ParrotTTS: Text-to-speech synthesis exploiting disentangled self-supervised representations

Anonymous ARR submission

Abstract

We present ParrotTTS, a modularized text-to-speech synthesis model leveraging disentangled self-supervised speech representations. It can train a multi-speaker variant effectively using transcripts from a single speaker. ParrotTTS adapts to a new language in low resource setup and generalizes to languages not seen while training the self-supervised backbone. Moreover, without training on bilingual or parallel examples, ParrotTTS can transfer voices across languages while preserving the speaker-specific characteristics, e.g., synthesizing fluent Hindi speech using a French speaker's voice and accent. We present extensive results in monolingual and multi-lingual scenarios. ParrotTTS outperforms state-of-the-art multi-lingual TTS models using only a fraction of paired data as latter. Speech samples from ParrotTTS can be found at `https://parrot-tts.github.io/tts/`

1 Introduction

Vocal learning forms the first phase of infants starting to talk (Locke, 1996, 1994) by simply listening to sounds/speech. It is hypothesized (Kuhl and Meltzoff, 1996) that infants listening to ambient language store perceptually derived representations of the speech sounds they hear, which in turn serve as targets for the production of speech utterances. Interestingly, in this phase, the infant has no conception of text or linguistic rules, and speech is considered sufficient to influence speech production (Kuhl and Meltzoff, 1996) as can parrots (Locke, 1994).

Our proposed ParrotTTS model follows a similar learning process. Our idea mimics the two-step approach, with the first learning to produce sounds capturing the whole gamut of phonetic variations. It is attained by learning quantized representations of sound units in a self-supervised manner using the raw audio data. The second phase builds on top of the first by learning a content mapping from text to quantized speech representations (or embeddings). Only the latter step uses paired text-speech data. The two phases are analogous to first *learning to talk* followed by *learning to read*.

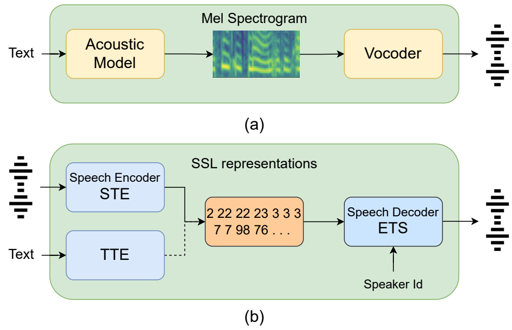


Figure 1: (a) Traditional mel-based TTS and (b) Proposed TTS model

Figure 1 illustrates ParrotTTS contrasting it with the traditional mel-based TTS. The SSL module includes a speech-to-embedding (STE) encoder trained on masked prediction task to learn an embedding representation of the input raw audio (Baevski et al., 2020; Hsu et al., 2021; Van Den Oord et al., 2017). An embedding-to-speech (ETS) decoder is independently trained to invert embeddings to synthesize audio waveforms and is additionally conditioned on speaker identity. This *learning to talk* is the first of the two-step training pipeline. In the subsequent *learning to read* step, a separate text-to-embedding (TTE) encoder is trained to generate embeddings from text (or equivalent phonetic) inputs. This step requires labeled speech with aligned transcriptions.

ParrotTTS offer several advantages over the traditional mel-based neural TTS models (Ren et al., 2020; Wang et al., 2017). For instance, (a) Quantized speech embedding has lower variance than that of Mel frames reducing the complexity to train TTE (b) Direct waveform prediction bypasses potential vocoder generalization issues (Kim et al., 2021). (c) Reduced complexity helps in stabler training of TTE encoder for either autoregressive or non-autoregressive choice. For example, we observe at least eight-fold faster convergence in training iterations of our TTE module compared to that of Ren et al. (2020) and Wang et al. (2017).

While our work closely relates with recent works (Du et al., 2022; Wang et al., 2023; Siuzdak et al., 2022) utilizing self-supervised representations for text-to-speech synthesis, our focus differs by aiming to achieve a unified multi-speaker, multi-lingual TTS system in low-resource scenarios (Xu et al., 2020). In our work, low-resource refers to the scarcity of paired text-to-speech data. Here are the key distinctions of our model compared to recent efforts:

- Unlike contemporary efforts concentrated on large scale training (Wang et al., 2023), we focus on low resource adaptation.
- We employ disentangled self-supervised representations (pol) paired with independently trained STE. This allows us to train multi-speaker TTS using paired data from a single speaker and still adapt it to novel voices with untranscribed speech alone. In contrast, prior efforts either limit to a single speaker TTS (Du et al., 2022) or require paired text-audio data from multiple speakers during training (Siuzdak et al., 2022).
- We show that the ParrotTTS can be extended to a new language with as little as five hours of paired data from a single speaker. The model generalizes to languages unseen during the learning of self-supervised representation.
- Moreover, without training on any bilingual or parallel examples, ParrotTTS can transfer voices across languages while preserving the speaker-specific characteristics. We present extensive results on six languages in terms of speech naturalness and speaker similarity in parallel and cross-lingual synthesis.

Additionally, it's worth mentioning that certain methods (Wang et al., 2023) depend partially or entirely on Automatic Speech Recognition (ASR) to obtain paired data. It should be noted that these ASR models are trained using substantial amounts of supervised data, inaccessible in low resource settings. While architecturally similar to other SSL-based TTS (Wang et al., 2023; Siuzdak et al., 2022), our primary contribution lies in achieving promising outcomes in the low resource scenario, where minimal paired data from a single speaker per language is accessible for TTS training.

2 Related work

2.1 Foundational Neural TTS models

Traditional neural TTS model encodes text or phonetic inputs to hidden states, followed by a decoder that generates Mels from the hidden states. Predicted Mel frames contain all the necessary information to reconstruct speech (Griffin and Lim, 1984) and an independently trained vocoder (Oord et al., 2016; Kong et al., 2020) transforms them into time-domain waves. Mel predicting decoders could be autoregressive/sequential (Wang et al., 2017; Valle et al., 2020; Shen et al., 2018) or non-autoregressive/parallel (Ren et al., 2019, 2020; Łańcucki, 2021). Non-autoregressive models additionally predict intermediate features like duration, pitch, and energy for each phoneme. They are faster at inference and robust to word skipping or repetition errors (Ren et al., 2020). Multi-speaker capabilities are often achieved by conditioning the decoder on speaker embeddings (one-hot embeddings or ones obtained from speaker verification networks (Jia et al., 2018)). Training multi-speaker TTS models requires paired text-audio data from multiple speakers. Methods relying on speaker-embeddings can, in theory, perform zero-shot speaker adaptation; however, the rendered speech is known to be of poorer quality, especially for speakers not sufficiently represented in the train set (Tan et al., 2021).

2.2 Raw-audio for TTS

Unsupervised speech synthesis (Ni et al., 2022) does not require transcribed text-audio pairs for training. They typically employ unsupervised ASR (Baevski et al., 2021; Liu et al., 2022a) to transcribe raw speech to generate pseudo labels. However, their performance tends to be bounded by the performance of the unsupervised ASR model, which still has to close a significant gap compared to supervised counterparts (Baevski et al., 2021). Switching to a multi-speaker setup further widens this quality gap (Liu et al., 2022b).

Some prior works have looked at adapting TTS to novel speakers using untranscribed audio (Yan et al., 2021; Luong and Yamagishi, 2019; Taigman et al., 2017). Unlike ours, their methods require a large amount of paired data from multiple speakers during initial training. Some of these (Luong and Yamagishi, 2019; Taigman et al., 2017) jointly train the TTS pipeline and the modules for speaker adaptation but model training's convergence is trickier.

In contrast, ParrotTTS benefits from the disentanglement of linguistic content from speaker information, making adaptation easier with stabler training as we observe in our experiments.

2.3 Self-supervised learning

Self-supervised learning (SSL) methods are becoming increasingly popular in speech processing due to their ability to utilize abundant unlabeled data. Techniques like masked prediction, temporally contrastive learning, and next-step prediction are commonly used to train SSL models. Popular models like Wav2vec2 (Baevski et al., 2020), VQ-VAE (Van Den Oord et al., 2017), AudioLM (Borsos et al., 2022) and HuBERT (Hsu et al., 2021) have been successfully deployed in tasks like ASR (Baevski et al., 2020), phoneme segmentation (Kreuk et al., 2020), spoken language modeling (Lakhotia et al., 2021), and speech resynthesis (pol).

Our work is related to recent efforts (Du et al., 2022; Wang et al., 2023; Siuzdak et al., 2022) that utilize self-supervised audio embeddings in text-to-speech synthesis. However, those of Du et al. (2022) and Siuzdak et al. (2022) require speaker-specific SSL embeddings while we use generic HuBERT embeddings (Hsu et al., 2021; Lee et al., 2022) train for multiple speakers.

2.4 Multi-lingual TTS

It is challenging to build an unified TTS model supporting multiple languages and speakers, even more so for cross lingual synthesis, *i.e.*, allowing multiple languages to be spoken in each of the speaker's voices. The primary challenge is in acquiring paired data to train language dependent components that often includes its embeddings. The trick ParrotTTS employs to break this data dependence is to decouple acoustics from content handling, of which only the latter is language dependent and requires paired data while the former is deferred to self-supervised models.

Initial attempts (Liu and Mak, 2019; Zhang et al., 2019) address these by conditioning the decoder on language and speaker embeddings, but the results were subpar due to entanglement of text representation with language/speaker information. Recent approaches (Zhang et al., 2019; Cho et al., 2022; Nekvinda and Dušek, 2020) addressed this issue by incorporating an explicit disentanglement loss term, using reverse gradients through a language or speaker classification branch.

Nekvinda and Dušek (2020) propose MetaTTS, that uses a contextual parameter generation through language-specific convolutional text encoders. Cho et al. (2022) extend MetaTTS with a speaker regularization loss and investigate different input formats for text. Knowledge sharing (Prakash et al., 2019) and distillation (Xu et al., 2020) have been explored for multi-lingual TTS. Recently, Wu et al. (2022) employ a data augmentation technique based on a cross-lingual voice conver- sion model trained with speaker-invariant features extracted from a speech representation.

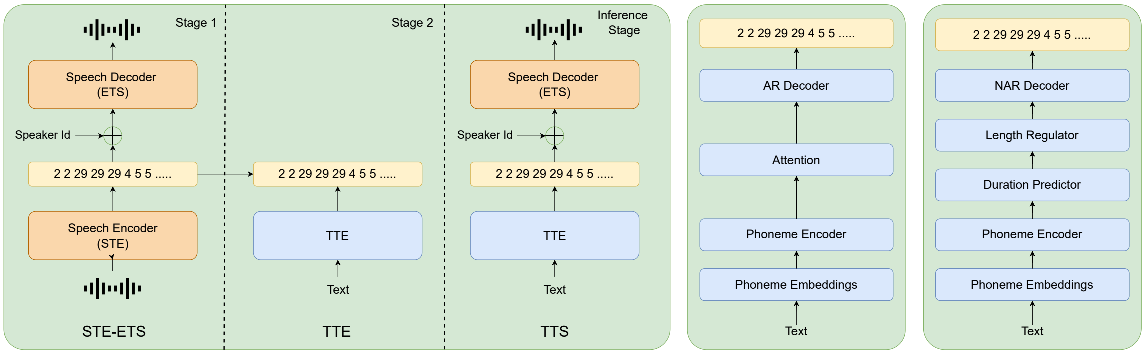
Certain limitations still persist in existing approaches (Nekvinda and Dušek, 2020; Chen et al., 2019; Zhang et al., 2019; Zhang and Lin, 2020). For example, many of them rely on Tacotron (Wang et al., 2017) as their backbone, which is prone to word alignment and repetition errors. Prior multilingual TTS models typically support only 2-3 languages simultaneously or require extensive training data as noted by Nekvinda and Dušek (2020). Additionally, they have not yet capitalized on self-supervised embeddings and our efforts aim to address this gap.

3 ParrotTTS architecture

ParrotTTS has three modules; two encoders that map speech or text inputs to common embedding space (referred to as STE and TTE respectively) and a decoder (ETS) that renders speech signal from these embeddings. Our speech encoder-decoder choices are borrowed from (pol). Our speech decoder ETS is a modified version of HiFi-GAN (Kong et al., 2020). Text encoder TTE is an encoder-decoder architecture and we experiment with both autoregressive (AR) and non-autoregressive (NAR) choices for the same.

3.1 Speech encoder STE

The self-supervised HuBERT model we use for our STE is pre-trained on large raw audio data from English, on BERT-like masked prediction task (Devlin et al., 2018) to learn "combined acoustic and language model over the continuous inputs" of speech. It borrows the base architecture from Wav2vec 2.0 (Baevski et al., 2020) with convolutions on raw inputs followed by a few transformer layers, however, replaces its contrastive loss with a BERT-like classification. The "noisy" classes for this classification are derived by clustering MFCC features of short speech signals. Encoder input is



(a) ParrotTTS (b) Autoregressive TTE (c) Non Autoregressive TTE

Figure 2: (a) ParrotTTS performs a two stage training. In stage1, ETS is trained to synthesize speech from discrete units obtained though an independently trained STE module. In Stage2, TTE learns to map text sequence to corresponding speech units obtained from STE. (b) and (c) illustrate the explored TTE architectures.

audio signal $X = (x_1, ...x_T)$ sampled at a rate of 16kHz. Let E_r denote the raw-audio encoder, and its output be,

$$\mathbf{h}_r = (h_1, ..., h_{\widehat{T}}) := E_r(X),$$

where $\widehat{T} = T/320$ indicates downsampling and each $h_i \in \{1, \ldots, K\}$ with K being the number of clusters in HuBERT's clustering step, set to 100 in our experiments. For multi-lingual experiments, instead of using HuBERT, we utilize mHuBERT (Lee et al., 2022), which is trained on a multi-lingual corpus. We use K=1000 for mHuBERT embeddings.

3.2 Speech decoder ETS

We adapt the HiFiGAN-v2 vocoder for our ETS to decode from $\mathbf{h} = (\mathbf{h}_r, \mathbf{h}_s)$ to speech, where $\mathbf{h}_s$ is the one-hot speaker embedding. It has a generator G and a discriminator D. G runs $\mathbf{h}$ through transposed convolutions for upsampling to recover the original sampling rate followed by residual block with dilations to increase the receptive field to synthesize the signal, $\widehat{X} := G(\mathbf{h})$.

The discriminator distinguishes synthesized $\widehat{X}$ from the original signal X and is evaluated by two sets of discriminator networks. Multi-period discriminators operate on equally spaced samples, and multi-scale discriminators operate at different scales of the input signal. Overall, the model attempts to minimize $D(X, \widehat{X})$ over all its parameters to train ETS.

3.3 Text encoder TTE

The third module we train, TTE is a text encoder that maps phoneme/character sequence $P = (p_1, \ldots, p_N)$ to embedding sequence $\mathbf{h}_p = (h_1, \ldots, h_{\widehat{N}})$. We train a sequence-to-sequence architecture to achieve this $\mathbf{h}_p := E_p(P)$. E_p initially encodes P into a sequence of fixed dimensional vectors (phoneme embeddings), conditioned upon which its sequence generator produces variable dimensional $\mathbf{h}_p$. Embedding $\mathbf{h}_p$ is intended to mimic $\mathbf{h}_r := E_r(X)$ extracted from the audio X corresponding to the text P. Hence, the requirement of transcribed data (X, P) to derive the target $\mathbf{h}_r$ for training TTE by optimizing over the parameters of E_p.

One could model E_p to generate $\mathbf{h}_p$ autoregressively one step at a time, which we refer to as AR-TTE model (Figure 2(b)). Input phoneme sequence is encoded through a feed-forward transformer block that stacks self-attention layers (Vaswani et al., 2017) and 1D convolutions similar to FastSpeech2 (Ren et al., 2019). Decoding for $\mathbf{h}_p$ uses a transformer module with self-attention and cross-attention. Future-masked self-attention attends to ground truth at train and to previous decoder predictions at inference. Cross-attention attends to phoneme encoding in both cases.

Alternatively, for a non-autoregressive choice of E_p, the model NAR-TTE determines the output length $\widehat{N}$ followed by a step to simultaneously predict all $\widehat{N}$ entries of $\mathbf{h}_p$. Figure 2(c) depicts NAR-TTE where the input phoneme sequence encoding is similar to that of AR-TTE. The duration predictor and length regulator modules are responsible for determining $\widehat{N}$ followed by the decoding step to generate $\mathbf{h}_p$. In multi-lingual scenario, we investigate both character and phoneme sequences for representing the input text. For character representation, we extract the tokens using a dictionary created by iterating over the entire text corpus. In contrast, for phoneme representation, we utilize an off-the-shelf phonemizer (version: 3.2.1) (Bernard and Titeux, 2021) to extract phonemes belonging to the IPA vocabulary, which are common across languages.

4 Experiments

We perform experiments in monolingual and multi-lingual scenarios. Details of various ParrotTTS models trained and of those each of them is compared to is covered below.

4.1 ParrotTTS training

Datasets (monolingual) For single language experiments, we use two public datasets. LJSpeech (Ito and Johnson, 2017) provides 24 hours high quality transcribed data from a single speaker. Data are split into two, with 512 samples set aside for validation and the remaining available for model training. VCTK (Veaux et al., 2017) with about 44 hours of transcribed speech from 108 different speakers is used for the multi-speaker setup. It has a minimum, average, and maximum of 7, 22.8, and 31 minutes per speaker speech length, respectively.

Datasets (multi-lingual) We collate our multi-lingual dataset using publicly available corpora containing samples from multiple speakers in six languages: (1) 80.76 hours of Hindi and Marathi from (in INdian languages , SYSPIN) from 2 speakers, respectively; (2) 71.69 hours of German (GmbH., 2017) from 3 speakers; (3) 83.01 hours of Spanish (GmbH., 2017) from 3 speakers; (4) 10.70 hours of French (Honnet et al., 2017) from 1 speaker; (5) 23.92 hours of English (Ito and Johnson, 2017) from 1 speaker. Overall the dataset comprises of 354.12 hours of paired TTS data from 12 speakers across all six languages. We resample all speech samples to 16 kHz.

STE training. We use a 12 layer transformer model for HuBERT training. It is trained using 960 hour-long LibriSpeech corpus (Panayotov et al., 2015). The multi-lingual variant mHuBERT is trained using 13.5k hours of English, Spanish and French data from VoxPopuli unlabelled speech corpus (Lee et al., 2022; Wang et al., 2021). In both cases, the model splits each T seconds long audio into units of $T/320$ seconds and maps each of the obtained units to a 768 dimensional vector.

TTE training (monolingual). We use LJSpeech to train two different TTE encoder modules; TTE_{LJS} that uses all the data from our LJSpeech train set and a second, $\text{TTE}_{\frac{1}{2}\text{LJS}}$ with only half the data. This is used to understand the effect of training data size on TTS performance. All variants of TTE we experiment with are trained only on samples from the single speaker in LJSpeech data.

Text converted to phoneme sequence as described by Sun et al. (2019) are inputs with $\mathbf{h}_r$ targets extracted from STE for training. Additionally, NAR-TTE requires phonetic alignment to train the duration predictor. We use Montreal forced-aligner (McAuliffe et al., 2017) to generate them for its training. We use cross-entropy loss with the 100 clusters derived from discretization codebook of HuBERT units as classes.

TTE training (multi-lingual). Focusing on low-resource setting, we use only 5 hours of paired data for a single speaker in each language to train the TTE that totals to merely 30 hours of paired data across all languages. We report the evaluation metrics for *seen speakers* where the model has seen the speaker paired data and *unseen speakers* whose paired data is not used to train the TTE. To evaluate the performance on various text representations, we train two variants of the TTE , the character TTE (CTE) and the phoneme TTE (PTE). CTE uses character tokens across the languages to learn sound units while PTE uses phoneme tokens. Additionally, we employ Deep Forced Aligner (in INdian languages , SYSPIN) to align ground-truth speech and input text representations to train the duration predictor. Cross-entropy loss with 1000 clusters of mHuBERT are used as classes to predict $\mathbf{h}_p$.

ETS training. We train a single-speaker ETS, SS-ETS using only speech clips from LJSpeech since its training does not require transcriptions. Similarly, our multi-speaker ETS, MS-ETS decoder model uses only raw audio of all speakers from VCTK data (Veaux et al., 2017). So only embeddings $\mathbf{h}_r$ extracted from VCTK audio clips are used along with one-hot speaker vector $\mathbf{h}_s$. We emphasize that VCTK data were used only in training the multi-speaker-ETS module, and the TTE has not seen any from this set. For multi-lingual scenario, we train a multi-speaker ETS using speech-only data with 12 speakers from all six languages.

4.2 Comparison to prior art

Single Speaker TTS: We train Tacotron2 (Wang et al., 2017) and FastSpeech2 (Ren et al., 2020) using the ground truth transcripts of LJspeech and referred to as SS-Tacotron2 and SS-FastSpeech2. We additionally trained an unsupervised version of FastSpeech2 by replacing the ground truth transcripts with transcriptions obtained from the ASR model. FastSpeech2-SupASR uses supervised ASR model (Radford et al., 2022) to generate the transcripts while Tacotron2-UnsupASR (Ni et al., 2022) alternatively uses unsupervised ASR Wav2vec-U 2.0 (Liu et al., 2022a). We further adapt WavThruVec (Siuzdak et al., 2022) to our setup and train a model (SS-WavThruVec) using intermediate embeddings extracted from Wav2Vec 2.0 (Baevski et al., 2020). Additionally, we apply a similar approach to the embeddings obtained from VQ-VAE (Van Den Oord et al., 2017) and term it as SS-VQ-VAES. We compare against three variants of ParrotTTS;

1. AR-TTE$_{\text{LJS}}$+SS-ETS that is autoregressive TTE trained on full LJSpeech with single speaker ETS,
2. NAR-TTE$_{\text{LJS}}$+SS-ETS that pairs TTE with non-autoregressive decoding trained on full LJSpeech with single speaker ETS, and
3. NAR-TTE$_{\frac{1}{2}\text{LJS}}$+SS-ETS that uses TTE with non-autoregressive decoding trained on half LJSpeech with single speaker ETS.

Multi-speaker TTS: We compare against a fully supervised Fastspeech2 baseline trained on VCTK using paired data from all speakers and that we refer to as MS-FastSpeech2. For ParrotTTS we borrow the TTE module trained on LJSpeech and use the raw audio of VCTK to train the multi-speaker ETS module. We refer to this multi-speaker variant of our ParrotTTS model as NAR-TTE$_{\text{LJS}}$+MS-ETS that uses non-autoregressive decoding.

For a fair comparison, we also curate a multi-speaker TTS baseline using a combination of single-speaker TTS and a voice cloning model. We use FastSpeech2 trained on LJspeech with state-of-the-art voice cloning model (pol) in our experiments and refer to this model as VC-FastSpeech2. We also compare against multi-speaker TTS trained by obtaining pseudo labels from a supervised ASR called MS-FastSpeech2-SupASR. Additionally, we also report numbers from GT-Mel+Vocoder that converts ground truth Mels from actual audio clip back to speech using a vocoder (Kong et al., 2020) for a perspective of best achievable with ideal Mel frames.

Multi-lingual TTS: We compare against, (a) FastSpeech2-MLS which is a fully-supervised FastSpeech2 model and (b) state-of-the-art meta learning-based multi-lingual TTS model MetaTTS (Nekvinda and Dušek, 2020). Both these models are trained on the entirety of train data (354 hours of transcribed speech). In contrast, the TTE training in ParrotTTS model (our sole module that needs paired data) uses only $1/12^{th}$ of this *i.e*, a total of 30 hours of paired text-speech (5 hours per language). The remaining data is used for evaluation purposes, serving as the test set. We refer to this model as NAR-TTE$_{\frac{1}{12}\text{MLS}}$+ML-ETS. We also compare character (CTE) and phoneme (PTE) tokenization for encoding text in this setting.

4.3 Evaluation metrics

We evaluate the intelligibility of various models using Word Error Rate (WER) with the pre-trained Whisper *small* model (Radford et al., 2022). We validate the speaker adaptability using Equal Error Rate (EER) from a pre-trained speaker verification network proposed in (Desplanques et al., 2020) and trained on VoxCeleb2 (Chung et al., 2018). The WER and EER metrics are computed on entire validation set. We perform subjective evaluations using Mean Opinion Score (MOS) with five native speakers per language, rating samples synthesized by different models, where five sentences from the test set are randomly selected for evaluation.

5 Results

5.1 Single-speaker TTS

Naturalness and intelligibility. As shown in Table 1, ParrotTTS is competitive to state-of-the-art in the single-speaker setting. In the autoregressive case, our AR-TTE$_{\text{LJS}}$+SS-ETS has a statistically insignificant drop (of about 0.05 units) on the MOS scale relative to the Tacotron2 baseline. The non-autoregressive case has a similar observation (with a 0.01 drop) on MOS in our NAR-TTE$_{\text{LJS}}$+SS-ETS model relative to FastSpeech2. This empirically establishes that the naturalness of the speech rendered by ParrotTTS is on par with the currently established methods. The WER scores show a similar trend with a statistically insignificant drop (of

	Model	MOS ↑	WER ↓
Traditional TTS	SS-FastSpeech2	3.87	4.52
	SS-Tacotron2	3.90	4.59
	FastSpeech2-SupASR	3.78	4.72
	Tacotron2-UnsupASR	3.50	11.3
WavThruVec	SS-WavThruVec	3.57	6.27
VQ-VAE	SS-VQ-VAES	3.12	21.78
ParrotTTS	AR-TTE$_{\text{LJS}}$+SS-ETS	3.85	4.80
	NAR-TTE$_{\text{LJS}}$+SS-ETS	3.86	4.58
	NAR-TTE$_{\frac{1}{2}\text{LJS}}$+SS-ETS	3.81	6.14

Table 1: Subjective and objective comparison of TTS models in the single speaker setting.

Model	VCTK	MOS ↑	WER ↓	EER ↓
GT-Mel+Vocoder	Yes	4.12	2.25	2.12
MS-FastSpeech2	Yes	3.62	5.32	3.21
MS-FastSpeech2-SupASR	No	3.58	6.65	3.85
VC-FastSpeech2	No	3.41	7.44	8.18
WavThruVec-MS	No	3.17	6.79	5.08
NAR-TTE$_{\text{LJS}}$+MS-ETS	No	3.78	6.53	4.38

Table 2: Comparison of the multi-speaker TTS models on the VCTK dataset. Column 2 indicates if the corresponding method uses VCTK transcripts while training.

under 0.2pp[1]) among the autoregressive and non-autoregressive model classes. The performance of SS-WavThruVec and SS-VQ-VAES is lower in both naturalness and intelligibility, indicating that the utilization of Wav2Vec 2.0 and VQ-VAE embeddings results in a decrease in performance.

Supervision and data efficiency. In the study to understand how the degree of supervision affects TTS speech quality, we see a clear drop by 0.28 MOS units in moving from the FastSpeech2-SupASR model that employs supervised ASR for transcriptions to Tacotron2-UnsupASR model using unsupervised ASR. Despite some modeling variations, this is generally indicative of the importance of clean transcriptions on TTS output quality, given that all other models are within 0.05 MOS units of each other.

The data requirement for TTS supervision needs to be understood in light of this impact on output quality, and we show how ParrotTTS helps cut down on this dependence. TTE is the only module that needs transcriptions to train, and we show that by reducing the size of the train set by half in NAR-TTE$_{\frac{1}{2}\text{LJS}}$+SS-ETS the MOS is still comparable to that of the model trained on all data NAR-TTE$_{\text{LJS}}$+SS-ETS (with only about 0.04 units MOS drop). Finally, the MOS numbers of FastSpeech2-SupASR, need to be read with some caution since the supervised ASR model used, Whisper, is itself trained with plenty of transcriptions (spanning over 600k hours) from the web, including human and machine transcribed data achieving very low WERs on various public and test sets. So, the machine transcriptions used in FastSpeech2-SupASR are indeed close to ground truth.

5.2 Multi-speaker TTS

Naturalness and intelligibility. Table 2 summarizes results from our multi-speaker experiments. NAR-TTE$_{\text{LJS}}$+MS-ETS clearly outperforms all other models ranking only below GT-Mel+Vocoder that re-synthesizes from ground truth Mels. Interestingly, ParrotTTS fares even better than MS-FastSpeech2, which is, in turn, better than other models that ignore transcripts at the train, namely, MS-FastSpeech2-SupASR and VC-FastSpeech2. On the WER metric for intelligibility, ParrotTTS is about 1pp behind supervised MS-FastSpeech2 but fares better than the other two models that discard VCTK transcripts for training. WavThruVec-MS model leveraging Wav2Vec 2.0 embeddings has a noticeable quality drop in the multi-speaker setting with lowest MOS.

Speaker adaptability. VC-FastSpeech2 is the closest in terms of experimental setup since it is limited to transcriptions from LJSpeech for training similar to ours, with VCTK used only for adaptation. In this case, EER of NAR-TTE$_{\text{LJS}}$+MS-ETS is about twice as good as that of VC-FastSpeech2. However, improvements are visible when VCTK transcripts are part of training data but remain within 1pp relative to ParrotTTS while GT-Mel+Vocoder continues to dominate the scoreboard leaving room for further improvement.

5.3 Multi-lingual TTS

The results from our multi-lingual experiments are in Tables 3, 4, 5, and 6. It is notable that speech rendered by ParrotTTS has superior naturalness compared to baselines that are trained with twelve times more paired samples stressing its viability for low-resource languages. Further, the naturalness also changes with the text tokenization method. Choosing character tokens for Indic languages outperformed phoneme tokens while it was the opposite for the European languages. ParrotTTS with the best performing tokeniser in each language was superior to FastSpeech2-MLS and MetaTTS for both *seen speakers* (Table 3) as well as *unseen speakers* (Table 4). It is interesting to note that scores for ParrotTTS were better than groundtruth and this is possibly due to noise in original sample

[1]Percentage points abbreviated as pp.

	GT	CTE (Ours)	PTE (Ours)	FS2-MLS	MetaTTS
Hindi	3.78 ± 0.14	$\mathbf{3.33 \pm 0.19}$	3.22 ± 0.15	3.33 ± 0.12	2.12 ± 0.12
Marathi	4.81 ± 0.07	$\mathbf{3.78 \pm 0.12}$	3.04 ± 0.19	3.59 ± 0.15	2.13 ± 0.15
German	3.54 ± 0.20	3.33 ± 0.19	$\mathbf{3.58 \pm 0.12}$	3.21 ± 0.16	1.8 ± 0.15
French	3.83 ± 0.19	2.23 ± 0.14	$\mathbf{4.17 \pm 0.19}$	3.50 ± 0.16	1.7 ± 0.16
English	4.20 ± 0.12	3.11 ± 0.11	$\mathbf{3.50 \pm 0.10}$	2.50 ± 0.18	1.6 ± 0.17
Spanish	3.67 ± 0.12	3.5 ± 0.21	$\mathbf{3.67 \pm 0.20}$	2.50 ± 0.21	2.1 ± 0.15

Table 3: Comparison of naturalness MOS on seen speakers with FastSpeech2-MLS (FS2-MLS) and MetaTTS model

	GT	CTE (Ours)	PTE (Ours)	FS2-MLS	MetaTTS
Hindi	4.22 ± 0.18	$\mathbf{3.28 \pm 0.19}$	3.05 ± 0.20	3.22 ± 0.21	2.02 ± 0.18
Marathi	4.48 ± 0.13	$\mathbf{3.63 \pm 0.18}$	3.11 ± 0.18	3.15 ± 0.19	1.91 ± 0.19
German	3.17 ± 0.22	2.72 ± 0.23	$\mathbf{3.55 \pm 0.20}$	2.05 ± 0.22	1.8 ± 0.17
Spanish	3.67 ± 0.19	3.17 ± 0.17	$\mathbf{3.33 \pm 0.18}$	3.17 ± 0.19	1.3 ± 0.16

Table 4: Comparison of naturalness MOS on unseen speakers with FastSpeech2-MLS (FS2-MLS) and MetaTTS model

Language	Our model	FS2-MLS	MetaTTS
Hindi	$\mathbf{4.29 \pm 0.18}$	3.92 ± 0.21	2.23 ± 0.19
Marathi	$\mathbf{4.21 \pm 0.16}$	3.83 ± 0.08	2.12 ± 0.16
German	$\mathbf{4.09 \pm 0.11}$	3.25 ± 0.14	2.05 ± 0.14
French	$\mathbf{3.87 \pm 0.20}$	3.50 ± 0.19	2.24 ± 0.17
English	$\mathbf{3.94 \pm 0.18}$	3.00 ± 0.19	2.32 ± 0.19
Spanish	$\mathbf{4.33 \pm 0.17}$	3.50 ± 0.19	2.0 ± 0.18

Table 5: Comparison of speaker similarity MOS with FastSpeech2-MLS (FS2-MLS) and MetaTTS model

Speaker-Text	Our model	FS2-MLS	MetaTTS
Hindi-Spanish	$\mathbf{3.87 \pm 0.22}$	3.25 ± 0.19	1.26 ± 0.15
Marathi-English	$\mathbf{3.63 \pm 0.21}$	3.5 ± 0.22	1.23 ± 0.19
French-Hindi	$\mathbf{4.07 \pm 0.12}$	2.71 ± 0.21	1.23 ± 0.16
Spanish-German	$\mathbf{4.14 \pm 0.20}$	2.29 ± 0.21	1.45 ± 0.19
English-German	$\mathbf{3.57 \pm 0.15}$	2.43 ± 0.18	1.56 ± 0.16
English-Hindi	$\mathbf{3.57 \pm 0.19}$	2.57 ± 0.18	1.23 ± 0.19
French-German	$\mathbf{3.93 \pm 0.17}$	2.71 ± 0.18	1.18 ± 0.17
Spanish-French	$\mathbf{3.71 \pm 0.18}$	2.57 ± 0.17	1.4 ± 0.16
Hindi-Marathi	$\mathbf{4.13 \pm 0.21}$	3.25 ± 0.19	1.3 ± 0.18
Marathi-French	$\mathbf{2.87 \pm 0.19}$	2.75 ± 0.18	1.25 ± 0.19

Table 6: Comparison of naturalness MOS for cross-lingual speech synthesis with FastSpeech2-MLS (FS2-MLS) and MetaTTS model

that was suppressed by HuBERT embeddings that are known to discard ambient information.

Speaker similarity. Results in Table 5 consistently demonstrate the superiority of ParrotTTS over FastSpeech2-MLS and MetaTTS, indicating its effectiveness in separating speaker and content information. This is attributed to the decoder being conditioned solely on speaker ID while sharing the acoustic space across all languages.

Cross lingual synthesis. We also assess the model's performance in synthesizing samples of a speaker in a language different from native language. Table 6 presents these results comparing naturalness of MOS in a cross-lingual setting. The first column lists a pair of languages of which the first is the speaker's native language while the second is language of text that is rendered. ParrotTTS achieved higher MOS demonstrating strong decoupling of content from speaker characteristics that is controlled in the decoder. Further, more than 90% of the participants were able to discern the nativity of the synthesized speech.

5.4 Stabler training and faster inference

We observe that NAR-TTE converges (in 20k steps) about eight times faster than FastSpeech2 (160k steps) during training. Similarly, AR-TTE model converges 10-times faster than the corresponding Tacotron2 counterpart. The proposed NAR-TTE system also improves inference latency and memory footprint. On NVIDIA RTX 2080 Ti GPU, we observe ParrotTTS serves 15% faster than FastSpeech2. Furthermore, the TTE module uses 17M parameters in contrast to 35M parameters of the Mel synthesizer module in Fastspeech2. More details are provided in the supplementary material.

6 Conclusion, limitations and future work

We investigate a data-efficient ParrotTTS model that leverages audio pre-training from self-supervised models and ties it to separately trained speech decoding and text encoding modules. We evaluate this architecture in various settings. Quality of rendered speech with as little as five hours of paired data per language is on par with or superior to competitive baselines. This is the key result from our experiments that we believe will help scale TTS training easily to new languages by bringing low-resource ones into the same quality range as the resource-rich ones.

In the future, we plan to fine-tune the Hubert-based embeddings on diverse set of languages (South Asian, Latin, English, etc.) to create a more comprehensive set of sound units. Another direction being to improve upon the data efficiency for speaker adaptability (Wang et al., 2023). Investigations into emotive speech and controllable generation is another aspect. For example, Hubert embeddings are known to skip prosody information (Kharitonov et al., 2021) and hence giving emotive affect to speech would be a challenge in this setup. Finally, we aim to release an open-source, multi-lingual TTS model to enable the wider application of our findings to resource-scarce and less privileged languages.

7 Ethical Considerations

Our research is grounded in ethical considerations. We recognize the potential of text-to-speech synthesis in various domains, such as accessibility, human-computer interaction, telecommunications, and education. However, we acknowledge the risk of misuse, particularly with regards to unethical cloning and the creation of false audio recordings. Our experiments strictly use publicly available datasets and our method does not aim to synthesize someone's voice without their consent. We are mindful of the negative consequences associated with these actions. While the benefits currently outweigh the concerns, we strongly advocate for the research community to actively explore methods for detecting and preventing misuse.

It is important to note that our approach is trained on a limited set of languages and has not been validated on different languages or individuals with speech impediments. Therefore, the dataset and results may not be representative of the entire population. A comprehensive understanding of this issue necessitates further studies in conjunction with linguistic and socio-cultural insights.

References

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. *Advances in Neural Information Processing Systems*, 34:27826–27839.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

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, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2022. Audiolm: a language modeling approach to audio generation. *arXiv preprint arXiv:2209.03143*.

Mengnan Chen, Minchuan Chen, Shuang Liang, Jun Ma, Lei Chen, Shaojun Wang, and Jing Xiao. 2019. Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding. In *Interspeech*, pages 2105–2109.

Hyunjae Cho, Wonbin Jung, Junhyeok Lee, and Sang Hoon Woo. 2022. SANE-TTS: Stable And Natural End-to-End Multilingual Text-to-Speech. In *Proc. Interspeech 2022*, pages 1–5.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chenpeng Du, Yiwei Guo, Xie Chen, and Kai Yu. 2022. VQTTS: High-Fidelity Text-to-Speech Synthesis with Self-Supervised VQ Acoustic Feature. In *Proc. Interspeech 2022*, pages 1596–1600.

Munich Artificial Intelligence Laboratories GmbH. 2017. The m-ailabs speech dataset. https://github.com/imdatsolak/m-ailabs-dataset.

Daniel Griffin and Jae Lim. 1984. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243.

Pierre-Edouard Honnet, Alexandros Lazaridis, Philip N Garner, and Junichi Yamagishi. 2017. The siwis french speech synthesis database? design and recording of a high quality french database for speech synthesis. Technical report, Idiap.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

SYnthesizing SPeech in INdian languages (SYSPIN). 2017. Deep forced alligner. https://github.com/bloodraven66/DeepForcedAligner.

SYnthesizing SPeech in INdian languages (SYSPIN). 2022. Text-to-speech synthesizer in nine indian languages. https://syspin.iisc.ac.in/datasets.

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

Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu-Anh Nguyen, Morgane Rivière, Abdelrahman Mohamed, Emmanuel Dupoux, et al. 2021. Text-free prosody-aware generative spoken language modeling. *arXiv preprint arXiv:2109.03264*.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033.

Felix Kreuk, Joseph Keshet, and Yossi Adi. 2020. Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation. In *Proc. Interspeech 2020*, pages 3700–3704.

Patricia K Kuhl and Andrew N Meltzoff. 1996. Infant vocalizations in response to speech: Vocal imitation and developmental change. *The journal of the Acoustical Society of America*, 100(4):2425–2438.

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

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE.

Ann Lee, Hongyu Gong, Paul-Ambroise Duquenne, Holger Schwenk, Peng-Jen Chen, Changhan Wang, Sravya Popuri, Yossi Adi, Juan Pino, Jiatao Gu, and Wei-Ning Hsu. 2022. Textless speech-to-speech translation on real data. In *NAACl-HLT*.

Alexander H Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. 2022a. Towards end-to-end unsupervised speech recognition. *arXiv preprint arXiv:2204.02492*.

Alexander H. Liu, Cheng-I Lai, Wei-Ning Hsu, Michael Auli, Alexei Baevski, and James Glass. 2022b. Simple and Effective Unsupervised Speech Synthesis. In *Proc. Interspeech 2022*, pages 843–847.

Zhaoyu Liu and Brian Mak. 2019. Cross-lingual multi-speaker text-to-speech synthesis for voice cloning without using parallel corpus for unseen speakers. *arXiv preprint arXiv:1911.11601*.

John L Locke. 1994. Phases in the child's development of language. *American Scientist*, 82(5):436–445.

John L Locke. 1996. Why do infants begin to talk? language as an unintended consequence. *Journal of child language*, 23(2):251–268.

Hieu-Thi Luong and Junichi Yamagishi. 2019. A unified speaker adaptation method for speech synthesis using transcribed and untranscribed speech with backpropagation. *arXiv preprint arXiv:1906.07414*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

Tomáš Nekvinda and Ondřej Dušek. 2020. One model, many languages: Meta-learning for multilingual text-to-speech. *arXiv preprint arXiv:2008.00768*.

Junrui Ni, Liming Wang, Heting Gao, Kaizhi Qian, Yang Zhang, Shiyu Chang, and Mark Hasegawa-Johnson. 2022. Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition. *arXiv preprint arXiv:2203.15796*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Anusha Prakash, A Leela Thomas, S Umesh, and Hema A Murthy. 2019. Building multilingual end-to-end speech synthesisers for indian languages. In *Proc. of 10th ISCA Speech Synthesis Workshop (SSW'10)*, pages 194–199.

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

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 32.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Hubert Siuzdak, Piotr Dura, Pol van Rijn, and Nori Jacoby. 2022. WavThruVec: Latent speech representation as intermediate features for neural speech synthesis. In *Proc. Interspeech 2022*, pages 833–837.

Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. 2019. Token-Level Ensemble Distillation for Grapheme-to-Phoneme Conversion. In *Proc. Interspeech 2019*, pages 2115–2119.

Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. 2017. Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

Rafael Valle, Kevin Shih, Ryan Prenger, and Bryan Catanzaro. 2020. Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. *arXiv preprint arXiv:2005.05957*.

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. 2017. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *ACL*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.

Jilong Wu, Adam Polyak, Yaniv Taigman, Jason Fong, Prabhav Agrawal, and Qing He. 2022. Multilingual text-to-speech training using cross language voice conversion and self-supervised learning of speech representations. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8017–8021. IEEE.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812.

Yuzi Yan, Xu Tan, Bohan Li, Tao Qin, Sheng Zhao, Yuan Shen, and Tie-Yan Liu. 2021. Adaspeech 2: Adaptive text to speech with untranscribed data. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6613–6617. IEEE.

Haitong Zhang and Yue Lin. 2020. Unsupervised Learning for Sequence-to-Sequence Text-to-Speech for Low-Resource Languages. In *Proc. Interspeech 2020*, pages 3161–3165.

Yu Zhang, Ron J. Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, R.J. Skerry-Ryan, Ye Jia, Andrew Rosenberg, and Bhuvana Ramabhadran. 2019. Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning. In *Proc. Interspeech 2019*, pages 2080–2084.

Supplementary Material:
ParrotTTS: Text-to-speech synthesis exploiting disentangled self-supervised representations

Anonymous ARR submission

1 User Study

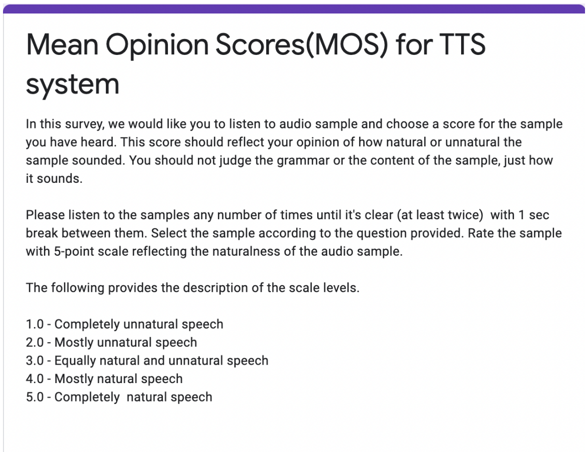
In this document, we present the supplementary material to support the submission titled 'ParrotTTS: Text-to-speech synthesis exploiting disentangled self-supervised representations' We present more details of the user study in section 5.

This section details the methodology followed in our user studies to evaluate the perceptual-quality/naturalness of TTS synthesized samples. The raters are fellow researchers who are professional English speakers. Their written consent to publish the survey results was obtained prior to rolling out the survey. We evaluate the ethical aspect of the survey with our peer group.

The following figures show the screenshot for the instructions given for the surveys: Figure 1 - perceptual-quality/naturalness. While rating MOS, the subjects are asked to listen to the sample at least twice and choose a score that reflects their opinion. They were also asked not to judge the grammar or the content of the sample but just how it sounds.

The following provides the discription of the scale levels.

- 1.0 - Completely unnatural speech
- 2.0 - Mostly unnatural speech
- 3.0 - Equally natural and unnatural speech
- 4.0 - Mostly natural speech
- 5.0 - Completely natural speech



Mean Opinion Scores(MOS) for TTS system

In this survey, we would like you to listen to audio sample and choose a score for the sample you have heard. This score should reflect your opinion of how natural or unnatural the sample sounded. You should not judge the grammar or the content of the sample, just how it sounds.

Please listen to the samples any number of times until it's clear (at least twice) with 1 sec break between them. Select the sample according to the question provided. Rate the sample with 5-point scale reflecting the naturalness of the audio sample.

The following provides the description of the scale levels.

1.0 - Completely unnatural speech
2.0 - Mostly unnatural speech
3.0 - Equally natural and unnatural speech
4.0 - Mostly natural speech
5.0 - Completely natural speech

Figure 1: Survey for MOS

2 Stabler training and faster inference

In Figure 2, we compare training profiles of Tacotron2 and AR-TTE keeping batch size the same. As visualized in Figure 2(a), the attention matrix in Tacotron2 takes about 20k iterations to stabilize with an anti-diagonal structure and predict a phoneme-aligned Mel sequence. AR-TTE, in contrast, is about ten times faster at predicting a discrete HuBERT unit sequence that aligns with input phonemes taking only about 2k iterations to arrive at a similar-looking attention plot. While the snapshots are illustrative, we use the guided-attention loss described by Tachibana et al. (2018) as a metric to quantify the evolution of the attention matrix through training steps. As shown in Figure 2(b), the loss dives down a lot sooner for ParrotTTS relative to its Tacotron2 counterpart. In a similar comparison, we observe that NAR-TTE converges (20k steps) about eight times faster than FastSpeech2 (160k steps).

We suppose that the faster convergence derives from the lower variance of discrete embeddings in ParrotTTS as opposed to the richness of Mels that are complete with all acoustic variations, including speaker identity, prosody, etc. The output speech is independent of inputs given the Mel-spectrogram unlike ParrotTTS embeddings that further need cues like speaker identity in later ETS module. We hypothesize that segregating content mapping away from learning acoustics like speaker identity helps improve training stability, convergence, and data efficiency for the TTE encoder.

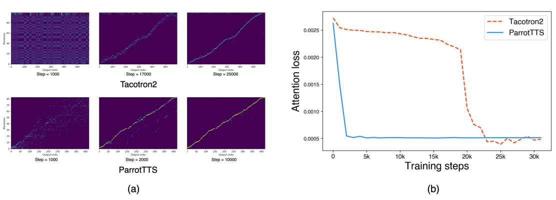


Figure 2: Visualization of attention between output units and phonemes. (a) Evolution of attention matrix with training steps. (b) Attention loss plotted against training steps.

The proposed NAR-TTE system also improves inference latency and memory footprint, which are crucial factors for real-world deployment. On NVIDIA RTX 2080 Ti GPU, we observe ParrotTTS serves 15% faster than FastSpeech2, reducing the average per utterance inference time to 11ms from 13 ms. Furthermore, the TTE module uses 17M parameters in contrast to 35M parameters of the Mel synthesizer module in Fastspeech2.

3 Choices of hyper-parameters

Our proposed ParrotTTS backbones are derived from existing models (Ren et al., 2020),(Hsu et al., 2021), (pol),and (Wang et al., 2017) as mentioned in the main text. Hence all hyper-parameters and optimization methods are same unless explicitly mentioned otherwise. We do not tune hyper-parameter for performance and train the model only once for the proposed design. For evaluation metrics, we use pretrained speaker verification and ASR networks.

References

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. 2018. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4784–4788. IEEE.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*.