ParrotTTS: Text-to-Speech synthesis by exploiting self-supervised representations

Anonymous ACL submission

Abstract

Text-to-speech (TTS) systems are modelled as mel-synthesizers followed by speech-vocoders since the era of statistical TTS that is carried forward into neural designs. We propose an alternative approach to TTS modelling referred to as ParrotTTS borrowing from self-supervised learning (SSL) methods. ParrotTTS takes a two-step approach by initially training a speech-to-speech model on unlabelled data that is abundantly available, followed by a text-to-embedding model that leverages speech with aligned transcriptions to extend it to TTS. ParrotTTS achieves competitive mean opinion scores on naturalness compared to traditional TTS models but significantly improves over the latter's data efficiency of transcribed pairs and speaker adaptation without transcriptions. This further paves the path to training TTS models on generically trained SSL speech models. Speech samples from ParrotTTS can be found at https://parrottts.github.io/tts/

1 Introduction

Vocal learning forms the first phase of infants starting to talk (Locke, 1996, 1994). In this phase, the learning happens by simply listening to sounds/speech. Studies show that vocal learning begins in the final trimester of pregnancy; the normally developing fetus can hear its mother's voice within the womb (Kolata, 1984). Several studies show that the best way to promote language development for babies is to talk to them. 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). Eventually, if parrots can talk without understanding language, there is no reason human infants should need to possess grammatical capability either to comprehend and produce speech (Locke, 1994).

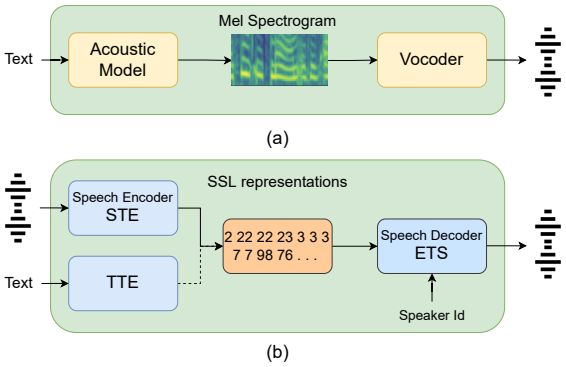


Figure 1: (a) Traditional TTS and (b) Proposed TTS model

We propose a novel design for text-to-speech synthesis called ParrotTTS that follows a similar learning process. Our idea mimics the-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. The second phase builds on top of the first by learning a mapping from text to the quantized representations (embeddings). This step uses paired text-speech data. The two phases are analogous to first *learning to talk* followed by *learning to read*.

Our proposed ParrotTTS is illustrated in Figure 1(b) distinguishing it from traditional design in Figure 1(a). The self-supervised module learns discrete speech representations using raw audio data from multiple speakers without aligned transcriptions similar to Wav2Vec 2.0 (Baevski et al., 2020) or Hubert (Hsu et al., 2021). The SSL module includes a speech-to-embedding (STE) encoder trained on masked prediction task to generate the intermediate representation of audio input. An embedding-to-speech (ETS) decoder is indepen-

dently 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. However, the data requirement in this step is very low in terms of volume and number of speakers. We show that transcribed samples from even a single speaker suffices to learn phonetic mapping (TTE) sufficiently well for generalization on a large number of speakers. Further, the decoder ETS can be conditioned on speaker identity to change the voice of rendered speech. In our model, the speech embeddings can be obtained either from the text (using TTE) or directly from audio (using STE), providing a unified model for speech synthesis, of which we limit the scope of this work to only text-to-speech.

Overall, the restructuring of learning components has effectively changed the data dependence equation in our favor, cutting down the amount of transcribed data needed by leveraging abundant raw audio to achieve similar speech quality. This further makes it easy to extend the model to *de novo* voices unseen in initial training by independently fine-tuning the ETS decoder module on untranscribed audio from the corresponding speakers. Also, the ParrotTTS' components are functionally different from that of traditional synthesizer-vocoder design. This offers several other benefits.

1. For instance, our speech embedding has lower variance than that of Mel frames reducing the complexity to train TTE and increasing capacity of downstream ETS. We observe that, for example, our embeddings are speaker agnostic, requiring ETS conditioning on speaker identity for speaker adaptation.

2. Speaker agnostic speech embeddings paired with independently trained STE disentangled speaker handling from content. This enabled adaptation to novel voices with untranscribed speech alone. The data requirement is placed between zero-shot methods that use speaker-embedding but are poor in quality and traditional TTS requiring fully transcribed speech while its quality matches the latter.

3. Segregation of functions pushed acoustic handling into ETS module towards the end that directly infers the speech signal without going through Mel frames. This bypasses potential vocoder generalization issues (Kim et al., 2021) similar to FastSpeech2s (Ren et al., 2020).

4. 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).

The main contribution of this work is the novel ParrotTTS architecture detailed in Section 3. It redesigns the standard synthesizer-vocoder neural TTS to leverage self-supervised learning from which the various benefits listed above flow. We train multiple models of the proposed ParrotTTS approach with different choices and study their effects like the quality of rendered speech, data efficiency, word-error rates upon transcription of speech output, etc., see Section 4. Experimental results reported in Section 5 consistently point to the competitive or superior performance of ParrotTTS relative to the current state-of-the-art for TTS. While these observations are of significant value to practitioners in evaluating the adoption of ParrotTTS approach for speech synthesis, numerous questions need further investigation. We conclude in Section 6 with a discussion of these questions and the related topics that need further exploration to better understand the proposed approach.

2 Related work

TTS systems have been studied for decades now, with the concatenative statistical models from earlier attempts (Hunt and Black, 1996; Cohn and Zellou, 2020) being increasingly replaced by neural variants in recent years (Oord et al., 2016). We specifically review the popular and better-performing supervised models in Section 2.1 and their unsupervised counterparts in Section 2.2. These references help understand data challenges for TTS training and how their quality is observed to vary with the degree of supervision. Towards the end of this section, we review the self-supervised learning approach that ParrotTTS leverages with pointers to its application in other domains.

2.1 Supervised TTS

A typical neural TTS model has an acoustic synthesizer that generates frequency-domain Mel-spectrogram frames. The synthesizer has an encoder that maps 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 models (Wang et al., 2017; Valle et al., 2020; Shen et al., 2018) that generate the Mel frames in sequential order. It conditions the generation of a Mel frame at any time instant on all preceding predictions and the encoder output using attention modules (Graves, 2013). In contrast, non-autoregressive or parallel models (Ren et al., 2019, 2020; Łańcucki, 2021) predict intermediate features like duration, pitch, and energy for each phoneme. Mel frames of all time instants are then generated simultaneously from these predicted intermediate features. Non-autoregressive models are quicker at inference and robust to word skipping or repetition errors (Ren et al., 2020).

The quality and quantity of transcribed audio used in TTS training are known to impact the quality of speech rendered. Public data with about 24 hours of studio recorded audio is known to train reasonable quality single-speaker models (Ito and Johnson, 2017). This becomes more demanding in a multi-speaker setting requiring sufficient per-speaker audio to learn all voices well (Veaux et al., 2017). Speaker conditioning of the decoder is commonly achieved by one-hot-encoding of those seen at train time. Alternatively, speaker embeddings (Jia et al., 2018) could be used for decoder conditioning which in theory could render speech for *de novo* voices not part of the training set. However, speech rendered through this method is known to be of poorer quality and naturalness, 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 the TTS acoustic modeling. They typically employ unsupervised automatic speech recognition (ASR) model (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). Furthermore, switching to a multi-speaker setup worsens quality relative to fully supervised models (Liu et al., 2022b).

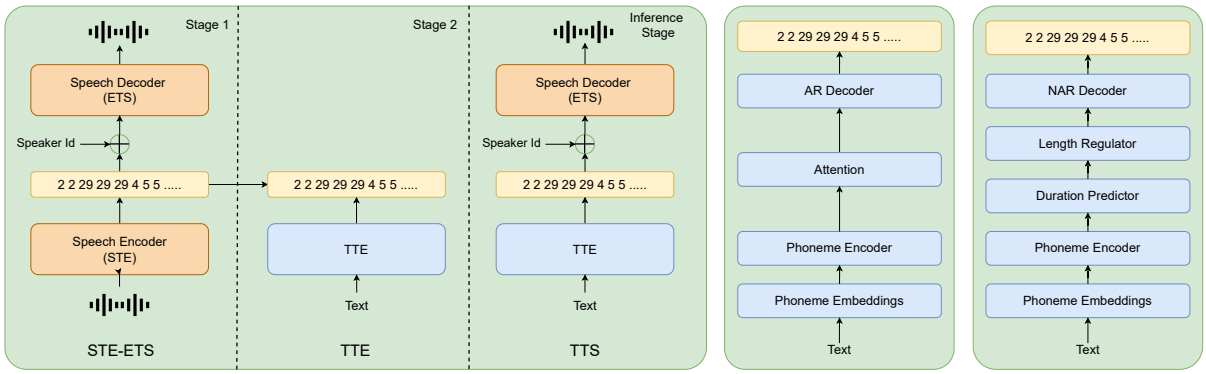
Some prior works have looked at adapting TTS to novel speakers using untranscribed audio (Yan et al., 2021; Luong and Yamagishi, 2019; Taigman et al., 2017). Unlike ours, these methods require a large amount of paired data from multiple speakers during initial training. Some of these (Luong and Yamagishi, 2019; Taigman et al., 2017) jointly train the TTS pipeline and the modules for speaker adaptation but the model convergence gets tricky. In contrast, ParrotTTS benefits from the disentanglement of linguistic content from speaker information, making adaptation easier.

2.3 Self-supervised learning

Self-supervised learning (SSL) methods have become increasingly popular in numerous applications owing to their ability to leverage copious amounts of unlabeled data to learn large models that can be fine-tuned for multiple tasks later. They are reported to achieve results better than supervised models trained on fewer labeled samples and have found applications in computer vision (He et al., 2022), natural language processing (Devlin et al., 2018; Vaswani et al., 2017) and audio processing (Schneider et al., 2019). Mask prediction, temporally contrastive learning, next-step prediction, etc., are some common techniques to train SSL models. Wav2vec2 (Baevski et al., 2020), Hubert (Hsu et al., 2021) are popular SSL models for speech processing and ASR (Baevski et al., 2020), phoneme segmentation (Kreuk et al., 2020), and spoken language modeling (Lakhotia et al., 2021), speech resynthesis (Polyak et al., 2021) are tasks that gained from leveraging them. In the same spirit, our work explores SSL, specifically pre-trained Hubert (Hsu et al., 2021), for TTS. To the best of our knowledge, there are no known TTS models trained on SSL, and our efforts fill this gap.

3 ParrotTTS architecture

As mentioned earlier, ParrotTTS has three modules; two encoders, STE and TTE that map audio and text respectively to embedding, and a decoder



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

Figure 2: Schematic diagram of the proposed model.

ETS that maps the embedding to the speech signal. Our speech encoder-decoder choices are borrowed from (Polyak et al., 2021). The speech encoder STE is HuBERT (Hsu et al., 2021) that maps input audio clip to discrete vectors with entries called HuBERT units. Our speech decoder ETS is a modified version of HiFiGan (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. We give architectural details of these three modules below.

3.1 Speech encoder STE

The self-supervised HuBERT model we use for our STE is pre-trained on large raw audio data on masked prediction task very similar to the BERT model for text (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 are derived by clustering MFCC features of short speech signals. Encoder input is 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,

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

Where $\widehat{T} = T/320$ indicating downsampling and each $h_i \in \{1, \ldots, K\}$ with K being a number of clusters in HuBERT's clustering step, set to 100 in our experiments.

3.2 Speech decoder ETS

We use a modified version of HiFiGAN (Kong et al., 2020) 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, $\widehat{X} \coloneqq G(\mathbf{h})$.

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 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 $\mathbf{h}_p \coloneqq E_p(P)$. 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 $\mathbf{h}_r \coloneqq E_r(X)$ 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. See Figure 2(b) for an illustra-

tion. 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 $\mathbf{h}_p$.

4 Experiments

We train multiple models of the ParrotTTS under different settings and benchmark them against comparable models in the literature. Specifically, we train single-speaker and multi-speaker models to evaluate naturalness, intelligibility, and speaker adaptability. Naturalness is measured by mean-opinion scores (MOS) from human judgments. Intelligibility is measured by word-error rates from an ASR model on the rendered speech output. Speaker adaptability is measured using Equal-Error-Rate from a pre-trained speaker verification system. We perform these experiments with both autoregressive and non-autoregressive choices of TTE.

4.1 ParrotTTS training

We use two public data sets for our experiments. LJSpeech (Ito and Johnson, 2017) provides about 13k high-quality English transcribed audio clips totaling about 24 hours 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. All audio samples are resampled to 16kHz before use.

STE training. We use 12 layer transformer model for HuBERT trained for two epochs on 960 hour-long LibriSpeech corpus (Panayotov et al., 2015) as our STE module to extract $\mathbf{h}_r$ embeddings. 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. The vectors are drawn from the network's activation units on the sixth layer similar to that of Lakhotia et al. (2021). Continuous vectors are then discretized to $\mathbf{h}_r$ embeddings using a codebook made from applying k-means (with k set to 100) to 100 hour subset of the data called LibriSpeech-clean (Panayotov et al., 2015).

TTE training. 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 our metrics. 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. Unlike standard TTS systems that predict Mel spectrograms, TTE generates discrete units. Hence, we replace the mean-square error loss used in Mels with cross-entropy with as many classes as clusters in the discretization codebook.

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.

4.2 Comparison to prior art

Single Speaker TTS. We compare against state-of-the-art TTS models from the literature of both kinds; Tacotron2 (Wang et al., 2017) from among autoregressive models and FastSpeech2 (Ren et al., 2020) from the non-autoregressive models. Both models are trained using the ground truth transcripts of LJspeech and referred to as SS-Tacotron2

	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
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 studied TTS models in the single speaker setting.

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 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. In the multi-speaker setting, we compare against a fully supervised Fastspeech2 baseline trained on VCTK with all its speakers using the entire paired audio-transcript data that we refer to as MS-FastSpeech2. 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 TTE similar to the FastSpeech2 baseline trained on LJSpeech alone and multi-speaker ETS trained on VCTK alone.

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 (Polyak et al., 2021) 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. In all multi-speaker experiments, we use one-hot encoding for speaker identity. 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.

4.3 Evaluation metrics

Naturalness is measured by mean opinion scores (MOS) from subjective listening tests on a five-point Likert scale, with 1 being "completely unnatural" speech to 5 indicating "completely natural" output. We randomly sample five clips per model from the validation set for each of our forty subjects who are proficient English speakers. They are asked to make quality judgments by rating the naturalness of the synthesized speech samples. The average rating of MOS is calculated and reported. Intelligibility is measured by the word error rate of ASR transcriptions of rendered speech. We use pre-trained Whisper *small* model (Radford et al., 2022) for this.

We validate the speaker adaptability by reporting Equal Error Rate (EER) from a pre-trained speaker verification network. Specifically, we use the verification model proposed in (Desplanques et al., 2020) trained on VoxCeleb2 (Chung et al., 2018) with a 0.8% EER on the test split of VoxCeleb1 (Chung et al., 2018).

5 Results

Quantitative and qualitative results evaluating the proposed ParrotTTS system are shown in Tables 1 and 2 for single-speaker and multi-speaker models, respectively.

Model	VCTK Transcripts	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
NAR-TTE$_{\text{LJS}}$+MS-ETS	No	3.78	6.53	4.38

Table 2: Comparison of the studied multi-speaker TTS models on the VCTK dataset. The second column suggests if the corresponding method uses the ground truth VCTK transcripts while training.

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 under 0.2pp[1]) among the autoregressive and non-autoregressive model classes.

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 very close to ground truth.

[1]Percentage points abbreviated as pp.

5.2 Multi-speaker TTS

Naturalness and intelligibility. Table 2 summarizes results from our multi-speaker experiments. Among all methods listed in it, NAR-TTE$_{\text{LJS}}$+MS-ETS clearly outperform all other models ranking only below re-synthesizing from ground truth Mels, GT-Mel+Vocoder. 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.

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 under 1pp relative to ParrotTTS while GT-Mel+Vocoder continues to dominate the scoreboard leaving room for further improvement.

5.3 Stabler training and faster inference

In Figure 3, we compare training profiles of Tacotron2 and AR-TTE keeping batch size the same. As visualized in Figure 3(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

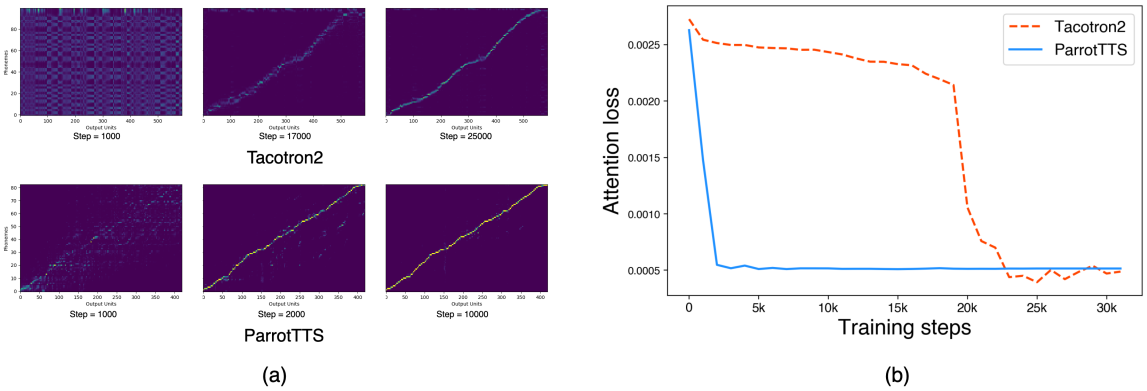


Figure 3: Visualization of attention between output units and phonemes. (a) Evolution of attention matrix with training steps. (b) Attention loss plotted against training steps.

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 3(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.

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.

6 Conclusion, limitations and future work

In this work, we proposed ParrotTTS, a fast, high quality, and efficient to train TTS system. The two-stage learning process of ParrotTTS is designed to leverage untranscribed speech data and the corresponding self-supervised embeddings. We show that even when trained using transcribed data of a single speaker from the LJSpeech dataset, ParrotTTS can synthesize speech in 108 different voices of the VCTK corpus. In terms of naturalness of speech, ParrotTTS outperforms the established prior art and alternative baselines by a noticeable margin in the multi-speaker setup. On single speaker benchmarks, ParrotTTS provides competitive performance compared to the prior art. Overall, our work paves the way for further explorations towards exploiting SSL in TTS models.

Our experiments are limited to a single language. A deeper study exploring multiple languages, effects of background noise, accents, and other demographic variations is left for future work. The current pre-trained HuBERT model skips prosody information (Kharitonov et al., 2021), so the model has no levers to control these prosodic variations. We want to study ways to bring prosodic controllability into ParrotTTS. Further, it would be essential to improve TTE training to use noisy samples that the current model does not work well with to leverage weak supervision to scale.

7 Ethical Considerations

Our research is founded on ethical considerations. We are excited about the potential of text-to-speech synthesis to push the frontier of technology, such as in accessibility (giving voice to the voiceless), human-computer interaction, telecommunications, and education. However, there is the potential for misuse. Notably, multi-speaker text-to-speech sys-

tems have raised concerns about unethical cloning. Our experiments limit to publicly available datasets, and our method is not intended for synthesizing someone's voice without their permission. Another potential misuse is creating an audio file of someone supposedly speaking words they never actually uttered. We are keenly aware of these negative consequences. While the benefits outweigh the concerns at this point, we firmly believe that the research community should proactively continue to identify methods for detecting and preventing misuse.

Our approach is trained on western speech data and has yet to be validated on different languages or people with speech impediments. As such, the dataset and results are not representative of the population. A deeper understanding of this issue requires future studies in tandem 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.

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

Michelle Cohn and Georgia Zellou. 2020. Perception of concatenative vs. neural text-to-speech (tts): Differences in intelligibility in noise and language attitudes. In *Proceedings of Interspeech*.

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*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009.

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.

Andrew J Hunt and Alan W Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

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.

Gina Kolata. 1984. Studying learning in the womb: behavioral scientists are using established experimental methods to show that fetuses can and do learn. *Science*, 225(4659):302–303.

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. *Interspeech*.

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.

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 Jeff Lai, Wei-Ning Hsu, Michael Auli, Alexei Baevskiv, and James Glass. 2022b. Simple and effective unsupervised speech synthesis. *Interspeech*.

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.

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.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. *Interspeech*.

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

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.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *Interspeech*.

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.

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.

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.

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*.

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.

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*.

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.