.81_{\pm 0.06}$
VoiceCraft	6.1	$\mathbf{4.05}_{\pm 0.08}$	$\mathbf{4.14}_{\pm 0.07}$	$\mathbf{4.12}_{\pm 0.07}$	$\mathbf{4.11}_{\pm 0.05}$	$\mathbf{3.68}_{\pm 0.10}$	$\mathbf{4.25}_{\pm 0.09}$	$\mathbf{4.16}_{\pm 0.08}$	$\mathbf{4.03}_{\pm 0.05}$
Original	5.4	$4.22_{\pm 0.07}$	$4.30_{\pm 0.07}$	$4.16_{\pm 0.08}$	$4.22_{\pm 0.05}$	$3.84_{\pm 0.09}$	$4.35_{\pm 0.08}$	$4.29_{\pm 0.08}$	$4.17_{\pm 0.05}$

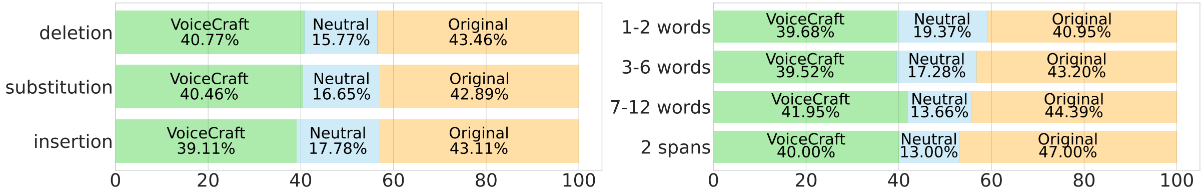


Figure 3: Breakdown of side-by-side human preference on naturalness comparing VoiceCraft edited speech and the original speech. Grouped by edit type (left) and edit span length (right).

Table 5: Side-by-side naturalness comparison of VoiceCraft (VCr) v.s. Original (Orig.) and FluentSpeech (FS).

Comparison	VCr better	Tie	VCr worse
VoiceCraft v. FS	56.1%	19.7%	24.1%
VoiceCraft v. Orig.	40.3%	16.2%	43.6%

use mel-ceptral distortion (MCD), F0 distance (F0) and energy distance (Energy). These are all objective metrics and their definitions are detailed in §C. For speech editing and zero-shot TTS evaluation, we use a combination of objective and subjective metrics. For the objective metrics, we used WER and speaker similarity (SIM) following prior works(Wang et al., 2023a; Kharitonov et al., 2023). SIM is calculated using the WavLM-TDCNN (Chen et al., 2021b). WER and SIM are calculated on all 310 utterances in RealEdit, and 250 utterances in the zero-shot TTS dataset. For our subjective evaluation, we used the Amazon Mechanical Turk platform to conduct human listening tests. For speech editing, the outputs of our model on all 310 utterances from RealEdit are evaluated by Turkers in terms of naturalness and intelligibility, and we use a 5-point Likert scale where 1 means poor and 5 means excellent. We also performed side-by-side A/B testing of VoiceCraft's output against the original (non-edited) speech, as well as the edited speech produced by FluentSpeech. In both cases, Turkers were asked to determine which utterance sounds more natural. The Turkers can choose either one of the two, or indicate that they are equally natural. Each evaluation received 5 ratings from 5 different Turkers. For zero-shot TTS, we randomly sampled 80 utterances (40 from LibriTTS and 40 from YouTube) from the original evaluation set, and asked Turkers to rate the naturalness, intelligibility, and speaker similarity of the generated speech to the reference prompt on a 5-point Likert scale. Each evaluation received 10 ratings. For all evaluations except the side-by-side comparison, Mean-Opinion-Score (MOS) with 95% confidence interval are reported. For the side-by-side comparison, we report the percentage of the time one model is preferred over the other. 64 and 59 Turkers participated in speech editing and TTS evaluation respectively. Please refer to §E for instructions and participants description.

5.2 Ablations

In table 3, we see that larger model sizes lead to better performance across all metrics . In addition, we see a bigger gap between the bigger models, indicating the potential of further scaling model (and possibly training data) sizes. For the impact of codebook re-weighting, and we see that weighting earlier codebook heavier leads to better performance on intelligibility related metrics WER and MCD, while worse performance on prosody related metrics F0 and Energy[5]. We choose weight $(5, 1, 0.5, 0.1)$ in our final 830M model because anecdotally, we found that VoiceCraft is stronger in prosody compared to intelligibility (similar properties about NCLMs are also found in (Jiang et al., 2023a; Song et al., 2024; Du et al., 2024b))

5.3 Speech Editing Results

Table 4 shows the results of speech editing evaluation in terms of WER, and human preference

[5] This can be regarded as a probing results that shows the properties of different codebooks in RVQ models. Since this is not the focus of our work, we do not conduct further experiment on this direction.

Table 6: On the zero-shot TTS task, comparing VOICECRAFT with other models.

Model	WER	SIM	Intelligibility MOS			Naturalness MOS			Speaker Similarity MOS		
			Libri.	YouTube	Total	Libri.	YouTube	Total	Libri.	YouTube	Total
YourTTS	6.6	0.41	3.28±0.11	3.01±0.12	3.14±0.08	2.99±0.12	2.59±0.12	2.79±0.08	3.10±0.12	2.49±0.12	2.79±0.09
FluentSpeech	**3.5**	0.47	3.70±0.11	3.65±0.12	3.67±0.08	3.34±0.11	3.43±0.12	3.38±0.08	4.10±0.09	3.92±0.11	4.01±0.07
VALL-E	7.1	0.50	4.05±0.09	3.94±0.10	4.00±0.07	3.85±0.10	3.86±0.10	3.86±0.07	4.12±0.10	4.02±0.10	4.07±0.07
XTTS v2	3.6	0.47	4.29±0.09	3.97±0.10	4.13±0.07	4.02±0.09	3.90±0.10	3.96±0.07	3.64±0.12	3.25±0.12	3.44±0.08
VOICECRAFT	4.5	**0.55**	**4.38**±0.08	**4.08**±0.10	**4.23**±0.06	**4.16**±0.08	**4.18**±0.09	**4.17**±0.06	**4.35**±0.08	**4.33**±0.09	**4.34**±0.06
Ground Truth	3.8	0.76	4.37±0.08	4.42±0.08	4.39±0.06	4.32±0.08	4.64±0.06	4.48±0.05	4.26±0.10	4.62±0.08	4.44±0.06

on intelligibility and naturalness. Our VOICECRAFToutperforms FluentSpeech on both intelligibility and naturalness MOS across different sources. Interestingly, FluentSpeech achieves a WER lower than the original recording (4.5 v.s. 5.4), although its intelligibility MOS (3.97) is worse than both VOICECRAFT (4.11) and original recording (4.22). This suggests that ASR model and human judgement diverge on FluentSpeech's intelligibility. Anecdotally, we observe that FluentSpeech tends to produce dull and sometimes robotic speech [6], and we hypothesize that this type of speech tends be more easily recognized by ASR, but is less intelligible to human ears. We notice this same phenomenon in our results on zero-shot TTS.

Human listeners rate LibriTTS's naturalness lower than YouTube and Spotify on original speech (results on TTS is consistent with this). This suggests that to better evaluate speech synthesis in general, the research community should consider evaluating on other speech domains besides audiobooks as is commonly done.

Table 5 presents side-by-side utterance naturalness comparison of VOICECRAFT vs. FluentSpeech and VOICECRAFT vs. the original, unedited speech. We observe that VOICECRAFT is preferred over FluentSpeech 56.1% of the time, with an additional 19.7% of the time the two are tied. This means that 75.9% of the time, human listeners' think VOICECRAFT produces equal or more natural speech than FluentSpeech. Impressively, human listeners judge the edited speech produced by VOICECRAFT to be equally or more natural than the original unedited speech 56.4% of the time. Fig. 4 shows the breakdown of the side-by-side comparisons by edit type and edit span length. We see that compared to the original speech, VOICECRAFT performs consistently well across different edit types, but human listeners think its outputs are slightly less natural with longer edit span(s).

5.4 Zero-Shot TTS Results

Table 6 shows both objective and subjective evaluation on zero-shot TTS. We observe that VOICECRAFT achieves the best results in both automatic speaker similarity metric SIM, and all human evaluation metrics. In particular, VOICECRAFT is only slightly worse than ground truth in terms of intelligibility MOS (4.23 v.s. 4.39), and speaker similarity MOS (4.34 v.s. 4.44). The gap on naturalness is larger between VOICECRAFT and ground truth (4.17 v.s. 4.48), especially on YouTube utterances, which highlights the challenges of zero-shot TTS on noisy, in-the-wild data. The commercial model XTTS v2 comes second in terms of intelligibility and naturalness, and second to last on speaker similarity MOS. VALL-E achieves the second best on both automatic metric SIM and subjective metric speaker similarity MOS. Similarly to the speech editing results, ground truth YouTube utterances receive higher MOS scores than ground truth LibriTTS utterances in Table 6, which again suggests that we should consider using more diverse data for future speech synthesis model evaluation. Lastly, we again observe that FluentSpeech achieves lower WER than the ground truth, but receives much lower ratings in terms of intelligibility MOS from human listeners, indicating that WER could be misleading in evaluating intelligibility of speech synthesis systems[7].

6 Conclusion

We introduce a neural codec language model VOICECRAFT that achieves state-of-the-art performance on speech editing and zero-shot TTS on in-the-wild data. The key lies in an innovative token rearrangement procedure which enables efficient and effective autoregressive codec generation with bidirectional context. In addition, we introduce a first-of-its-kind high quality, challenging, and re-

[6] please refer to our demo page for examples

[7] we also tried Whisper Large-v3, it gets WER of 4.1 for ground truth, and 2.7 for FluentSpeech.

alistic speech editing dataset REALEDIT, which we believe can reliably measure the practicality of speech editing models.

7 Limitations

Given the advancement of made by VOICECRAFT, there are still limitations. First and foremost is the long silence and scratching sound that occasionally occur during generation. Although in this work, we overcome it with sampling multiple utterances and selecting the shorter ones, more elegant and efficient methods are needed. Another important aspect is AI safety, how can we watermark and detect synthesized speech? While watermarking and deepfake detection has attracted increasing attention in the research community, and remarkable progress has been made (Zhang et al., 2020; Yamagishi et al., 2021; Chen et al., 2023; Roman et al., 2024), more advanced models such as VOICECRAFT presents new opportunities and challenges to safety research. To facilitate speech synthesis and AI safety research, we fully open source our codebase and model weights.

8 Ethical Implications

The speech synthesis model VOICECRAFT introduced in this work has both positive and negative implications.

On the positive side, VOICECRAFT holds the promise of significant benefits across several domains. For individuals with speech impairments or who have lost the use of their voice, VOICECRAFT could be transformative, enabling these individuals new ways to communicate with ease and clarity that were previously not possible. Content creators, whether they work in education, video production, or podcasting, could leverage VOICECRAFT to streamline their editing processes, making it easier to produce high-quality content without the need to re-record takes when they contain a small mistake. Furthermore, VOICECRAFT's ability to handle diverse accents without compromising on quality opens up new possibilities for creating synthetic data. This could, in turn, enhance speech recognition systems, such as Voicebox (Le et al., 2023), by providing them with a richer and more varied dataset to learn from, thereby improving their accuracy and accessibility to users worldwide.

However, the potential negative impacts of VOICECRAFT cannot be overlooked. One of the primary concerns is the model's potential to exacerbate existing biases, particularly those related to ethnicity. If not carefully monitored and corrected, these biases could lead to unequal performance across different groups, perpetuating and possibly even worsening existing disparities. Moreover, the ease with which voices can be cloned raises serious concerns about misuse, including impersonation and fraud. The ability to replicate someone's voice with only a few seconds of reference audio could be exploited to commit crimes or spread misinformation, posing significant ethical and security challenges. As such, while the benefits of VOICECRAFT are clear and substantial, it is imperative to approach its deployment with caution, ensuring that measures are in place to mitigate these risks and protect against potential misuse.

Despite the concerns regarding impersonation and fraud associated with VOICECRAFT, there are compelling reasons to advocate for its release. Foremost among these is the opportunity it presents for the broader research community and technology developers to better understand and mitigate these negative impacts. By making these methods open source, we can catalyze the development of more robust countermeasures against the misuse of voice cloning technologies. This collaborative approach allows for the rapid identification of vulnerabilities and the exploration of innovative strategies to address them. Moreover, the authors of this work fully committed to advancing the field responsibly. We are actively working on pioneering deepfake detection and watermarking algorithms specifically designed for synthetic speech. By doing so, we not only acknowledge the potential risks associated with our technology but also take concrete steps to ensure its ethical use. This dual approach of open collaboration and dedicated research into safeguarding mechanisms reflects our commitment to fostering a technological ecosystem where the benefits of voice cloning can be realized while minimizing its potential for harm.

9 Acknowledgements

We thank Ziyue Jiang for providing guidance in running inference with and scaling up the FluentSpeech model. We thank students at SALT Lab of UT Austin for helpful discussions. This work is supported in part by the National Science Foundation under Grant No. 2238605.

References

Armen Aghajanyan, Po-Yao (Bernie) Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. 2022. Cm3: A causal masked multimodal model of the internet. *ArXiv*, abs/2201.07520.

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. 2023. Musiclm: Generating music from text. *ArXiv*, abs/2301.11325.

He Bai, Renjie Zheng, Junkun Chen, Xintong Li, Mingbo Ma, and Liang Huang. 2022. A3t: Alignment-aware acoustic and text pretraining for speech synthesis and editing. In *International Conference on Machine Learning*.

Mohammad Bavarian, Heewoo Jun, Nikolas A. Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. Efficient training of language models to fill in the middle. *ArXiv*, abs/2207.14255.

Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to phones transcription for multiple languages in python. *Journal of Open Source Software*, 6(68):3958.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022a. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533.

Zalán Borsos, Matthew Sharifi, and Marco Tagliasacchi. 2022b. Speechpainter: Text-conditioned speech inpainting. In *Interspeech*.

Zalán Borsos, Matthew Sharifi, Damien Vincent, Eugene Kharitonov, Neil Zeghidour, and Marco Tagliasacchi. 2023. Soundstorm: Efficient parallel audio generation. *ArXiv*, abs/2305.09636.

Edresson Casanova, Julian Weber, Christopher Dane Shulby, Arnaldo Cândido Júnior, Eren Gölge, and Moacir Antonelli Ponti. 2021. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*.

Guang Chen, Yu Wu, Shujie Liu, Tao Liu, Xiaoyong Du, and Furu Wei. 2023. Wavmark: Watermarking for audio generation. *ArXiv*, abs/2308.12770.

Guoguo Chen, Shuzhou Chai, Guan-Bo Wang, Jiayu Du, Weiqiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. 2021a. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *Proc. Interspeech 2021*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Micheal Zeng, and Furu Wei. 2021b. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16:1505–1518.

Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020. 100,000 podcasts: A spoken English document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5903–5917, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Defossez. 2023. Simple and controllable music generation. *ArXiv*, abs/2306.05284.

COQUI. 2023. Xtts v2. `https://huggingface.co/coqui/XTTS-v2`.

Alexandre Defossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *ArXiv*, abs/2210.13438.

Chris Donahue, Antoine Caillon, Adam Roberts, Ethan Manilow, Philippe Esling, Andrea Agostinelli, Mauro Verzetti, Ian Simon, Olivier Pietquin, Neil Zeghidour, and Jesse Engel. 2023. Singsong: Generating musical accompaniments from singing. *ArXiv*, abs/2301.12662.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Chenpeng Du, Yiwei Guo, Feiyu Shen, Zhijun Liu, Zheng Liang, Xie Chen, Shuai Wang, Hui Zhang, and K. Yu. 2024a. Unicats: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding. In *AAAI*.

Chenpeng Du, Yiwei Guo, Hankun Wang, Yifan Yang, Zhikang Niu, Shuai Wang, Hui Zhang, Xie Chen, and Kai Yu. 2024b. Vall-t: Decoder-only generative transducer for robust and decoding-controllable text-to-speech.

Hugo Flores Garcia, Prem Seetharaman, Rithesh Kumar, and Bryan Pardo. 2023. Vampnet: Music generation via masked acoustic token modeling. *ArXiv*, abs/2307.04686.

Zhifang Guo, Yichong Leng, Yihan Wu, Sheng Zhao, and Xuejiao Tan. 2022. Promptts: Controllable text-to-speech with text descriptions. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Wei-Ning Hsu, David Harwath, Tyler Miller, Christopher Song, and James R. Glass. 2021. Text-free image-to-speech synthesis using learned segmental units. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Shengpeng Ji, Jia li Zuo, Minghui Fang, Ziyue Jiang, Feiyang Chen, Xinyu Duan, Baoxing Huai, and Zhou Zhao. 2023. Textrolspeech: A text style control speech corpus with codec language text-to-speech models. *ArXiv*, abs/2308.14430.

Ziyue Jiang, Jinglin Liu, Yi Ren, Jinzheng He, Zhenhui Ye, Shengpeng Ji, Qian Yang, Chen Zhang, Pengfei Wei, Chunfeng Wang, Xiang Yin, Zejun MA, and Zhou Zhao. 2024. Mega-TTS 2: Boosting prompting mechanisms for zero-shot speech synthesis. In *The Twelfth International Conference on Learning Representations*.

Ziyue Jiang, Yi Ren, Zhe Ye, Jinglin Liu, Chen Zhang, Qiang Yang, Shengpeng Ji, Rongjie Huang, Chunfeng Wang, Xiang Yin, Zejun Ma, and Zhou Zhao. 2023a. Mega-tts: Zero-shot text-to-speech at scale with intrinsic inductive bias. *ArXiv*, abs/2306.03509.

Ziyue Jiang, Qiang Yang, Jia li Zuo, Zhe Ye, Rongjie Huang, Yixiang Ren, and Zhou Zhao. 2023b. Fluentspeech: Stutter-oriented automatic speech editing with context-aware diffusion models. In *Annual Meeting of the Association for Computational Linguistics*.

Zeyu Jin, Gautham J. Mysore, Stephen DiVerdi, Jingwan Lu, and Adam Finkelstein. 2017. Voco: text-based insertion and replacement in audio narration. In *International Conference on Computer Graphics and Interactive Techniques*.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu Nguyen, Morgane Rivière, Abdel rahman Mohamed, Emmanuel Dupoux, and Wei-Ning Hsu. 2021a. Text-free prosody-aware generative spoken language modeling. *ArXiv*, abs/2109.03264.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu Nguyen, Morgane Rivière, Abdel rahman Mohamed, Emmanuel Dupoux, and Wei-Ning Hsu. 2021b. Text-free prosody-aware generative spoken language modeling. *ArXiv*, abs/2109.03264.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matthew Sharifi, Marco Tagliasacchi, and Neil Zeghidour. 2023. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. *ArXiv*, abs/2106.06103.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *ArXiv*, abs/2010.05646.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Defossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. 2022. Audiogen: Textually guided audio generation. *ArXiv*, abs/2209.15352.

Robert F. Kubichek. 1993. Mel-cepstral distance measure for objective speech quality assessment. *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, 1:125–128 vol.1.

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu Nguyen, Jade Copet, Alexei Baevski, Adel Ben Mohamed, and Emmanuel Dupoux. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Matt Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu. 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *ArXiv*, abs/2306.15687.

Feiteng Li. 2023. An unofficial pytorch implementation of vall-e. https://github.com/lifeiteng/vall-e.

Guanghou Liu, Yongmao Zhang, Yinjiao Lei, Yunlin Chen, Rui Wang, Zhifei Li, and Linfu Xie. 2023. Promptstyle: Controllable style transfer for text-to-speech with natural language descriptions. *ArXiv*, abs/2305.19522.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Daniel Lyth and Simon King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *ArXiv*, abs/2402.01912.

Matthias Mauch and Simon Dixon. 2014. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *SciPy*.

Max Morrison, Lucas Rencker, Zeyu Jin, Nicholas J. Bryan, Juan Pablo Cáceres, and Bryan Pardo. 2021. Context-aware prosody correction for text-based speech editing. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7038–7042.

Tu Nguyen, Eugene Kharitonov, Jade Copet, Yossi Adi, Wei-Ning Hsu, Ali Mamdouh Elkahky, Paden Tomasello, Robin Algayres, Benoît Sagot, Abdel rahman Mohamed, and Emmanuel Dupoux. 2022. Generative spoken dialogue language modeling. *Transactions of the Association for Computational Linguistics*, 11:250–266.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *ArXiv*, abs/2212.04356.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Robin San Roman, Pierre Fernandez, Alexandre Defossez, Teddy Furon, Tuan Tran, and Hady ElSahar. 2024. Proactive detection of voice cloning with localized watermarking. *ArXiv*, abs/2401.17264.

Yakun Song, Zhuo Chen, Xiaofei Wang, Ziyang Ma, and Xie Chen. 2024. Ella-v: Stable neural codec language modeling with alignment-guided sequence reordering.

Daxin Tan, Liqun Deng, Yu Ting Yeung, Xin Jiang, Xiao Chen, and Tan Lee. 2021. Editspeech: A text based speech editing system using partial inference and bidirectional fusion. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 626–633.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. *ArXiv*, abs/1711.00937.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Zi-Hua Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023a. Neural codec language models are zero-shot text to speech synthesizers. *ArXiv*, abs/2301.02111.

Tao Wang, Jiangyan Yi, Liqun Deng, Ruibo Fu, Jianhua Tao, and Zhengqi Wen. 2022. Context-aware mask prediction network for end-to-end text-based speech editing. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6082–6086.

Tianrui Wang, Long Zhou, Zi-Hua Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei. 2023b. Viola: Unified codec language models for speech recognition, synthesis, and translation. *ArXiv*, abs/2305.16107.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. 2023c. Speechx: Neural codec language model as a versatile speech transformer. *ArXiv*, abs/2308.06873.

Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92).

Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md. Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong-Aik Lee, Tomi H. Kinnunen, Nicholas W. D. Evans, and Héctor Delgado. 2021. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection. *ArXiv*, abs/2109.00537.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Guangzhi Lei, Chao Weng, Helen M. Meng, and Dong Yu. 2023. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *ArXiv*, abs/2301.13662.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. In *ICLR*.

Dacheng Yin, Chuanxin Tang, Yanqing Liu, Xiaoqiang Wang, Zhiyuan Zhao, Yucheng Zhao, Zhiwei Xiong, Sheng Zhao, and Chong Luo. 2022. Retrievertts: Modeling decomposed factors for text-based speech insertion. In *Interspeech*.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Heiga Zen, Viet-Trung Dang, Robert A. J. Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Z. Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. In *Interspeech*.

You Zhang, Fei Jiang, and Zhiyao Duan. 2020. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941.

Zi-Hua Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *ArXiv*, abs/2303.03926.

A Additional Experiments

A.1 Comparing ScaledAdam and AdamW

The hyperparameters settings of ScaledAdam can be found in table 9. For AdamW (Loshchilov and Hutter, 2017), we tried 3 settings:

- setting1: peak learning rate: 1e-5, batch size: 3.3 min, update steps: 500k
- setting2: peak learning rate: 1e-4, batch size: 33.3 min (same as ScaledAdam), update steps: 80k
- setting3: peak learning rate: 1e-4, batch size: 3.3 min, update steps: 500k

For all settings, we use a linear scheduler which linear ramp up the learning rate to peak in first 8% steps, and linearly decay it afterwards. We use the common default values for other hyperparameters, setting $\beta_1 = 0.9, \beta_2 = 0.999$, weight-decay $= 0.01$. All experiments are done on 4 A40 GPUs. Results are shown in table 7.[8] We see that ScaledAdam achieves better performance in all metrics while using less compute. However we note that due to limitation in computational resources, we could not exhaust hyperparameter search for AdamW, therefore we do not over-generalize our finding here.

A.2 Breakdown of side-by-side human preference comparison.

The comparison breakdown between VOICECRAFT and FluentSpeech is shown in figure 4. We see that VOICECRAFT outperforms FluentSpeech across the board, especially for substitution edits and when the edit span length is long.

A.3 Spectrograms Comparison

Spectrogram level comparison between FluentSpeech and VOICECRAFTare shown in figure 5, 6, 7 with the edited part marked in dark green rectangle. The three examples have increasing difficulty in terms of accents and recording conditions, in particular, the examples in figure 7 appears to be in low bandwidth transmission. In all 3 examples, we see that VOICECRAFT is able to generated more detailed frequency patterns. The corresponding audio can be found in the demo page.

[8] We early stopped AdamW setting 2 at step 57k to save the compute, as it has already taken more time than the finished ScaledAdam job while the performance was worse.

Table 7: ScaledAdam consistently outperforms AdamW across all metrics, while taking 10% less time to train.

Optimizer	Setting	Training Time	WER	MCD	F0	Energy
AdamW	lr=1e-5, bsz=13.3min, steps=500k	262 hours	16.45	8.91	196.15	5.94
AdamW	lr=1e-4, bsz=133.2min, steps=57k	273 hours	10.77	8.45	117.38	4.91
AdamW	lr=1e-4, bsz=13.3min, steps=500k	262 hours	7.58	8.32	82.73	3.70
ScaledAdam	lr=3e-2, bsz=133.2min, steps=50k	**237 hours**	**7.30**	**8.13**	**73.41**	**3.19**

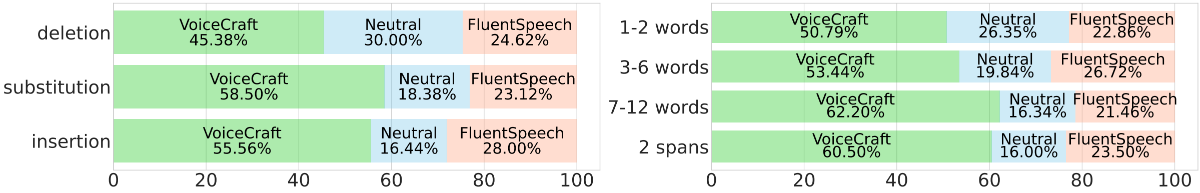


Figure 4: Breakdown of side-by-side human preference on naturalness comparing of VOICECRAFT and FluentSpeech on speech editing. Grouped by edit type (left) and edit span length (right).

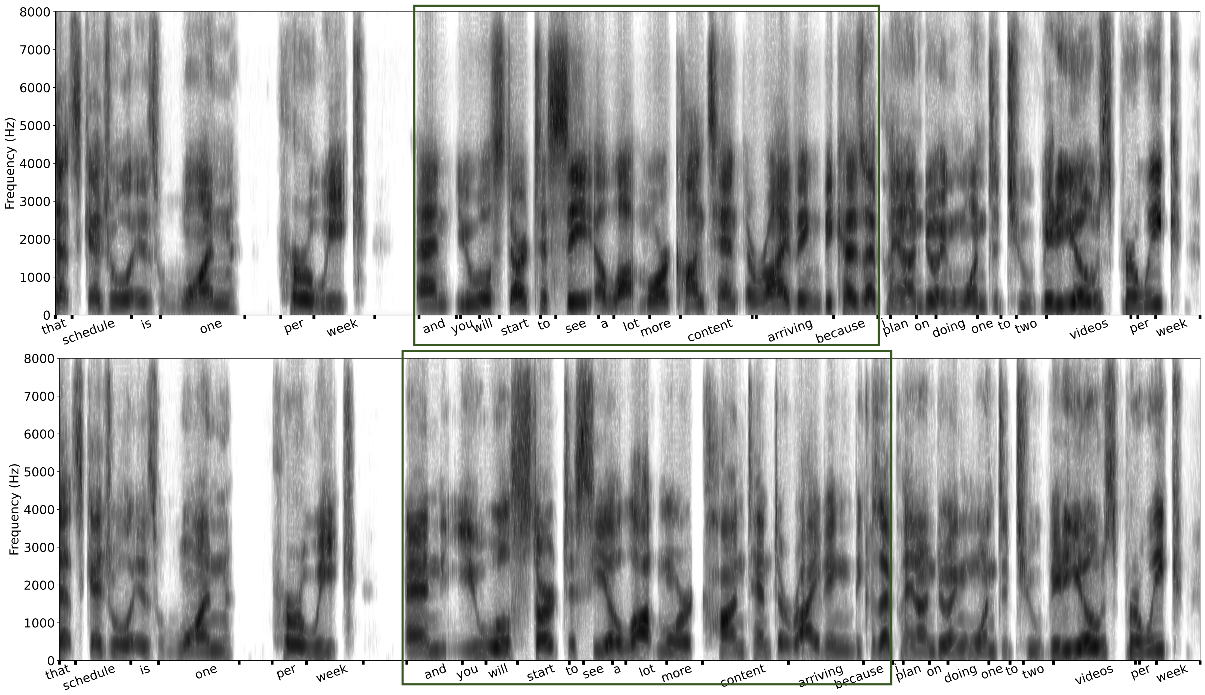


Figure 5: Upper: FluentSpeech; lower: VOICECRAFT

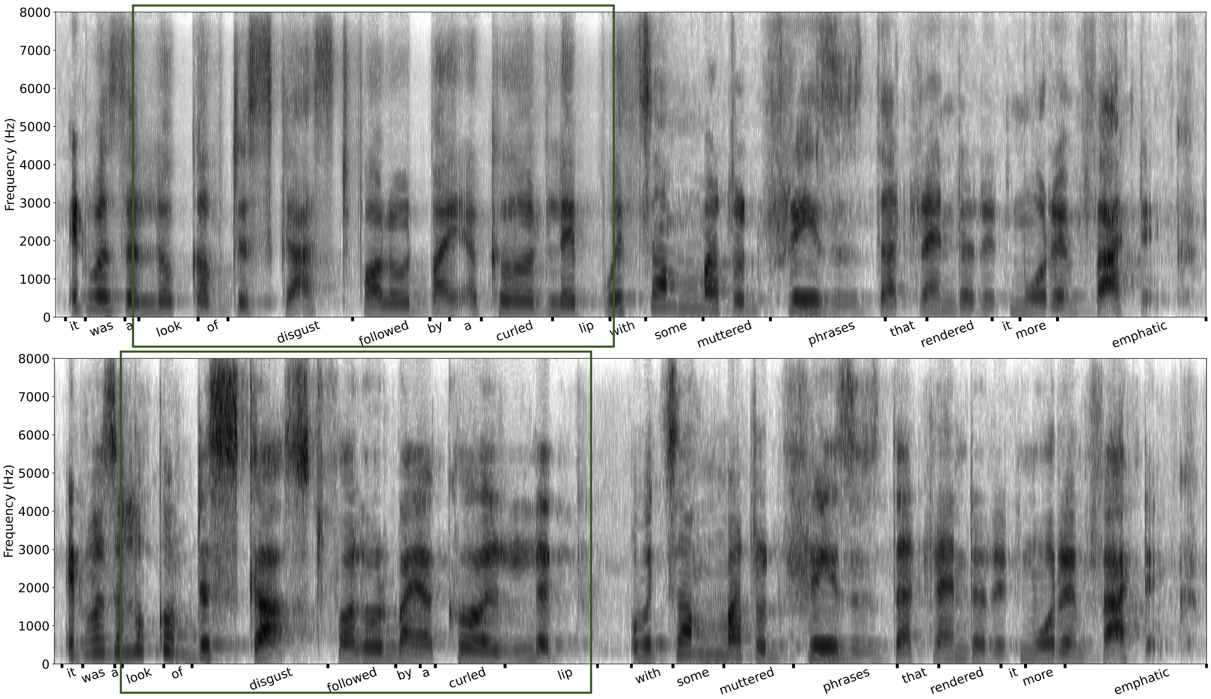


Figure 6: upper: FluentSpeech; lower: VOICECRAFT

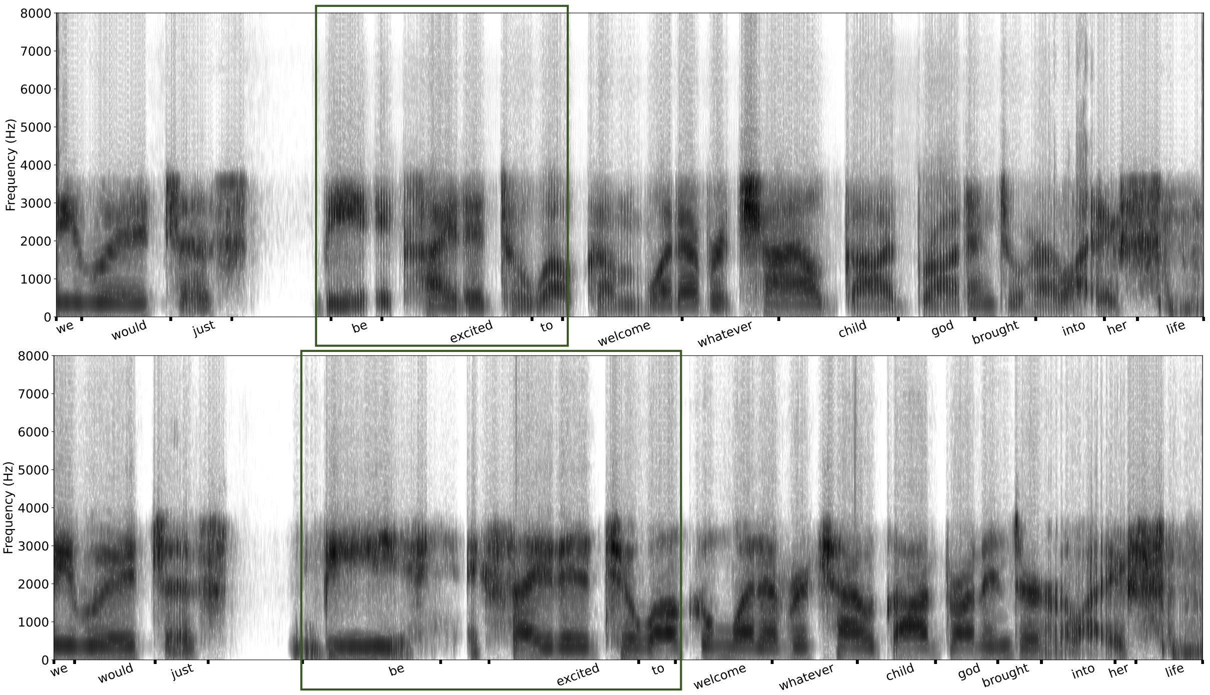


Figure 7: upper: FluentSpeech; lower: VOICECRAFT. Note that since the speech is recorded in very challenging condition, the word alignment is not very accurate. We see that for FluentSpeech's result, since the entire mel-spectrogram are passed to HiFi-GAN for resynthesis, even the unedited speech contains high frequency noise.

B Examples of the Speech Editing Dataset REALEDIT

Examples of REALEDIT are shown in table 8.

C Implementational Details

The Encodec model. The Encodec model we use has a stride of 320 samples, which means the codec framerate is 50Hz for recording of sample rate 16kHz. The base dimension is 64, doubling at each of the 5 convolutional layer in the encoder. Following (Copet et al., 2023), we use the open-sourced audiocraft repo[9] for Encodec model training. 1 second speech segments sampled from Gigaspeech over a total of 160 epochs (320k steps) with a batch size of 240. The model is trained with the Adam (Kingma and Ba, 2014) with base learning rate of 3e-4.

Eden Scheduler (Yao et al., 2024). the scheduler adjust the learning rate α_t at step t using the following formula:

$$\begin{aligned}\alpha_t =&\alpha_{\text{base}} \cdot \left(\frac{t^2 + \alpha^2_{\text{step}}}{\alpha^2_{\text{step}}}\right)^{-0.25} \cdot \\ &\cdot \left(\frac{e^2 + \alpha^2_{\text{epoch}}}{\alpha^2_{\text{epoch}}}\right)^{-0.25} \cdot \\ &\cdot \text{linear}(\alpha_{\text{start}}, t_{\text{warmup}}, t).\end{aligned}$$

Where α_{base} base learning rate, t is the step index, e is the epoch index, and α_{step} and α_{epoch} controls the amount of data the model has seen before significantly reducing the learning rate. $\text{linear}(\alpha_{\text{start}}, t_{\text{warmup}}, t)$ linearly increase the outcome from α_{start} to 1 over t_{warmup} steps, and stays at 1. In our experiment, we set

$$\begin{aligned}\alpha_{\text{base}} = 0.05, \alpha_{\text{step}} = 3000, \alpha_{\text{epoch}} = 4, \\ \alpha_{\text{start}} = 0.5, t_{\text{warmup}} = 500\end{aligned}$$

Since our dataset is quite large, we use pseudo-epoch instead of the actual epoch, and 1 pseudo-epoch is set to be 3000 training steps. Note that the choice of these hyperparameters are inspired by Yao et al. (2024); Li (2023), and if computation resources permitted, a grid search might find better hyperparameters settings.

Configuration in ablation studies. Configuration of different models are shown in table 9. Note that we use base learning rate 3e-2 for 430M model instead of 5e-2 because the latter gave a NaN error.

Task and Data for ablation studies. The evaluation task is masked reconstruction, where for each utterance, we randomly select a span of length 1 to 15 words to mask, and ask VOICECRAFT to reconstruct the masked speech based on the transcript and unmasked speech. We use a 1000-utterance random subset of the Gigaspeech validation set, which contains YouTube videos and podcast data. We ensure that each utterance in the subset has a WER lower than 15% when decoded by Whisper medium.en (Radford et al., 2022).

Metrics for ablation studies. Since ground truth is available for masked reconstruction evaluation, in addition to WER (measured from Whisper medium.en's output), we also measure the mel-cepstral distortion (MCD) (Kubichek, 1993), F0 distance (F0), and energy distance (Energy) WER and MCD are better correlated with intelligibility of the speech, and F0 and Energy are better correlated with prosody similarity between the generated and ground truth. MCD measures the difference of Mel Frequency Cepstrum Coefficients (MFCC) between generated and ground truth, defined as

$$\text{MCD} = \frac{10}{\ln 10}\sqrt{\frac{1}{2}\sum_{i=1}^{L}(m_i^g - m_i^r)^2}$$

where L is the order of MFCC, which we set to be 13. m_i^g is the ith MFCC of ground truth recording and m_i^r is the ith MFCC of the generated. We use pymcd package [10] for calculating MCD. For F0 estimation, we use the pYIN (Mauch and Dixon, 2014) algorithm implemented in librosa (McFee et al., 2015) with minimal frequency 80hz and maximal frequency 600hz. For energy calculation, we use the root mean square of magnitude of spectrogram, which is extracted using short time Fourier transform with window length of 640, hop size of 160. Note that since generated speech might have a different length compared to ground truth, dynamic time wrapping is first applied to time aligned the extracted MFCC/F0/energy before calculating their euclidean distances. For each model in the ablation study, we use 3 different random seeds and report the averaged results.

Scaling FluentSpeech. The original FluentSpeech (Jiang et al., 2023b) is trained on LibriTTS, and we made our best effort in scaling it for a fair comparison. Taking guidance from the authors of FluentSpeech. We scale the batch size

[9] Encodec training doc can be here

[10] https://github.com/chenqi008/pymcd

Table 8: Examples of the speech editing dataset REALEDIT.

Edit Types	Original	Edited
substitution, substitution	See why it's extremely **valuable to it's kind of like** it's kind of like having a **wall hack** to watch a demo.	See why it's extremely **important right?** it's kind of like having a **rough time** to watch a demo.
deletion	I wrote the title **of the course many years ago, ah,** when I created this course.	I wrote the **title when** I created this course.
insertion	Fast cars, that had the nice **clothes, that** had the money, they was criminals.	Fast cars, that had the nice clothes, **that had expensive gold watches,** that had the money, they was criminals.
substitution	When the CEO of blockbuster heard that, he promptly had **a kitchen sink** delivered to the netflix office, a fairly creative way of declaring war.	When the CEO of blockbuster heard that, he promptly had **five hundred pounds of glitter divided into five thousand manilla envelopes** delivered to the netflix office, a fairly creative way of declaring war.
substitution	So if you've been following my story, you will remember that I said earlier **in this podcast that the** Grammy nominations came out.	So if you've been following my story, you will remember that I said earlier **that this week we had super exciting stuff to talk about because** Grammy nominations came out.
insertion	No to the chemical pollution, air **pollution, and** the destruction of the environment caused by factories and the manufacturing industry.	No to the chemical pollution, air pollution, **no to the killing of plants and wildlife** and the destruction of the environment caused by factories and the manufacturing industry.
substitution, substitution	because we can include so many other characters if we **just** expand the definitions to any **sword** wielder, who's a little spicy.	because we can include so many other participants if we **are brave enough to** expand the definitions to any **blade** wielder, who's a little spicy.
insertion	So for more craziness now that French was **conquered we** have to join forces to Great Britain.	So for more craziness now that French was conquered **by the Germans,** we have to join forces to Great Britain.
substitution	economic development remains one of the most **effective ways to increase the capacity** to adapt to climate change.	economic development remains one of the most **promising options that we have left on the table** to increase the capacity to adapt to climate change.
insertion	And **we're at** this point.	And we're **all extremely excited** at this point.
insertion	Steve also co-founded pixar animation studios. Which has revolutionized the film industry in it's short history **with brilliant** use of technology.	Steve also co-founded pixar animation studios. Which has revolutionized the film industry in it's short history with **films like toy story that showcase** brilliant use of technology.
substitution, deletion	this is just so cozy **up** here, **and having that skylight is just lovely** isn't it.	this is just so cozy **and warm here, isn't** it.
substitution	It was a **glance of inquiry, ending in a look of chagrin,** with some muttered phrases that rendered it more emphatic.	It was a **look of disgust followed by a curled lip,** with some muttered phrases that rendered it more emphatic.
substitution	More of a base and infrastructure to **tell those stories rather than doing it** out of a out of a tent with solar power.	More of a base and infrastructure to **fight these battles instead of** out of a tent with solar power.

Params	codebook dim	Trm hidden dim	FFN dim	Trm layers	Base LR	Update Steps
830M	2048	2048	8192	16	5e-2	50k
430M	2048	2048	8192	8	3e-2	50k
120M	1024	1024	4196	8	5e-2	50k

Table 9: Hyperparameters settings for the different model sizes. Trm stands for Transformer.

from 16 utterances to 256 utterances. Diffusion base hidden dimension from 320 to 1024, residual layers from 20 layers to 30 layers, residual channels from 256 to 512. The final model contains 330M parameters, which is roughly the same as the Voicebox model (Le et al., 2023). The model was trained on Gigaspeech training set on 1 A40 GPU for 626k steps which took 10 days. The HiFi-GAN vocoder is also retrained on Gigaspeech training set for 400k steps using hyperparameters used on Voicebox (Le et al., 2023) (they also use Hifi-GAN as vocoder to decode to 16kHz speech)

Baselines for zero-shot TTS. For zero-shot TTS, we compare our VOICECRAFT with VALL-E (Wang et al., 2023a), XTTS v2 (COQUI, 2023), YourTTS (Casanova et al., 2021), and FluentSpeech. Since the original VALL-E is not open-sourced, we use the code from the popular open-source implementation by Li (2023), and also trained the model on Gigaspeech. Both the AR and NAR model are trained for 50k steps using the ScaledAdam optimizer and Eden scheduler, same as our VOICECRAFT. The commercial model XTTS v2 is composed of three modules, VQ-VAE (van den Oord et al., 2017) for speech tokenization, a GPT-2 (Radford et al., 2019) model for speech token modeling and a customized HiFi-GAN (Kong et al., 2020) model for token to waveform generation. XTTS v2 is trained on a mixture of publicly available data and web-crawled data, but the exact data sources are unknown. YourTTS is a zero-shot TTS model built upon the adversarial VAE model VITS (Kim et al., 2021), with novel zero-shot multi-speaker and multilingual training. The model is trained on VCTK, LibriTTS, and also French and Portugese corpora. The FluentSpeech model we used for TTS is the same as in speech editing, as the model can be configured to do zero-shot TTS similar to Voicebox (Le et al., 2023).

Licenses of the speech corpora. Licenses: LibriTTS: CC BY 4.0; Gigaspeech: Apache-2.0; Spotify Podcast dataset: CC BY 4.0.

D The Conditional Independence Assumption

To better explain the rational behind the conditional independent assumption in equation 3, we go back to sequence Y produced by causal masking. The assumption we are making for equation 3 to hold is equivalent to the assumption that given W and $H_{s,t}$, $Y_{s,t,k}$ is independent of $I^{(1)}_{s,t,k}$ and $I^{(2)}_{s,t,k}$ defined as

$$I^{(1)}_{s,t,k} \triangleq (Y_{s,t+k-1,1}, Y_{s,t+k-2,2}, \cdots, Y_{s,t+1,k-1})$$

$$I^{(2)}_{s,t,k} \triangleq (Y_{s,t-1,k+1}, Y_{s,t-2,k+2}, \cdots, Y_{s,t-K+k,K})$$

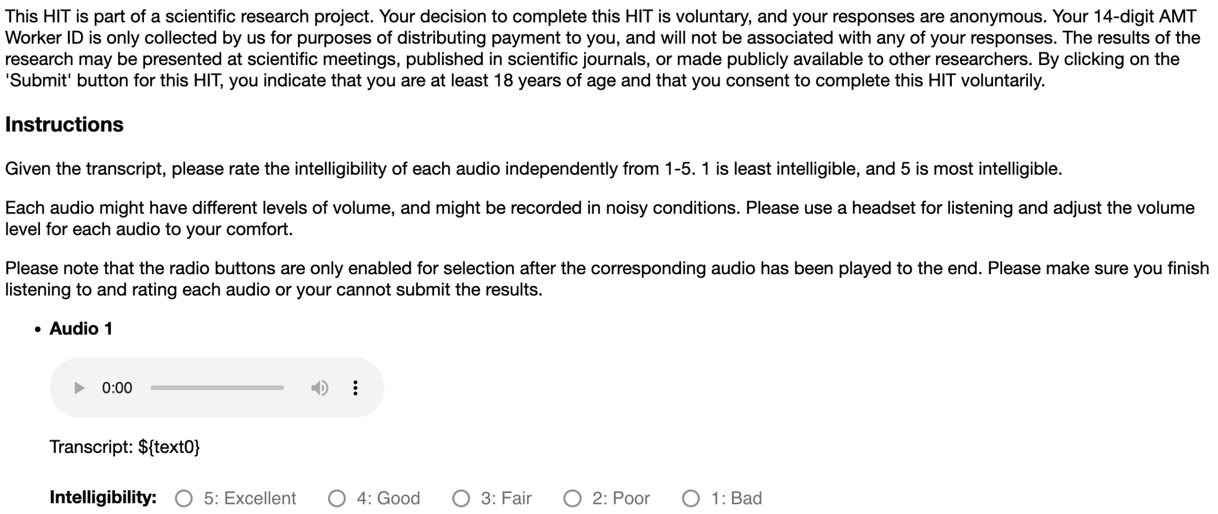
We argue that this assumption is mild, because 1) $I^{(1)}_{s,t,k}$ are tokens from timestep after t and therefore should have less impact on the distribution of $Y_{t,k}$ given past tokens $H_{s,t}$ ($H_{s,t}$ might also contain also future tokens in physical time if $Z_{s,t}$ is in the masked spans); 2) although $I^{(2)}_{s,t,k}$ are tokens from timestep before t, they are from codebooks that are later than codebook k in the residual quantization chain, meaning that they model the residual left by codebook k (at the corresponding timesteps). Given the fact that $\{Y_{s,t-1,k}, Y_{s,t-2,k+1}, \cdots, Y_{s,t-K+k,K-1}\} \subset H_{s,t}$[11], meaning that the "fitted parts" are given, and therefore the "unfitted parts" (which is the residual) should have miner impact on the distribution of $Y_{s,t,k}$. Empirically, MusicGen shows that a codec language model trained with the Delay Pattern enjoys the efficiency of the naive parallel pattern, while achieving similar modeling performance as completely flattened sequence.

E Instructions for human listening test

Screenshots of instructions for the human listening test we used on Amazon Mechanical Turk is shown in figure 8 (speech editing - intelligibility), figure 9(speech editing - naturalness), figure 10 (speech editing - side-by-side comparison), figure 11 (zero-shot TTS - intelligibility), figure 12

[11] A weaker condition holds for the first K tokens in unmasked spans (which accounts for at most 0.08s of speech for our models), but we omit the discussion here for simplicity

(zero-shot TTS - speaker similarity), figure 13 (zero-shot TTS - naturalness). For speech editing evaluation, 64 Turkers participated and we paid 474.3 USD in total; for zero-shot TTS evaluation, 59 Turkers participated and we paid 457.6 USD. We only allow Turkers who are resident of the U.S. to do the tasks, and the goal is to increase the probability of Turkers being native English speakers. We acknowledge that this is a perfect approach and might need to bias in judgement, but since Amazon Mechanical Turk doesn't allow selection on native language, this is the best approach we could think of as a proxy to constraining the native language.



This HIT is part of a scientific research project. Your decision to complete this HIT is voluntary, and your responses are anonymous. Your 14-digit AMT Worker ID is only collected by us for purposes of distributing payment to you, and will not be associated with any of your responses. The results of the research may be presented at scientific meetings, published in scientific journals, or made publicly available to other researchers. By clicking on the 'Submit' button for this HIT, you indicate that you are at least 18 years of age and that you consent to complete this HIT voluntarily.

Instructions

Given the transcript, please rate the intelligibility of each audio independently from 1-5. 1 is least intelligible, and 5 is most intelligible.

Each audio might have different levels of volume, and might be recorded in noisy conditions. Please use a headset for listening and adjust the volume level for each audio to your comfort.

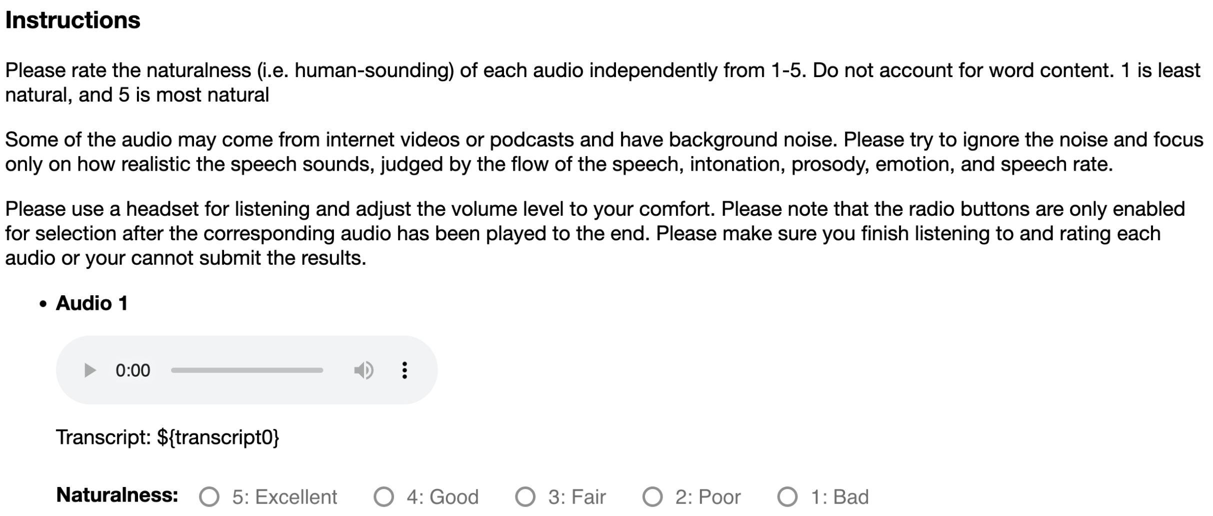
Please note that the radio buttons are only enabled for selection after the corresponding audio has been played to the end. Please make sure you finish listening to and rating each audio or your cannot submit the results.

- **Audio 1**

Transcript: ${text0}

Intelligibility: ○ 5: Excellent ○ 4: Good ○ 3: Fair ○ 2: Poor ○ 1: Bad

Figure 8: Instruction for speech editing-intelligibility preference. Each task contains 5 recordings. Since the first paragraph is also presented in all other tasks in the instruction page, we only show it in this screenshot.



Instructions

Please rate the naturalness (i.e. human-sounding) of each audio independently from 1-5. Do not account for word content. 1 is least natural, and 5 is most natural

Some of the audio may come from internet videos or podcasts and have background noise. Please try to ignore the noise and focus only on how realistic the speech sounds, judged by the flow of the speech, intonation, prosody, emotion, and speech rate.

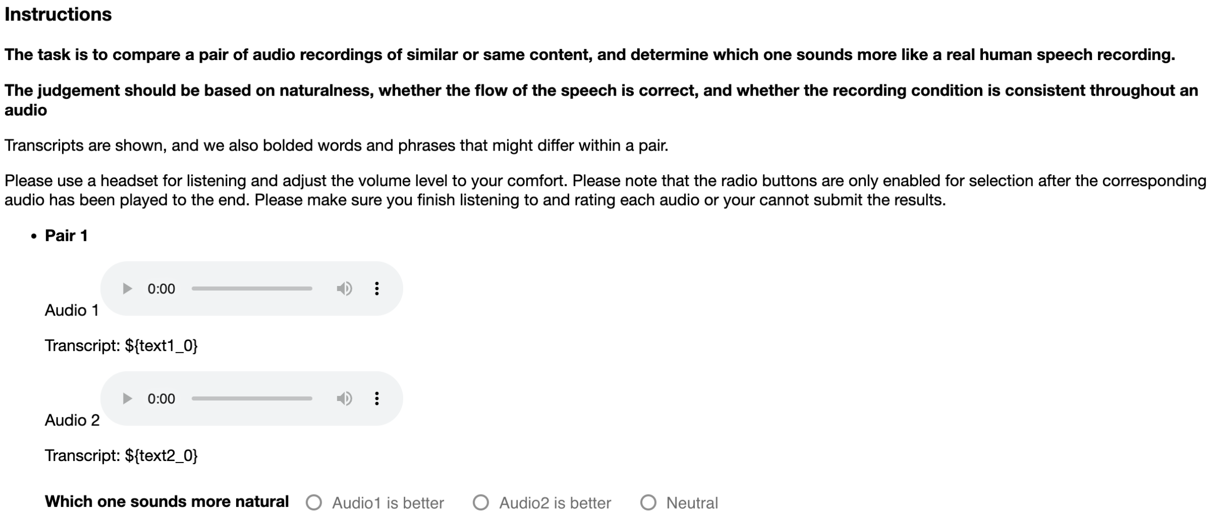
Please use a headset for listening and adjust the volume level to your comfort. Please note that the radio buttons are only enabled for selection after the corresponding audio has been played to the end. Please make sure you finish listening to and rating each audio or your cannot submit the results.

- **Audio 1**

Transcript: ${transcript0}

Naturalness: ○ 5: Excellent ○ 4: Good ○ 3: Fair ○ 2: Poor ○ 1: Bad

Figure 9: Instruction for speech editing-naturalness preference. Each task contains 5 recordings.



Instructions

The task is to compare a pair of audio recordings of similar or same content, and determine which one sounds more like a real human speech recording.

The judgement should be based on naturalness, whether the flow of the speech is correct, and whether the recording condition is consistent throughout an audio

Transcripts are shown, and we also bolded words and phrases that might differ within a pair.

Please use a headset for listening and adjust the volume level to your comfort. Please note that the radio buttons are only enabled for selection after the corresponding audio has been played to the end. Please make sure you finish listening to and rating each audio or your cannot submit the results.

- **Pair 1**

Audio 1

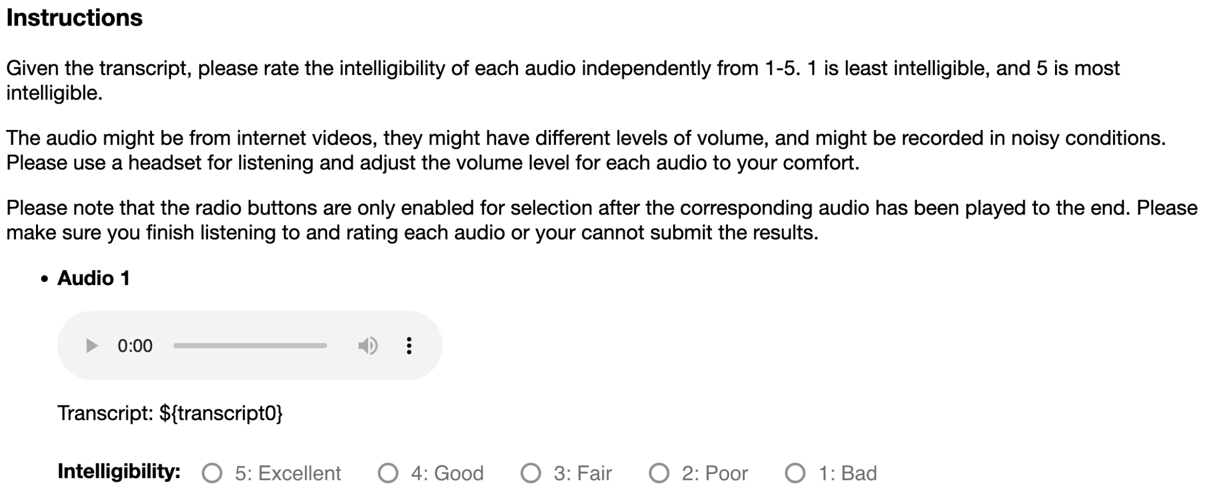
Transcript: ${text1_0}

Audio 2

Transcript: ${text2_0}

Which one sounds more natural ○ Audio1 is better ○ Audio2 is better ○ Neutral

Figure 10: Instruction for speech editing, side-by-side naturalness preference. Each task contains 3 pairs of recordings.



Instructions

Given the transcript, please rate the intelligibility of each audio independently from 1-5. 1 is least intelligible, and 5 is most intelligible.

The audio might be from internet videos, they might have different levels of volume, and might be recorded in noisy conditions. Please use a headset for listening and adjust the volume level for each audio to your comfort.

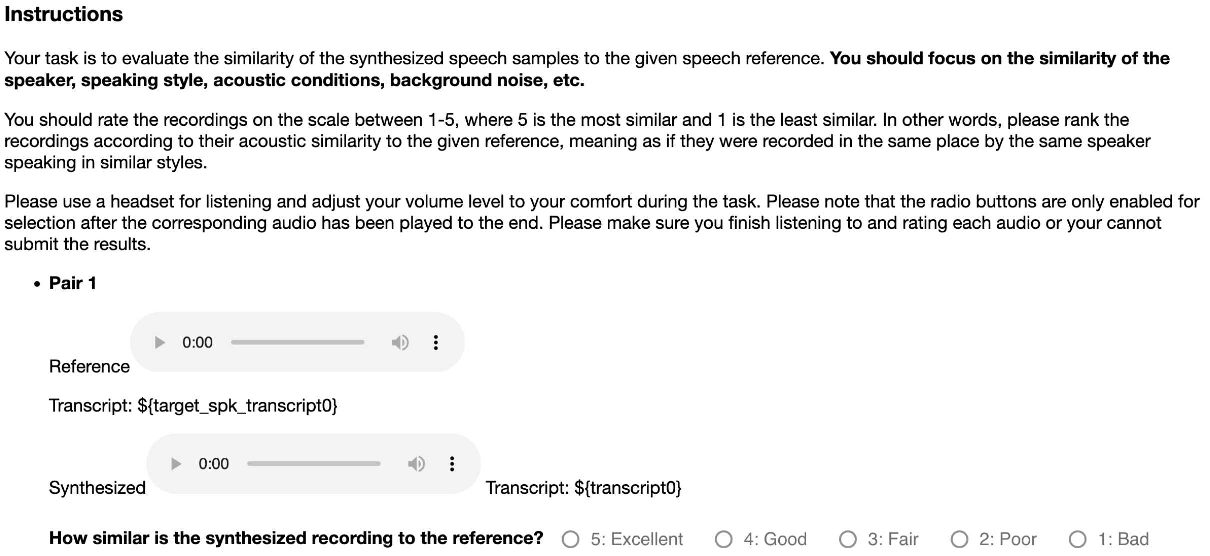
Please note that the radio buttons are only enabled for selection after the corresponding audio has been played to the end. Please make sure you finish listening to and rating each audio or your cannot submit the results.

- **Audio 1**

Transcript: ${transcript0}

Intelligibility: ○ 5: Excellent ○ 4: Good ○ 3: Fair ○ 2: Poor ○ 1: Bad

Figure 11: Instruction for zero-shot TTS, intelligibility preference. Each task contains 5 recordings.



Instructions

Your task is to evaluate the similarity of the synthesized speech samples to the given speech reference. **You should focus on the similarity of the speaker, speaking style, acoustic conditions, background noise, etc.**

You should rate the recordings on the scale between 1-5, where 5 is the most similar and 1 is the least similar. In other words, please rank the recordings according to their acoustic similarity to the given reference, meaning as if they were recorded in the same place by the same speaker speaking in similar styles.

Please use a headset for listening and adjust your volume level to your comfort during the task. Please note that the radio buttons are only enabled for selection after the corresponding audio has been played to the end. Please make sure you finish listening to and rating each audio or your cannot submit the results.

- **Pair 1**

Reference

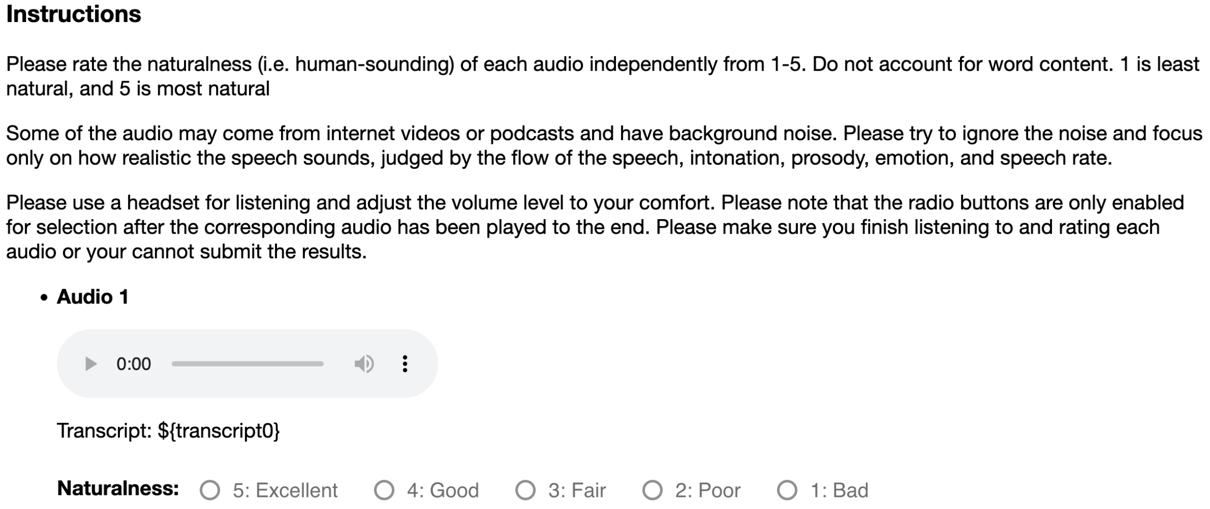
Transcript: ${target_spk_transcript0}

Synthesized

Transcript: ${transcript0}

How similar is the synthesized recording to the reference? ○ 5: Excellent ○ 4: Good ○ 3: Fair ○ 2: Poor ○ 1: Bad

Figure 12: Instruction for zero-shot TTS, speaker similarity preference. Each task contains 3 pairs.



Instructions

Please rate the naturalness (i.e. human-sounding) of each audio independently from 1-5. Do not account for word content. 1 is least natural, and 5 is most natural

Some of the audio may come from internet videos or podcasts and have background noise. Please try to ignore the noise and focus only on how realistic the speech sounds, judged by the flow of the speech, intonation, prosody, emotion, and speech rate.

Please use a headset for listening and adjust the volume level to your comfort. Please note that the radio buttons are only enabled for selection after the corresponding audio has been played to the end. Please make sure you finish listening to and rating each audio or your cannot submit the results.

- **Audio 1**

Transcript: ${transcript0}

Naturalness: ○ 5: Excellent ○ 4: Good ○ 3: Fair ○ 2: Poor ○ 1: Bad

Figure 13: Instruction for zero-shot TTS, naturalness preference. Each task contains 5 recordings.