



UnitSpeech: Speaker-adaptive Speech Synthesis with Untranscribed Data

Heeseung Kim¹, Sungwon Kim¹, Jiheum Yeom¹, Sungroh Yoon^{*1,2}

¹ Data Science and AI Lab, ECE, Seoul National University, Seoul 08826, Korea

² Interdisciplinary Program in AI, Seoul National University, Seoul 08826, Korea

{gmltmd789, ksw0306, quilava1234, sryoon}@snu.ac.kr

Abstract

We propose UnitSpeech, a speaker-adaptive speech synthesis method that fine-tunes a diffusion-based text-to-speech (TTS) model using minimal untranscribed data. To achieve this, we use the self-supervised unit representation as a pseudo transcript and integrate the unit encoder into the pre-trained TTS model. We train the unit encoder to provide speech content to the diffusion-based decoder and then fine-tune the decoder for speaker adaptation to the reference speaker using a single <unit, speech> pair. UnitSpeech performs speech synthesis tasks such as TTS and voice conversion (VC) in a personalized manner without requiring model re-training for each task. UnitSpeech achieves comparable and superior results on personalized TTS and any-to-any VC tasks compared to previous baselines. Our model also shows widespread adaptive performance on real-world data and other tasks that use a unit sequence as input.

Index Terms: speaker adaptation, text-to-speech, voice conversion, diffusion model, self-supervised unit representation

1. Introduction

As text-to-speech (TTS) models have shown significant advances in recent years [1, 2], there have also been works on adaptive TTS models which generate personalized voices using reference speech of the target speaker [3, 4, 5, 6, 7]. Adaptive TTS models mostly use a pre-trained multi-speaker TTS model and utilize methods such as using target speaker embedding [3, 4, 5] or fine-tuning the model with few data [3, 6, 7]. While the former allows easier adaptation compared to the latter, it suffers from relatively low speaker similarities.

Most fine-tuning-based approaches require a small amount of target speaker speech data and may also require a transcript paired with the corresponding speech. AdaSpeech 2 [7] proposes a pluggable mel-spectrogram encoder (mel encoder) to fine-tune the pre-trained TTS model with untranscribed speech. Since the mel encoder is introduced to replace the text encoder during fine-tuning, AdaSpeech 2 does not require a transcript when fine-tuning the decoder on the target speaker. However, its results are bounded only to adaptive TTS and show limitations such as requiring a relatively large amount of target speaker data due to its deterministic feed-forward decoder.

Recent works on diffusion models [8, 9] show powerful results on text-to-image generation [10] and personalization with only a few images [11, 12], and such trends are being extended to speech synthesis [13, 14] and adaptive TTS [15, 16]. Guided-TTS 2 leverages the fine-tuning capability of the diffusion model and the classifier guidance technique to build high-quality adaptive TTS with only a ten-second-long untranscribed

speech. However, Guided-TTS 2 requires training of its unconditional generative model, which results in more challenging and time-consuming training compared to typical TTS models.

In this work, we propose UnitSpeech, which performs personalized speech synthesis by fine-tuning a pre-trained diffusion-based TTS model on a small amount of untranscribed speech. We use the multi-speaker Grad-TTS as the backbone TTS model for speaker adaptation which requires transcribed data for fine-tuning. Likewise AdaSpeech 2, we introduce a new encoder model to provide speech content to the diffusion-based decoder without transcript. While AdaSpeech 2 directly uses mel-spectrogram as the input of the encoder, we use the self-supervised unit representation [17] which contains speech content disentangled with the speaker identity to better replace the text encoder. The newly introduced encoder, named unit encoder, is trained to condition the speech content into the diffusion-based decoder using the input unit. For speaker adaptation, we fine-tune the pre-trained diffusion model conditioned on the unit encoder output with a <unit, speech> pair of the target speaker. By customizing the diffusion decoder to the target speaker, UnitSpeech is capable of performing multiple adaptive speech synthesis tasks that receive transcript or unit as input.

We show that UnitSpeech is comparable to or outperforms baseline models on adaptive TTS and any-to-any VC tasks. We further ablate how each factor of UnitSpeech affects the pronunciation and speaker similarity for adaptive speech synthesis. In addition to samples for evaluation, we provide samples for a wide range of scenarios, including various real-word reference data from YouTube and other tasks using units on demo page¹.

Our contributions are as follows:

- To the best of our knowledge, this is the first work that introduces unit representation to utilize untranscribed speech for speaker adaptation.
- We propose a pluggable unit encoder for pre-trained TTS model, enabling fine-tuning using untranscribed speech.
- We introduce a simple guidance technique to improve pronunciation accuracy in adaptive speech synthesis.

2. Method

Our aim is the personalization of existing diffusion-based TTS models using only untranscribed data. To personalize a diffusion model [8, 9] without any transcript, we introduce a unit encoder that learns to encode speech content for replacing the text encoder during fine-tuning. We use the trained unit encoder to adapt the pre-trained TTS model to the target speaker on various tasks. We briefly explain the pre-trained TTS model in Section 2.1, explain methods used for unit extraction and unit encoder

* Corresponding Author

¹Demo: <https://unitspeech.github.io/>

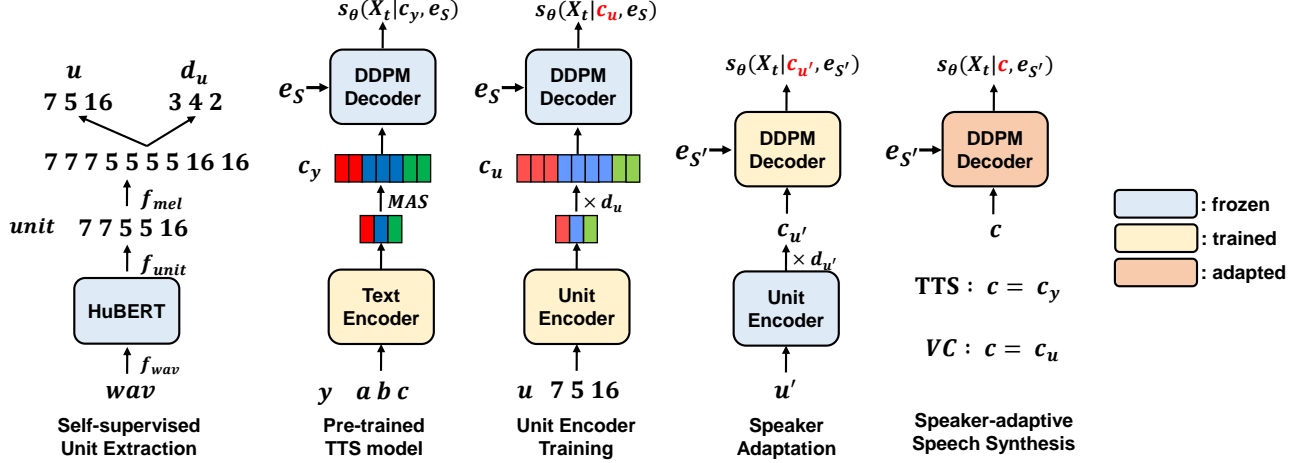


Figure 1: The overall procedure of UnitSpeech.

training in Section 2.2, and show how the trained UnitSpeech is used to perform various tasks in Section 2.3.

2.1. Diffusion-based Text-to-Speech Model

Following the success of Grad-TTS [14] in single-speaker TTS, we adopt a multi-speaker Grad-TTS as our pre-trained diffusion-based TTS model. It consists of a text encoder, a duration predictor, and a diffusion-based decoder, just like Grad-TTS, and we additionally provide speaker information for multi-speaker TTS. To provide speaker information, we use a speaker embedding extracted from a speaker encoder.

The diffusion-based TTS model defines a forward process that gradually transforms mel-spectrogram X_0 into Gaussian noise $z = X_T \sim N(0, I)$, and generates data by reversing the forward process. While Grad-TTS defines the prior distribution using mel-spectrogram-aligned text encoder output, we use the standard normal distribution as the prior distribution. The forward process of the diffusion model is as follows:

$$dX_t = -\frac{1}{2}X_t\beta_t dt + \sqrt{\beta_t}dW_t, \quad t \in [0, T], \quad (1)$$

where the β_t is a pre-defined noise schedule, and W_t denotes the Wiener process. We set T to 1 as in [14].

The pre-trained diffusion-based decoder predicts the score which is required when sampling through the reverse process. For pre-training, the data X_0 is corrupted into noisy data $X_t = \sqrt{1 - \lambda_t}X_0 + \sqrt{\lambda_t}\epsilon_t$ through the forward process, and the decoder learns to estimate the conditional score given the aligned text encoder output c_y and the speaker embedding e_S with the training objective in Eq. 2.

$$L_{grad} = \mathbb{E}_{t, X_0, \epsilon_t} [\|(\sqrt{\lambda_t}s_\theta(X_t, t|c_y, e_S) + \epsilon_t)\|_2^2], \quad (2)$$

where $\lambda_t = 1 - e^{-\int_0^t \beta_s ds}$, and $t \in [0, 1]$. Using the estimated score s_θ , the output of the diffusion-based decoder, the model can generate mel-spectrogram X_0 given the transcript and speaker embedding using the discretized reverse process which is as follows:

$$X_{t-\frac{1}{N}} = X_t + \frac{\beta_t}{N} \left(\frac{1}{2}X_t + s_\theta(X_t, t|c_y, e_S) \right) + \sqrt{\frac{\beta_t}{N}}z_t, \quad (3)$$

where N denotes the number of sampling steps.

In addition to L_{grad} in Eq. 2, the pre-trained TTS model aligns the output of the text encoder with the mel-spectrogram

using monotonic alignment search (MAS) proposed in Glow-TTS [2] and minimizes the distance between the aligned text encoder output c_y and the mel-spectrogram X_0 using the encoder loss $L_{enc} = MSE(c_y, X_0)$. To disentangle the text encoder output with speaker identity, we minimize the distance between the speaker-independent representation c_y and X_0 without providing the speaker embedding e_S to the text encoder.

2.2. Unit Encoder Training

While we aim to fine-tune the pre-trained TTS model for high-quality adaptation given minimal amounts of untranscribed reference data, the pre-trained TTS model alone is structurally challenging of doing so. Our pre-trained TTS model is restricted only to training with transcribed speech data, whereas the larger half of real-world speech data is occupied by untranscribed data. As a solution to this problem, we combine a unit encoder with the pre-trained TTS model to expand the generation capabilities for adaptation.

The unit encoder is a model identical to the text encoder of the TTS model in both architecture and role. In contrast to the text encoder which uses transcripts, the unit encoder uses a discretized representation known as unit, which broadens the model's generation capabilities, enabling adaptation on untranscribed speech. Specifically, unit is a discretized representation obtained by HuBERT [17], a self-supervised model for speech. The leftmost part of Fig. 1 shows the unit extraction process, where speech waveform is used as input of HuBERT, and output representation is discretized by K -means clustering into unit clusters, resulting in a unit sequence. Note that by setting an appropriate number of clusters, we can constrain the unit to contain mainly the desired speech content. The obtained unit sequence from HuBERT is upsampled to mel-spectrogram length, where we then compress into unit duration d_u and squeezed unit sequence u .

The center of Fig. 1 shows the training process of the unit encoder. With squeezed unit sequence u as input, the unit encoder, plugged into the pre-trained TTS model, plays the same role as the text encoder. The unit encoder is trained with the same training objective $L = L_{grad} + L_{enc}$, only having c_y replaced with c_u , an extended unit encoder output using ground-truth duration d_u . This results in c_u being placed in the same space as c_y , enabling our model to replace the text encoder with the unit encoder during fine-tuning. Note that the diffusion decoder is frozen, and only the unit encoder is to be trained.

2.3. Speaker-Adaptive Speech Synthesis

Combining the pre-trained TTS model and the pluggable unit encoder, we are able to perform various speech synthesis tasks in an adaptive fashion by using a single untranscribed speech of the target speaker. Using squeezed unit u' and unit duration $d_{u'}$ extracted from the reference speech as in the previous section, we fine-tune the decoder of the TTS model using the unit encoder. When doing so, the unit encoder is frozen to minimize pronunciation deterioration, and we only train the diffusion decoder using the objective in Eq. 2 with c_y switched into $c_{u'}$.

Our trained model is capable of synthesizing adaptive speech using either transcript or unit as input. For TTS, we provide c_y as a condition to the fine-tuned decoder to generate personalized speech with respect to the given transcript. When performing tasks using units including voice conversion or speech-to-speech translation, squeezed unit u and unit duration d_u are extracted from the given source speech using HuBERT. The extracted two are inputted into the unit encoder, which outputs c_u , and the adaptive diffusion decoder uses c_u as a condition to generate voice-converted speech.

To further enhance the pronunciation of our model, we leverage a classifier-free guidance method [18] during sampling, which amplifies the degree of conditioning for the target condition using an unconditional score. Classifier-free guidance requires a corresponding unconditional embedding e_Φ to estimate the unconditional score. Since the encoder loss drives the encoder output space close to mel-spectrogram, we set the e_Φ to the mel-spectrogram mean of the dataset c_{mel} instead of training e_Φ as in other works [10]. The modified score we utilize for classifier-free guidance is as follows:

$$\begin{aligned}\hat{s}(X_t, t|c_c, e_S) &= s(X_t, t|c_c, e_S) + \gamma \cdot \alpha_t, \\ \alpha_t &= s(X_t, t|c_c, e_S) - s(X_t, t|c_{mel}, e_S).\end{aligned}\quad (4)$$

c_c here indicates the aligned output of text or unit encoder while γ denotes the gradient scale that determines the amount of provided condition information.

3. Experiments

3.1. Experimental Setup

3.1.1. Datasets

We use LibriTTS [19] to train the multi-speaker TTS model and the unit encoder. LibriTTS is a TTS dataset consisting of 2,456 different speakers, and we use the entire train subset. For training the speaker encoder, we use VoxCeleb 2 [20], a dataset consisting of 6,112 speakers. To show the unseen speaker adaptation capability of UnitSpeech on TTS, we select 10 speakers and a reference speech for each speaker from the `test-clean` subset of LibriTTS following YourTTS [3]. For evaluation on any-to-any VC, we randomly choose 10 reference speakers from the `test-clean` subset of LibriTTS, and randomly select 50 source samples from the `test-clean` subset. The reference samples are all 7 ~ 32 seconds long.

3.1.2. Training and Fine-tuning Details

Our pre-trained TTS model shares the same architecture and hyperparameters with Grad-TTS except for the doubled number of channels for multi-speaker modeling. The architecture of the unit encoder is equal to that of the text encoder. We train the TTS model on 4 NVIDIA RTX 8000 GPUs for 1.4M iterations and train the unit encoder for 200K iterations. We use the

Adam optimizer [21] with the learning rate $1e-4$ and batch size 64. The transcript is converted into the phoneme sequence using [22]. When extracting unit sequences, we utilize textless-lib [23]. We also train the speaker encoder on VoxCeleb2 [20] with GE2E [24] loss to extract the speaker embedding e_S of each reference speech. For fine-tuning, we use Adam optimizer [21] with learning rate $2 \cdot 10^{-5}$. We set the number of fine-tuning steps to 500 as a default, which only requires less than a minute on a single NVIDIA RTX 8000 GPU.

3.1.3. Evaluation

To evaluate the performance on adaptive TTS, we compare UnitSpeech with Guided-TTS 2 [16], Guided TTS 2 (zero-shot), and YourTTS [3]. For baselines on voice conversion, we used DiffVC [25], YourTTS [3], and BNE-PPG-VC [26]. As for the vocoder, we use the officially released pre-trained model of universal HiFi-GAN [27]. We use the official implementations and pre-trained models for each baseline. Only a single reference speech is used for the adaptation of all the models, and generated audios are downsampled to 16khz for fair comparison. For all the diffusion-based models, we fix the number of sampling steps N to 50. We set the gradient scale γ of UnitSpeech to 1.0 for TTS and 1.5 for VC.

We select 5 sentences from `text-clean` subset of LibriTTS each for the 10 reference speakers chosen in 3.1.1 and set the total of 50 sentences as test set for TTS. 50 source speeches for evaluation of VC are selected as explained in 3.1.1. We use three metrics for model evaluation: the 5-scale mean opinion score (MOS) on audio quality and naturalness, the character error rate (CER) indicating pronunciation accuracy, and speaker encoder cosine similarity (SECS) to measure how similar the generated sample is to the target speaker. When calculating CER, we use the CTC-based conformer [28] of NEMO toolkit [29] as Guided-TTS 2. We also use the speaker encoder of Resemblyzer [30] for SECS evaluation as YourTTS. We generate adapted samples for each corresponding test sample and measure the CER and SECS values. We report the average values by repeating this measurement 5 times.

3.2. Results

3.2.1. Adaptive Text-to-Speech

In Table 1, we compare UnitSpeech to other adaptive TTS baselines. The MOS results indicate that our model generates high-quality speech comparable to Guided-TTS 2, a model for adaptive TTS only. UnitSpeech also shows superior performance compared to YourTTS, a model capable of both adaptive TTS and voice conversion similar to our model. Furthermore, we show that UnitSpeech is capable of generating speech with accurate pronunciation through the CER results.

We also confirm that our model is on par with Guided-TTS 2, which is also fine-tuned on the reference speech and outperforms zero-shot adaptation baselines on target speaker adaptation from the SECS results. Through these results, we show that even though our model is capable of various tasks using either unit or transcript inputs in a personalized manner, it shows reasonably comparable TTS quality against single-task-only baselines. Samples of each model can be found on our demo page.

3.2.2. Any-to-Any Voice Conversion

As shown in Table 2, UnitSpeech performs reasonably on VC task. Our model outperforms baselines regarding naturalness and speaker similarity, with a slight decline in pronunciation ac-

Table 1: MOS, CER, and SECS for TTS experiments on LibriTTS. Guided-TTS 2 (zs) indicates Guided-TTS 2 that performs zero-shot adaptation without fine-tuning.

	5-scale MOS	CER(%)	SECS
Ground Truth	4.49 ± 0.06	0.7	0.933
Mel + HiFi-GAN [27]	4.09 ± 0.10	0.75	0.927
UnitSpeech	4.13 ± 0.10	1.75	0.935
Guided-TTS 2 [16]	4.16 ± 0.10	0.84	0.937
Guided-TTS 2 (zs) [16]	4.10 ± 0.11	0.8	0.873
YourTTS [3]	3.57 ± 0.13	2.38	0.866

Table 2: MOS, CER, and SECS for VC experiments on LibriTTS. Mel + HiFi-GAN indicates samples obtained by inputting source speech mel-spectrogram into HiFi-GAN.

	5-scale MOS	CER(%)	SECS
Source	4.47 ± 0.06	0.7	-
Mel + HiFi-GAN [27]	4.24 ± 0.08	0.75	-
UnitSpeech	4.26 ± 0.09	3.55	0.923
DiffVC [25]	3.97 ± 0.09	3.67	0.909
YourTTS [3]	3.88 ± 0.10	2.20	0.763
BNE-PPG-VC [26]	3.86 ± 0.10	1.37	0.851

curacy as a trade-off. This result demonstrates that our model is capable of both high-quality adaptive TTS and any-to-any VC. We include samples of our model and baselines on demo page.

3.2.3. Other Data and Tasks

In the previous section, we explained that by fine-tuning the model with a single reference speech of the target speaker, we were able to obtain results either comparable or superior to the baselines on both TTS and VC tasks. UnitSpeech is capable of not only TTS and VC but also any other speech synthesis task that may use unit, providing a sense of personalization to each task. On speech-to-speech translation (S2ST), one of the most general tasks that can utilize unit, we replace the speech synthesis part, which generally uses a single speaker unit-HiFi-GAN [31], with UnitSpeech, and show possibilities of personalized S2ST on CoVoST-2 [32]. Samples are on our demo page.

UnitSpeech also maintains reasonable fine-tuning quality even on real-world data for various tasks. To show the real-world availability, we use 10-second-long real-world data extracted from Youtube. Due to copyright issues, we do not explicitly upload these data, but instead, post the Youtube link and start time/end time of each data. We post various adaptation samples on our demo page.

3.2.4. Analysis

We show the effects of several factors of our model in Table 3.

The number of unit clusters We observed that the number of clusters K does not significantly affect TTS results. In the case of voice conversion, however, which directly uses units as inputs, the increase in K allows a more precise segmentation of pronunciation, leading to better pronunciation accuracy.

Fine-tuning Our results demonstrate that the more we fine-tune, speaker similarity increases gradually and eventually converges around 500 iterations. We also observe that the pronunciation accuracy decreases when fine-tuning over 2,000 iterations. Thus, we have set the default number of iterations for fine-tuning to 500, which only takes less than a minute in a single NVIDIA RTX 8000 GPU.

We also measure pronunciation accuracy and speaker simi-

Table 3: CER, SECS regarding the number of unit clusters, fine-tuning iterations, length of untranscribed speech used for fine-tuning, and the gradient scale in classifier-free guidance.

		Text-to-Speech		Voice Conversion	
		CER (%)	SECS	CER (%)	SECS
K (# Units)	50	1.94	0.932	12.64	0.928
	100	1.87	0.930	5.69	0.920
	200	1.75	0.935	3.55	0.923
	500	2.10	0.932	3.80	0.918
# Iters	0	1.89	0.849	3.65	0.845
	50	2.15	0.905	3.78	0.893
	200	1.96	0.925	3.92	0.924
	500	1.75	0.935	3.55	0.923
	2000	2.04	0.937	3.78	0.925
Length (secs)	3	2.16	0.916	3.82	0.926
	5	1.96	0.921	3.44	0.925
	30	1.88	0.949	3.07	0.946
Gradient scale γ	0.0	2.83	0.941	5.02	0.939
	0.5	2.04	0.939	4.15	0.936
	1.0	1.75	0.935	3.86	0.93
	1.5	1.74	0.929	3.55	0.923
	2.0	1.79	0.923	3.74	0.918

larity according to the amount of reference speech used for fine-tuning. Our results show that both metrics improve as the length of reference speech increases. Furthermore, our model can still achieve sufficient pronunciation accuracy and speaker similarity even with a 5-second-long short reference speech.

Gradient scale in classifier-free guidance The results in Table 3 indicate that the proposed guidance method improves pronunciation at the cost of a minor decrease in speaker similarity. Therefore, we choose the gradient scale γ that maximizes the pronunciation improvement while minimizing the reduction in speaker similarity, which is 1 for TTS and 1.5 for VC.

4. Conclusion

We proposed UnitSpeech, a diffusion model that enables various adaptive speech synthesis tasks by fine-tuning a small amount of untranscribed speech. UnitSpeech consists of a unit encoder in addition to the text encoder, eliminating the need for a transcript during fine-tuning. We also introduce a simple guidance technique that allows UnitSpeech to perform high-quality adaptive speech synthesis with accurate pronunciation. We showed that UnitSpeech is on par with the TTS baselines and outperforms VC baselines regarding audio quality and speaker similarity. Our demo results also indicate that UnitSpeech can robustly adapt to untranscribed speech of real-world data and we can substitute UnitSpeech for speech synthesis modules of tasks that take the unit as input.

5. Acknowledgements

This work was supported by SNU-Naver Hyperscale AI Center, Samsung Electronics (IO221213-04119-01), Institute of Information & communications Technology Planning & Evaluation grant funded by the Korea government (MSIT) [2021-0-01343, AI Graduate School Program (SNU)], National Research Foundation of Korea grant funded by MSIT (2022R1A3B1077720), and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, SNU in 2023.

6. References

- [1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
- [2] J. Kim, S. Kim, J. Kong, and S. Yoon, “Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [3] 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>
- [4] E. Cooper, C.-I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, “Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6184–6188.
- [5] Y. Wu, X. Tan, B. Li, L. He, S. Zhao, R. Song, T. Qin, and T.-Y. Liu, “Adaspeech 4: Adaptive text to speech in zero-shot scenarios,” *arXiv preprint arXiv:2204.00436*, 2022.
- [6] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, sheng zhao, and T.-Y. Liu, “Adaspeech: Adaptive text to speech for custom voice,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=Drynvt7gg4L>
- [7] Y. Yan, X. Tan, B. Li, T. Qin, S. Zhao, Y. Shen, and T.-Y. Liu, “Adaspeech 2: Adaptive text to speech with untranscribed data,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6613–6617.
- [8] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2256–2265. [Online]. Available: <https://proceedings.mlr.press/v37/sohl-dickstein15.html>
- [9] J. Ho, A. Jain, and P. Abbeel, “Denosing Diffusion Probabilistic Models,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Curran Associates, Inc., 2020, vol. 33.
- [10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.
- [11] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, “Multi-concept customization of text-to-image diffusion,” *arXiv*, 2022.
- [12] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, “Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation,” 2022.
- [13] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “DiffWave: A Versatile Diffusion Model for Audio Synthesis,” in *International Conference on Learning Representations*, 2021.
- [14] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, “Grad-TTS: A Diffusion Probabilistic Model for Text-to-Speech,” in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 8599–8608.
- [15] M. Kang, D. Min, and S. J. Hwang, “Any-speaker adaptive text-to-speech synthesis with diffusion models,” 2022. [Online]. Available: <https://arxiv.org/abs/2211.09383>
- [16] S. Kim, H. Kim, and S. Yoon, “Guided-tts 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data,” 2022. [Online]. Available: <https://arxiv.org/abs/2205.15370>
- [17] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [18] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. [Online]. Available: <https://openreview.net/forum?id=qw8AKxfYbI>
- [19] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, “LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech,” in *Proc. Interspeech 2019*, 2019, pp. 1526–1530.
- [20] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” in *INTERSPEECH*, 2018.
- [21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [22] J. Park, Kyubyong Kim, “g2pe,” <https://github.com/Kyubyong/g2p>, 2019.
- [23] E. Kharitonov, J. Copet, K. Lakhota, T. A. Nguyen, P. Tomasello, A. Lee, A. Elkahky, W.-N. Hsu, A. Mohamed, E. Dupoux, and Y. Adi, “textless-lib: a library for textless spoken language processing,” 2022.
- [24] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.
- [25] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. S. Kudinov, and J. Wei, “Diffusion-based voice conversion with fast maximum likelihood sampling scheme,” in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=8c50f-DoWau>
- [26] S. Liu, Y. Cao, D. Wang, X. Wu, X. Liu, and H. Meng, “Any-to-many voice conversion with location-relative sequence-to-sequence modeling,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1717–1728, 2021.
- [27] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative Adversarial networks for Efficient and High Fidelity Speech Synthesis,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [28] 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 *Proc. Interspeech 2020*, 2020, pp. 5036–5040.
- [29] O. Kuchaiev, J. Li, H. 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 preprint arXiv:1909.09577*, 2019.
- [30] G. Louppe, “Resemblyzer,” <https://github.com/resemble-ai/Resemblyzer>, 2019.
- [31] S. Popuri, P.-J. Chen, C. Wang, J. Pino, Y. Adi, J. Gu, W.-N. Hsu, and A. Lee, “Enhanced Direct Speech-to-Speech Translation Using Self-supervised Pre-training and Data Augmentation,” in *Proc. Interspeech 2022*, 2022, pp. 5195–5199.
- [32] C. Wang, A. Wu, J. Gu, and J. Pino, “CoVoST 2 and Massively Multilingual Speech Translation,” in *Proc. Interspeech 2021*, 2021, pp. 2247–2251.