MIXER-TTS: NON-AUTOREGRESSIVE, FAST AND COMPACT TEXT-TO-SPEECH MODEL CONDITIONED ON LANGUAGE MODEL EMBEDDINGS

Oktai Tatanov, Stanislav Beliaev, Boris Ginsburg

NVIDIA, Santa Clara

ABSTRACT

This paper describes Mixer-TTS, a non-autoregressive model for mel-spectrogram generation. The model is based on the MLP-Mixer architecture adapted for speech synthesis. The basic Mixer-TTS contains pitch and duration predictors, with the latter being trained with an unsupervised TTS alignment framework. Alongside the basic model, we propose the extended version which additionally uses token embeddings from a pre-trained language model. Basic Mixer-TTS and its extended version achieve a mean opinion score (MOS) of 4.05 and 4.11, respectively, compared to a MOS of 4.27 of original LJSpeech samples. Both versions have a small number of parameters and enable much faster speech synthesis compared to the models with similar quality.

Index Terms— speech synthesis, mel-spectrogram generation, MLP-Mixer

1. INTRODUCTION

Recent neural text-to-speech (TTS) models have significantly improved the speed, robustness, and quality of generated speech. The improvement in training and inference speed is mostly related to switching from sequential, autoregressive models [1, 2, 3]) to parallel, non-autoregressive models [4, 5, 6, 7]. Non-autoregressive models can generate speech two orders of magnitude faster than auto-regressive models with similar quality. For example, FastPitch generates mel-spectrograms 60x faster than Tacotron 2 [6].

The robustness of TTS models was significantly improved by using an explicit duration predictor [8, 9, 4, 5, 6, 7] that practically eliminates skipping and repeating words, which were common issues in popular models like Tacotron 2. Traditionally, models with duration predictors have been trained in a supervised manner with external ground truth alignments. For example, TalkNet used the alignment from auxiliary ASR models, while FastSpeech and FastPitch used alignments from a teacher TTS model. Glow-TTS [10] proposed a flow-based algorithm for unsupervised alignment training. This algorithm has been improved in RAD-TTS [11] and modified for non-autoregressive models in [12]. This new alignment framework greatly simplifies TTS training pipeline.

The quality of speech generated by the first non-autoregressive models was inferior to state-of-the-art autoregressive models. FastPitch [6], a non-autoregressive model, closed the gap in quality by a adding pitch predictor for fundamental frequency (F0). Hayashi et al [13] proposed to augment TTS model with input representation from a pre-trained BERT [14] language model (LM). The authors hypothesized that text embeddings contain information about the importance of each word, which helped to improve speech prosody and pronunciation. The usage of semantic context for TTS was extended in [15].

In this paper, we present Mixer-TTS, a non-autoregressive model for text to mel-spectrogram synthesis. The model backbone is based on the MLP-Mixer [16] architecture from computer vision adapted for speech. The new backbone makes the model significantly smaller and faster than Transformer-based TTS models [5, 6]. Our model uses an explicit duration predictor, which is trained by the unsupervised alignment framework proposed in [12]. Mixer-TTS combines two methods to improve the prosody of generated speech. The basic version has an explicit pitch predictor similar to FastPitch [6]. The extended version adds token embeddings from an external pre-trained LM to improve speech prosody and pronunciation. Using token embeddings is significantly less expensive than inferring BERT outputs as in [13]. They notably improve speech quality with a very modest increase in the model size and inference speed.

We evaluate the quality of the proposed models combined with a HiFi-GAN [17] vocoder on LJSpeech [18]. Mixer-TTS achieves a mean opinion score (MOS) of 4.05 compared to a MOS of 4.27 for the original speech samples. The extended model with LM embeddings improves MOS to 4.11. The basic version has 19.2M parameters, and the extended model has 24M. Mixer-TTS samples are published online[1].

2. MODEL ARCHITECTURE

The model architecture is shown in Figure 1. We encode the text and align it by using audio features in a separate module to get "ground truth" durations. Then, we calculate character or phoneme-level pitch values and feed them all into the

[1] https://mixer-tts.github.io/

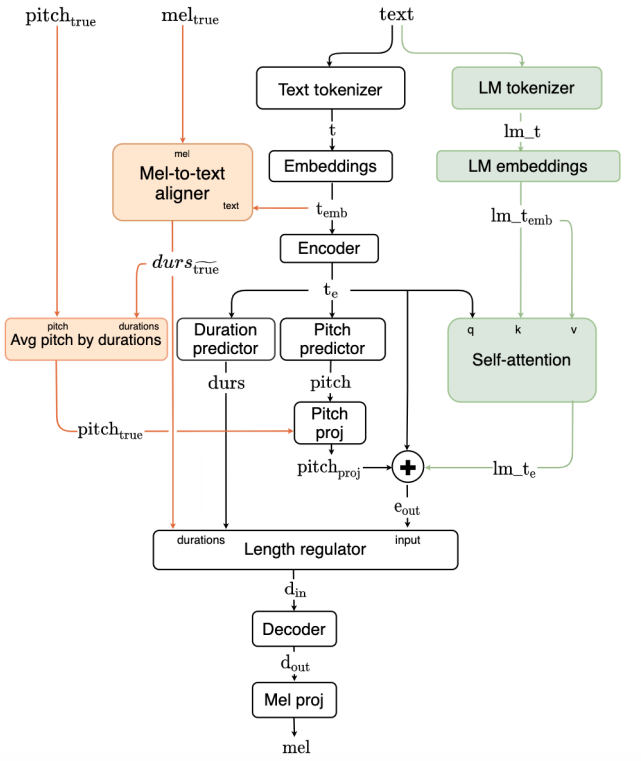

Fig. 1. Training and inference pipeline of Mixer-TTS. During training, the decoder uses durations from mel-to-text aligner and ground truth pitch. During inference, durations and pitch are obtained from predictors. Extended Mixer-TTS contains additional blocks for LM embeddings.

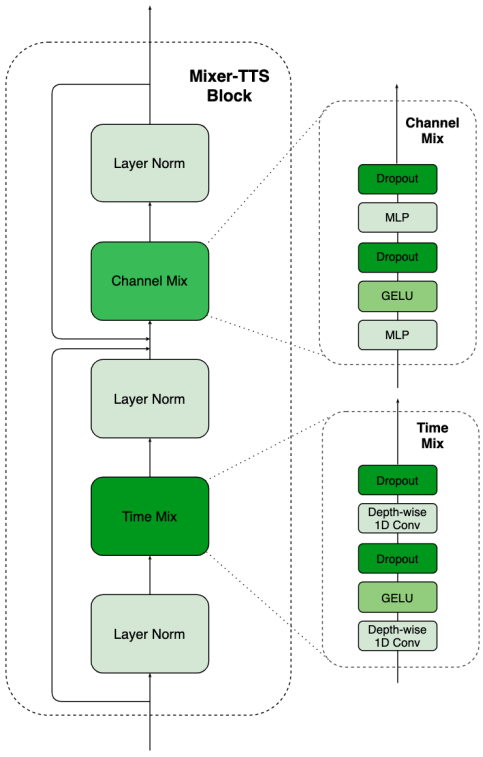

Fig. 2. Mixer-TTS block consists of time and channel mix blocks. Channel-mix block includes two MLPs and a GELU. Time-mix block is similar but with depth-wise 1D convolutions instead of MLPs.

length regulator module to expand each character or phoneme feature along with their corresponding durations. Next, the decoder generates mel-spectrogram from the encoded representations.

The basic Mixer-TTS is structurally similar to FastPitch with two major changes. First, we replaced all feed-forward Transformer-based blocks in the encoder and decoder with new Mixer-TTS blocks (see subsection 2.1). Second, we used an unsupervised speech-to-text alignment framework to train the duration predictor (see subsection 2.2). The extended Mixer-TTS additionally includes conditioning on embeddings from pretrained LM (see subsection 2.3). We use the duration and pitch predictor architectures described in FastPitch.

The model is trained with loss function combined from aligner loss and mean-squared errors between ground-truth and predicted values for mel-spectrogram, duration and pitch:

$$L = L_{aligner} + L_{mel} + 0.1 \cdot L_{durs} + 0.1 \cdot L_{pitch}$$

2.1. Mixer-TTS Block

The MLP-Mixer architecture, introduced in [16] for computer vision, is based exclusively on multi-layer perceptrons (MLPs). MLP-Mixer performs two key actions over input: "mixing" the per-location features and "mixing" spatial information. Both operations are implemented by the stack of two MLPs layers. The first MLP layer increases number of channels by an "expansion factor", and the second MLP layer reduces channels to the original value. However, such an approach is only possible when the input size for a layer is fixed by every dimension. To use this architecture for text-to-speech (i.e. case where one of the input's dimensions has dynamic size), we use "time-mixing", replacing MLPs with depth-wise 1D convolutions[2] and borrowed the original layer for channel "mixing". The rest of the structure of the original MLP-Mixer remains unchanged, including layer normalization and residual connections (see Figure 2). During mini-batch training and inference, when sequences in a batch are padded to match the longest sequence, we use sequence masking after MLP and depth-wise 1D convolution layers.

The encoder is composed of six stacked Mixer-TTS

[2]In this case, model consists not only of MLPs, so we decided to call our block as *Mixer-TTS*, because the idea of "mixing" still remained.

blocks with convolution kernel in time-mix growing linearly from 11 to 21 with step 2. The decoder is composed of nine stacked Mixer-TTS blocks with kernel sizes growing from 15 to 31 in the same manner. Feature dimension is constant 384 for all blocks used, channel-mix expansion factor is 4 and there is no expansion factor in time mix. We used a dropout of 0.15 in each block.

2.2. Speech-to-text alignment framework

Most non-autoregressive TTS models with duration prediction rely on durations extracted from external sources. However, in our work, we train the speech-to-text alignments jointly with the decoder by using adaptation of unsupervised alignment algorithm [12] which was proposed in implementation of FastPitch 1.1[3]. This aligner encodes text and mel-spectrogram using 1D convolutions and projects them to a space with the same dimensionality. The "soft" alignment is computed using a pair-wise L_2 distance of the encoded text and mel representations, then Connectionist Temporal Classification (CTC) loss is used to learn these alignments. To obtain monotonic binarized alignments (i.e "ground-truth" durations), the Viterbi algorithm is used to find the most likely monotonic path. More details about the alignment framework can be found in [12].

2.3. Extended Mixer-TTS

The extended model takes advantage of token embeddings from external LM. We used ALBERT [19] model from HuggingFace [20] pretrained on large corpus of English text. We kept the LM tokenization method and utilized frozen embeddings for input tokens. The lengths of original and tokenized text from external LM are different because they are produced by different tokenizers. To align two sequences, we use a single head self-attention block applied to LM embeddings lm_{emb} and encoder output t_e, which mixes their features while preserving the lengths of "basic" text embeddings. Text features for self-attention aligning are extracted with convolutional layers preceded by separate positional embedding layers.

3. RESULTS

3.1. Training details

The model was trained on the LJSpeech dataset which was split into three sets: $12,500$ samples for training, 100 samples for validation, and 500 samples for testing. The text was lower-cased while leaving all punctuation intact. We experimented with two tokenization approaches: character-based and phoneme-based. For grapheme-to-phoneme conversion we used the ARPABET representation in the *CMUdict*[4] vocabulary and left ambiguous words and heteronyms in character representation. We converted ground truth 22050Hz sampling rate audios to mel-spectrograms using a Short-Time Fourier Transform (STFT) with 50 ms Hann window and 12.5 ms frame hop. Ground truth pitch was extracted using the *librosa* library [21] with values-aligned along mel-spectrogram frames.

The model was trained for 1000 epochs using the LAMB optimizer [22] with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-8}$, a weight decay of 10^{-6} and gradient clipping of 1000.0. A Noam annealing learning rate policy was used with a learning rate of 0.1 and a 1000 steps warmup. We used a total batch of 128 for four GPUs with gradients accumulation of 2. The training takes around 12 hours on four V100 GPUs in mixed precision mode [23].

LM embeddings	Tokenizer	MOS
✓	chars	$\mathbf{4.11 \pm 0.06}$
✓	phonemes	4.06 ± 0.06
✗	phonemes	4.06 ± 0.06
✗	chars	4.03 ± 0.06

Table 1. **Mixer-TTS Ablation Studies** with 95% confidence interval. Four options of Mixer-TTS were used: with or without LM embeddings and with char-based or phoneme-based tokenizers.

3.2. Speech quality evaluation

We have conducted several mean opinion score (MOS) studies for generated speech quality comparison using Amazon Mechanical Turk[5]. For evaluation, we selected Mturk workers with top performance ($\geq$ 95% HITS Approval, $\geq$ 5000 HITS Total), from the US only and with minimum high school degree. We tested 50 audio samples per model with 15 people per sample. The scores ranged from 1.0 to 5.0 with a step of 0.5.

First, we compared Mixer-TTS with different tokenization approaches in combination with LM embeddings conditioning. The results are in Table 1. Phonetic input representation slightly outperforms characters for the basic model. But for the extended model, the combination of character-based tokenization with LM embeddings performs better.

As the main study, we compare Mixer-TTS with the following popular and relevant models: Tacotron 2, FastPitch, and TalkNet 2. The last two models were trained with the same aligner mechanism instead of the usual external set of durations to match the approach presented in Mixer-TTS. According to the results, the basic version of Mixer-TTS with phonemes achieves a comparable to FastPitch level of speech

[3] https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/SpeechSynthesis/FastPitch

[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[5] https://www.mturk.com

Model	MOS
Ground truth audios	4.27 ± 0.05
Ground truth mels	4.18 ± 0.07
FastPitch	4.05 ± 0.06
Tacotron 2	3.95 ± 0.06
TalkNet 2	3.95 ± 0.07
Mixer-TTS-X	$\mathbf{4.11 \pm 0.06}$
Mixer-TTS	$\mathbf{4.06 \pm 0.06}$

Table 2. **Mean Opinion Scores (MOS)** with 95% confidence interval. We used the HiFi-GAN vocoder which was trained on ground-truth mel-spectrograms and additionally fine-tuned for 100k steps on outputs from every model. TalkNet 2 and FastPitch were trained with an unsupervised aligner framework to match the Mixer-TTS approach.

quality, and the extended version (Mixer-TTS-X) exceeds the quality of all the examined models (see Table 2).

3.3. Inference performance

We compared Mixer-TTS inference with FastPitch as the fastest examined model. The measurement was done with a variable-length text generator based on snippets from the LJSpeech test set and batch size of one. Evaluation was done using AMP (Automatic Mixed Precision) in PyTorch 1.11 with CUDA 11.3, cuDNN 8.2 and NVIDIA's A100 GPU. We measured the wall-time of mel-spectrogram generation starting from the raw text processing step and averaged results over 10 consecutive runs with a warmup for cuDNN to adjust algorithms to input sizes.

Mixer-TTS inference is notably faster than FastPitch and it scales better with increasing the input length (see Figure 3). Furthermore, the best version of our model has only 24 million parameters while FastPitch has 45 million parameters.

4. CONCLUSION

We present Mixer-TTS, a non-autoregressive model for speech synthesis. Both the encoder and decoder of the proposed model are based on the MLP-Mixer architecture, adapted to work with variable size input. It has pitch conditioning and a duration predictor which is trained with an unsupervised TTS alignment framework. Together with the basic model, we propose the extended version which additionally utilizes token embeddings from a pretrained LM.

The quality of the generated speech is on par with the current state-of-the-art TTS models. The basic Mixer-TTS with HiFi-GAN vocoder achieves a MOS of 4.05, while the extended Mixer-TTS-X reaches a MOS of 4.11 (the ground truth speech has a MOS of 4.27). Thanks to the new design, the proposed model is fast in training and inference, which

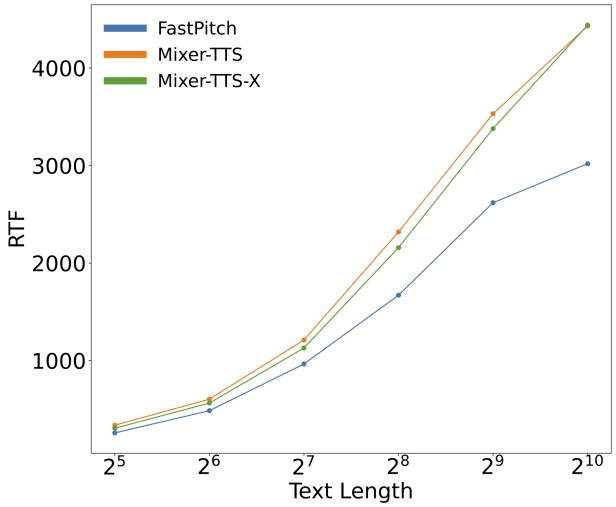

Fig. 3. **Real-time factor (RTF) for mel-spectrogram generation.** We used a batch size equal to one with mixed precision on A100 GPU.

makes it an attractive candidate for speech synthesis on low-resource devices.

The model will be released as part of NeMo toolkit [24].

5. ACKNOWLEDGMENTS

The authors thank Jocelyn Huang, Jason Li, Sang-Gil Lee, Rohan Badlani, Rafael Valle and the NVIDIA AI Applications team for the helpful feedback and review.

6. REFERENCES

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017.

[2] J. Shen, R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning Wavenet on mel spectrogram predictions," in *ICASSP*, 2018.

[3] W. Ping, K. Peng, A. Gibiansky, S. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: 2000-speaker neural text-to-speech," in *ICLR*, 2018.

[4] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *NeurIPS*, 2019.

[5] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," *arXiv:2006.04558*, 2020.

[6] A. Lańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," in *ICASSP*, 2021.

[7] S. Beliaev and B. Ginsburg, "Talknet: Non-autoregressive depth-wise separable convolutional model for speech synthesis with explicit pitch and duration prediction," in *INTERSPEECH*, 2021.

[8] S. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep Voice: Real-time neural text-to-speech," *arXiv:1702.07825*, 2017.

[9] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-speaker neural text-to-speech," in *NIPS*, 2017.

[10] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *NeuRIPS*, 2020.

[11] K. J. Shih, R. Valle, R. Badlani, A. Lancucki, W. Ping, and B. Catanzaro, "RAD-TTS: Parallel flow-based TTS with robust alignment learning and diverse synthesis," in *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

[12] R. Badlani, A. Lancucki, K. Shih, R. Valle, W. Ping, and B. Catanzaro, "One TTS alignment to rule them all," *arXiv:2108.10447*, 2021.

[13] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pre-Trained Text Embeddings for Enhanced Text-to-Speech Synthesis," in *INTERSPEECH*, 2019.

[14] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[15] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving prosody modelling with cross-utterance bert embeddings for end-to-end speech synthesis," in *ICASSP*, 2021.

[16] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic, et al., "MLP-Mixer: An all-MLP architecture for vision," *arXiv:2105.01601*, 2021.

[17] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *arXiv:2010.05646*, 2020.

[18] K. Ito and L. Johnson, "The LJ speech dataset," `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2020.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *EMNLP*, 2020.

[21] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *14th Python in Science conference*, 2015.

[22] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," *arXiv:1904.00962*, 2019.

[23] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, Ol. Kuchaev, G. Venkatesh, et al., "Mixed precision training," *arXiv:1710.03740*, 2017.

[24] O. Kuchaiev, J. Li, H.n Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, et al., "Nemo: a toolkit for building ai applications using neural modules," *arXiv:1909.09577*, 2019.