STArK: Towards Synthesizing Articulatory Kinematics from Text

Xavier Yin **, *Carlos Busso*

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA

xoy@cs.cmu.edu, busso@cmu.edu

Abstract

Recent deep learning approaches in modeling articulatory kinematics have shown success in mapping speech audio to *electromagnetic articulography* (EMA) kinematics, but these methods are limited by the scarcity of high-quality articulatory datasets. We introduce *synthesizing text to articulatory kinematics* (STArK), which offers a novel approach in generating articulatory kinematics directly from text. We train STArK on the LibriTTS-R dataset with pseudo-labeled EMA features derived from a pre-trained speech-to-articulation model. STArK is the first model to demonstrate high-quality articulatory kinematics generation from text. STArK achieves comparable performance to existing multi-speaker text-to-speech models at speech synthesis, while supporting voice cloning without seeing speaker embeddings during training. STArK aims to enable the generation of scalable articulatory data from text corpora for downstream tasks.

Index Terms: text-to-articulatory synthesis, electromagnetic articulography, articulatory representation, speech synthesis, text-to-speech, voice cloning

1. Introduction

Humans produce speech by manipulating the articulators in the vocal tract. Articulation is the key to understanding human speech production. It is used as the foundation of the source-filter model [1], guides the understanding of articulatory phonetics [1–3], and has been demonstrated in sensorimotor cortex studies on speech production [4, 5]. Despite the importance of articulatory models and data in understanding human speech, limited research on the usage of articulation-based features in speech synthesis models has been conducted until recently [6]. Ultimately, articulation is still an underexplored feature space in speech processing.

Previous studies have demonstrated high-quality speech synthesis from articulatory features obtained by *electromagnetic articulography* (EMA) recordings [6]. Others have utilized articulatory information to condition *text-to-speech* (TTS) generation, showing improved speech synthesis quality [7]. Recently, Cho et al. [6] demonstrated the capability to predict accurate EMA trajectories from raw speech audio, significantly increasing the possible use-cases of articulatory data in the speech processing domain. Nevertheless, the aforementioned techniques are fundamentally limited by the lack of available high-quality articulatory data to use for training.

Although other studies have demonstrated the synthesis of speech from articulatory representations, the synthesis of articulatory representations directly from text remains an open research question. Such a capability would greatly expand the feasibility of articulation-based speech models. Simultaneously, this would provide much-needed interpretability compared to high-dimensional speech embeddings or mel-spectrogram-based features. Articulatory synthesis has downstream applications in phonetics research, cross-linguistic analysis, disordered speech enhancement, and speech avatar simulation.

We introduce *synthesizing text to articulatory kinematics* (STArK), a novel approach that generates high-fidelity articulation-based features directly from text. STArK addresses the capability gap of previous spoken articulatory synthesis methods, as well as the need for scalable articulatory data generation. In this work, we demonstrate the following:

1. STArK generates high-quality articulatory kinematics in a normalized articulation feature space and synthesizes intelligible speech;
2. STArK accurately models the articulatory processes in the human vocal tract; and
3. STArK does not require speaker embeddings during training to achieve multi-speaker speech synthesis and voice cloning.

We believe that STArK can be used to significantly broaden the range of datasets available to researchers using articulatory kinematics data, and enable research towards understanding human speech biomechanics, developing assistive technologies, and creating interpretable speech models. Code for reproducing STArK and demo samples are available at `https://github.com/Lab-MSP/STArK/`.

2. Related Work

Using articulation to assist TTS models has been attemped in previous studies. Ling et al. [8] introduce the idea of using articulatory features for speech synthesis using *hidden Markov models* (HMMs). Since then, Csapó et al. [9] have combined *deep neural network* (DNN) architectures with text and tongue image prompts to generate speech. More recently, Wang et al. [7] conditioned mel-spectrogram generation on articulatory features to improve voice timbre and speaking prosody in the multi-speaker setting. However, these studies generally assume that articulatory information is available for conditioning generation rather than synthesizing articulatory information from input text.

STArK uses the articulatory coding framework introduced in SPARC [6]. SPARC is designed to invert speech audio to 15 vocal tract parameters (12 EMA features, pitch, periodicity, and loudness) that represent the biomechanical implementation of phonetic contents. SPARC then resynthesizes these articulatory features back to the corresponding audio waveform. SPARC predicts 6 x–y coordinate pairs for its EMA features: Upper Lip (ULX/ULY), Lower Lip (LLX/LLY), Lower

** indicates the corresponding author.

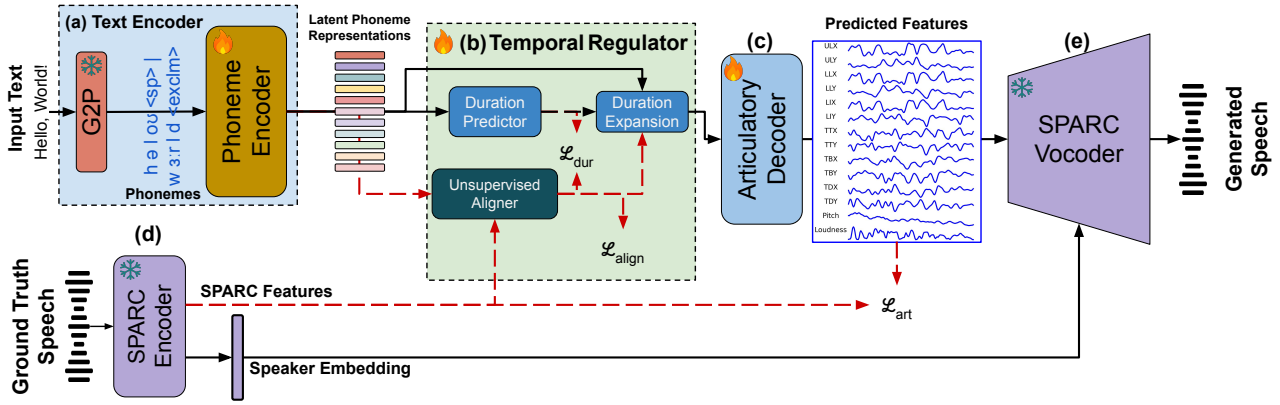


Figure 1: *STArK model diagram showing the pipeline from text graphemes to articulatory kinematics and speech output. Red dotted arrows represent operations performed during training only. Black dotted arrow represents operations performed during inference only. During training, Duration Expansion uses the output of the Unsupervised Alignment Module rather than the Duration Predictor.*

Incisor (LIX/LIY), Tongue Tip (TTX/TTY), Tongue Blade (TBX/TBY), and Tongue Dorsum (TDX/TDY). SPARC utilizes a template articulatory space, disentangled from the voice timbre, which is separately modeled by a speaker encoder. Cho et al. [6] demonstrated that EMA feature spaces between different speakers are separated by an affine transform, allowing for generalizable multispeaker articulatory synthesis. Given that SPARC is grounded in vocal tract physiology, disentangled from speaker identity, and low-dimensional, SPARC features are an effective representation for modeling human speech from text.

3. Model Architecture

STArK follows a *non-autoregressive* (NAR) *text-to-articulation* (TTA) pipeline that is inspired by similar NAR TTS models such as FastSpeech 2 [10] or FastPitch [11]. These models generate mel-spectrograms, whereas STArK uses this architecture to predict articulatory features: 12 EMA features, loudness, and pitch. Note that we do not predict SPARC's periodicity feature, as this is not used by the SPARC vocoder, but only used by SPARC's speaker encoder [6].

Fig. 1 shows the STArK architecture. Following DelighfulTTS [12] and FastSpeech 2 Conformer [13], we opted for *feed-forward Conformer* (FFConformer) blocks [14] in the Text Encoder (Fig. 1(a)) and for the Articulatory Decoder (Fig. 1(c)), as it has shown slight performance gains over Transformer-based variants on the TTS task [13]. Between the Encoder and the Decoder, we retain the Temporal Regulator module as described in previous NAR TTS studies [10–12, 15, 16].

3.1. Text Encoder

The Text Encoder (Fig. 1(a)) consists of two components: (1) an off-the-shelf *grapheme-to-phoneme* (G2P) module and (2) an FFConformer Phoneme Encoder module.

The G2P module uses Phonemizer[1] implemented with the eSpeak NG[2] backend with the US English model. The phonemes are first converted to phoneme IDs, and then into vector embeddings. These embeddings are fed into the position-wise FFConformer that uses *rotary positional embeddings* (RoPE) [17] to encode positional information. RoPE has shown improved performance over other positional embedding techniques such as *relative positional embeddings* (RPE) [18].

[1] https://github.com/bootphon/phonemizer
[2] https://github.com/espeak-ng/espeak-ng

STArK retains the reversed order of the depth-wise convolution and multi-head self-attention layers as introduced by Liu et al. [12]. Inspired by Luo et al. [16], the convolutional layers are replaced with *separable convolution* (SepConv) layers. The final outputs of the Phoneme Encoder are the FFConformer phoneme representations for each phoneme input, as shown in Fig. 1.

3.2. Feature Extraction and Speech Synthesis

We use the SPARC [6] (`en+`) model to extract articulatory feature targets for each training speech sample: 12 EMA features, pitch, and loudness. We freeze the SPARC encoder and use it to generate speaker embeddings for voice cloning purposes (Fig. 1(d)). Similarly, we freeze and use the SPARC vocoder (Fig. 1(e)), which is a HiFi-GAN [19] modified to use articulatory features and a SPARC speaker embedding as input, to synthesize speech from the predicted articulatory kinematics.

To improve training convergence and increase the speaker independence of the predicted articulatory features generated by STArK, the SPARC pitch value is logarithmically normalized by the median pitch of the utterance (Eq. 1). The predicted pitch that is output by STArK is then denormalized according to the median pitch of the target speaker, facilitating voice cloning.

$$\hat{f}_0[i] = \frac{\log f_0[i]}{\log\left(\text{median}\left\{f_0[k] | k \in [0, T]\right\}\right)} \tag{1}$$

3.3. Temporal Regulator

An important concept in many NAR TTS systems is temporal regulation. The Temporal Regulator (Fig. 1(b)) in STArK operates in two stages: (1) generate durations for each phoneme representation from the Phoneme Encoder, and (2) replicate each of the phoneme representations by their associated duration.

Stage (1) involves an Unsupervised Aligner and a Duration Predictor (Fig. 1(b)). Previous studies have used external aligners such as the Montreal Forced Aligner [20]. In the Unsupervised Aligner, STArK employs the One TTS Alignment method [21] to extract and learn ground-truth phoneme durations without external alignment dependencies during the training process. We adapt this approach to use pre-processed extracted SPARC features rather than mel-spectrogram frames. Furthermore, we use the Phoneme Encoder output rather than the phoneme embeddings so that the Unsupervised Aligner and Duration Predictor receive the same phoneme representations as input.

Following the monotonic alignment search algorithm [22], the "soft" probabilistic alignment map output by the Unsupervised Aligner is converted into "hard" alignment by using the Viterbi algorithm [23], extracting the most probable monotonic alignment path between phonemes and SPARC frames. This hard alignment is then used to extract ground-truth phoneme durations.

The Duration Predictor follows FastSpeech-based models [10–12, 15, 16] with three stacked SepConv layers, substituting the convolutional layers with SepConv. It receives the Phoneme Encoder's output and outputs a single duration value that represents the number of audio frames each phoneme is pronounced for. For numerical stability, logarithmic durations are predicted and then exponentiated to get the predicted durations.

Stage (2), "Duration Expansion," involves taking the predicted durations and repeating the Phoneme Encoder representations for each phoneme the corresponding number of times. During training, we utilize the Unsupervised Aligner durations for this process, whereas during inference, we use the Duration Predictor's output.

3.4. Articulation Decoder and Loss Function

The Articulation Decoder (Fig. 1(c)) follows the same FFConformer architecture as the Phoneme Encoder. It receives the duration-regulated representations from the Temporal Regulator and predicts a position-wise articulatory feature representation for each. These predicted articulatory features are then fed into the frozen SPARC vocoder to synthesize speech.

During training, the overall loss is calculated as follows. We used the *mean squared error* (MSE) between the 14 generated articulatory features and the extracted (pitch-normalized) SPARC features to compute the articulation loss ($\mathcal{L}_{\text{art}}$). We use the alignment loss ($\mathcal{L}_{\text{align}}$) for NAR TTS models as described in One TTS Alignment [21], and the MSE loss between the Duration Predictor's log-durations and the Unsupervised Aligner's log-durations ($\mathcal{L}_{\text{dur}}$). Therefore, the overall loss function to optimize can be represented as:

$$\mathcal{L} = \mathcal{L}_{\text{art}} + \alpha\mathcal{L}_{\text{align}} + \beta\mathcal{L}_{\text{dur}}, \tag{2}$$

where α and β represent loss weights that are chosen hyperparameters.

4. Experimental Methodology

4.1. Dataset and Preprocessing

To demonstrate the performance of the STArK architecture, we train and run inference on LibriTTS-R [24], a commonly used multi-speaker English-language TTS dataset. We used the original train-clean-100, dev-clean, `test-clean` splits of LibriTTS-R. We used SPARC to extract articulatory features for all audio samples, as well as per-utterance median pitch and speaker embeddings.

4.2. Training Details

STArK was trained on a single node with 4 L40S GPUs with 48GB of memory each. The base model has 73.7M parameters and was trained for 32,000 steps which took approximately 96 hours. We used the AdamW [25] optimizer with an initial learning rate of 2×10^{-4} and a cosine-decay scheduler with 10,000 warm-up steps.

4.3. Evaluation Metrics

For articulatory metrics, we compare our results against SPARC (`en+`). We reconstruct articulatory feature sequences from text using STArK with the following setups: Setup (1) predicted durations, Setup (2) providing aligner-generated durations, and Setup (3) providing aligner-generated durations along with original pitch and loudness features from SPARC. The aim of using Setups (2) and (3) is to evaluate text-to-articulation performance without bias towards prosodic accuracy.

For speech metrics, we compare our synthesized speech against YourTTS [26], which has zero-shot voice cloning capability to demonstrate that STArK inherently has similar capability by using the SPARC vocoder. For all subsequent speech metrics, the audio is resampled to 16 kHz.

For speech quality, we use automatic *mean opinion score* (MOS) metrics (UTMOSv2 [27] and DNSMOS [28, 29]), and for intelligibility we use the *word error rate* (WER) between the ground truth transcriptions and Whisper [30] (`whisper-large-v3-turbo`) transcriptions. We report MOS values for each evaluated model as a mean value with a 95% confidence interval over the LibriTTS-R `test-clean` subset. We calculate WER as a single value over the LibriTTS-R `test-clean` subset.

In addition, we report the *naturalness* MOS (NMOS) collected from evaluations completed by a total of 10 participants proficient in English. No *personal identifiable information* (PII) was collected. Participants were asked to rate each audio sample independently on a scale of 1-5, with 1 being the least natural and 5 the most natural. We report NMOS as a mean value with 95% confidence interval for each model over 24 randomly selected samples from the LibriTTS-R `test-clean` subset. The samples were shown to the participants in a random order with no two utterances having the same lexical content. Due to the small sample size of 40 ratings for each of the six evaluated models, the 95% confidence interval is calculated using a bootstrap distribution over 10,000 samples of the 10 participants.

We also measure speaker similarity to ground truth audio in order to quantify speaker similarity degradation as a result of multi-speaker generalization. We use Pyannote's [31, 32] embedding model[3] to extract speaker embeddings. Then, we report the *speaker embedding cosine similarity* (SECS) between the ground truth audio synthesized audio with a 95% confidence interval over the entire LibriTTS-R `test-clean` subset.

Although Cho et al. [6] used the *Pearson correlation coefficient* (PCC) to evaluate articulatory trace similarity, we only report this metric for Setup (2). This is because in Setup (1), the durations of SPARC and STArK sequences will not be identical. We also report *dynamic time warping* (DTW) [33] scores between the extracted SPARC features and the synthesized articulatory features from STArK. This metric indicates how similar, in terms of articulation, the two sequences are over each utterance. We report the *difference in DTW scores* (ΔDTW) between Setup (1) and Setup (2), where a negative ΔDTW indicates an improvement in DTW by adding aligner durations. Following Cho et al. [6], we report the EMA, pitch, and loudness metrics separately.

5. Results

Table 1 shows speech synthesis evaluation results. DNSMOS gives four values: DNSMOS P.808, *signal quality* (SIG), *background noise* (BAK), and *overall* (OVR), with the last three forming the DNSMOS P.835 audio evaluation metric. We see that baseline STArK performs similarly to the *ground-truth* (GT) audio from LibriTTS-R on DNSMOS. Providing alignment information (Setup (2)) or both alignment and prosody

[3] `https://huggingface.co/pyannote/embedding`

Table 1: *Speech synthesis evaluation results. The best model result for each metric is* ***bolded*** *and second best is* underlined*. Multiple* ***bolded*** *and* underlined *values indicate results that are not statistically different,* $p \geq 0.05$.

Model	DNSMOS				UTMOS ↑	NMOS ↑	SECS ↑	WER ↓
	P.808 ↑	SIG ↑	BAK ↑	OVR ↑	v2			
GT	3.71±0.01	3.465±0.007	3.925±0.008	3.135±0.009	3.25±0.01	3.9±0.4	1.000	2.39
SPARC	3.77±0.01	3.543±0.005	4.024±0.006	3.240±0.007	**3.03±0.01**	**3.7±0.3**	**0.493±0.006**	**3.31**
STArK	3.71±0.01	3.504±0.006	3.902±0.009	3.142±0.009	2.97±0.01	2.8±0.6	0.426±0.005	6.25
+ aligner	**3.84±0.01**	3.540±0.005	3.983±0.007	3.216±0.008	2.85±0.01	2.9±0.4	0.446±0.005	6.30
+ prosody	3.79±0.01	3.537±0.005	4.006±0.006	3.224±0.007	**3.03±0.01**	3.3±0.3	0.478±0.006	6.01
YourTTS	3.81±0.01	**3.568±0.005**	**4.100±0.004**	**3.307±0.006**	2.80±0.01	2.0±0.5	0.410±0.005	4.58

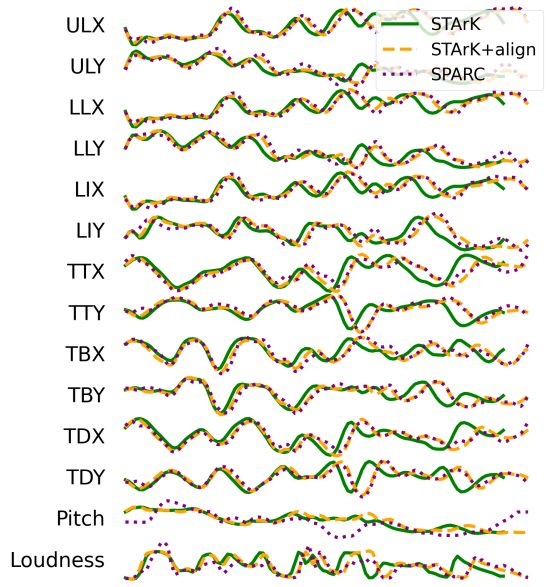


Figure 2: *Example comparison between generated articulatory features using STArK (green solid), STArK with aligner durations (orange dashed), and extracted SPARC features from the SPARC model (purple dotted). Y-axis labels are x–y positions for Upper Lip, Lower Lip, Lower Incisor, Tongue Tip, Tongue Blade, and Tongue Dorsum, in addition to Pitch and Loudness.*

information (Setup (3)) allows STArK to match SPARC in performance over P.835 metrics and exceed SPARC in P.808 MOS score. In the UTMOSv2 metric, STArK outperforms YourTTS and achieves similar results to SPARC when ground truth prosody is provided. The NMOS results are largely in agreement with the UTMOSv2 results, placing STArK between SPARC and YourTTS, with Setup (3) performing the best of the STArK setups.

We also observe that despite training on significantly less data and not having access to speaker embeddings during training, STArK performs admirably against multilingual YourTTS on DNSMOS metrics and outperforms YourTTS on SECS. We attribute this advantage in voice cloning capability to the strong speaker–content disentanglement provided by SPARC. YourTTS outperforms STArK with respect to WER, but the drop in performance from SPARC to STArK is expected. Adding ground truth prosody improves WER, indicating that future work should focus on improving prosody generation.

Fig. 2 shows an example of an articulatory trajectory for a single utterance generated by STArK Setup (1) (green), STArK

Table 2: *Articulatory feature similarity results for EMA, pitch, and loudness between STArK and SPARC. Pearson correlation coefficient (PCC) is calculated based on STArK with aligner durations and* Δ*DTW is the difference in dynamic time warping (DTW) distance between STArK with and without aligner durations.*

STArK	PCC ↑	ΔDTW ↓
EMA	0.905±0.001	−0.81±0.09
Pitch	0.533±0.009	−0.04±0.04
Loudness	0.796±0.002	−0.20±0.06

Setup (2) with aligner durations (orange), and SPARC inversion (purple). In this figure, base STArK does not use aligner durations and thus predicts phoneme durations without external conditioning. This results in a temporal warping of the trajectories that can be observed in the figure. After adding aligner durations the EMA features and the loudness feature appear to closely follow the SPARC-predicted trajectories, with only the pitch feature having noticeable deviation from the extracted SPARC trajectory. This observation is further supported in Table 2, with Setup (2) showing improved DTW metrics over Setup (1) for both EMA and loudness, but not for pitch. The lower PCC performance for pitch, as well as the lack of improvement in DTW after adding aligner durations, is expected as there is no prosody conditioning in the model. Importantly, these results demonstrate that we can inject durations into STArK for finer-grained control over output without loss of speech quality, which would be a useful functionality for tasks such as disfluent speech enhancement or reconstruction.

6. Conclusions

In this work, we introduced STArK, a text-to-articulatory synthesis model that aims to help bridge the articulatory data gap for the training of articulation-grounded speech models. STArK demonstrates strong articulatory trajectory prediction, high speech synthesis quality, and innate voice cloning capabilities. For future work, we believe that improving the time-domain features such as prosody, emotion, and accent will prove vital towards demonstrating the usability of articulatory representations in speech synthesis. We also plan to streamline the training process and enhance the model architecture to use more recent techniques such as flow matching to generate higher fidelity articulatory representations. Future work will also improve human and automated evaluation robustness and explore using STArK for downstream speech processing tasks, such as speech disorder assessment.

7. Acknowledgments

The authors thank Cheol Jun Cho and Prof. Gopala Anumanchipalli for their assistance in the usage of the SPARC model. Early stage experiments for STArK were conducted by the primary author, Xavier Yin, during his M.S. degree at the University of California, Berkeley under Prof. Gopala Anumanchipalli.

8. Generative AI Use Disclosure

Generative AI use was limited to editing and polishing the manuscript's content. Generative AI tools were not used to write drafts, analyze results, create citations, or otherwise in the writing process of this manuscript.

9. References

[1] G. Fant, *Acoustic Theory of Speech Production*. Berlin, Boston: De Gruyter Mouton, 1971. [Online]. Available: https://doi.org/10.1515/9783110873429

[2] C. P. Browman and L. M. Goldstein, "Articulatory Phonology: An Overview," *Phonetica*, vol. 49, pp. 155 – 180, 1992. [Online]. Available: https://doi.org/10.1159/000261913

[3] International Phonetic Association, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, 1999. [Online]. Available: https://doi.org/10.1017/9780511807954

[4] J. Chartier, G. K. Anumanchipalli, K. Johnson, and E. F. Chang, "Encoding of Articulatory Kinematic Trajectories in Human Speech Sensorimotor Cortex," *Neuron*, vol. 98, no. 5, pp. 1042–1054.e4, 2018. [Online]. Available: https://doi.org/10.1016/j.neuron.2018.04.031

[5] G. K. Anumanchipalli, J. Chartier, and E. F. Chang, "Speech synthesis from neural decoding of spoken sentences," *Nature*, vol. 568, no. 7753, pp. 493–498, Apr. 2019. [Online]. Available: https://doi.org/10.1038/s41586-019-1119-1

[6] C. J. Cho, P. Wu, T. S. Prabhune, D. Agarwal, and G. K. Anumanchipalli, "Coding Speech Through Vocal Tract Kinematics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 8, pp. 1427–1440, 2024. [Online]. Available: https://doi.org/10.1109/JSTSP.2024.3497655

[7] Z. Wang, Y. Wang, M. Li, and H. Huang, "ArtSpeech: Adaptive Text-to-Speech Synthesis with Articulatory Representations," in *Proceedings of the 32nd ACM International Conference on Multimedia*, ser. MM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 535–544. [Online]. Available: https://doi.org/10.1145/3664647.3681097

[8] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Integrating Articulatory Features Into HMM-Based Parametric Speech Synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009. [Online]. Available: https://doi.org/10.1109/TASL.2009.2014796

[9] T. G. Csapó, L. Tóth, G. Gosztolya, and A. Markó, "Speech Synthesis from Text and Ultrasound Tongue Image-based Articulatory Input," in *Speech Synthesis Workshop*, 2021.

[10] 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," in *International Conference on Learning Representations*, 2021.

[11] A. Łańcucki, "Fastpitch: Parallel Text-to-Speech with Pitch Prediction," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6588–6592, 2020. [Online]. Available: https://doi.org/10.1109/ICASSP39728.2021.9413889

[12] Y. Liu, Z. Xu, G. Wang, K. Chen, B. Li, X. Tan, J. Li, L. He, and S. Zhao, "DelightfulTTS: The Microsoft Speech Synthesis System for Blizzard Challenge 2021," in *The Blizzard Challenge 2021, virtual, October 23, 2021*. ISCA, 2021. [Online]. Available: https://doi.org/10.21437/Blizzard.2021-14

[13] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent Developments on Espnet Toolkit Boosted By Conformer," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5874–5878, 2020. [Online]. Available: https://doi.org/10.1109/ICASSP39728.2021.9414858

[14] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech 2020*, 2020, pp. 5036–5040. [Online]. Available: https://doi.org/10.21437/Interspeech.2020-3015

[15] 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 *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019. [Online]. Available: https://dl.acm.org/doi/10.5555/3454287.3454572

[16] R. Luo, X. Tan, R. Wang, T. Qin, J. Li, S. Zhao, E. Chen, and T.-Y. Liu, "Lightspeech: Lightweight and Fast Text to Speech with Neural Architecture Search," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5699–5703, 2021. [Online]. Available: https://doi.org/10.1109/ICASSP39728.2021.9414403

[17] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "RoFormer: Enhanced transformer with Rotary Position Embedding," *Neurocomputing*, vol. 568, p. 127063, 2024. [Online]. Available: https://doi.org/10.1016/j.neucom.2023.127063

[18] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-Attention with Relative Position Representations," in *North American Chapter of the Association for Computational Linguistics*, 2018. [Online]. Available: https://doi.org/10.18653/v1/N18-2074

[19] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3495724.3497152

[20] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502. [Online]. Available: https://doi.org/10.21437/Interspeech.2017-1386

[21] R. Badlani, A. Lancucki, K. J. Shih, R. Valle, W. Ping, and B. Catanzaro, "One TTS Alignment to Rule Them All," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6092–6096. [Online]. Available: https://doi.org/10.1109/ICASSP43922.2022.9747707

[22] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: a generative flow for text-to-speech via monotonic alignment search," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020. [Online]. Available: https://dl.acm.org/doi/10.5555/3495724.3496400

[23] G. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973. [Online]. Available: https://doi.org/10.1109/PROC.1973.9030

[24] Y. Koizumi, H. Zen, S. Karita, Y. Ding, K. Yatabe, N. Morioka, M. Bacchiani, Y. Zhang, W. Han, and A. Bapna, "LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus," in *Interspeech 2023*, 2023, pp. 5496–5500. [Online]. Available: https://doi.org/10.21437/Interspeech.2023-1584

[25] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019.

[26] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 2709–2720. [Online]. Available: https://proceedings.mlr.press/v162/casanova22a.html

[27] K. Baba, W. Nakata, Y. Saito, and H. Saruwatari, "The T05 System for the voicemos challenge 2024: Transfer Learning from Deep Image Classifier to Naturalness MOS Prediction of High-Quality Synthetic Speech," *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 818–824, 2024. [Online]. Available: https://doi.org/10.1109/SLT61566.2024.10832315

[28] C. K. A. Reddy, V. Gopal, and R. Cutler, "DNSMOS: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6493–6497, 2020. [Online]. Available: https://doi.org/10.1109/ICASSP39728.2021.9414878

[29] ——, "DNSMOS P.835: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 886–890, 2021. [Online]. Available: https://doi.org/10.1109/ICASSP43922.2022.9746108

[30] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[31] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 2020. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9052974

[32] J. M. Coria, H. Bredin, S. Ghannay, and S. Rosset, "A Comparison of Metric Learning Loss Functions for End-To-End Speaker Verification," in *Statistical Language and Speech Processing*, L. Espinosa-Anke, C. Martín-Vide, and I. Spasić, Eds. Springer International Publishing, 2020, pp. 137–148. [Online]. Available: https://doi.org/10.1007/978-3-030-59430-5_11

[33] H. Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 159–165, 1978. [Online]. Available: https://doi.org/10.1016/B978-0-08-051584-7.50016-4