IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 November 2024, accepted 6 December 2024, date of publication 11 December 2024,
date of current version 23 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3515206

RESEARCH ARTICLE

Leveraging Low-Rank Adaptation for Parameter-Efficient Fine-Tuning in Multi-Speaker Adaptive Text-to-Speech Synthesis

CHANGI HONG[1], (Graduate Student Member, IEEE), JUNG HYUK LEE[2], (Graduate Student Member, IEEE), AND HONG KOOK KIM[1,2,3], (Senior Member, IEEE)

[1]AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[2]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
[3]AunionAI Company Ltd., Gwangju 61005, Republic of Korea

Corresponding author: Hong Kook Kim (hongkook@gist.ac.kr)

This work was supported in part by Gwangju Institute of Science and Technology (GIST)-Massachusetts Institute of Technology (MIT) Research Collaboration Grant; and in part by the "Practical Research and Development Support Program supervised by the GIST Technology Institute (GTI)" Grant funded by the GIST, in 2024.

ABSTRACT Text-to-speech (TTS) technology is commonly used to generate personalized voices for new speakers. Despite considerable progress in TTS technology, personal voice synthesis remains problematic in achieving high-quality custom voices. In addressing this issue, fine-tuning a TTS model is a popular approach. However, it must be applied once for every new speaker, which results in both time-consuming model training and excessive storage of the TTS model parameters. Therefore, to support a large number of new speakers, a parameter-efficient fine-tuning (PEFT) approach must be used instead of full fine-tuning, as well as an approach to accommodate multiple speakers with a small number of parameters. To this end, this work first incorporates a low-rank adaptation-based fine-tuning method for variational inference with adversarial learning for end-to-end TTS (VITS) model. Next, the approach is extended with conditional layer normalization for multi-speaker fine-tuning, and the residual adapter is further applied to the text encoder outputs of the VITS model to improve the intelligibility and naturalness of the speech quality of personalized speech. The performance of the fine-tuned TTS models with different combinations of fine-tuning modules is evaluated using the Libri-TTS-100, VCTK, and Common Voice datasets, as well as a Korean multi-speaker dataset. Objective and subjective quality comparisons reveal that the proposed approach achieves speech quality comparable to that of a fully fine-tuned model, with around a 90% reduction in the number of model parameters.

INDEX TERMS Text-to-speech synthesis, low-rank adaptation, multi-speaker adaptation, parameter-efficient fine-tuning, residual adapter, conditional layer normalization, variational inference with adversarial learning.

I. INTRODUCTION

Text-to-speech (TTS) technology synthesizes speech waveforms from input texts through several processes, including text analysis, linguistic feature extraction, acoustic feature prediction, and waveform generation [1]. With recent advances in deep learning, TTS models have significantly improved the quality of synthesized speech compared with traditional statistical parametric models. At present, they can generate natural and human-level quality speech after being trained for several hours on single-speaker or multi-speaker recordings [2], [3], [4], [5]. This advancement has made TTS technology attractive across diverse speech-related applications.

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea De Marcellis.

Recently, there has been a growing interest in using TTS for personalized voice assistants and broadcasting [6] with personalized custom voices. In these applications, generating personalized voices for new speakers not included in training data presents a challenge. This challenge arises because of the quality gap between the synthesized speeches of a trained speaker and those of a new speaker. This gap is often caused by factors such as the lack of training data for the new speaker or characteristics of the speaker that do not match the data used during training [7]. To address the issue associated with such a quality gap, speaker adaptation techniques have been applied to better adapt to new speakers not included in training data.

Two main methods are being studied for generating speech for new speakers: speaker adaptation methods based on zero-shot learning [8] or fine-tuning a pretrained TTS model to personalize the natural voices of new speakers [9], [10]. Zero-shot learning utilizes a single pretrained model to imitate unseen speech patterns and features. Significant advances have been achieved by applying zero-shot learning to TTS [11], [12], [13], [14]. However, the zero-shot approach generates a relatively inconsistent personalized voice with distorted naturalness for a given new speaker. Additionally, when the speakers pronounce strong accents or nonstandard pronunciations, the similarity of the synthesized speech further decreases [15]. In contrast, the fine-tuning approach generally adapts a pretrained TTS model by optimizing all the parameters of the TTS model using a limited amount of new-speaker data.

Although adapting the TTS model for a target speaker can improve the synthesized speech quality, several problems arise. First, fine-tuning all the parameters of the TTS model incurs significant computational cost and time consumption [6]. Second, the adapted TTS model for each target speaker needs to be stored individually, which requires considerable storage space [16], [17]. Therefore, reducing the number of adaptation parameters is necessary for fine-tuning.

To mitigate the abovementioned problems, parameter-efficient fine-tuning (PEFT) approaches have been proposed. For instance, AdaSpeech leverages acoustic condition modeling and conditional layer normalization (CLN) at the mel-decoder stage to achieve parameter efficiency while fine-tuning TTS models [6]. Meanwhile, Meta-StyleSpeech [18] employs metalearning techniques for style modeling, enabling fast adaptation to a new speaker's style with minimal data. Furthermore, adapter-based methods have been introduced as PEFT [19], [20], [21], and they achieve efficiency by selectively fine-tuning only a subset of parameters rather than the entire model, thereby reducing the computational load and storage requirements. However, these approaches have typically focused on fine-tuning the acoustic models of two-stage TTS models [4], [22], [23]. Because acoustic feature representation and waveform synthesis in two-stage TTS models are processed independently, the TTS performance is limited because of the independence of the fine-tuned intermediate features [24].

In recent years, end-to-end (E2E) TTS models have been widely studied to provide higher-quality expression compared with two-stage TTS models. One representative E2E TTS model is variational inference with adversarial learning for E2E TTS (VITS) model [24], which mainly comprises a variational autoencoder (VAE) augmented with normalizing flow (NF) [25], [26] and is trained through adversarial training [27]. Another notable E2E model is Your-TTS [11], which builds upon the VITS framework and incorporates a speaker encoder for zero-shot multi-speaker adaptation and multilingual training. Additionally, NaturalSpeech [12] achieves high-quality single-speaker TTS by modifying the VITS model structure, introducing a bidirectional NF alongside differentiable duration modeling and phoneme pretraining, which significantly enhances the synthesized speech's expressiveness and naturalness. However, an issue persists when PEFT is applied to these VITS-based models. The connection between the modules in the VITS-based model is represented by a probability distribution. Thus, applying PEFT to a specific module in the VITS model can change the probability distribution of the output of the module. However, whether this updated probability distribution is suitable for the input of the subsequent module is uncertain. Without more sophisticated fine-tuning, high-quality synthesized speech cannot be guaranteed.

To address this issue, a recent study [15] proposed a zero-shot learning and PEFT method for VITS-based models, which improved the zero-shot adaptation performance by altering the VAE model structure to prevent overfitting and introducing a specific discriminator for speaker information, thereby enhancing the overall model performance. In addition, the speaker encoder was based on the ECAPA-TDNN architecture [28], which was modified to extract speaker embeddings and pretrained to effectively capture speaker characteristics. In this model [15], the baseline TTS model was trained using speaker embeddings extracted from the pretrained speaker encoder to aid the model's flow and duration predictor during training. PEFT was applied through adapters to the prior encoder, specifically targeting the flow-based decoder and text encoder. This approach demonstrated impressive performance in speaker adaptation. However, this method relied on a pretrained speaker encoder, did not consider multi-speaker adaptation, and only applied the adapter to the prior encoder.

Thus, this paper presents a PEFT approach in VITS models and demonstrates the effectiveness of applying PEFT to multiple specific modules within the E2E architecture, providing a new method for improving TTS performance for multi-speaker adaptation. To further enhance this approach, we propose three specific strategies to realize PEFT for the VITS model. First, we incorporate low-rank adaptation (LoRA) [29] for fine-tuning the VITS model. LoRA is a method for reducing the complexity of neural network

parameters by decomposing them into lower-dimensional representations [30]. Consequently, it adapts only a subset of model parameters using a low-rank matrix rather than the entire model parameters. In this study, LoRA is applied to several modules: the attention network of the text encoder, the WaveNet [2] structure in both the flow network and posterior encoder, the HiFi-GAN generator [31], and two linear projection layers. Second, LoRA-based fine-tuning is expanded with CLN [6] for multi-speaker fine-tuning. This is because LoRA alone does not capture diverse speaker-specific variations, resulting in suboptimal performance in multi-speaker adaptation. CLN uses a small conditioning layer to obtain scale and bias vectors for normalization instead of standard layer normalization, and it is applied to the text encoder and the stochastic duration predictor (SDP) of the VITS model by replacing layer normalization (LN). Lastly, to achieve intelligibility and naturalness of speech quality as in full fine-tuning, the degree of expressiveness of the prior distribution should be increased [12]. Therefore, this work additionally applies the modified version of the residual adapter [22], [32], [33], which can be flexibly inserted into the output of any module. In our model, we inserted the residual adapter into the text encoder outputs of the VITS model to enhance the representation of the prior distribution of the text encoder output.

We conducted experimental evaluations on the widely adopted multi-speaker VCTK [34] and Libri-TTS-100 [35] datasets to measure the voice quality of the proposed fine-tuning method against several objective and subjective metrics. These datasets were chosen to test the robustness of our process to different data characteristics. The VCTK dataset was characterized by many audio samples per speaker and a generally calm and consistent tone of voice. In contrast, the Libri-TTS-100 dataset comprised significantly more speakers despite similar sample numbers, with variations in the tone of each speaker. Because VCTK and Libri-TTS-100 were composed of controlled and stable speeches, we repeated experiments using the Common Voice datasets [36] to evaluate the performance of the proposed PEFT method under various accent conditions, which was essential for building personalized custom voices. Moreover, we conducted additional experiments using a Korean multi-speaker dataset to further investigate the model's adaptability to different languages. Using these datasets, we verified the performance of our multi-speaker fine-tuning method with four speakers. The speech performances of different models, where fine-tuned TTS models were evaluated according to different combinations of fine-tuning modules (e.g., LoRA, CLN, and residual adapter), were compared in terms of the number of tuning parameters and speech quality measures. To measure speech quality, we used five objective metrics: speaker embedding cosine similarity (SECS) [37], word error rate (WER), character error rate (CER) [38], nonintrusive objective speech quality assessment for TTS (NISQA-TTS) [39], and mean opinion score (MOS) prediction by a fine-tuned wave2vec2.0 model (WV-MOS) [40]. In addition, to measure reliable TTS perception quality in terms of human-level quality, we used a comparative mean opinion score (CMOS) as a subjective metric [41].

The main contributions of this study are as follows:

- To implement PEFT in the VITS model, we applied LoRA to the prior encoder and other specific modules within the E2E model, achieving speech quality comparable to that of a fully fine-tuned model with a 90% reduction in model parameters.
- To handle speaker-specific variation with improved multi-speaker PEFT performance, CLN replaced the LN in the text encoder and the SDP, allowing the model to train an additional speaker with only 0.02M parameters.
- To improve the expressiveness of the prior distribution, the residual adapter was integrated into the text encoder output. With only 0.15M parameters, this integration improved the WER, CER, and NISQA-TTS scores.

The remainder of this paper is organized as follows. Section II provides helpful background knowledge to help readers understand our work. Section III describes the VITS model architecture used as the baseline TTS model. Section IV proposes the PEFT method using LoRA, CLN, and residual adapter for multi-speaker adaptation. Section V evaluates the performance of the VITS models with the proposed PEFT, including several ablation studies and visualization experiments. Finally, Section VI concludes the paper.

II. BACKGROUND

This section provides helpful background knowledge to help readers understand our work. First, we give a general review of TTS models. Next, we explain the flow-based generative models used in TTS systems.

A. OVERVIEW OF TEXT-TO-SPEECH MODELS

Recently, neural TTS systems have made significant advances in terms of performance. Two-stage TTS structures are commonly used to generate speech. These systems use acoustic models to predict predetermined acoustic features, such as mel-spectrograms, and then synthesize waveforms using a vocoder [31], [42]. When predicting these acoustic features, acoustic models can be categorized into two groups: autoregressive (AR) and non-autoregressive (NAR) TTS systems. Typically, sequence-to-sequence AR-TTS systems include models such as WaveNet [2] and Tacotron1, 2 [3], [22]. Transformer TTS [23] is the first model to use a transformer network in TTS. These AR-TTS systems sequentially generate frames of a mel-spectrogram by relying on the previous frame to effectively capture long-term dependencies. However, such a system can lead to a compromise in terms of inference speed and robustness errors, such as missing words and repetition. Thus, NAR-TTS systems have been developed to address these problems. For instance, FastSpeech [43] overcomes problems such as repetition in AR-TTS and parallelizes the process with a duration predictor to improve the speed and robustness of speech synthesis. FastSpeech2

[4] refines this setup using a variance adaptor for pitch and energy, although it still depends on an external text and speech alignment tool. Meanwhile, Glow-TTS [5] advances the field by learning alignment directly during training using monotonic alignment search (MAS).

Despite the progress in NAR-TTS systems, the above-mentioned cascaded acoustic/vocoder model pipeline still has problems. In two-stage models, the latter model is trained on samples generated by earlier models or leverages pretrained models without modification. In addition, fine-tuning for high-quality speech synthesis is problematic because the two models must be trained separately. Furthermore, training–inference mismatches occur for both the mel-spectrogram and the duration as the models are trained with ground-truth values but rely on predicted values during inference. High-quality speech synthesis requires fine-tuning. Because of this problem, E2E models utilizing efficient training methods have been widely studied [44], [45]. Among these models, VITS [24] has succeeded in producing more natural speech than two-stage models by integrating the TTS model and a neural vocoder within an E2E framework using a VAE to enhance the synthetic speech quality. Moreover, VITS addresses the one-to-many problem of TTS by employing an SDP, enabling the generation of varied rhythms. Consequently, there have been widely adopted E2E models based on the VITS architecture [11], [12], [46].

B. FLOW-BASED GENERATIVE MODEL

Flow-based models are increasingly being used in different models because of their ability to compute the exact likelihood of data by applying inverse transformations [47]. To estimate the exact density, the latent variable of a generative model should be as simple as a Gaussian distribution. This leads to NF [25], which transforms a simple distribution into a complex distribution by applying a sequence of invertible transformations. This transform is iteratively replaced by changing the following variables:

$$log\, p_\theta\,(c) = log\, p_\theta\,(z) + \sum_{i=1}^{k} log \left| det\left(J\left(f_i^{-1}\,(c)\right)\right)\right|, \quad (1)$$

$$z = f_k^{-1} o f_{k-1}^{-1} o \ldots f_1^{-1}\,(c) \quad (2)$$

where K is the number of layers in the flow-based decoder, o is a composition operator, and $J\,(\cdot)$ is a Jacobian operator.

When implementing NF, two conditions must be satisfied. The Jacobian matrix of the transformation should be easily calculated, and the NF should be able to perform the inverse transformation easily. These requirements have been effectively addressed using the affine coupling layer proposed previously [48], simplifying the Jacobian computation and ensuring invertibility. The affine coupling layer operates by partitioning the input into two parts, transforming one part conditionally according to the other, facilitating the simple calculation of the Jacobian determinant. Additionally, the limitations of the unchanging dimensions of the affine coupling layer have been overcome upon the introduction of the

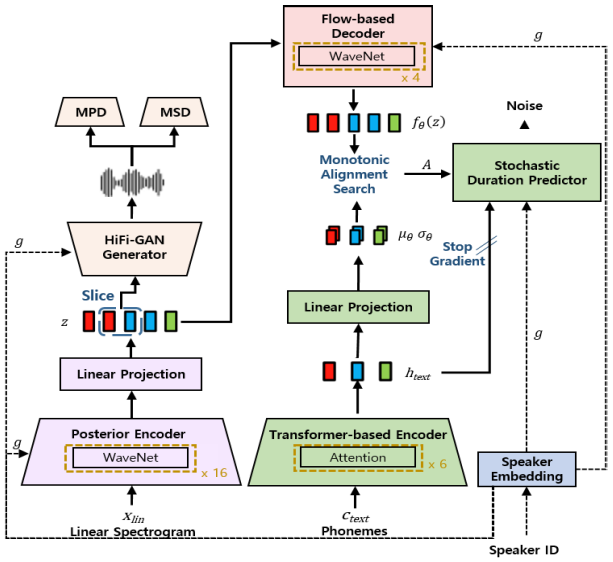

FIGURE 1. Block diagrams of the baseline VITS model.

1×1 invertible convolution method [49], which facilitates feature permutation (mixing between channels) and enhances the model's flexibility.

The WaveGlow model [50] further extended these structures by incorporating the WaveNet architecture and significantly enhancing its capabilities in modeling complex audio signals. The model structure was utilized to compose the baseline model, VITS, incorporating VAE with its NF framework. This integration improved the expressiveness of the prior distribution and significantly improved the quality of speech synthesis by leveraging the ability of flow, allowing the construction of complex probability distributions with a simple distribution. More detailed explanations are provided in Section III.

III. BASELINE TTS MODEL

In this section, we explain the VITS model [24], which is employed as the baseline model in this work, with a focus on the network architecture and training process. VITS is a parallel E2E model that utilizes a VAE to learn latent variables that serve as intermediate representations between the acoustic model and the waveform generator in a fully integrated training process. This integration improves the smooth flow of information from the acoustic model to the waveform generator, resulting in the consistency of the personalized voice quality.

Fig. 1 depicts the training procedure of the baseline VITS model, which comprises three primary components: a prior encoder, a posterior encoder, and a HiFi-GAN generator [31]. The prior encoder comprises a transformer-based text encoder, a flow-based decoder, MAS [5], and an SDP. The text encoder uses multiple feed-forward transformer blocks [51] to transform the input phonemes c_{text} into hidden representations h_{text}. These representations are then

processed by a linear projection layer to generate $f_\theta(c)$ with the mean $\mu_\theta(c)$ and variance $\sigma_\theta^2(c)$, which are used to construct the prior distribution. Here, c is defined as $[c_{\text{text}}, A]$, where A is the alignment between the text c and target speech x, selected from all potential alignments through MAS. In parallel, the SDP is trained using speaker embedding g, the alignment A, and the hidden representation h_{text} of the text encoder output, as depicted in Fig. 1. During inference, the SDP estimates the alignment based on the text. The SDP aligns the text by incorporating two random variables, u and v, into the duration d, in view of variational Bayes estimation. The two random variables are sampled from an approximate posterior distribution of the text to optimize a variational lower bound on the log-likelihood of the phoneme duration. The training loss L_{dur} is the lower bound of the calculated negative variation:

$$\log p_\theta(d|C_{\text{text}}) \geq \mathbb{E}_{q_\emptyset(u,v|d,C_{\text{text}})}\left[\log \frac{p_\theta(d-u, v|C_{\text{text}})}{q_\emptyset(u,v|d,C_{\text{text}})}\right]. \tag{3}$$

The flow-based decoder is constructed by arranging a stack of WaveNet [2] residual blocks in a stack of affine coupling layers [47]. The probability of the latent variables conditioned on the text, $p_\theta(z\,|\,c)$, can be expressed as

$$p_\theta(z|c) = N\left(f_\theta(z); \mu_\theta(c), \sigma_\theta(c)\right)\left|\det \frac{\partial f_\theta(z)}{\partial z}\right| \tag{4}$$

where $f_\theta(z)$ is the output of the flow-based decoder. To calculate the inverse probability, the Jacobian determinant in equation (4) is computed as $\left|\det \frac{\partial f_\theta(z)}{\partial z}\right|$.

The posterior encoder and the Hi-Fi GAN generator, as shown in Fig. 1, correspond to the encoder and decoder of VAE, respectively. The former extracts the latent representation z from the waveform x, whereas the latter generates the reconstructed waveform $\hat{x}$ according to z:

$$z = Enc(x) \sim q(z\,|\,x), \tag{5}$$

$$\hat{x} = Dec(z) \sim p(x\,|\,z). \tag{6}$$

The training loss for a conditioned VAE is derived from the evidence lower bound of the marginal log-likelihood $p_\theta(x|c)$ and maximized as

$$\log p_\theta(x|c) \geq \mathbb{E}_{q_\emptyset(z|x)}\left[\log p_\theta(x|z) - \log \frac{q_\emptyset(z|x)}{p_\theta(z|c)}\right] \tag{7}$$

where $p_\theta(z|c)$ represents the prior distribution of z in equation (4), $q_\emptyset(z|x)$ is an approximate posterior distribution, and $\log p_\theta(x|z)$ is the likelihood function for a data point x. Equation (7) is decomposed into a reconstruction loss measured in the output of the HiFi-GAN and a Kullback–Leibler (KL) divergence loss. The reconstruction loss L_{recon} is defined as the L1 loss between the target and the predicted mel-spectrograms—x_{mel} and $\hat{x}_{\text{mel}}$, respectively—as follows:

$$L_{\text{recon}} = \left\|x_{\text{mel}} - \hat{x}_{\text{mel}}\right\|_1. \tag{8}$$

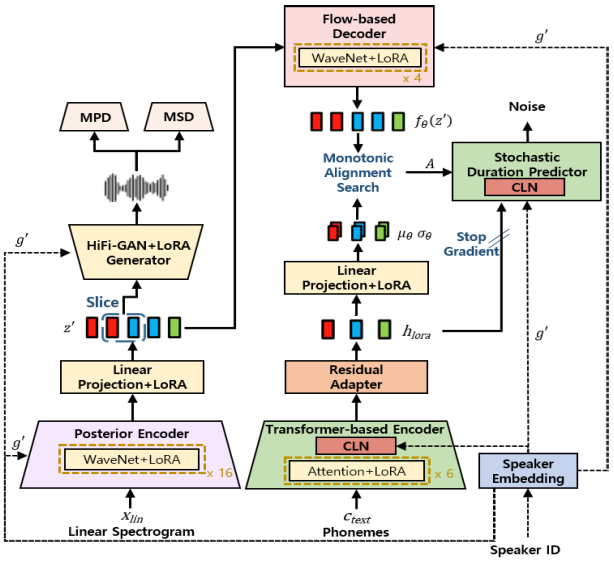

FIGURE 2. Block diagrams of the proposed fine-tuned VITS.

In addition, the KL loss in the latent space is defined using the output of the priority distribution p_θ and the posterior distribution q_ϕ of the baseline model as follows:

$$L_{\text{KL}} = \log q_\phi(z|x_{\text{lin}}) - \log p_\theta(z\,|\,c) \tag{9}$$

where x_{lin} is a linear spectrogram of x, as shown in the bottom left part of Fig. 1.

Finally, the HiFi-GAN generator G synthesizes the predicted speech $\hat{x}$ according to the intermediate representation z. In the VITS framework, G comprises a series of transposed convolutions, each followed by a multi-receptive field fusion module (MRF). The adversarial loss of the HiFi-GAN generator G is defined as

$$L_{\text{adv}}(G) = \mathbb{E}_{(z)}\left[(D(G(z)) - 1)^2\right] \tag{10}$$

where D is the discriminator for GAN, composed of a multi-period discriminator and a multiscale discriminator, as shown in the top right part of Fig. 1, and it is trained using the adversarial loss of

$$L_{\text{adv}}(D) = \mathbb{E}_{(x,z)}\left[(D(x) - 1)^2 + (D(G(z)))^2\right]. \tag{11}$$

In addition to $L_{\text{adv}}(G)$, a feature-matching loss $L_{\text{fm}}(G)$ is used as a reconstruction loss of the discriminator of the HiFi-GAN by summing all the L1 losses between the feature maps extracted from the intermediate layers of each discriminator. Consequently, the total loss function of the VITS model is a combination of VAE and GAN loss, which is configured to facilitate E2E learning; it is expressed as follows:

$$L_{total} = L_{recon} + L_{kl} + L_{dur} + L_{adv}(G) + L_{fm}(G). \tag{12}$$

IV. PROPOSED METHOD

This section proposes three approaches to fine-tuning the baseline VITS model. First, we incorporate LoRA for fine-tuning the VITS model to reduce the complexity of

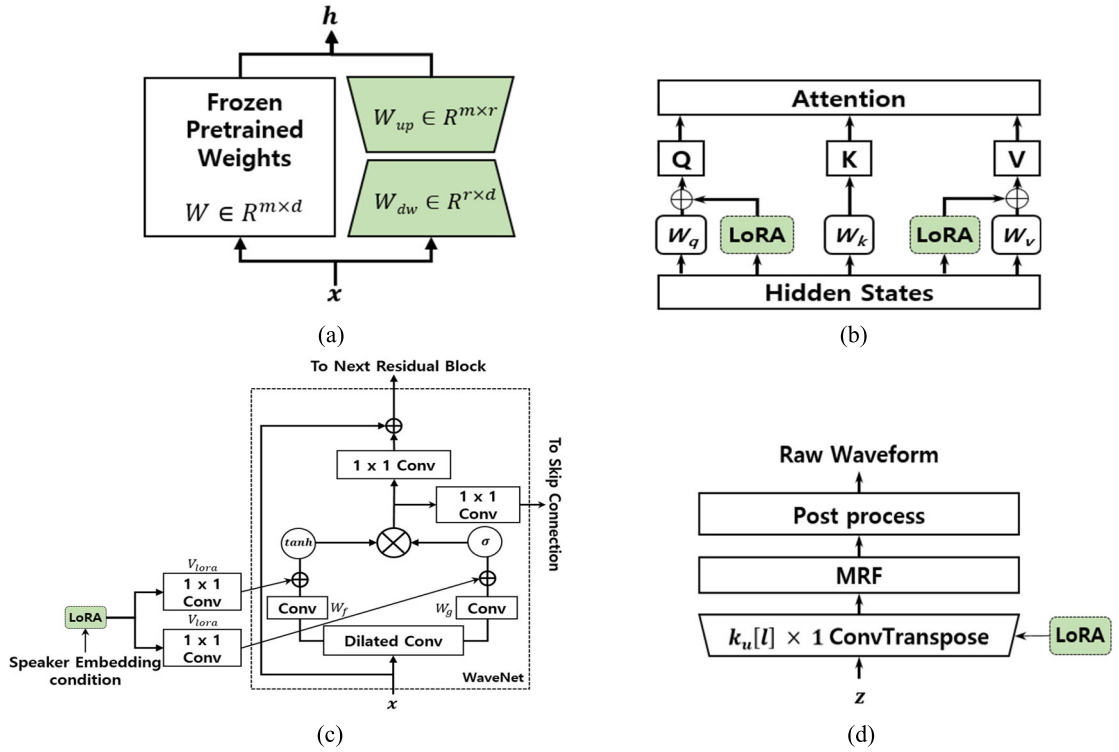

FIGURE 3. Network architectures of (a) LoRA applied to a weight matrix, (b) LoRA applied to the attention matrices in the transformer-based text encoder, (c) LoRA applied to a WaveNet residual block, and (d) LoRA applied to the MRF in the HiFi-GAN generator.

neural network parameters by decomposing them into lower dimensional representations. Second, LoRA-based fine-tuning is expanded with CLN for multi-speaker fine-tuning. Third, we apply the residual adapter to the text encoder outputs of the VITS model, which can enhance the representation of the prior distribution of the text encoder output. Fig. 2 illustrates how the parameter-efficient modules—LoRA, CLN, and residual adapter—are integrated into the VITS architecture, with specific colors used for each module. Compared with Fig. 1, Fig. 2 also indicates that the latent variable changes from z to z' for a given new-speaker embedding g' after applying such modules in the baseline VITS model.

A. REDUCTION OF MODEL PARAMETERS BASED ON LoRA

Instead of optimizing all model parameters, the LoRA-based fine-tuning method optimizes the parameters of the low-rank model [29]. Assuming that the pretrained model parameter is Φ_0, the fine-tuning process involves finding Φ that maximizes$\Phi = \Phi_0 + \Delta\Phi$, where $\Delta\Phi$ is the changed parameter during the fine-tuning. If we can replace $\Delta\Phi$ with a low-rank model, $\Theta\,(\ll \Phi)$, Φ is expressed as $\Phi = \Phi_0 + \Delta\Phi\,(\Theta)$.

Fig. 3(a) illustrates an example of the application of LoRA to a matrix as if it is applied to our fine-tuning process. For a pretrained weight matrix $W \in R^{m\times d}$, its update can be constrained by representing it with a low-rank decomposition $W + \Delta W = W + W_{\text{up}}W_{\text{dw}}$ where$W_{\text{up}} \in R^{m\times r}$ and $W_{\text{dw}} \in R^{r\times d}$ with the rank$r \ll \min(m, d)$. During training,W is frozen and does not receive gradient updates, where W_{up} and W_{dw} contain the trainable parameters. Both W and $\Delta W = W_{\text{up}}W_{\text{dw}}$ are multiplied by the same input, $x \in R^d$, and their respective output vectors $h \in R^m$ can be expressed as follows:

$$h = Wx + \Delta Wx = Wx + W_{up}W_{dw}x. \tag{13}$$

In this study, the LoRA module is integrated into six different modules of the baseline VITS model, as illustrated in Fig. 2. In particular, there are two LoRAs for each linear projection layer and four LoRAs for the attention matrices in the transformer-based text encoder, each of the two WaveNets, and an upsampling layer of the generator. The linear projection layer is an important part of the VITS architecture because it projects the distribution of the posterior and prior encoders. Each layer uses a 192 × 384 matrix to split the 384 output channels and derive the mean and variance. LoRA is applied to the matrix $W \in R^{192\times 384}$ with $W_{\text{up}} \in R^{192\times r}$ and $W_{\text{dw}} \in R^{r\times 384}$, where r is set to 8 throughout this paper according to the setting described previously [29].

In addition to the linear projection layer, Fig. 3(b) shows the network architecture of LoRA applied to the attention matrices in the transformer-based text encoder. As shown in Fig. 3(b), the self-attention module of each transformer block includes four weight matrices: W_{q}, W_{k}, W_{v}, and W_{o}. Of these, W_{q}, W_{k}, and W_{v} have a size of 192 × 192 and

project the input features c_{text}into queries, keys, and values, respectively, to handle the dimensionality of the input and output effectively. Among these four matrices, LoRA is applied to W_{q} and W_{v}, which are crucial for generating queries and values.

As mentioned in Section III, the VITS architecture contains the WaveNet structure in two modules. The posterior encoder comprises 16 noncausal WaveNet residual blocks, whereas the flow-based decoder consists of four affine coupling layer stacks [48], each containing four WaveNet residual blocks. WaveNet fundamentally works through stacked residual blocks, each containing a dilated convolution layer, two activation functions, and a 1×1 convolutional layer, as shown in Fig. 3(c). WaveNet uses the gate activation unit method [52], where each layer consists of a gate, $\sigma(\cdot)$ that looks at a feature of the input value as a filter, $\tanh(\cdot)$ and decides the magnitude of this information to be passed to the next layer. In the VITS model, WaveNet is responsible for embedding conditional information, which is important for generating specific speakers' voices. This is achieved by incorporating a new speaker embedding g' as a global condition within the WaveNet structure. To fine-tune the conditioning part, LoRA is applied to a 1×1 convolution layer, V_{lora}, on each block of the WaveNet that takes the information of g'. The operation of the conditional WaveNet can be formulated as follows:

$$z = tanh\left(W_f * x + V_{lora}g'\right)\sigma(W_g * x + V_{lora}g') \quad (14)$$

where $*$ denotes a convolution operation, $\odot$ represents element-wise multiplication, and x is the input.

Finally, we apply LoRA to the generator whose MRF model structure is described in Fig. 3(d). MRF facilitates the formation of several different receptive field patterns to enrich speech with details and textures. Therefore, MRF fine-tuning is crucial for generating personalized voice; thus, LoRA is applied to the ConvTranspose layer connected to the MRF.

B. CONDITIONAL LAYER NORMALIZATION FOR MULTI-SPEAKER TTS

To handle speaker-specific variations with improved multi-speaker PEFT performance, we incorporate CLN by replacing LN [53] in the text encoder and SDP. Fig. 4(a) shows the conditional network comprising two linear layers: W_γ and W_β. These layers project the extracted speaker representation onto a scale vector γ_s and a bias vector β_b, which are essential components of CLN. Specifically, the speaker embedding vector g' is processed through W_γ and W_β, which are responsible for producing γ_s and β_b, respectively. Consequently, the normalization is defined as

$$\gamma_s = g' \times W_\gamma, \beta_b = g' \times W_\beta. \quad (15)$$

Without CLN, all model parameters for each new speaker must be stored. However, by adjusting the normalization parameters for each speaker, the model can achieve high-quality adaptation during multispeaker optimization

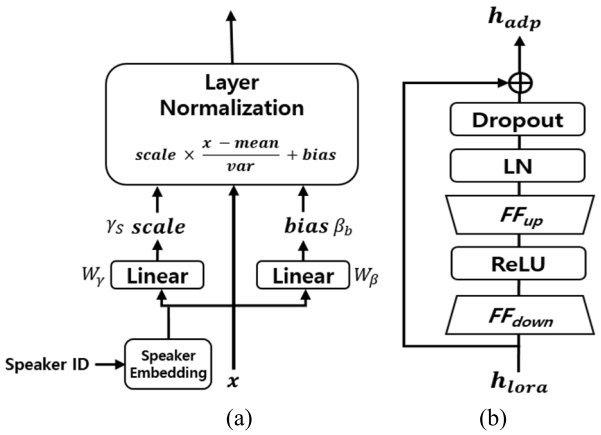

FIGURE 4. Network architecture of the (a) conditional normalization layer and (b) residual adapter used for the proposed fine-tuning approach.

while significantly reducing the number of parameters. Specifically, a storage of only 0.02M parameters for each speaker is required by applying CLN during fine-tuning, which corresponds to ∼0.05% of the model parameters required for full fine-tuning.

C. RESIDUAL ADAPTER FOR EXPRESSIVE TTS

Although the application of LoRA and CNL provided enhanced performance, limitations in naturalness and pronunciation compared with those of full fine-tuning persisted. To address this issue, the expressiveness of the prior distribution of the new-speaker data during fine-tuning must be enhanced [12]. Accordingly, we attempted to increase the rank of the LoRA matrix applied to the text encoder; however, it did not yield performance improvement. Therefore, a residual adapter [22], [32] was integrated into the text encoder output.

Fig. 4(b) shows the network architecture of a residual adapter, a modified version of the vanilla adapter [33], used for the proposed fine-tuning approach. As shown in Fig. 4(b), the residual adapter operates by initially projecting the text encoder output h_{lora} through a down-projection feed-forward network FF_{down}, which reduces the dimensionality to a lower-dimensional bottleneck. A rectified linear unit activation function [54] is then applied to add the nonlinearity to the output of FF_{down}. Next, the dimension is restored by an up-projection feed-forward network FF_{up}. This residual adapter requires only 0.15M parameters.

The adapter incorporates a residual connection to ensure stable training and minimize the disruption to the original model architecture. This connection enables the original input h_{lora} to bypass the adapter and merge with the adapter output. This effectively allows the network to start training from a near-identity state, which is crucial for maintaining the initial performance level. Note that we also incorporate dropout and LN with zero initialization of the final layer to make this residual adapter operate as an identity function. According to

the description thus far, the residual adapter can be described as follows:

$$h_{adp} = h_{lora} + LN\left(ReLU\left(FF_{down}\left(h_{lora}\right)\right) FF_{up}\right). \quad (16)$$

V. EXPERIMENTS AND RESULTS

A. DATASET

We utilized four datasets—VCTK [34], Libri-TTS-100 [35], Common Voice [36], and the Korean Multi-Speaker Speech Synthesis (KMSSS)[1]—to evaluate the performance of the TTS model using the proposed fine-tuning approaches. These datasets were selected for their different characteristics. For instance, the VCTK dataset comprised around 400 sentences spoken by 109 speakers. The audio format was a 16-bit PCM with a sampling rate of 48 kHz. This dataset was characterized by a similar number of speech samples per speaker and low variability in speech. Meanwhile, the Libri-TTS-100 dataset had a similar number of speech samples as VCTK but comprised 247 speakers. The total length of the audio data was approximately 54 h, with a sampling rate of 24 kHz. This dataset had fewer samples per speaker, an inconsistent number and length of speeches per speaker, and more variability in speech. The Common Voice dataset consisted of mono-channel, 16-bit MPEG-3 audio files at a sampling rate of 48 kHz. In this experiment, we organized a subset of 144 English speakers, each with ∼1,000 samples, to ensure balanced data for fine-tuning. Compared with VCTK and Libri-TTS-100, this dataset offered a greater variation in speech, including various accents and dialects, recorded by volunteers from diverse linguistic backgrounds. Lastly, to investigate the model's adaptability to non-English languages, a dataset was constructed from the KMSSS dataset by taking 184 speakers, where each speaker spoke 500 utterances at a sampling rate of 48 kHz.

For multi-speaker fine-tuning, a VITS model was pretrained using 100 speakers from the VCTK dataset, with five speakers for validation and four for fine-tuning and testing. In contrast, we pretrained, validated, and tested the VITS model with 220, 14, and 13 speakers, respectively, for the Libri-TTS-100 dataset, where we selected the four speakers with the highest number of samples in the test data for fine-tuning. For the Common Voice dataset, the VITS model was pretrained using 130 speakers, whereas ten and four speakers were used for validation and testing, respectively. Note that two out of the four speakers recorded speeches in environments with slight background noise, adding diversity to the data. Similarly, for the Korean dataset, we used 160 speakers for training, 20 for validation, and 4 for testing.

B. EXPERIMENTAL SETUP

In our experimental setup, we resampled all the speech data at a sampling rate of 22 kHz. Then, we normalized the raw text sequences and converted the normalized sequences into the International Phonetic Alphabet sequence using an open-source phonemizer[2] [55].

To obtain our pretrained VITS model, we utilized the AdamW optimizer [56] with the hyperparameters set as β_1 =0.8, β_2 =0.99, and the weight decay $\lambda = 0.01$. The learning rate was initially set at 2×10^{-4} and followed a decay schedule of $0.991^{1/8}$. The pretraining phase was conducted over 400 epochs using four NVIDIA A100 graphics processing units (GPUs). In the fine-tuning phase, all hyperparameters were maintained except for the learning rate and batch size, which were adjusted to 1×10^{-5} and 32, respectively. Each fine-tuning process was run for 150 epochs on a single A100 GPU. To evaluate the model performance, we excluded 15 speech samples per speaker from the test dataset and used the remaining data for fine-tuning. We fine-tuned four speakers to evaluate the multi-speaker fine-tuning performance.

C. EVALUATION METRICS

To evaluate the performance of the proposed fine-tuning approaches, we compared the synthesized speech with the reference speech using five objective metrics: SECS, WER, CER, NISQA-TTS[3], and WV-MOS[4].

SECS measured the cosine similarity between the speaker embedding of the synthesized speech and the reference speech audio. This value, which ranged from −1 to 1, indicated how closely the speaker's vocal characteristics match. We computed the speaker embedding using the H/ASP model [37], a publicly available speaker verification model[5] trained on VoxCeleb2 [57], a large-scale speech dataset.

WER (%) and CER (%) respectively indicated the percentages of recognized word and character errors in the synthesized transcript to the ground-truth text. For synthesized speech transcription, we used NeMo's stt_en_conformer_transdu-cer_large_model[6] [38], which was based on the conformer transducer architecture, and computed these error rates using the Levenshtein distance algorithm[7] [58]. A lower value suggests fewer pronunciation errors in the synthesized speech, indicating higher fidelity of the synthesized audio in adhering to the provided transcription.

NISQA-TTS was designed to predict the naturalness of synthetic speech, providing a nonintrusive evaluation without needing a reference signal in TTS systems. This metric predicted the naturalness score on a five-point scale consistent with the human MOS evaluation. Our work used the NISQA-TTS model to estimate the naturalness of the synthetic speech generated by our TTS system.

WV-MOS evaluated the overall quality of the utterances generated by each model and provided a score that ranged

[1] https://aihub.or.kr

[2] https://github.com/bootphon/phonemizer
[3] https://github.com/gabrielmittag/NISQA
[4] https://github.com/AndreevP/wvmos
[5] https://github.com/clovaai/voxceleb_trainer
[6] https://huggingface.co/nvidia/stt_en_conformer_transducer_xlarge
[7] https://pypi.org/project/python-Levenshtein

TABLE 1. Comparison of the number of trained model parameters and objective quality between the fine-tuned TTS models according to different combinations of the proposed fine-tuning approaches applied to the VCTK and Libri-TTS-100 datasets.

TTS Model	Proj	AT	WN	MRF	CLN	ADP	SEPL	SECS (↑)	WER (↓)	CER (↓)	NISQA-TTS (↑)	WV-MOS (↑)	No. of Trained Parameters
Model 1	×	✓	×	×	×	×	×	0.300	62.3	51.54	2.37 ± 0.12	3.19	0.02M
Model 2	×	✓	✓	×	×	×	×	0.342	22.6	16.10	2.69 ± 0.16	3.44	0.04M
Model 3	✓	✓	✓	✓	×	×	×	0.388	15.6	10.79	2.84 ± 0.18	3.59	0.33M
Model 4	×	✓	×	×	×	×	✓	0.439	40.7	33.24	2.54 ± 0.21	3.30	3.17M
Model 5	×	✓	✓	×	×	×	✓	0.558	10.6	6.02	2.82 ± 0.17	3.67	3.19M
Model 6	✓	✓	✓	×	×	×	✓	0.572	7.8	4.60	2.98 ± 0.18	3.72	3.21M
Model 7	×	×	×	×	✓	×	✓	0.502	39.9	28.74	2.54 ± 0.21	3.22	3.23M
Model 8	✓	✓	✓	×	✓	×	✓	0.634	7.0	4.50	3.23 ± 0.17	3.78	3.29M
Model 9	✓	✓	✓	×	✓	✓	✓	0.637	6.3	3.56	3.28 ± 0.17	3.84	3.44M
Model 10	✓	✓	✓	✓	✓	×	✓	0.697	6.9	3.92	3.26 ± 0.23	4.04	3.56M
Model 11	✓	✓	✓	✓	✓	✓	✓	0.714	5.5	2.92	3.30 ± 0.27	4.05	3.71M
Full fine-tuning	—	—	—	—	—	—	—	0.755	4.1	1.72	3.36 ± 0.27	4.12	37.7M
Full fine-tuning + CLN	—	—	—	—	✓	—	—	0.770	3.2	1.18	3.40 ± 0.26	4.14	37.8M
Ground truth	—	—	—	—	—	—	—	—	2.0	0.64	3.44 ± 0.23	4.27	—

from 1 to 5 points. For MOS prediction, the WV-MOS model utilized a neural network architecture, wav2vec2.0, which was pretrained in a contrastive self-supervised manner, making it useful for various downstream tasks. The pretrained wav2vec2.0 model was fine-tuned using listening evaluation results from the Voice Conversion Challenge 2018 dataset [59]. In our study, we used WV-MOS to measure the overall quality of the generated speech for each fine-tuning method.

Objective metrics are not always reliable for measuring the perceived quality of synthesized speech from TTS models. Therefore, subjective evaluation is required to accurately assess speech quality. In this study, we compared the quality of synthesized speech obtained using our fine-tuning approach with that of the original speech using a CMOS on a seven-point scale ranging from −3 to 3. Ten people participated in the subject test by listening to 10 randomly selected pairs of original and synthesized speeches.

D. PERFORMANCE EVALUATION

To examine the effectiveness of the proposed different fine-tuning approaches on the objective and subjective quality of synthesized speech, we generated speech samples from the TTS models after applying different combinations of the proposed approaches. Table 1 compares the objective quality between the fine-tuned TTS models according to different combinations of the proposed fine-tuning approaches. The rightmost column of Table 1 compares the number of model parameters trained by each fine-tuned TTS model. In Table 1, Proj, AT, WN, and MRF denote the LoRA approach applied to the linear projection layers, attention matrix in the transformer-based text encoder (shown in Fig. 3(b)), WaveNet (shown in Fig. 3(c)), and MRF in the generator (shown in Fig. 3(d)), respectively. In addition, CLN and ADP signify the proposed approach using the CLN and residual adapter, as described in Figs. 3(a) and 3(b), respectively. Moreover, to investigate the effect of the fine-tuning of the projection layers connected to the speaker embeddings g' on the performance, four speaker embedding projection layers in the VITS model were fine-tuned (✓) or frozen (✗), denoted as SEPL (speaker embedding projection layer) in the table.

TABLE 2. Comparison of the subjective scores of the top six fine-tuned models measured in terms of CMOS.

TTS Model		CMOS
Ground truth		0
Full fine-tuning + CLN		−0.32
Full fine-tuning		−0.37
Proposed Model	Model 11	−0.49
	Model 10	−0.55
	Model 9	−0.74
	Model 8	−0.69

As revealed by Table 1, Model 1, where AT was only fine-tuned, performed deficiently overall. However, the metric scores of Model 2 showed that tuning the WN was necessary to improve the overall speech quality, naturalness, and intelligibility. Model 3, trained by fine-tuning only LoRA, indicated that it was difficult to capture the unique characteristics of the speaker's voice, making it challenging to represent the speaker accurately. Next, we fine-tuned the SEPLs in Model 4, which showed that tuning the SEPL increased the SECS score. Subsequently, we fine-tuned the VITS model with AT, WN, and SE together to create Model 5, which showed that fine-tuning the critical parts in the VITS model resulted in a higher overall quality of the synthesized speech. Further, Table 1 indicates that Model 6 improved the overall performance of the generated speech, particularly in terms

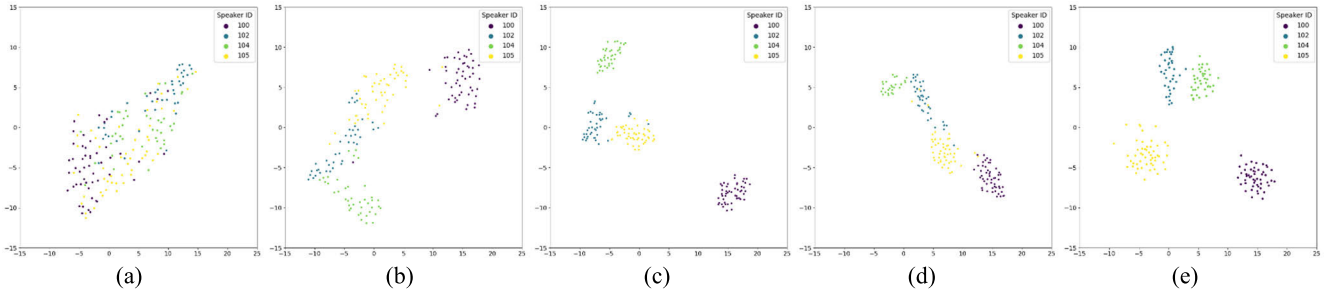

FIGURE 5. Comparison of the t-SNE plots of latent vectors predicted by different models: (a) Model 1, (b) Model 7, (c) Model 11, (d) fully fine-tuned model, and (e) fully fine-tuned model with CLN, where the latent vectors were obtained from the test speakers on the VCTK dataset.

of naturalness. However, Model 7 provided a lower overall quality but a higher SECS score than Models 1 to 4, which implied that SEPL and CLN could contribute to speaker similarity.

In the observation from Model 7, we applied SEPL and CLN to the following fine-tuned models from Models 8 to 11. As shown in Table 1, Model 8 demonstrated higher performance than Model 6, highlighting the importance of CLN in the multi-speaker fine-tuning process with an additional increase of 0.02M parameters. Meanwhile, Model 9 demonstrated higher performance in terms of WER, CER, and NISQA-TTS than Model 8 because of the addition of ADP. Moreover, Model 10 further improved speaker similarity and overall speech quality compared with Model 9 because the MRF was fine-tuned.

Lastly, we employed all the proposed approaches to fine-tune the VITS model, referred to as Model 11. As shown in the 11th row of Table 1, Model 11 showed the best objective performance among other models from Models 1–10, with a 10% increase in the number of model parameters. Interestingly, Model 11 achieved a slightly lower performance than the full fine-tuning method. Finally, to investigate the effect of the CLN on the multi-speaker TTS, we fully fine-tuned the VITS model with the CLN. The last two rows of Table 1 show that the CLN considerably contributed to improving all the objective metrics compared with the model with full fine-tuning.

Additionally, we performed a subjective test on the synthesized speeches with the models whose NISQA-TTS was higher than 3.0. In particular, we chose Models 8–11 and two models with full fine-tuning with/without CLN adaptation. Table 2 compares the CMOS of the top six fine-tuned models, revealing that CMOS was closely related to either NISQA-TTS or WV-MOS. Although the proposed approaches had slightly lower CMOS values than the case of full fine-tuning, the participants' survey confirmed that they demonstrated comparable listening results.

E. EFFECT OF SPEAKER-RELATED TECHNIQUES ON SPEAKER REPRESENTATION

We conducted a series of experiments to understand the effect of the fine-tuning of speaker-related modules on speaker

TABLE 3. Performance comparison of different LoRA ranks r = 1, 8, and 96 using the NISQA-TTS and WV-MOS metrics averaged across the VCTK and Libri-TTS datasets.

TTS Model	Performance Based on Different Ranks					
	$r = 1$		$r = 8$		$r = 96$	
Model 11	NISQA TTS	WVMOS	NISQA TTS	WVMOS	NISQA TTS	WVMOS
	3.24 ± 0.22	3.97	3.30 ± 0.27	4.05	3.26 ± 0.31	4.01

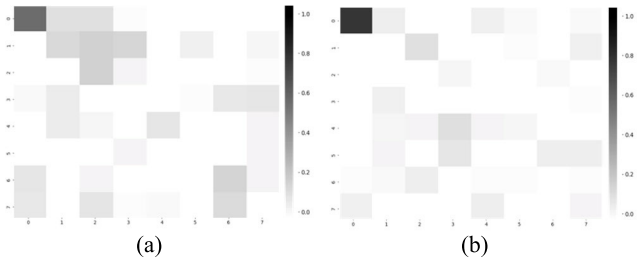

FIGURE 6. Similarity of eigenvectors between $\Delta W(8)$ and $\Delta W(96)$ for (a) the attention query matrix W_q and (b) the attention value matrix W_v. The attention map illustrates the similarity between the eigenvectors of $\Delta W(8)$ and top 8 eigenvectors of ΔW (96).

representations. Fig. 5 illustrates the t-distributed stochastic neighbor embedding (t-SNE) [60] plots of the latent vectors z of synthesized speeches from the test speakers in the VCTK dataset to compare the speaker clustering performance according to different VITS models. As shown in Fig. 5(a), the latent vectors of Model 1 were distributed randomly, implying that the speakers were not clustered anymore. Instead, Model 7 (as shown in Fig. 5(b)) provided better speaker clustering than Model 1, which implied that CLN was effective for speaker representation. Then, we plotted the latent vectors from Model 11, which showed the best subjective and objective performance, as presented in Tables 1 and 2, and achieved better speaker clustering than that of Model 7. Next, we compared the t-SNE plots of the fully fine-tuned models with and without CLN, as shown in Figs. 5(d) and 5(e), respectively. Thus, CLN was also demonstrated as effective in speaker clustering, resulting in better objective and subject quality scores.

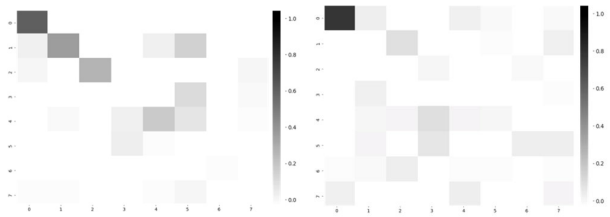

FIGURE 7. Similarity of eigenvectors between $\Delta W(8)$ and $\Delta W(96)$ for (a) a 1 × 1 convolution layer V_{lora} on a block of the WaveNet and (b) the ConvTranspose layer C_{MRF} connected to the MRF. The attention map illustrates the similarity between the singular vectors of $\Delta W(8)$ and top 8 singular vectors of ΔW (96) .

TABLE 4. Complexity comparison of the average fine-tuning speed and the RTF according to different TTS models.

TTS Model	Added Module	Average Fine-Tuning Speed/epoch (sec)	RTF
Model 5	AT+WN	23.06	0.0138
Model 6	+Proj	23.76	0.0136
Model 8	+CLN	26.36	0.0162
Model 10	+MRF	28.54	0.0161
Model 11	+ADP	29.12	0.0166
Full fine-tuning	—	41.73	0.0134
Full fine-tuning + CLN	+CLN	46.49	0.0159

F. SETTING THE RANK OF LoRA

This section describes the performance when the rank $r = 8$ for LoRA-based fine-tuning. To this end, we chose $r = 8$, 96 for the comparison. As shown in Fig. 3(b), the LoRA was first applied to W_q and W_v in an attention module in the transformer-based text encoder because they were crucial for generating queries and values. Note that W_q and W_v both had (192 × 192) matrices. Then, the LoRA matrix, $\Delta W(r) = W_{up}W_{dw}$ with rank $= r$ was applied to W_q or W_v.Then, either $\Delta W(r)$ for W_q or $\Delta W(r)$ for $W_v$was processed through singular value decomposition or eigenvalue decomposition to obtain singular vectors or eigenvectors. We computed the similarity between the pairs of two vectors obtained from $\Delta W(8)$ and $\Delta W(96)$ using the following equation:

$$\emptyset\left(\Delta W(8), \Delta W(96), i, j\right) = \frac{\left\|(\Delta W(8)^i)^T \Delta W(96)^j\right\|_F^2}{min(i, j)} \tag{17}$$

where $\Delta W(8)^i$ is the i-th singular or eigenvector of $\Delta W(8)$ and $\Delta W(96)^j$ is the j-th largest singular or eigenvector of $\Delta W(96)$. Here, we compared $r = 8$ with $r = 96$ because $r = 96$ provided the same number of elements between W_q (or W_v) and $\Delta W(96)$ as $192 \times 192 = 192 \times 96 + 96 \times 192$. Fig. 6 depicts the similarity of eigenvectors between $\Delta W(8)$ and $\Delta W(96)$ for the attention query matrix W_q and attention value matrix W_v. Accordingly, it seems that the rank $r = 96$ should be reduced to $r = 8$. Next, we repeated this experiment by applying 1) LoRA to a 1 × 1 convolution layer V_{lora} on each block of the WaveNet and 2) the ConvTranspose layer C_{MRF} connected to the MRF. Even if V_{lora} and C_{MRF}were (192 × 384) and (192 × 512) matrices, respectively, we compared $r = 8$ with $r = 96$, whereas the sophisticated selection of r could be 128 or 139 as $192 \times 384 = 192 \times 128 + 128 \times 384$ and $192 \times 512 \cong 192 \times 139 + 139 \times 512$. Fig. 7 also depicts the similarity of the singular vectors between $\Delta W(8)$ and $\Delta W(96)$ for the WaveNet layer matrix V_{lora} and the ConvTranspose layer matrix C_{MRF},which implies that rank $r = 8$ can be more reduced into smaller r. Consequently, we set $r = 8$ according to a previous recommendation [29] and these supporting experiments.

Lastly, we applied the proposed method to fine-tune the models with LoRA ranks $r = 1$ to further evaluate the performance. We measured each model's performance using the NISQA-TTS and WV-MOS metrics. Table 3 compares the NISQA-TTS and WV-MOS metrics of the TTS models when different LoRA ranks $r = 1, 8$, and 96 were applied on the VCTK and Libri-TTS-100 datasets. The results demonstrated that increasing the rank did not improve the fine-tuning performance of the TTS model; instead, it led to a performance decline. Consequently, the TTS model with $r = 8$ achieved the highest performance among the three different ranks. This confirmed that the rank $r = 8$ was the better choice among our tested ranks.

G. COMPARISON OF SYNTHESIS AND TRAINING SPEED

We evaluated the speech synthesis and training speed of our model, focusing on complexity when a new module was added. We measured two indices: the average fine-tuning speed per epoch (measured in seconds) and the real-time factor (RTF). All the measurements were performed on a single A100 GPU with a batch size of 1, and the fine-tuning speed was measured using 1,546 sentences over 20 epochs. Table 4 compares the average fine-tuning speed and RTFs of the different models. A comparison of Model 8 with Models 5 and 6 indicated that CLN increased its average fine-tuning speed from 23.76 to 26.36 s. MRF also increased the average fine-tuning speed of 2.18 s, as revealed by the comparison of Models 10 and 8. In particular, the average fine-tuning speed of Model 11, which was fine-tuned with all the proposed approaches, was much faster than that of the fully fine-tuned model. In contrast, the RTF was proportional to the number of added modules that corresponded to the additional model parameters given in the rightmost column of Table 1. As expected, the fully fine-tuned model had the lowest RTF among all the models compared in Table 4. However, Model 11, which had the highest RTF among the proposed PEFT methods, remained at a real-time level. When we inferred Model 11 on lower-resource GPUs, such as NVIDIA TITAN X and RTX 2080 Ti, it was confirmed that the proposed method could operate in real time under low-resource conditions.

H. OBJECTIVE QUALITY ACCORDING TO DIFFERENT DATASETS

In this section, we decompose the performance evaluation results shown in Table 1 into the results according to each dataset: VCTK and Libri-TTS-100. We conducted an ablation

TABLE 5. Comparison of the number of trained model parameters and objective quality between the fine-tuned TTS models according to different combinations of the proposed fine-tuning approaches applied to the publicly available VCTK dataset.

TTS Model	Proj	AT	WN	MRF	CLN	ADP	SEPL	SECS (↑)	WER (↓)	CER (↓)	NISQA-TTS (↑)	WV-MOS (↑)	No. of Trained Parameters
Model 1	×	✓	×	×	×	×	×	0.364	65.2	54.23	2.32 ± 0.17	3.34	0.02M
Model 2	×	✓	✓	×	×	×	×	0.409	25.8	18.38	2.51 ± 0.26	3.66	0.04M
Model 3	✓	✓	✓	✓	×	×	×	0.444	16.5	11.15	2.71 ± 0.14	3.78	0.33M
Model 4	×	✓	×	×	×	×	✓	0.465	48.6	39.76	2.41 ± 0.32	3.43	3.17M
Model 5	×	✓	✓	×	×	×	✓	0.591	10.1	5.61	2.63 ± 0.25	3.85	3.19M
Model 6	✓	✓	✓	×	×	×	✓	0.612	6.2	3.82	2.84 ± 0.17	3.94	3.21M
Model 7	×	×	×	×	✓	×	✓	0.523	39.5	28.36	2.43 ± 0.24	3.36	3.23M
Model 8	✓	✓	✓	×	✓	×	✓	0.662	5.7	3.91	3.04 ± 0.29	3.93	3.29M
Model 9	✓	✓	✓	×	✓	✓	✓	0.660	5.2	3.19	3.07 ± 0.23	4.05	3.44M
Model 10	✓	✓	✓	✓	✓	×	✓	0.721	6.2	3.68	3.05 ± 0.33	4.23	3.56M
Model 11	✓	✓	✓	✓	✓	✓	✓	0.734	3.9	2.21	3.08 ± 0.36	4.24	3.71M
Full fine-tuning	—	—	—	—	—	—	—	0.771	2.1	0.96	3.18 ± 0.33	4.36	37.7M
Full fine-tuning + CLN	—	—	—	—	✓	—	—	0.783	1.6	0.73	3.20 ± 0.28	4.34	37.8M
Ground truth	—	—	—	—	—	—	—	—	0.8	0.26	3.23 ± 0.13	4.50	—

TABLE 6. Comparison of the number of trained model parameters and objective quality between the fine-tuned TTS models according to different combinations of the proposed fine-tuning approaches applied to the publicly available Libri-TTS-100 dataset.

TTS Model	Proj	AT	WN	MRF	CLN	ADP	SEPL	SECS (↑)	WER (↓)	CER (↓)	NISQA-TTS (↑)	WV-MOS (↑)	No. of Trained Parameters
Model 1	×	✓	×	×	×	×	×	0.237	59.4	48.85.	2.41 ± 0.18	3.04	0.02M
Model 2	×	✓	✓	×	×	×	×	0.276	19.4	13.81	2.87 ± 0.17	3.21	0.04M
Model 3	✓	✓	✓	✓	×	×	×	0.332	14.7	10.42	2.97 ± 0.22	3.40	0.33M
Model 4	×	✓	×	×	×	×	✓	0.413	32.8	26.72	2.66 ± 0.27	3.18	3.17M
Model 5	×	✓	✓	×	×	×	✓	0.526	11.1	6.43	3.01 ± 0.22	3.49	3.19M
Model 6	✓	✓	✓	×	×	×	✓	0.531	9.5	5.38	3.11 ± 0.31	3.51	3.21M
Model 7	×	×	×	×	✓	×	✓	0.481	40.4	29.12	2.64 ± 0.34	3.08	3.23M
Model 8	✓	✓	✓	×	✓	×	✓	0.606	8.3	5.08	3.42 ± 0.18	3.63	3.29M
Model 9	✓	✓	✓	×	✓	✓	✓	0.614	7.4	3.93	3.49 ± 0.25	3.62	3.44M
Model 10	✓	✓	✓	✓	✓	×	✓	0.673	7.7	4.16	3.48 ± 0.33	3.85	3.56M
Model 11	✓	✓	✓	✓	✓	✓	✓	0.695	7.1	3.63	3.52 ± 0.39	3.86	3.71M
Full fine-tuning	—	—	—	—	—	—	—	0.739	6.1	2.48	3.54 ± 0.42	3.89	37.7M
Full fine-tuning + CLN	—	—	—	—	✓	—	—	0.756	4.8	1.63	3.62 ± 0.44	3.93	37.8M
Ground truth	—	—	—	—	—	—	—	—	3.2	1.02	3.64 ± 0.32	4.04	—

study using different fine-tuning methods and assessed the performance of the proposed method. We compared the results with those of full fine-tuning and ground truth. Table 5 presents the evaluation results using the objective metrics of different methods on the VCTK dataset, whereas Table 6 provides the results using the objective metrics on the Libri-TTS-100 dataset.

As shown in Tables 5 and 6, Model 5, which fine-tuned AT, WN, and SEPL together, showed a higher overall quality of the synthesized speech, achieving SECS values of 0.591 and 0.526 and WV-MOS scores of 3.85 and 3.49. By fine-tuning Proj to Model 5, Model 6 improved the overall performance of the generated speech with NISQA-TTS scores of 2.84 ± 0.17 and 3.11 ± 0.31. However, when it was applied to fine-tuning using four speakers, the performance was lower than that for fine-tuning using a single speaker. Model 7 involved fine-tuning using four speakers, demonstrating higher performance and highlighting the importance of CLN in the multi-speaker fine-tuning process. Moreover, its objective performance was improved compared with that of Model 6.

Model 10 involved fine-tuning the MRF by LoRA, which showed an improvement in the overall quality across both datasets. Model 11 further enhanced this performance by incorporating ADP to improve the expressiveness of the prior distribution, resulting in the best performance. Overall, the VCTK dataset outperformed the Libri-TTS-100 dataset; however, because of its nature, the Libri-TTS-100 dataset had a higher naturalness score, and CLN had a more significant impact during the fine-tuning process.

TABLE 7. Comparison of the number of trained model parameters and objective quality between the fine-tuned TTS models according to different combinations of the proposed fine-tuning approaches applied to the publicly available Common Voice dataset.

TTS Model	Proj	AT	WN	MRF	CLN	ADP	SEPL	SECS (↑)	WER (↓)	CER (↓)	NISQA-TTS (↑)	WV-MOS (↑)	No. of Trained Parameters
Model 1	×	✓	×	×	×	×	×	0.227	63.8	50.21	2.31 ± 0.15	2.79	0.02M
Model 2	×	✓	✓	×	×	×	×	0.254	23.6	18.07	2.67 ± 0.17	3.06	0.04M
Model 3	✓	✓	✓	✓	×	×	×	0.313	21.7	16.33	2.81 ± 0.13	3.13	0.33M
Model 4	×	✓	×	×	×	×	✓	0.406	38.5	28.91	2.57 ± 0.21	2.87	3.17M
Model 5	×	✓	✓	×	×	×	✓	0.517	19.4	12.37	2.94 ± 0.28	3.23	3.19M
Model 6	✓	✓	✓	×	×	×	✓	0.522	16.6	9.86	3.03 ± 0.21	3.21	3.21M
Model 7	×	×	×	×	✓	×	✓	0.473	43.9	31.85	2.58 ± 0.24	2.84	3.23M
Model 8	✓	✓	✓	×	✓	×	✓	0.589	13.2	7.11	3.26 ± 0.37	3.38	3.29M
Model 9	✓	✓	✓	×	✓	✓	✓	0.608	12.1	6.87	3.30 ± 0.13	3.36	3.44M
Model 10	✓	✓	✓	✓	✓	×	✓	0.661	11.8	6.22	3.32 ± 0.27	3.41	3.56M
Model 11	✓	✓	✓	✓	✓	✓	✓	0.655	10.6	5.67	3.44 ± 0.25	3.46	3.71M
Full fine-tuning	—	—	—	—	—	—	—	0.681	8.1	4.71	3.51 ± 0.31	3.49	37.7M
Full fine-tuning + CLN	—	—	—	—	✓	—	—	0.696	8.2	4.97	3.55 ± 0.28	3.54	37.8M
Ground truth	—	—	—	—	—	—	—	—	5.1	2.34	3.66 ± 0.40	3.78	—

TABLE 8. Comparison of the number of trained model parameters and objective quality between the fine-tuned TTS models according to different combinations of the proposed fine-tuning approaches applied to the publicly available KMSSS dataset.

TTS Model	Proj	AT	WN	MRF	CLN	ADP	SEPL	SECS (↑)	WER (↓)	CER (↓)	NISQA-TTS (↑)	WV-MOS (↑)	No. of Trained Parameters
Model 1	×	✓	×	×	×	×	×	0.252	68.6	48.02	2.48 ± 0.19	3.18	0.02M
Model 2	×	✓	✓	×	×	×	×	0.304	29.3	16.51	2.77 ± 0.26	3.41	0.04M
Model 3	✓	✓	✓	✓	×	×	×	0.357	27.8	15.46	3.03 ± 0.29	3.55	0.33M
Model 4	×	✓	×	×	×	×	✓	0.433	42.8	25.77	2.53 ± 0.33	3.19	3.17M
Model 5	×	✓	✓	×	×	×	✓	0.554	22.5	13.82	3.18 ± 0.38	3.64	3.19M
Model 6	✓	✓	✓	×	×	×	✓	0.568	20.1	10.34	3.31 ± 0.26	3.68	3.21M
Model 7	×	×	×	×	✓	×	✓	0.509	50.3	35.21	2.71 ± 0.31	3.24	3.23M
Model 8	✓	✓	✓	×	✓	×	✓	0.642	17.8	6.01	3.49 ± 0.27	3.81	3.29M
Model 9	✓	✓	✓	×	✓	✓	✓	0.651	15.3	4.65	3.68 ± 0.21	3.87	3.44M
Model 10	✓	✓	✓	✓	✓	×	✓	0.703	14.7	4.11	3.87 ± 0.22	4.01	3.56M
Model 11	✓	✓	✓	✓	✓	✓	✓	0.697	13.1	3.96	4.04 ± 0.35	3.98	3.71M
Full fine-tuning	—	—	—	—	—	—	—	0.736	11.4	2.66	4.15 ± 0.37	4.04	37.7M
Full fine-tuning + CLN	—	—	—	—	✓	—	—	0.749	10.8	2.92	4.24 ± 0.41	4.19	37.8M
Ground truth	—	—	—	—	—	—	—	—	8.2	2.21	4.38 ± 0.22	4.41	—

In addition to the VCTK and Libri-TTS-100 datasets, the performance of the fine-tuned models was evaluated on the Common Voice dataset to assess the robustness of the proposed PEFT method against diverse accents and speech styles. Table 7 presents the evaluation results using the objective metrics of the different methods on the Common Voice dataset. Apparently, the WV-MOS score of the ground truth samples was 3.78, which was lower than the scores obtained from both the VCTK and Libri-TTS-100 datasets. This was due to the characteristics of the Common Voice dataset, such as its various accents and slight background noise, which increased CER and WER. Compared with the results in Tables 5 and 6, the tendency of performance variations according to different combinations of the proposed fine-tuning approaches was similar to those in the VCTK and Libri-TTS-100 datasets. In other words, the full fine-tuning with CLN provided better performance than the conventional full fine-tuning and also a comparable overall performance to the ground truth with an SECS score of 0.696, demonstrating the effectiveness of the fine-tuning process. Moreover, the performance of Model 11 was the best among the fine-tuned models using the proposed PEFT method. It also maintained WV-MOS and NISQA-TTS scores comparable to those by the full fine-tuning, suggesting that the proposed PEFT method could be effective even when applied to more challenging speech samples.

I. ADAPTATION RESULTS WITH THE KOREAN DATASET

In this section, we apply the proposed PEFT method to the KMSSS dataset to examine the effect of variations in

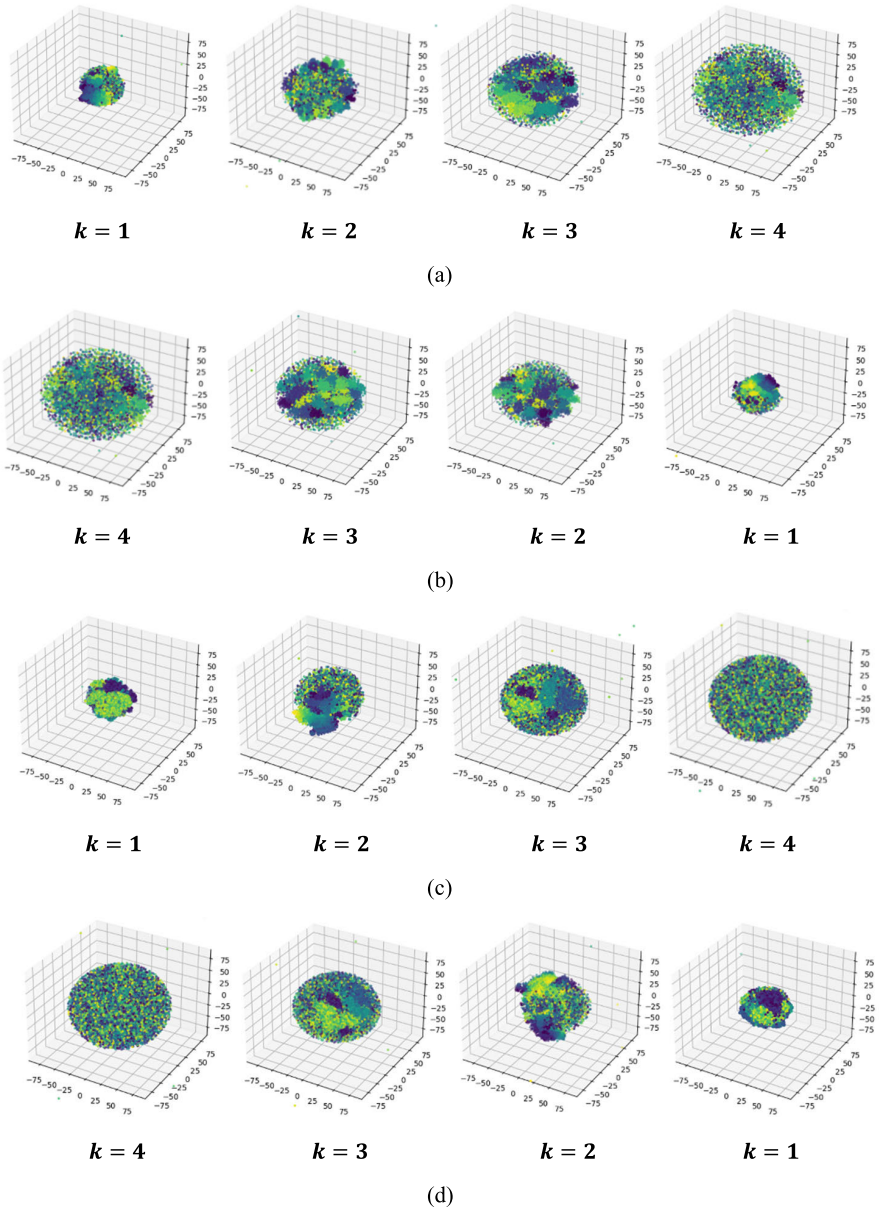

FIGURE 8. 3D t-SNE plots of the latent variables in the kth forward flow and backward flow (k= 1, $\cdots$, 4): (a) forward flows and (b) backward flows of the flow-based decoder after full fine-tuning and (c) forward flows and (d) backward flows of the flow-based decoder after applying LoRA in Model 11. The plots were obtained using 32 samples from the VCTK dataset.

pronunciation and tone across languages on the model's adaptability. Table 8 presents the evaluation results using the objective metrics of the different methods on the KMSSS dataset. According to the results, although the proposed PEFT method was applied to fine-tune the pretrained Korean TTS model, the difference in terms of performance between Model 11 and full fine-tuning with CLN for the Korean dataset was consistent with those for the English datasets, as shown in Tables 5–7. This implied that even if we developed a PEFT method using English datasets, the proposed PEFT method could be applied to any language. Instead, the most critical factor for applying PEFT lies in the ability of the pretrained TTS model to produce proper synthetic and personalized speech by applying the proposed fine-tuning method to achieve optimal performance. In conclusion, the proposed PEFT method can be effectively applied in various scenarios, regardless of the language.

J. COMPARISON WITH ZERO-SHOT TTS MODELS

In this section, we compare the performance of our model, which incorporated the proposed PEFT method, with two zero-shot TTS models: YourTTS [11] and XTTS [14]. YourTTS is a VITS-based E2E TTS model that utilizes the H/ASP model's output as speaker embedding and applies a speaker consistency loss to ensure high speaker similarity between synthetic and ground truth speech. Meanwhile,

TABLE 9. Comparison of the objective metrics between Model 11, Your-TTS, and XTTS. The results are averaged across the test samples from the VCTK and Libri-TTS.

TTS Model	SECS	WER	CER	NISQA-TTS	WV-MOS
Model 11	0.715	5.5	2.92	3.30 ± 0.37	4.05
YourTTS	0.658	7.1	3.65	3.21 ± 0.24	3.64
XTTS-v2	0.641	4.8	2.42	3.38 ± 0.31	3.77

XTTS builds on Tortoise [61] but introduces several novel modifications to enable multilingual training, enhance zero-shot TTS performance, and achieve faster training and inference. As we aimed to evaluate the performance of the zero-shot TTS, we utilized the open-source YourTTS[8] model and XTTS-v2[9] without any fine-tuning. To evaluate these three TTS models, including our Model 11, we prepared 120 samples by taking 60 samples from each VCTK and Libri-TTS-100.

Table 9 compares the objective metrics of Model 11, Your-TTS, and XTTS-v2. Apparently, Model 11 outperformed YourTTS in all objective metrics. Meanwhile, XTTS achieved slightly better WER, CER, and NISQA-TTS values than Model 11, but Model 11 showed much better performance in terms of SECS, which was the most important metric for personalized speech. Therefore, we concluded that the proposed PEFT method was more effective than zero-shot TTS models for generating personalized speech.

K. EFFECT OF LoRA ON INFORMATION FLOW IN THE FLOW-BASED DECODER

Herein, we investigate whether the application of LoRA to the flow-based decoder did not affect invertibility during inference. Fig. 8 illustrates the three-dimensional (3D) t-SNE of each k-th latent variable from $f_k(\cdot)$ and $f_k^{-1}(\cdot)$, as described in equation (3). The final output ($k = 4$) was then passed backward through the same layers to obtain the original output data, as depicted in Fig. 8(b) (full fine-tuning) and Fig. 8(d) (Model 11).

To measure the difference between the initial forward data distribution and the final backward distribution, we used the centered kernel alignment (CKA) [62]. CKA quantifies the similarity between pairs of neural network representations and effectively calculates the similarity of representation distributions invariant to isotropic scaling. Using CKA, we can robustly assess the similarity of data distributions before and after passing through flow layers. Note that the CKA calculations were performed using open-source data[10].

Table 10 compares the CKA accuracy between the latent variables of the first forward and the last backward layer outputs applied to the full fine-tuned model, Model 2, and Model 11. The reason why we compared Model 11 with Model 2 was that Model 2 was the first attempt to deal with LoRA to WaveNet in the flow network.

[8]https://github.com/Edresson/YourTTS
[9]https://github.com/coqui-ai/TTS
[10]https://github.com/yuanli2333/CKA-Centered-Kernel-Alignment

TABLE 10. Comparison of the CKA accuracy between the latent variables of the first forward and the last backward layer outputs, applied to the full fine-tuned model, Model 2, and Model 11.

TTS Model	CKA Accuracy
Full fine-tuning	96.31
Model 2	95.98
Model 11	96.19

As shown in the table, Model 11 achieved a CKA accuracy of 96.19, whereas the full CKA accuracy of the fine-tuned model was 96.31. Such a high similarity score of Model 11 demonstrated that the integration of LoRA into the flow-base encoder yielded flow invertibility. Furthermore, Model 2 achieved a CKA accuracy of 95.98. Although Model 2 showed lower speech performance because of its fewer tuning parameters, the CKA scores indicated that the invertibility of the flow transformations was also maintained. These results confirmed that integrating LoRA did not affect the invertibility of the flow-based transformations.

VI. CONCLUSION

In this paper, we proposed several fine-tuning approaches to improve the performance of an E2E multi-speaker TTS by efficiently adapting it to new speakers. To this end, we first proposed a LoRA-based fine-tuning approach to achieve speech quality comparable with that of a fully fine-tuned model by updating a smaller number of model parameters. Second, a CLN-based fine-tuning approach was proposed to handle speaker-specific variation with improved multi-speaker PEFT performance. Third, the residual adapter was integrated into the text encoder output to improve the expressiveness of the prior distribution. We constructed the VITS models using the VCTK, Libri-TTS-100, Common Voice, and Korean multi-speaker datasets according to different combinations of the proposed fine-tuning approaches (i.e., LoRA, CLN, and residual adapter). The model performance was evaluated using five objective measures, namely, SECS, WER, CER, NISQA-TTS, and WV-MOS, as well as a subjective listening test involving the measurement of CMOS. The performance comparison revealed that LoRA improved the overall objective measures but was limited in improving the subjective quality for multi-speaker TTS. However, combining LoRA and CLN improved the speech quality compared to that using only LoRA. In addition, the VITS model was fine-tuned using all the proposed approaches, which provided objective and subjective speech quality compared with the fully fine-tuned model. Next, we investigated the effect of the proposed fine-tuning approaches on speaker clustering. The t-SNE comparison showed that CLN was effective in separating speakers in the latent space. Finally, the comparison of complexity by measuring the average fine-tuning speed and RTF showed that the proposed fine-tuning approaches were realized with less complexity compared with the full fine-tuning approach.

Despite these promising results, the proposed approaches have limitations that require future work. First, although the

proposed PEFT method achieved good performance, it is still not as effective as full fine-tuning. Second, the baseline model structure has exhibited limited adaptability when dealing with challenging datasets such as Common Voice. These limitations can be mitigated by enhancing the adaptability of the pre-trained model through structural modifications or adding new modules to the baseline model architecture. Furthermore, additional adapters can be integrated into various components of the system beyond the prior encoder of the VITS to assess their potential for further performance enhancement. By focusing on these aspects, we aim to advance the adaptability and efficiency of PEFT approaches in multi-speaker TTS systems.

REFERENCES

[1] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," 2021, *arXiv:2106.15561*.
[2] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.
[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 4779–4783.
[4] 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 *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–11.
[5] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 8067–8077.
[6] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, "AdaSpeech: Adaptive text to speech for custom voice," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–12.
[7] F. Biadsy, Y. Chen, X. Zhang, O. Rybakov, A. Rosenberg, and P. Moreno, "A scalable model specialization framework for training and inference using submodels and its application to speech model personalization," in *Proc. Interspeech*, Incheon, South Korea, Sep. 2022, pp. 5125–5129.
[8] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2251–2265, Sep. 2019.
[9] S. O. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, 2018, pp. 10019–10029.
[10] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, A. van den Oord, O. Vinyals, and N. de Freitas, "Sample efficient adaptive text-to-speech," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 1–15.
[11] E. Casanova, J. Weber, C. D. Shulby, A. Candido Jr., E. Gölge, and M. A. Ponti, "YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Baltimore, MD, USA, 2022, pp. 2709–2720.
[12] X. Tan, J. Chen, H. Liu, J. Cong, C. Zhang, Y. Liu, X. Wang, Y. Leng, Y. Yi, L. He, S. Zhao, T. Qin, F. Soong, and T.-Y. Liu, "NaturalSpeech: End-to-end text-to-speech synthesis with human-level quality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 6, pp. 4234–4245, Jun. 2024.
[13] Z. Jiang, Y. Ren, Z. Ye, J. Liu, C. Zhang, Q. Yang, S. Ji, R. Huang, C. Wang, X. Yin, Z. Ma, and Z. Zhao, "Mega-TTS: Zero-shot text-to-speech at scale with intrinsic inductive bias," 2023, *arXiv:2306.03509*.
[14] E. Casanova, K. Davis, E. Gölge, G. Göknar, I. Gulea, L. Hart, A. Aljafari, J. Meyer, R. Morais, S. Olayemi, and J. Weber, "XTTS: A massively multilingual zero-shot text-to-speech model," in *Proc. Interspeech*, Kos Island, Greece, Sep. 2024, pp. 4978–4982.
[15] W. Wang, Y. Song, and S. Jha, "USAT: A universal speaker-adaptive text-to-speech approach," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 2590–2604, 2024.
[16] H. B. Moss, V. Aggarwal, N. Prateek, J. González, and R. Barra-Chicote, "BOFFIN TTS: Few-shot speaker adaptation by Bayesian optimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 7639–7643.
[17] Z. Zhang, Q. Tian, H. Lu, L.-H. Chen, and S. Liu, "AdaDurIAN: Few-shot adaptation for neural text-to-speech with DurIAN," 2020, *arXiv:2005.05642*.
[18] D. Min, Y. Kim, D. Yang, S. Yang, and S. Yoon, "Meta-StyleSpeech: Multi-speaker adaptive text-to-speech generation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Vienna, Austria, 2021, pp. 7748–7759.
[19] N. Morioka, H. Zen, N. Chen, Y. Zhang, and Y. Ding, "Residual adapters for few-shot text-to-speech speaker adaptation," 2022, *arXiv:2210.15868*.
[20] C.-P. Hsieh, S. Ghosh, and B. Ginsburg, "Adapter-based extension of multi-speaker text-to-speech model for new speakers," 2022, *arXiv:2211.00585*.
[21] A. Mehrish, A. R. Kashyap, L. Yingting, N. Majumder, and S. Poria, "ADAPTERMIX: Exploring the efficacy of mixture of adapters for low-resource TTS adaptation," in *Proc. INTERSPEECH*, Dublin, Ireland, Aug. 2023, pp. 312–316.
[22] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 4006–4010.
[23] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, 2019, pp. 6706–6713.
[24] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 5530–5540.
[25] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," 2015, *arXiv:1505.05770*.
[26] Z. Ziegler and A. Rush, "Latent normalizing flows for discrete sequences," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 7673–7682.
[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, 2014, pp. 2672–2680.
[28] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 3830–3834.
[29] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.
[30] A. Aghajanyan, S. Gupta, and L. Zettlemoyer, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics (ACL) 11th Int. Joint Conf. Natural Lang. Process. (IJCNLP)*, 2021, pp. 7319–7328.
[31] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 17022–17033.
[32] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 506–516.
[33] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 2790–2799.
[34] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," Dept. Centre Speech Technology Research, Univ. Edinburgh, Edinburgh, U.K., Tech. Rep., 2019, doi: 10.7488/ds/2645.
[35] 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*, Graz, Austria, Sep. 2019, pp. 1526–1530.
[36] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proc. 12th Lang. Resources Eval. Conf. (LREC)*, Marseille, France, 2020, pp. 4218–4222.

[37] H. S. Heo, B.-J. Lee, J. Huh, and J. S. Chung, "Clova baseline system for the VoxCeleb speaker recognition challenge 2020," 2020, *arXiv:2009.14153*.
[38] 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*, Shanghai, China, Oct. 2020, pp. 5036–5040.
[39] G. Mittag and S. Möller, "Deep learning based assessment of synthetic speech naturalness," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 1748–1752.
[40] P. Andreev, A. Alanov, O. Ivanov, and D. Vetrov, "HiFi++: A unified framework for bandwidth extension and speech enhancement," 2022, *arXiv:2203.13086*.
[41] P. C. Loizou, "Speech quality assessment," in *Multimedia Analysis, Processing and Communications* (Studies in Computational Intelligence), W. Lin, D. Tao, J. Kacprzyk, Z. Li, E. Izquierdo, and H. Wang, Eds., Berlin, Germany: Springer, 2011, pp. 623–654.
[42] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 14910–14921.
[43] 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 *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 3171–3180.
[44] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 1–7.
[45] J. Donahue, S. Dieleman, M. Bińkowski, E. Elsen, and K. Simonyan, "End-to-end adversarial text-to-speech," 2020, *arXiv:2006.03575*.
[46] S.-H. Lee, S.-B. Kim, J.-H. Lee, E. Song, M.-J. Hwang, and S.-W. Lee, "HierSpeech: Bridging the gap between text and speech by hierarchical variational inference using self-supervised representations for speech synthesis," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, 2024, pp. 16624–16636.
[47] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural spline flows," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, 2019, pp. 7509–7520.
[48] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, 2017, pp. 1–31.
[49] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, 2018, pp. 10215–10224.
[50] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," 2018, *arXiv:1811.00002*.
[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.
[52] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with PixelCNN decoders," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Barcelona, Spain, 2016, pp. 4797–4805.
[53] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[55] M. Bernard and H. Titeux, "Phonemizer: Text to phones transcription for multiple languages in Python," *J. Open Source Softw.*, vol. 6, no. 68, p. 3958, Dec. 2021.
[56] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 1–16.
[57] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, Hyderabad, India, Sep. 2018, pp. 1086–1090.
[58] S. Zhang, Y. Hu, and G. Bian, "Research on string similarity algorithm based on Levenshtein distance," in *Proc. IEEE 2nd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Chongqing, China, Mar. 2017, pp. 2247–2251.
[59] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," 2018, *arXiv:1804.04262*.
[60] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, Jan. 2008.
[61] J. Betker, "Better speech synthesis through scaling," 2023, *arXiv:2305.07243*.
[62] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 3519–3529.

CHANGI HONG (Graduate Student Member, IEEE) received the B.S. degree in electronics and computer engineering from Chonnam National University, South Korea, in 2022, and the M.S. degree from the AI Graduate School, Gwangju Institute of Science and Technology (GIST), South Korea, in 2024. His research interests include artificial intelligence in signal processing, focusing on text-to-speech (TTS) optimization, expressive TTS systems, and natural automatic dubbing.

JUNG HYUK LEE (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering and computer science from Ulsan National Institute of Science and Technology, South Korea, in 2015, and the M.S. degree in electrical engineering and computer science from Gwangju Institute of Science and Technology (GIST), South Korea, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include deep learning approaches to speech synthesis and voice cloning, as well as speech synthesis in the cross-lingual domain.

HONG KOOK KIM (Senior Member, IEEE) received the B.S. degree in control and instrumentation engineering from Seoul National University, South Korea, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, South Korea, in 1990 and 1994, respectively. From 1990 to 1998, he was a Senior Researcher with the Samsung Advanced Institute of Technology, South Korea. From 1998 to 2003, he was a Senior Technical Staff Member with the Voice-Enabled Services Research Laboratory with AT&T Labs-Research, Florham Park, NJ, USA. Since August 2003, he has been a Professor with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), South Korea. He is also jointly affiliated with the AI Graduate School, GIST. From 2014 to 2015, he was a Visiting Professor with The City University of New York, New York, NY, USA. Recently, he founded AunionAI Company Ltd., which provides AI-powered automatic subtitle generation and automatic dubbing solutions for media content creation. His research interests include statistical and deep learning approaches to speech recognition, sound event detection, unsupervised anomaly detection, speech/audio enhancement, and sound source separation. He has served as an Editorial Committee Member and an Area Editor for *Digital Signal Processing*. He is a member of the APSIPA Speech, Language, and Audio Technical Committee.