Journal of Computer Science and Cybernetics, V.39, N.2 (2023), 159–173
DOI no 10.15625/1813-9663/18136

ADAPT-TTS: HIGH-QUALITY ZERO-SHOT MULTI-SPEAKER TEXT-TO-SPEECH ADAPTIVE-BASED FOR VIETNAMESE

PHUONG PHAM NGOC[1,2], CHUNG TRAN QUANG[2,3], MAI LUONG CHI[4,*]

[1] *Thai Nguyen University, Tan Thinh Ward, Thai Nguyen City, Thai Nguyen Province, Viet Nam*
[2] *AIMed Artificial Intelligence Solution, 74 Lane 358/40 Bui Xuong Trach Street, Thanh Xuan District, Ha Noi, Viet Nam*
[3] *Japan Advanced Institute of Science and Technology (JAIST), Japan*
[4] *Institute of Information Technology, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet Street, Cau Giay District, Ha Noi, Viet Nam*

Abstract. Current adaptive-based speech synthesis techniques are based on two main streams: 1) Fine-tuning the model using small amounts of adaptive data; 2) Conditionally training the entire model through a speaker embedding of the target speaker. However, both of these methods require adaptive data to appear during training, which makes the training cost to generate new voices quite expensive. In addition, the traditional text to speech (TTS) model uses a simple loss function to reproduce the acoustic features. However, this optimization is based on incorrect distribution assumptions leading to noisy composite audio results. In this paper, we propose the Adapt-TTS model that allows high-quality audio synthesis from a small adaptive sample without training to solve these problems. The main contributions of the paper are: 1) The extracting mel-vector (EMV) architecture allows for a better representation of speaker characteristics and speech style; 2) An improved zero-shot model with a denoising diffusion model (mel-spectrogram denoiser) component allows for new voice synthesis without training with better quality (less noise). The evaluation results have proven the model's effectiveness when only needing a single utterance (1-3 seconds) of the reference speaker, the synthesis system gave high-quality synthesis results and achieved high similarity.

Keywords. Zero-shot TTS, multi-speaker, text-to-speech, diffusion models, mel-spectrogram denoiser, extracting mel-vector, EMV, adapt-TTS.

1. INTRODUCTION

Currently, speech synthesis techniques (TTS text-to-speech) based on neural networks have achieved the same naturalness as humans and are widely applied in real life. However, today's most popular and advanced synthesis models, such as Tacotron2 [1], Fastspeech2 [2], and VITS [3],... still require large amounts of data from a single speaker or multi-speaker. It also requires a long time to retrain the entire model every time a new speaker is added. The

*Corresponding author.
E-mail addresses: phuongpn@tnu.edu.vn (P.N. Phuong); chungtran@ai4med.vn (T.Q. Chung); lcmai@ioit.ac.vn (L.C Mai)

above TTS models can synthesize high quality with the voices in the training data or seen in training progress. However, without retraining, synthesis quality remains a significant challenge [4,5]. There is a great need for new speech learning applications with only a small amount of reference speaker data (target speaker), but still ensures that the synthesized voice achieves similarity with the sample voice, so adaptive techniques were proposed to solve these problems. Currently, two main adaptation techniques are popularly used: 1) Fine-tune all or part of the layer with adaptive data based on a pre-trained model (which has been trained with large amounts of data) [6]; 2) Use a vector to capture the representation of the speaker's characteristics with a small amount of adaptive sample [7,8]. These two methods give a good synthesis quality and a high similarity of the synthesized voices to the target voice. However, they require expensive computational resources, and besides, there are still two problems: First, speakers with too small sample data (target voice only one sentence or few seconds) are not adaptable, not good or not trainable; Second, to learn a new voice, it is still necessary to fine-tune a sample of the target voice to update the model parameters and the seen speaker training process for a long time (hours or even days). This leads to consuming computational resources and time-consuming to generate new voices, limiting many possibilities for practical application. A new approach called zero-shot is adopted to adopt a new voice with just one utterance or seconds of the sample without additional training. This technique allows the adaptation of the new voice without retraining; moreover, the data required for training is tiny (just one sentence or a few seconds of target voice data) [5]. Zero-shots in speech synthesis are techniques aimed at training a model that allows the generation of new voices under the condition that these voices have never appeared during training or are unknown during supervised learning (unseen speaker) [10]. These studies open up several useful applications, such as smart speaker systems (with small computational resources) that can tell stories or communicate with their voice, learn new voices on-site without retraining, and flexible speaker voice-over systems are provided on-site. Zero-shot multi-speaker TTS models typically use speaker embedding that can be easily adapted to the new speaker, allowing them to generate a new speaker's voice with much smaller data than other methods adapted by fine-tuning. These models have shown promising results regarding synthesizer quality and generalizability for new speakers. All in all, Zero-shot multi-speaker TTS is an exciting and rapidly growing area and has the potential to significantly impact how TTS systems are built and used in the future. The process of modeling speaker features in TTS consists of 3 steps: 1) Extracting the features of the target speaker; 2) Using these features as conditions for a synthetic TTS model, and 3) Generating the mel-spectrogram based on that representation. In the first step, the zero-shot TTS model typically uses a speaker embedding to represent the target speaker features best. Most of the research focuses on speaker encoder enhancement. However, it is difficult to accurately extract speaker characteristics in zero-shot conditions such as speaker characteristics, speaking style, and emotion. In steps 2 and 3, synthetic models such as non-autoregressive cannot produce diverse synthetic speech. It is because the model is often optimized using a simple regression loss function (e.g. L1, L2), and there is not any probabilistic model to reconstruct the acoustic features [11, 12]. The paper is structured into five main parts: The introduction presents an overview of TTS in the conditions of very little sample data, no training, and low computational cost, thereby posing the need for zero-shot adaptation; Related work presents related research on Zero-shot in TTS, diffusion model, and style vector; The main part presents the Adapt-TTS model

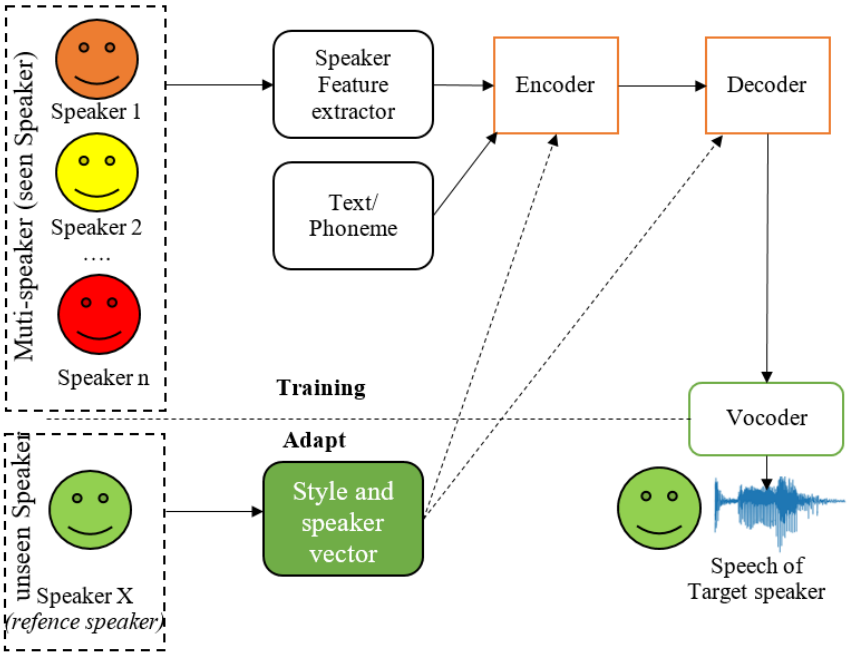

Figure 1: Basic speaking TTS multi-person zero-shot model

with two improvements applying for Multi-speaker TTS zero-shot: 1) Propose Extracting Mel-vector (EMV) architecture allows voice feature representation for better generalization. This architecture has effectively learned speaker characteristics from meager target voice samples. 2) Propose a Mel-spectrogram denoiser with kernel architecture using the denoising diffusion model for zero-shot Multi-speaker TTS model to improve synthesis quality and denoising ability; Finally, the experiments are evaluated and concluded.

2. RELATED WORKS

2.1. Zero-shot multispeaker TTS

Zero-shot multi-speaker TTS was first proposed by Arik et al., [8]. The idea of using a speaker encoder as a conditioning signal was further explored [4, 13], trying to close the quality gap between the speakers seen in the training set and those not in the training set (unseen) in the zero-shot Multi Speaker TTS model using embedding as extra information (Fig. 1). This study proposed a speaker embedding that uses neural network-based LDEs speaker embeddings to enhance the similarity and naturalness of voices and uses x-vectors to increase the scalability of the speaker verification task. With the use of embedding parts of the speaker, attention is given to encoding a more general speaking style instead of the speaker's audio [14]; [15] as well as methods that decode differently in the acoustic space such as generative flow [16], further efforts have been made to close the quality gap between seen speakers and unseen speakers. In addition, adapting Multi-speaker TTS models for voice transcription with few target voices requires diversity (including high-quality voice plurals and multiple speech attributes) of the speakers in the training data, and It is very important to achieve high generalization on the unseen-speaker dataset[8]. Therefore, these are still

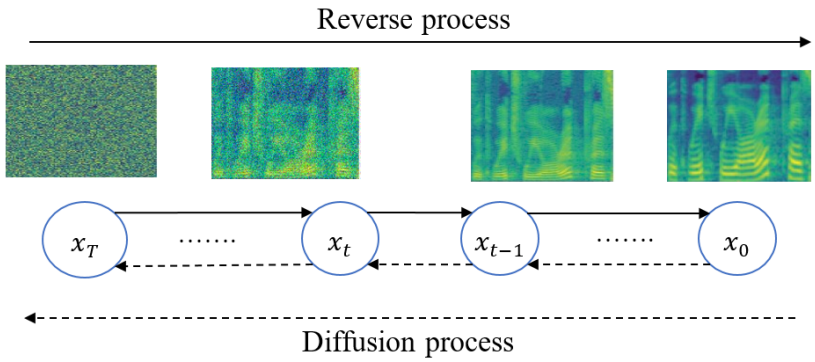

Figure 2: Visual depiction of the Diffusion model's reverse and diffusion process

major challenges needed to be solved.

2.2. Diffusion probabilistic models

The denoising diffusion probabilistic model (referred to as the denoising diffusion model) has shown high efficiency in image and sound generation [17, 18]. The denoising diffusion model is a Markov sequence that has been parameterized and trained using variable inference to generate matching patterns that resemble the original data after a finite time [17]. The transformations of this sequence are learned to reverse diffusion; the Markov series gradually add noise to the data in the direction opposite to the sampling direction until the signal is destroyed. When the diffusion includes a small amount of Gaussian noise, it is sufficient to set the sampling sequence transformations to Gaussian conditional, allowing for a particularly simple neural network parameterization. The diffusion model consists of two opposite processes, as depicted in Fig. 2: 1) The diffusion process is a Markov series with fixed parameters to convert complex data into an isotropic Gaussian distribution by gradually adding Gaussian noises; 2) The reverse process is a Markov sequence implemented by a neural network to learn how to recover the original data from repeated Gaussian white noise. The goal consists of two things, again the distance between the forward-diffusing noise x_t and the reverse diffusing decoder x_T, and argmax how to log-likelihood the maximum reverse diffusion probability between x_0 based on the noise decoder. The diffusion model is highly flexible and allows architecture with the same input and output sizes. That is essential in applying the diffusion model in speech synthesis to achieve the highest quality and likely-hook synthesized voice possible.

2.3. Style vector

An encoder is a component that encodes variable-length strings into fixed-dimensional representation vectors. In the basic multi-speaker TTS model [2, 7, 19], in the speaker encoder, an essential component is the speaker embedding to represent each speaker's voice signal as a feature vector. These vectors do not carry the speaker's features but carry the speaker's identity information. Adaptation-based TTS multi-speaker systems must use speaker features to train and refine the adaptive model. In order to do that, speech processing systems must first convert each variable-length audio clip into a fixed-length vector

representing the speaker's identity, called speaker embedding, and real now cluster based on these vectors. Speaker embedding is also widely used in speech-processing tasks, such as speaker recognition, speaker classification, speech tuning, and language synthesis [19–22]. Traditional methods often use the embedding module to extract a representative vector of the speaker's features. We can model the traditional method as the following formula

$$emb = Emb(Speaker\_ID). \tag{1}$$

However, it can be seen that this simple technique cannot represent the characteristics of each speaker (identity, gender, age, health) because it only uses speaker identifiers as input for the module. Some studies suggest another representative vector that carries information about the speaker's speaking style: style vector. Such as a study [14] that introduced GST (global style token) trained with unknown labels to learn how to model audio expressions and thereby control the synthesis in various styles such as speed, utterance, and textual independence. Sometimes the model shows a successful style transition. However, interleaved training only guarantees that some possible combinations of style classes are seen during training, resulting in a loss of representation of the speaker's style. The study of [11] used SALN (style-adaptive layer normalization) to align the gain and bias of the text input with style extracted from a reference short audio. Thus, it is possible to describe in general the style vector s representing the style of speaker X from the $Speech\_X$ reference audio input encoded by the style encoder as follows

$$s = Style\_encoder(Speech\_X). \tag{2}$$

3. ADAPT-TTS

3.1. Overall architecture

The adapt-TTS architecture consists of the main components: The architecture of Adapt-TTS consists of the following main components: EMV module to extract speaker features and styles of speech into a feature vector. Phoneme encoder module to convert phoneme sequences into phoneme hidden sequences. Then, the variance adapter will add duration, pitch, and energy information to the hidden sequences. Based on the diffusion model kernel, the mel-spectrogram denoiser will receive the hidden information from the previous steps to decode the output into high-quality mel-spectrograms. Finally, the vocoder module converts these mel-spectrograms into speech signals. The overall architecture is depicted in Fig. 3. The detailed architecture and functionality of the proposed enhancement modules are shown below.

3.2. Extracting mel-vector (EMV)

We propose a new module called "mel extraction vector" (EMV module), which can extract a fixed vector from the speaker's mel-spectrogram to accurately represent the speaker's features as the speaker and speaking style. EMV is to take the reference voice X as input. This block aims to extract an embedding stv vector containing the style and features of speaker X

$$stv = EMV(Mel). \tag{3}$$

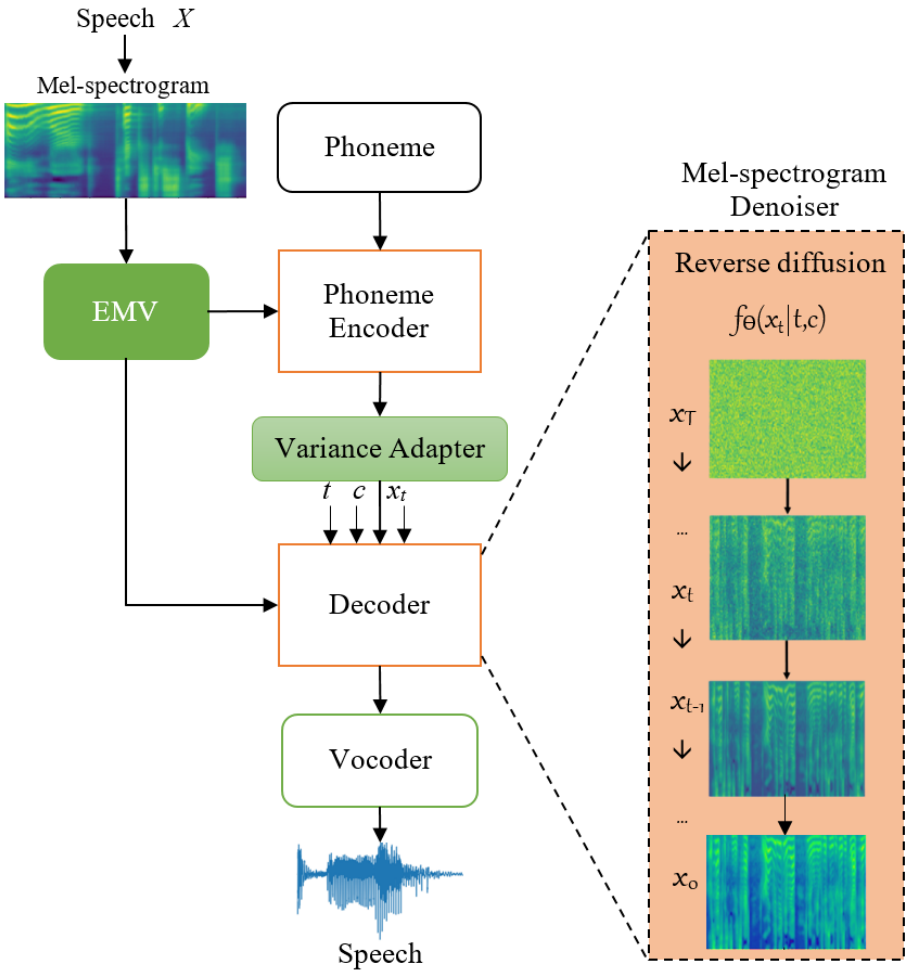

Figure 3: Overall architecture of adapt-TTS

In this module block, we use three main components, namely encoder feature, decoder feature, and embedding feature.

First, at the encoder feature module, the mel-spectrogram input is first fed to the fully connected (FC) layer, and the Mish activation functions convert each frame of the mel-spectrogram into the hidden sequence, which then passes through the two FC layers. The purpose of the encoder feature block is to convert the input feature into an encoder feature. Next, this vector will be passed through the decoder feature module. By using Conv1D +ReLu with the residual result to capture the information sequence from the given speech, this module aims to convert the decoder feature to the decoder feature. In addition, we also integrate skip connection, which will use the valuable features of the previous blocks. Finally, the decoder feature will be moved to the embedding feature module, which has a self-attention module with redundant connectivity plus the affine layer to encode the genetic information. We apply it at the frame level so that EMV can extract better style information even with a short speech sample. Then we temporarily average self-attention output to get a one-way style vector *emb*. Thus this module will generate a vector representing the Mel-

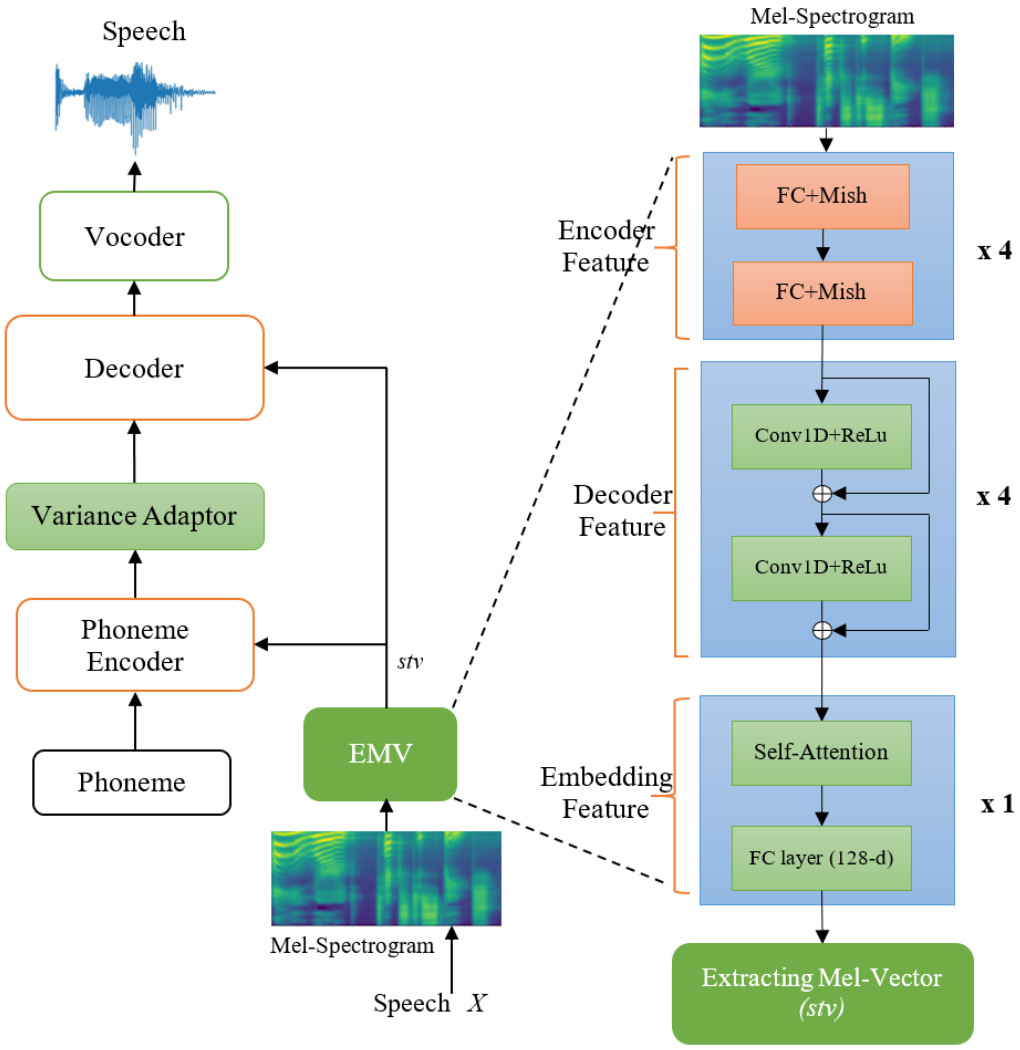

Figure 4: The detailed structure of the EMV module

spectrogram, and this vector will add to the text-to-speech model. The representation vector will drive the output of the TTS model and produce a synthetic voice similar to the input vector. Architectural details of EMV are shown in Table 1 and Fig. 4 respectively.

3.3. Mel-spectrogram denoiser

The decoder block takes input from the hidden phoneme sequence through the variance adapter to add variance information (e.g., duration, pitch, and energy) and then combines it with the EMV vector (representing human features). Then, Mel-spectrogram-denoiser module will take as input sequence x_t, text c, and time step t to perform high-quality audio denoising and synthesis based on the diffusion model. The inference process of the diffusion model for multi-speaker TTS will optimize the objective function $f_\theta(x_t|t, c)$ to convert the noise distributions into a mel-spectrogram distribution corresponding to the given text and the model. It includes two main processes:

Diffusion process. First, the mel-spectrogram is gradually corrupted with Gaussian noise and transformed into latent variables. This process is called the diffusion process. Assuming a sequence of variables x_1,..., x_T. with equal dimensions, where $t = 0, 1, ..., T$ is the index for diffusion time steps, the diffusion process transforms the mel-spectrogram x_0 into Gaussian noise x_T through a chain of Markov transitions. Each transition step is defined by a predetermined variance schedule β_1,β_2,..., β_T. Specifically, each transformation is performed using the Markov transition probability $q(x_t|x_{t-1},\ c)$, which is assumed to be independent of the text c and is defined as follows

$$q(x_t|x_{t-1}, c) = N(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{4}$$

The entire diffusion process $q(x_{1:T}|x_0, c)$ is a Markov process and can be analyzed as follows

$$q(x_1..., x_T|x_0, c) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{5}$$

Reverse process. The reverse process for generating a mel-spectrogram is the opposite of the diffusion process. Rather than introducing noise, the goal of the reverse process is to recover a mel-spectrogram from Gaussian noise. This process is defined by the conditional distribution $p_\theta(x_{0:T-1}|x_T, c)$ and can be decomposed into multiple transitions based on the Markov chain property

$$p_\theta(x_0..., x_{T-1}|x_T,\ c) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t,\ c). \tag{6}$$

Using the reverse transitions $p_\theta(x_{t-1}|x_t,\ c)$, the latent variables gradually reconstruct a mel-spectrogram corresponding to the diffusion time-step with the text condition. Mel-spectrogram denoiser thus learns a model distribution $p_\theta(x_0|c)$ via the reverse process. Let $q(x_0|c)$ be the mel-spectrogram distribution. To achieve a good approximation of $q(x_0|c)$, the reverse process aims to maximize the log-likelihood of the mel-spectrogram, $E_{\log q(x_0|c}[\log p_\theta(x_0|c)]$. As $p_\theta(x_0|c)$ is intractable, we use the parameterization trick demonstrated in [17] to calculate the variational lower bound of the log-likelihood in a closed form. Set $\alpha = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{T} \alpha_s$. The training objective of the mel-spectrogram denoiser is as follows

$$minL_\theta = E_{x_0,\epsilon,t}\Big[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_s x_0} + \sqrt{1-\overline{\alpha}_s}\epsilon, t, c||_1)\Big], \tag{7}$$

where t is uniformly taken from the entire diffusion time step. Mel-spectrogram denoiser only requires the L1 loss function between the model output $\epsilon_\theta(.)$ and Gaussian noise $\epsilon \sim N(0, I)$, without any auxiliary losses. In the inference phase, mel-spectrogram denoiser recovers a mel-spectrogram from a latent variable by iteratively predicting the diffusing noise added at each forward transition using $\epsilon_\theta(x_t, t, c)$ and then removing the corrupted portion in the following manner

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_s}}\epsilon_\theta(x_t, t, c)) + \delta_t z_t, \tag{8}$$

where $z_t \sim N(0, I)$ and $\delta_t = \eta\sqrt{\frac{1-\overline{\alpha}_{s-1}}{1-\overline{\alpha}_s}\beta_t}$. The scaling factor of the variance is represented by the temperature term η. In mel-spectrogram denoiser, the diffusion time-step t is used

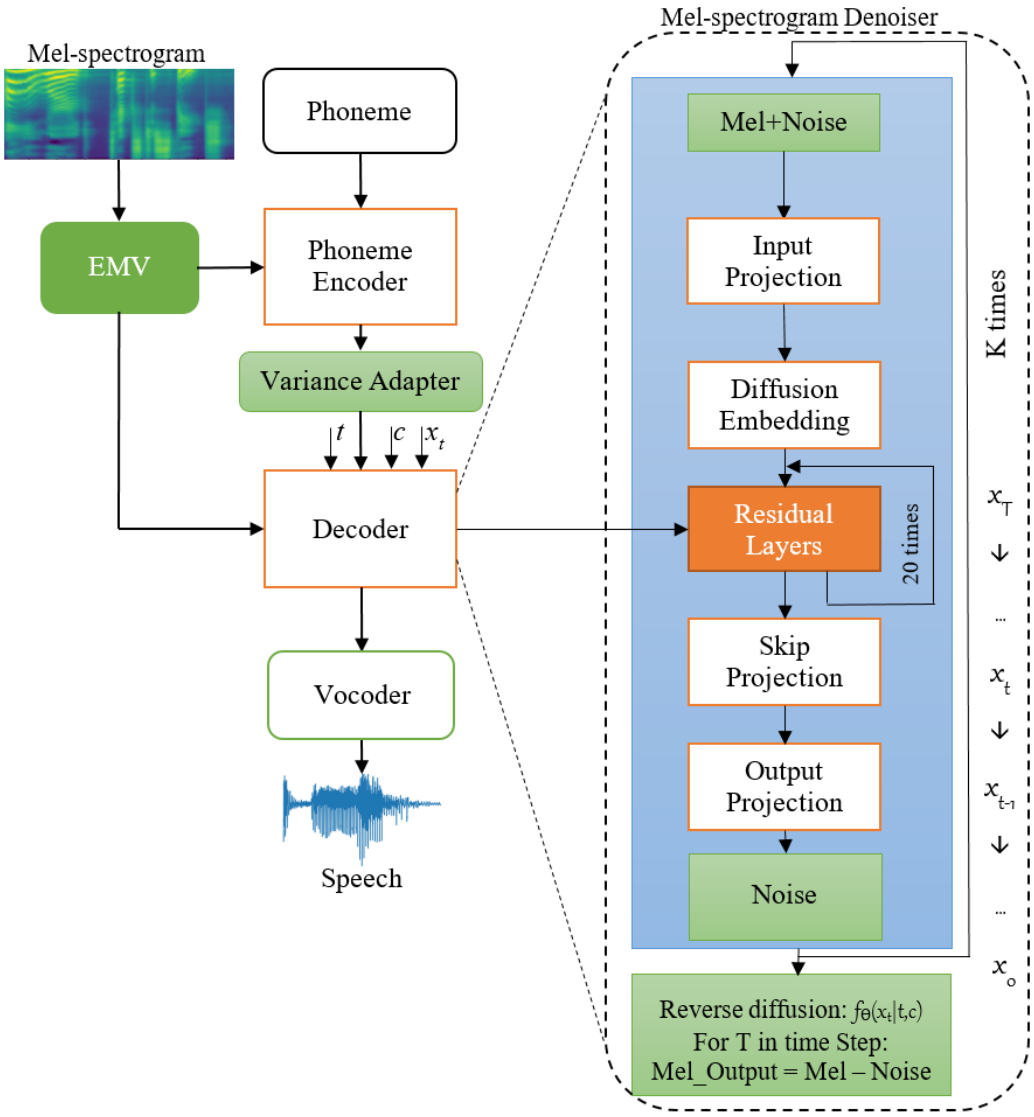

Figure 5: Detailed architecture of the Mel-spectrogram denoiser block

as input, allowing for shared parameters across all time-steps. This enables the iterative sampling over all preset time steps, ultimately resulting in the distribution $p(x_0|c)$ for the final mel-spectrogram.

Brief of training and inference

Training. Besides the sample reconstruction loss described above, to assess the quality of the predicted output in terms of pitch, energy, and duration, the loss values of the variation information are computed using the mean squared error (MSE) metric with respect to the ground truth. Additionally, to evaluate the similarity of the predicted mel-spectrogram to the actual audio, the loss is calculated using the mean absolute error (MAE) and structural similarity index measure (SSIM), which provide a measure of audio fidelity. The final loss

value during Mel-spectrogram denoiser training includes the following parts

$$L_{final} = L_\theta + L_{SSIM} + L_{duration} + L_{pitch} + L_{energy}. \tag{9}$$

1. L_θ (sample reconstruction loss): MSE mean square error between predicted and target mel-spectrogram sample;
2. L_{SSIM} (structural similarity index measure loss - SSIM): One minus the SSIM index between the predicted and target mel-spectrogram sample;
3. $L_{duration}, L_{pitch}, L_{energy}$ (variance reconstructs loss): Mean squared error between duration of syllables, pitch, and energy of prediction sample and target.

Inference: During inference, the mel-spectrogram denoiser predicts the input x_0 without noise and then re-adds the noise using the posterior distribution, thereby generating mel-spectrogram planes with increasing details. Specifically, the denoising model $f_\theta(x_t, t, c)$ first predicts x_t, then x_{t-1} is sampled using the posterior distribution $q(x_{t-1}|x_t, x_0)$ given by x_t and predicts x_{t-1}. Finally, a pre-trained vocoder converts the spectrogram plane generated from x_0 to a waveform.

4. EXPERIMENTS AND RESULTS

4.1. Experiments

Dataset. To evaluate the model, a labeled multi-speaker dataset of Vietnamese language was utilized. The dataset comprised 54 speakers, with 26 male and 28 female voices. The dataset also included both Northern and Southern dialects, with each speaker recording approximately 500 utterances. To evaluate the quality of the synthesized sound generated from the proposed models, we prepared 5 sets of data of Vietnamese: of which 4 sets were synthesized from audio references with durations of 1 second, 3 seconds, and 5 seconds, respectively, and 1 set includes the ground-truth audios for matching.

Evaluate results. We will use two models to synthesize: 1) Baseline model proposed by studies [2], and 2) Adapt-TTS model. We use 30 listeners who are Vietnamese officials, teachers, and students studying and working at universities in Vietnam to listen to and grade the sounds provided through a web-based assessment application. We evaluate the integrated system by combining the evaluation both by objective assessment method (Subjective using quantitative indicators such as WER) and subjective assessment (Objective using qualitative indicators) such as MOS/SIM).

Experiment 1: Assess the quality of speech synthesis MOS (mean opinion score) index evaluates audio or video quality based on human judgment. We conducted the MOS assessment by asking a group of listeners and rating their satisfaction with the sound quality synthesized by the models on a scale of 1 to 5. This scale includes 5 ratings as: 5: Excellent; 4: Good; 3: Medium; 2: Poor; 1: Bad. The results of the MOS are calculated by averaging the scores of all the reviews. To ensure objectivity, we also mix ground-truth sounds to determine the maximum scale for the speaker's voice. We also use the WER(word-error-rate) index to measure the percentage of misrecognized words in the synthetic audio word recognition text compared to the ground-truth audio recognition text. This WER index provides additional quality assessment information through the speech-to-text recognition capabilities of existing ASR systems [23].

Table 1: MOS/WER composite quality assessment results of baseline and proposed models with 95% confidence intervals.

Times/Models	Baseline		Adapt-TTS	
	MOS(↑)	WER(↓)	MOS(↑)	WER(↓)
Groundtruth	4.53	1.35	4.53	1.35
1 second	2.05	8.78	2.89	**3.38**
3 seconds	2.16	7.77	**3.29**	3.14
5 seconds	2.18	6.76	3.31	3.04

Experiment 2. Evaluate the similarity between the synthesized voice and the human voice: We use the SIM (similarity) index to measure the similarity between the synthesized and ground-truth audio of the target speaker. We ask listeners to listen and score the similarity synthesized by the models and the ground truth. These assessments are through listening to the corresponding pairs of sounds using the 4-scale similarity score based on the query and category suggested in [24]. This scale includes 4(four) ratings: 4. Definitely the same; 3. Maybe the same; 2. Maybe different; 1. Definitely different.

4.2. Results

4.2.1. Quality

Table 2 shows that with only 3 seconds of adaptation audio from the reference speaker, the Adapt-TTS model synthesized audio with a MOS score of 3.29 compared to 4.53 of the human voice without requiring retraining. This score is higher than the Baseline model's score of 2.16. The WER score also shows that with only 1 second of reference speaker audio, the system was able to synthesize audio with a WER of 3.38.

4.2.2. Similarity

Table 3 demonstrates that adapt-TTS achieved a SIM score of 2.22 compared to 3.9 of the speaker's voice with only 3 seconds of adaptation audio from the reference speaker. On the other hand, the baseline model only obtained a SIM score of 1.24. Moreover, by comparing the spectrogram in Fig. 6 with 3 seconds of adaptation samples, it can be seen that the mel-spectrogram image (highlighted in the rectangular box) between the audio generated by adapt-TTS and ground-truth has a significantly higher similarity than the audio produced by the baseline model. Additionally, the audio generated by the baseline model is blurry and contains a lot of noise.

Table 2: SIM similarity assessment results of baseline and proposed models with 95% confidence intervals.

Duration/Model	Baseline	Adapt-TTS
Groundtruth	3.90	3.90
1 second	1.16	1.71
3 seconds	1.24	**2.22**
5 seconds	1.31	2.6

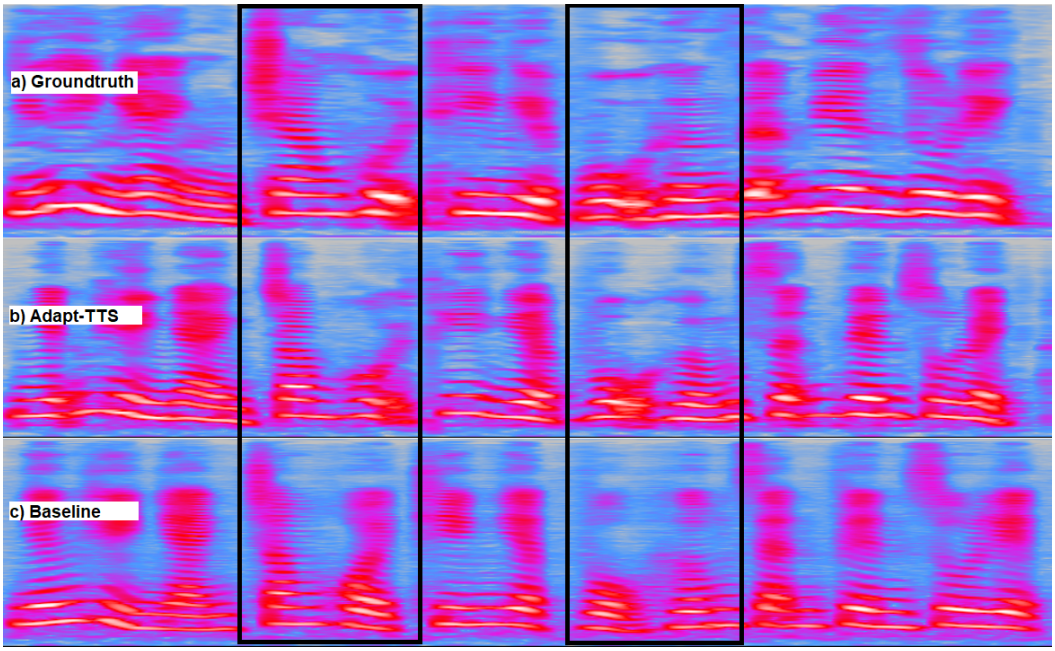

Figure 6: Mel spectrogram of 3 audio: a) Ground truth b) Audio generated by adapt-TTS and c) Audio generated by baseline model.

In order to gain a deeper understanding of the effectiveness of the adapt-TTS model, we illustrate the EMV vectors through the visualization method by computing the distance matrix between the data points of the synthesized audio and the human voice. Figure 7 presents the t-SNE [25] projection of EMV vectors obtained from unseen speakers in the Vietnamese multi-speaker dataset; Specifically, we chose 10 speakers (5 male and 5 female). Adapt-TTS shows an improved separation of the style vectors compared to the baseline model. The t-SNE chart by the adapt-TTS model (Fig. 7 a) shows that the synthesized and original sounds of the same speaker tend to cluster closely together. Gender characteristics are also clearly clustered in 2 different regions.

5. CONCLUSION

The article proposes an architecture that allows synthesizing a new voice using zero-shot speaker adaptation with only one utterance of the reference speaker without requiring retraining. The proposed approach utilizes EMV for better feature and speaking style representation and mel-spectrogram denoiser for synthesizing higher quality and less noisy speech. The experiments demonstrate that a single 1-3 second sample of the reference speaker's voice is sufficient to synthesize a voice with a MOS of 3.3/4.5 and a similarity score of 2.2/3.9. Although the sound quality produced by the proposed zero-shot multi-speaker TTS model cannot match or replace traditionally trained models. However, it allows for quick learning of new voices without retraining while maintaining acceptable sound quality and achieving high similarity with the target voice. The adapt-TTS model works well at cloning speakers with only a short sentence (several seconds), but if the sample data is increased significantly, the quality and similarity of the voice do not change much. The adapt-TTS model proposed

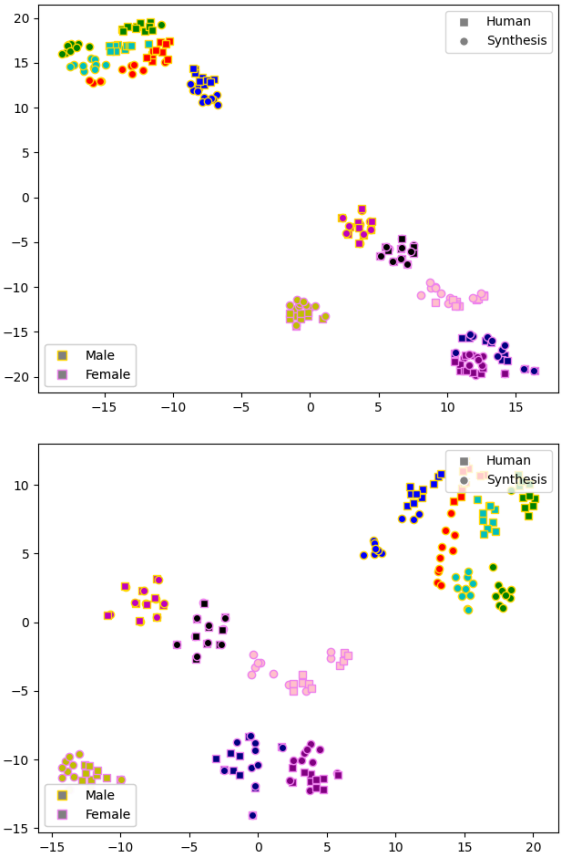

Figure 7: Modeling the spatial distribution of t-SNE between the synthesized voice of the proposed model on the human voice by 10 speakers by a) Adapt-TTS model and b) Baseline model.

in the article enables adaptive speech synthesis with the potential for diverse applications in daily life.

ACKNOWLEDGMENT

The authors would like to thank AIMED Co, ltd for funding this research. This work also is supported by the National Science Project under number KC4.0/19-25.

REFERENCES

[1] J. Shen et al., "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 4779-4783. Doi: 10.1109/ICASSP.2018.8461368.

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

[3] Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerrv-Ryan R, Saurous RA, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *International Conference on Machine Learning*, pp.5530–5540, 2021. PMLR.

[4] E. Cooper et al., "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 6184-6188. Doi: 10.1109/ICASSP40776.2020.9054535.

[5] Y. Wu, X. Tan, B. Li, L. He, S. Zhao, R. Song, T. Qin, T-Y. Liu, "Adaspeech 4: Adaptive text to speech in zero-shot scenarios," *arXiv preprint arXiv:2204.00436*, 2022.

[6] E. Cooper, C.I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Exploring transfer learning for low resource emotional TTS," *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), vol 1*, pp.52–60, 2020. Springer International Publishing.

[7] Q. Xie et al., "The multi-speaker multi-style voice cloning challenge 2021," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 8613-8617. Doi: 10.1109/ICASSP39728.2021.9414001.

[8] S. Arik, J. Chen, K. Peng, W. Ping, Y. Zhou, "Neural voice cloning with a few samples," *Advances in Neural Information Processing Systems (NeurIPS 2018),* vol 31, 2018.

[9] F. Pourpanah et al., "A review of generalized zero-shot learning methods," *in IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 4, pp. 4051-4070, 1 April 2023. Doi: 10.1109/TPAMI.2022.3191696.

[10] W. Ping, K. Peng, A. Gibiansky, S.O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *Proceedings 6th International Conference on Learning Representations (ICLR)*, pp.214–217, 2018.

[11] D. Min, D.B. Lee, E. Yang, S.J. Hwang, "Meta-stylespeech: Multi-speaker adaptive text-to-speech generation," *Proceedings of Machine Learning Research*, pp.7748–7759, 2021.

[12] J. Liu, C. Li, Y. Ren, F. Chen, Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol 36, no.10, pp.11020–11028, 2022. https://doi.org/10.1609/aaai.v36i10.21350

[13] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, Lopez I. Moreno, Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *Advances in Neural Information Processing Systems*, vol 31, pp.6184–6188, 2018.

[14] Y. Wang, D. Stanton, Y. Zhang, R.S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, R.A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *Proceedings of Machine Learning Research,* vol. 80, pp.5180–5189, 2018.

[15] S. Choi, S. Han, D. Kim, S. Ha, "Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding," *arXiv preprint arXiv:2005.08484*, 2020.

[16] E. Casanova, C. Shulby, E. Gölge, N.M. Müller, De F.S. Oliveira, A.C. Junior, A.D. Soares, S.M. Aluisio, M.A. Ponti, "SC-glowtts: An efficient zero-shot multi-speaker text-to-speech model," *arXiv preprint arXiv:2104.05557*, 2021.

[17] J. Ho, A. Jain, P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol 33 pp.6840–6851, 2020.

[18] A.Q. Nichol, P. Dhariwal, "Improved denoising diffusion probabilistic models," *Proceedings of Machine Learning Research*, vol. 139, pp.8162–8171, 2021.

[19] S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen, and H.-Y. Lee, "Meta-TTS: Meta-learning for few-shot speaker adaptive text-to-speech," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 30, pp. 1558-1571, 2022. Doi: 10.1109/TASLP.2022.3167258.

[20] Y. Liu, L. He, J. Liu, M.T. Johnson, "Introducing phonetic information to speaker embedding for speaker verification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol 2019, pp.1–17, 2019. https://doi.org/10.1186/s13636-019-0166-8

[21] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, USA, 2016, pp. 165-170. Doi: 10.1109/SLT.2016.7846260.

[22] Y. Kwon, J.W. Jung, H.S. Heo, Y.J. Kim, B.J. Lee, J.S. Chung, "Adapting speaker embeddings for speaker diarisation," *arXiv preprint arXiv:2104.02879*, 2021.

[23] S. Schneider, A. Baevski, R. Collobert, M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[24] M. Wester, Z. Wu, J. Yamagishi, "Analysis of the Voice Conversion Challenge 2016 Evaluation Results," *Interspeech*, pp.1637–1641, 2016.

[25] G. Hinton, van der L. Maaten, "Visualizing data using t-SNE," *Journal of Machine Learning Research,* vol. 9, pp. 2579–2605, 2008.

Received on March 01, 2023
Accepted on May 07, 2023