IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 18, 2021, accepted December 27, 2021, date of publication January 7, 2022, date of current version January 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141200

Transfer Learning, Style Control, and Speaker Reconstruction Loss for Zero-Shot Multilingual Multi-Speaker Text-to-Speech on Low-Resource Languages

KURNIAWATI AZIZAH, (Member, IEEE), AND WISNU JATMIKO, (Senior Member, IEEE)
Faculty of Computer Science Faculty, Universitas Indonesia, Depok 16424, Indonesia

Corresponding authors: Kurniawati Azizah (kurniawati.azizah@cs.ui.ac.id) and Wisnu Jatmiko (wisnuj@cs.ui.ac.id)

This work was supported in part by the Pendampingan Publikasi Internasional (PPI) Q1 from Universitas Indonesia under Grant NKB-547/UN2.RST/HKP.05.00/2021, and in part by the NVIDIA DGX-1 computing facilities at the Tokopedia-UI AI Center of Excellence.

ABSTRACT Deep neural network (DNN)-based systems generally require large amounts of training data, so they have data scarcity problems in low-resource languages. Recent studies have succeeded in building zero-shot multi-speaker DNN-based TTS on high-resource languages, but they still have unsatisfactory performance on unseen speakers. This study addresses two main problems: overcoming the problem of data scarcity in the DNN-based TTS on low-resource languages and improving the performance of zero-shot speaker adaptation for unseen speakers. We propose a novel multi-stage transfer learning strategy using a partial network-based deep transfer learning to overcome the low-resource problem by utilizing pre-trained monolingual single-speaker TTS and d-vector speaker encoder on a high-resource language as the source domain. Meanwhile, to improve the performance of zero-shot speaker adaptation, we propose a new TTS model that incorporates an explicit style control from the target speaker for TTS conditioning and an utterance-level speaker reconstruction loss during TTS training. We use publicly available speech datasets for experiments. We show that our proposed training strategy is able to effectively train the TTS models using a limited amount of training data of low-resource target languages. The models trained using the proposed transfer learning successfully produce intelligible natural speech sounds, while in contrast using standard training fails to make the models synthesize understandable speech. We also demonstrate that our proposed style encoder network and speaker reconstruction loss significantly improves speaker similarity in zero-shot speaker adaptation task compared to the baseline model. Overall, our proposed TTS model and training strategy has succeeded in increasing the speaker cosine similarity of the synthesized speech on the unseen speakers test set by 0.468 and 0.279 in native and foreign languages respectively.

INDEX TERMS Deep neural network, low-resource, multi-speaker, multilingual, partial network-based deep transfer learning, speaker reconstruction loss, style control, text-to-speech, zero-shot speaker adaptation.

I. INTRODUCTION

In line with the advancement of deep learning in various fields, deep neural network (DNN)-based text-to-speech (TTS) has become increasingly attractive to researchers in recent years [1]–[10]. Some advantages of this end-to-end DNN-based TTS system are more ease for conditioning on various attributes, such as speakers, language, prosody, speaking style, sentiment, and also more ease for new data adaptation [11]. Recent progress in the end-to-end DNN-based TTS such as Tacotron-2 [7] has resulted in highly realistic and natural-sounding synthetic speech close to that of human speech. This is not only for single-speaker TTS, multi-speaker TTS systems also show outstanding results. However, many studies of multi-speaker TTS only focus on producing speech utterances from target speakers seen in

The associate editor coordinating the review of this manuscript and approving it for publication was Felix Albu.

training data [12]–[18] including our previous work [19]. It is challenging to extend speaker adaptation capabilities that can synthesize speech from target speakers that are not seen during model training.

The zero-shot speaker adaptation approach has been used by recent studies to synthesize speech from unseen target speakers without requiring a retraining of the model for those speakers [20]–[23]. It utilizes a speaker encoder that has been trained separately for speaker recognition tasks such as automatic speaker identification (ASI) and automatic speaker verification (ASV). DNN-based speaker encoder provides an effective way to extract highly discriminatory speaker-specific features from audio recordings. It is widely used to model speaker representation. This includes zero-shot speaker adaptation in Tacotron-based TTS that uses a DNN-based x-vector embedding [20], [22], d-vector embedding [21], and ResNet34 speaker embedding [23].

However, the various advances in DNN-based TTS research mentioned previously are mostly on high-resource languages. Data-hungry deep learning methods generally give a good performance in high-resource languages but a poor performance in low-resource languages [24]. Low-resource languages are generally less studied, resource-scarce, less computerized, less privileged, less commonly taught, or low-density, among other denominations [25], [26]. The low-resource problem is challenging and considered as one of the four biggest open problems in natural language processing (NLP) research [27].

In this paper, we address the challenge to develop zero-shot speaker adaptation on multilingual multi-speaker TTS for low-resource languages. To overcome the low-resource problem we propose a novel partial network-based deep transfer learning (DTL) to train our TTS model in the low-resource target domain by utilizing auxiliary data from a high-resource language. In here, we also introduce the concept of partial network-based DTL in more detail and how it differs from the commonly used network-based DTL. This paper is an extended study of our previous work [19], [28] on transfer learning for TTS models. Different from our previous works, our current study focuses on transferring two pre-trained source models on a high-resource language to our zero-shot multilingual multi-speaker TTS for further fine-tuning on the low-resource target languages. The first source model is a monolingual single-speaker Tacotron-2-based TTS [7]. Without this source model, our TTS model fails to synthesize intelligible speech in the low-resource domain.

Meanwhile, for the second source model we use d-vector speaker encoder using a generalized-end-to-end (GE2E) [29]. Our strategy is to use a pre-trained d-vector speaker encoder on a high-resource language with a sufficient number of speakers for better speaker representation. The pre-trained speaker encoder generates the speaker representation in our TTS model on the low-resource languages. Transfer learning without fine-tuning the speaker encoder parameters allows our TTS to synthesize the speech sound from both seen and unseen target speakers. Seen target speakers refer to speakers that are included in our training data, while unseen target speakers refer to speakers that are not in the training data.

For the baseline model we use TTS model [21] because it uses the same d-vector speaker encoder as our model. However, we also apply methods proposed by [20], [22], [23] for comparative study. The study shows that the baseline models give poor performance on the low-resource languages. The speaker similarity is low, especially on unseen target speakers. Therefore, to improve the performance of the system on the unseen target speaker, we propose a new TTS model that incorporates: 1) a style encoder network; and 2) utterance-level speaker reconstruction loss.

The style encoder is used to explicitly control the prosody from the target speaker. It produces high-level style representation such as speaker style, pitch range, and speaking rate from a collection of voice data that is jointly trained with the overall TTS system. During inference time this style encoder generates the speaking style from the target speaker speech sample for TTS conditioning. We show that the speech style combined with d-vector speaker representation increases the speaker similarity between synthesized speech and the real target speaker speech.

The utterance-level speaker reconstruction loss is used to reconstruct d-vector speaker representation of the synthesized speech so that is similar to that of the target speaker voice. The training process minimizes this loss simultaneously with the frame-level acoustic loss. This perceptual loss for speaker representation is useful for increasing the speaker similarity between the synthesized speech and the real target speaker speech in zero-shot speaker adaptation task.

Our experiments use publicly available speech datasets. A joint multilingual multi-speaker dataset in Indonesian, Javanese, and Sundanese is used as the low-resource target domain and English datasets as the source domain. The results show that the proposed partial network-based DTL successfully trains a zero-shot multilingual multi-speaker TTS model on the low-resource domain. This is not the case for standard training schemes. The experiments also show that the proposed zero-shot speaker adaptation outperforms the baseline models. We conclude that adding the style encoder and the speaker reconstruction loss can significantly improve the speaker similarity on both seen and unseen target speakers for both native and foreign languages.

Overall, the contribution of this research can be summarized in the following points:

1. To overcome the low-resource problem, we propose a novel deep transfer learning strategy to train a zero-shot multilingual multi-speaker TTS model. We also introduce the concept of partial network-based DTL that is different from commonly used network-based DTL. The effectiveness of our strategy opens the research opportunity for low-resource languages to use the recently advanced data-hungry DNN technology.
2. To improve the performance of zero-shot speaker adaptation, we propose a new TTS model by adding

an explicit style control using GST network and an utterance-level speaker reconstruction loss combined with the frame-level acoustic reconstruction loss. To the best of our knowledge, our work is the first attempt to incorporate the speaking style of the target speaker and speaker reconstruction loss in zero-shot speaker adaptation task. Moreover, our work is the first study to apply multilingual model in this task.

The rest of this paper is organized as follows: Section II presents the previous related works. Section III introduces partial network-based DTL, the model architecture of zero-shot multilingual multi-speaker TTS, and the training scheme of the TTS model. Section IV provides our experiment details. Section V presents the results and analyses of the proposed method and Section VI concludes the study.

II. RELATED WORKS

A. NLP FOR LOW-RESOURCE

Most research in NLP involves only about 10 to 20 high-resource languages with a particular focus on English and ignores thousands of languages with billions of speakers [30]. Various obstacles arise that cause the low-resource problem to become one of the four biggest open problems in NLP research apart from the other three biggest problems: natural language understanding (NLU), reasoning from large-scale documents, and datasets and evaluation [27].

Three main approaches to overcome NLP for low-resource challenge are as follows: 1) resource recollection, such as data augmentation and distant supervision which is used to add data replacing gold standard human annotated data; 2) language-specific tools with linguistic knowledge; 3) machine learning approaches such as adversarial learning [31], meta-learning [32], and transfer learning [33]–[35].

Although these approaches have been widely researched, especially in the field of text processing, language modelling, word embedding, and machine translation [24], [36], there are not many low-resource studies in TTS area. Some of them focus on data efficiency for low-resource scenarios. Work [37] proposed semi-supervised training using word vectors generated from a large text corpus. Some other works used cycle consistency training using automatic speech recognition (ASR) models to train TTS [38]–[40], where in the training process, ASR was used to find transcripts of speech sounds and TTS reconstructed transcripts into speech sounds. Work [41] proposed a cross-lingual mapping from high-resource language domains. In contrast to these approaches, our study proposes deep transfer learning to train a DNN-based TTS model since it is the most widely used in low-resource NLP, including our previous works [19], [28].

Transfer learning in NLP has two aspects [24], [36]: 1) Unlabelled data that is used to obtain pre-trained language representation as well as representation training for domain-specific or multilingual; 2) Auxiliary data that is used to train and transfer models from related tasks in the same language or the same (or similar) tasks from another domain/language. Our study uses the latter aspect. We implement a transfer learning strategy using auxiliary data from a high-resource language. Unlike network-based DTL in general, which applies parameter transfer only to network layers with the same dimension, we propose partial network-based DTL that can transfer parameters to network layers with different dimensions. In our case it is from a lower dimensional network layer to a higher dimensional one.

B. TTS WITH SPEAKER ADAPTATION

There are three speaker adaptation approaches in neural TTS: 1) Using speaker encoder network that is jointly trained with the entire TTS network [12]–[14] or neural vocoder [15]; 2) Using speaker embedding generated by the pre-trained speaker encoder and fine-tuning the TTS model [14]–[18]; 3) Using speaker embedding generated by the pre-trained speaker encoder without fine-tuning the TTS model [20]–[22]. Among these approaches only the third one is able to handle zero-shot speaker adaptation.

In the development of speaker encoder for speaker recognition task itself, deep learning methods also mark a new phase in the evolution of ASI/ASV technology [42]–[46]. DNN-based speaker encoder including d-vector [29], [47]–[51] and x-vector [52]–[55] replaces factor analysis i-vector [56]–[58] which has dominated this field previously. The DNN-based x-vector approach, which is a state-of-the-art speaker embedding, can function well in the high-resource domain [52], while the DNN-based d-vector can function well in the low-resource domain [47].

In contrast to recent zero-shot speaker adaptation studies that used x-vector speaker encoder [20], [22], our approach adopts a DNN-based d-vector speaker encoder using the GE2E model [36]. This is because our TTS model aims to handle zero-shot speaker adaptation in low-resource languages as the target domain.

In addition, previous zero-shot speaker adaptation studies [20]–[22] reported that their performance for unseen target speakers was still unsatisfactory and it is a challenge to improve speaker similarity between synthesized speech and the unseen target speaker voice. Moreover, the synthetic speech produced still has incorrect accents and different prosodic characteristics from the target speaker speech.

Therefore, to address those problems our work proposes different strategies from those prior works by adding an explicit prosodic control using a style encoder to condition the TTS decoder. We use the GST network [59] in our zero-shot TTS model to transfer the speaking style from the target speaker. This is to increase the speaker similarity between the synthetic speech and the target speaker voice. Furthermore, unlike all prior works on zero-shot speaker adaptation we add a language encoder to handle multilingual speech synthesis of target speaker in both native and foreign languages.

C. TTS WITH PERCEPTUAL LOSS

Similar to Tacotron [6], [7], recent zero-shot multi-speaker adaptation TTS models [20]–[22] predict mel-spectrum features from input text sequence by minimizing frame-level reconstruction loss. This learning objective function focuses on the distance between spectral features. It does not take into account the quality of perception, such as prosody and temporal spectral information at the speech level. Therefore, this loss is not always consistent with human perception.

Work [60] introduces a perceptual loss that refers to the training loss derived from an auxiliary network that had been trained previously. The network is usually trained on a different task which provides the quality of perceptual evaluation at a higher level than the speech frame. Some previous studies described the perceptual loss as a style reconstruction loss in the context of prosodic modelling. Several speech processing systems have integrated the style reconstruction loss for speech enhancement [61]–[63], voice conversion [64], [65], phone recognition [2], multi-speaker speech synthesis [23], and expressive speech [66].

Following the same principle, we use a pre-trained speaker encoder network on the ASV task to extract the speaker representation. In contrast to the previous zero-shot speaker adaptation studies [20]–[22], we add a speaker reconstruction loss by comparing the d-vector speaker representation between synthesized speech and the target speaker voice. Similar to [23], we integrate frame-level mel-spectrogram reconstruction loss and utterance-level speaker representation loss to train our TTS model. The difference lies in the model architecture and training method for the auxiliary network. Work [23] used a learnable dictionary encoding-based (LDE-based) system, whereas we use generalized end-to-end (GE2E) loss.

D. D-VECTOR SPEAKER ENCODER

To develop multi-speaker TTS we adopt DNN-based d-vector speaker encoder trained for ASV using the GE2E loss [29]. The GE2E architecture consists of three layers of LSTM with projection for d-vector speaker embedding. ASV consists of three stages: training, enrollment, and evaluation.

In the training phase, the speaker encoder model parameters θ_{se} are constructed by training the model using GE2E loss function at the utterance level for text-independent. The model is trained to produce an utterance level d-vector speaker representation of the speech audio that is as close as possible to the centroid of the speaker and as far as possible from the centroid of the other speakers. For each speech utterance, a sliding window is applied to the mel-spectrogram which can be formulated as follows:

$$N_w = \frac{N_f - L_{win}}{L_{shift}} + 1, \tag{1}$$

where N_w is the number of mel-spectrogram windows, N_f is the number of mel-spectrogram frames, and L_{win} is the window size (in frames), and L_{shift} is the shift size (in frames). The speaker representation $\boldsymbol{d}_u$ is calculated from the average d-vector of all mel-spectrogram windows which can be formulated as follows:

$$\boldsymbol{d}_u = \text{SpeakerEncoder}_{\theta_{se}}(M) = \frac{1}{N_w}\sum_{i=1}^{N_w} f_{SE}(M_i), \tag{2}$$

where θ_{se} is the model parameters of the speaker encoder function f_{SE}, M is the overall speech mel-spectrogram, M_i is the i^{th} mel-spectrogram window, and N_w is the number of windows. The function f_{SE} is defined as the LSTM network used.

The enrollment stage is to find the speaker model for each speaker based on a number of each speaker's audio recordings. We use the average model by calculating the average d-vector of each speaker's speech audios as follows:

$$\boldsymbol{d}_{spk} == \frac{1}{N_{utt}}\sum_{u=1}^{N_{utt}} \text{SpeakerEncoder}_{\theta_{se}}(M_u), \tag{3}$$

where $\boldsymbol{d}_{spk}$ is the d-vector speaker model representation for speaker spk, N_{utt} is the number of audio recordings, and M_u is the mel-spectrogram of speech utterance u.

The evaluation stage is to calculate the similarity between speaker representation of sample speech and its target speaker model. The d-vector representation of speech synthesis at the utterance level is formulated in Eq. (2) while the speaker model is formulated in Eq. (3).

III. METHODS

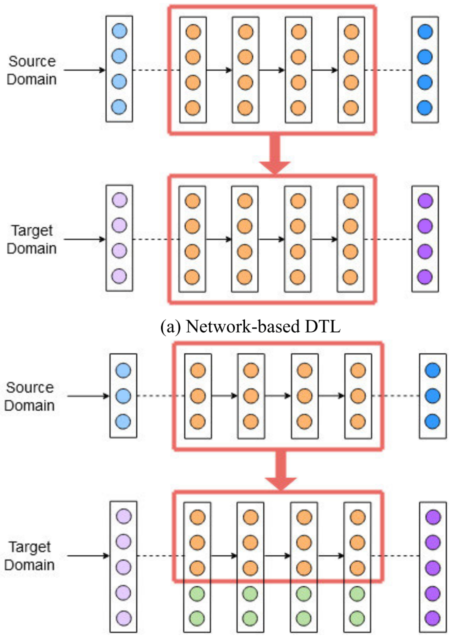
In this section, we firstly present our proposed concept of partial network-based DTL and how it differs from the commonly used network-based DTL. Secondly, we describe the proposed zero-shot multilingual multi-speaker TTS architecture. Finally, we present a novel strategy for training the model.

A. PARTIAL NETWORK-BASED DTL

In the realm of transfer learning there are two components that need to be defined, the domain D and the task T [35]. The task consists of a prediction function $f_T(x)$, considered as a conditional probability function $P(y|x)$, that predicts output y from input x. In traditional machine learning, transfer learning can be defined formally as follows: giving a learning task T_t in a domain D_t by getting help from a domain D_s for a learning task T_s to improve the predictive function f_T [33]. In deep learning, deep transfer learning (DTL) can be defined when f_T is a non-linear function that reflects a deep neural network [35].

A network-based DTL sketch map is illustrated in Fig. 1. A portion of the network that was previously trained for domain D_s and task T_s is transferred into part of a new network designed for task T_t on target domain D_t. It should be noted that the structure of each sub-network layer (see red box) of the target model has the same shape and parameter number as the source model.

Different from the commonly used network-based DTL as in Fig. 1(a), we introduce a partial network-based DTL scheme as illustrated in Fig. 1(b). The sub-networks of target



(a) Network-based DTL

(b) Partial Network-based DTL

FIGURE 1. **Network-based deep transfer learning (a) and partial network-based deep transfer learning (b). Red box indicates the sub-network that is transferred from the source mode to the target model.**

```
Algorithm 1 Network-Based DTL
Input: Model parameter $\theta_{target}$ and $\theta_{source}$,
1. For-each $\theta_w^s \subset \theta_{source}$ and $\theta_w^t \subset \theta_{t\,\arg et}$ do:
2. If Dimension($\theta_w^s$) = Dimension($\theta_w^t$) then:
3.     # full network-based DTL
4.     $\theta_w^t = \theta_w^s$
5. Else-if Dimension($\theta_w^s$) < Dimension($\theta_w^t$) then:
6.     # partial network-based DTL
7.     For-each $w^s_{d_1,d_2,\ldots,d_n} \in \theta_w^s$ do: # element-wise update
8.         $w^t_{d_1,d_2,\ldots,d_n} = w^s_{d_1,d_2,\ldots,d_n}$, where $w^t_{d_1,d_2,\ldots,d_n} \in \theta_w^t$
Output: New target model parameter $\theta_{target}$.
```

model which have the same or greater parameter dimensions can utilize the pre-trained sub-networks of source model on domain D_s for task T_s. This scheme has the flexibility to transfer sub-network layers from the source model that has different input dimensions from the corresponding sub-network layers of our target model.

The whole or partial network-based DTL process can be formulated using Algorithm 1. The source and target model parameters are denoted by θ_{source} and θ_{target} respectively. The sub-network parameters in the source model and the corresponding target model are denoted by θ_w^s and θ_w^t, where $\theta_w^s \subset \theta_{source}$ and $\theta_w^t \subset \theta_{t\,\arg et}$ respectively. These sub-network parameters are learnable weight vectors, matrices, or tensors.

This approach is based on the principle that transferring weight parameters, even if only partially, is more effective than training them from scratch. Partial transfer allows us to fine-tune the higher dimensional weight vector/matrix/tensor of the target model by making use of the low dimensional weight vector/matrix/tensor learned from the source model. In our case, we can transfer parameters from the monolingual single-speaker TTS source model to the target TTS model with the additional multilingual, multi-speaker, and style networks more optimally.

B. ZERO-SHOT TTS MODEL ARCHITECTURE

The TTS model in this study is an extension of Tacotron-2 [7] with additional networks to handle multilingual and zero-shot multi-speaker adaptation. Our model is an end-to-end TTS attention-based encoder decoder that predicts mel-spectrogram $Y' = \left(y'_1, \ldots, y'_{TY}\right)$ directly from the input grapheme-level text sequences $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{Tx})$, language identity *LangID*, and the mel-spectrogram of the target speaker speech sample $M = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{TM})$. The predicted mel-spectrogram is then converted into a speech waveform using WaveGlow vocoder [67]. The architecture of our TTS model as shown in Fig. 2 consists of several DNN blocks: text encoder, language encoder, style encoder, speaker encoder, attention, autoregressive decoder, pre-net, and post-net.

1) TEXT ENCODER

Our text encoder network is similar to Tacotron-2's. The text encoder processes the input text $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{Tx})$, where T_X is the number of characters in the normalized text, and converts it into a hidden representation $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx})$ as follows:

$$H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx}) = \text{TextEncoder}_{\theta_{te}}(X), \tag{4}$$

where θ_{te} is the text encoder network parameters.

2) LANGUAGE ENCODER

To model multilingual TTS we use a simple language encoder network consisting of one layer of fully connected neural network (FCNN) which is jointly trained with the whole TTS networks. The language encoder learns the low-dimensional language representation from discrete data as continuous vectors. The input is language label index *LangID* which is represented in a one-hot vector and the output is a d_L-dimensional language embedding $\boldsymbol{l}$ which is formulated as follows:

$$\boldsymbol{l} = \text{LanguageEncoder}_{\theta_{le}}(LangID), \tag{5}$$

where θ_{le} is the language encoder model parameters.

3) STYLE ENCODER

This study adds a style encoder using GST network which is jointly trained with the entire TTS model. The style encoder network produces style representation without requiring an explicit prosodic label, but is able to reveal various expressive styles, such as paralinguistic information (intention, attitude, and emotion), pitch, rhythm, intonation, stress, and speaker style. The d_L-dimensional style embedding $\boldsymbol{g}$ is generated

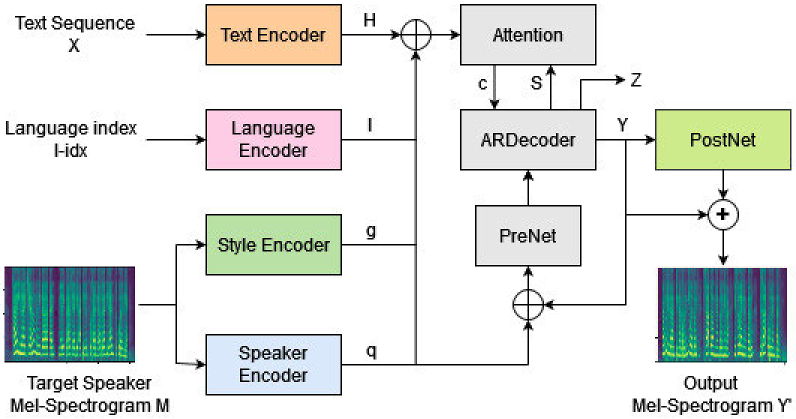


FIGURE 2. **Zero-shot multilingual multi-speaker TTS model inference. Given a text sequence *X*, language id *l-idx*, and the mel-spectrogram of the target speaker's speech sample *M*, the model predicts the output mel-spectogram *Y′* of the text *X*.**

by the style encoder network based on the mel-spectrogram of the target speaker speech sample $M = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{TM})$, which is formulated as follows:

$$\boldsymbol{g} = \text{StyleEncoder}_{\theta_{ge}}(M), \tag{6}$$

where θ_{ge} is the style encoder network parameter.

4) SPEAKER ENCODER

Speaker encoder produces a d_S-dimensional d-vector speaker embedding. Unlike other encoder networks such as text encoder, language encoder, and style encoder which are jointly trained with the TTS model, the speaker encoder network is trained separately using GE2E loss for speaker verification tasks. We use the pre-trained speaker encoder parameter weights without updating them during the TTS training process. D-vector speaker embedding $\boldsymbol{q}$ is generated by the speaker encoder from the target speaker mel-spectrogram $M = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{TM})$, which is formulated as follows:

$$\boldsymbol{q} = \text{SpeakerEncoder}_{\theta_{se}}(M), \tag{7}$$

where θ_{se} is the parameter of the speaker encoder network. Speaker embedding $\boldsymbol{q}$ is not only fed into attention network, it is also fed into pre-net.

5) MEL-SPECTROGRAM DECODER

Mel-spectrogram decoder in our TTS model consists of attention network, autoregressive (AR) decoder, pre-net, and post-net. In general it processes all encoder outputs and previous decoder output to produce mel-spectrogram acoustic features. Attention network and autoregressive decoder process the hidden representation $H = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{Tx})$ output from text encoder concatenated with language embedding $\boldsymbol{l}$, style embedding $\boldsymbol{g}$, and speaker embedding $\boldsymbol{q}$ to generate predicted mel-spectrogram $Y = \left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{TY}\right)$ and stop token $Z = (z_1, \ldots, z_{TY})$.

For each decoder step t, the output mel-spectrogram frame $\boldsymbol{y}_t$, the new hidden state decoder $\boldsymbol{s}_t$, and the stop token z_t are calculated based on the previous state $\boldsymbol{s}_{t-1}$, previous output $\boldsymbol{y}_{t-1}$, attention context vector $\boldsymbol{c}_t$, and speaker embedding $\boldsymbol{q}$ as follows:

$$\boldsymbol{s}_t, \boldsymbol{y}_t, z_t = \text{ARDecoder}_{\theta_d}\left(\boldsymbol{s}_{t-1}, \text{PreNet}_{\theta_{pn}}\left(\left[\boldsymbol{y}_{t-1}; \boldsymbol{q}\right]\right), \boldsymbol{c}_t\right), \tag{8}$$

where θ_d is the autoregressive decoder parameters and θ_{pn} is the pre-net parameters. The stop token is used to indicate when the decoder should stop producing mel-spectrogram frames. The context vector $\boldsymbol{c}_t$ is calculated with respect to all encoder outputs, as an attempt to determine the most important encoder step, using attention scheme as follows:

$$\boldsymbol{c}_t, \boldsymbol{\alpha}_t = \text{Attention}_{\theta a}\left([H; \boldsymbol{l}; \boldsymbol{g}; \boldsymbol{q}], \boldsymbol{s}_{t-1}, \boldsymbol{\alpha}_{t-1}\right), \tag{9}$$

where θ_a is the attention network parameters, $\boldsymbol{\alpha}_t$ is the weight of the attention map. Attention map is a mapping probability between decoder step $t \in T_Y$ with each encoder step $i \in T_X$ generated using location-sensitive attention mechanism. Furthermore, to improve the overall mel-spectrogram reconstruction, post-net layer consisting of stacked CNN consumes $Y = \left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{TY}\right)$ to get $Y' = \left(\boldsymbol{y}'_1, \ldots, \boldsymbol{y}'_{TY}\right)$ by adding the residual prediction as follows:

$$Y' = Y + postnet_{\theta_p}(Y), \tag{10}$$

where θ_p is the post-net parameters.

C. TTS MODEL TRAINING

We propose a novel training strategy for zero-shot multilingual multi-speaker TTS in the low-resource target domain by utilizing pre-trained monolingual single-speaker TTS in the high-resource domain and pre-trained speaker encoder for ASV tasks in the high-resource domain. We also propose to combine frame-level mel-spectrogram loss as in Tacotron-2 with utterance-level speaker reconstruction loss. The overall framework is illustrated in Fig. 3, which has three stages: 1) Monolingual single-speaker TTS training (top); 2) Speaker encoder training for ASV (bottom); 3) Zero-shot multilingual multi-speaker TTS training (middle).

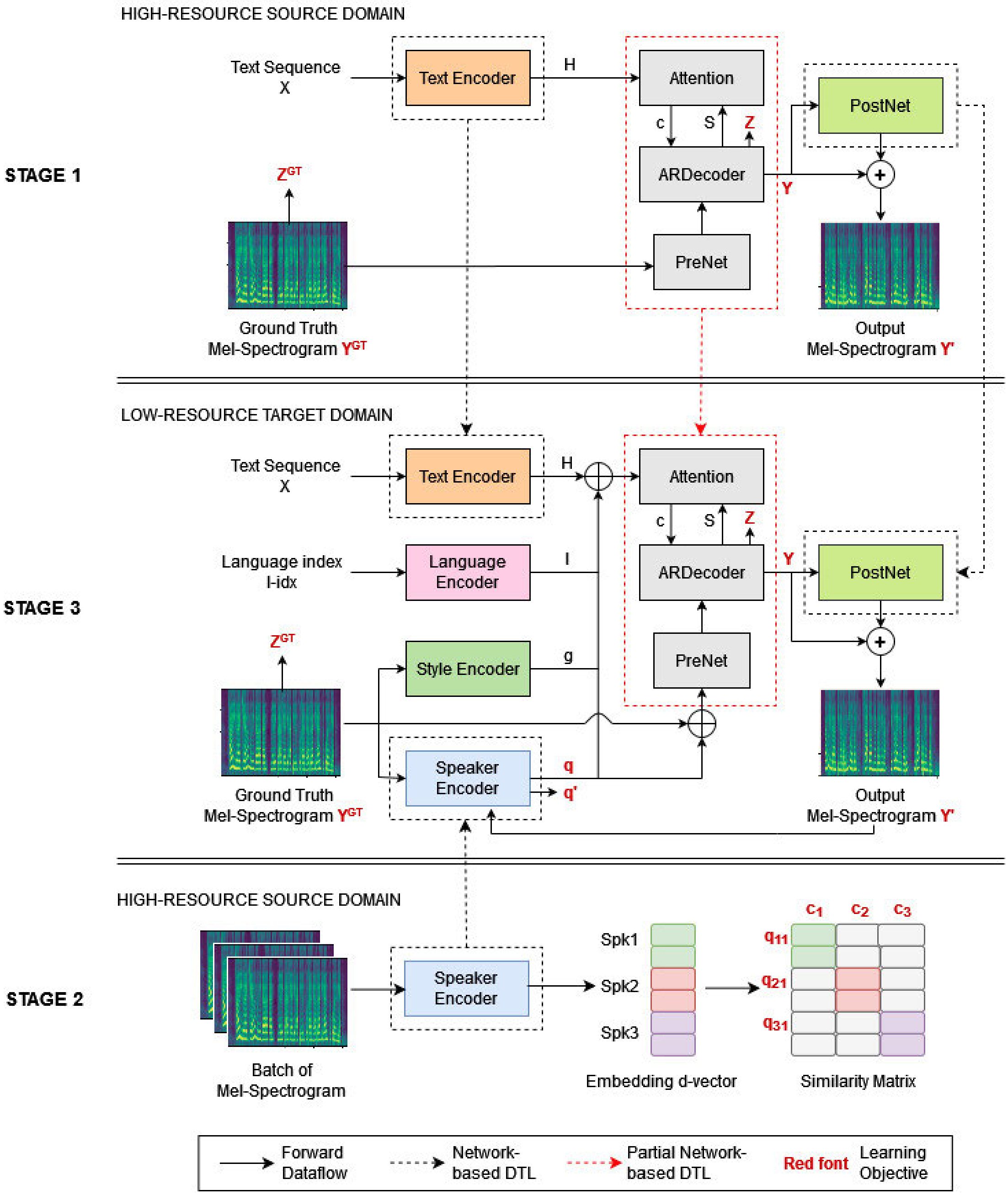


FIGURE 3. Zero-shot multilingual multi-speaker TTS model training. It consists of three training stages that include transfer learning. Stage 1 (Top) trains the monolingual single-speaker TTS on a high-resource language as the first source model. Stage 2 (Bottom) trains the speaker encoder on a high-resource language as the second source model. In Stage 3 (Middle), it transfers the trained sub-networks from the source models to the target model. The black dashed box indicates full network-based DTL, while the red dashed box indicates a partial network-based DTL. The target model is then trained on the low-resource languages as the target domain.

1) MONOLINGUAL SINGLE-SPEAKER TTS MODEL TRAINING ON HIGH-RESOURCE LANGUAGE

In Stage 1 the Tacotron2-based TTS is trained on a high-resource domain. The model is trained using a teacher-forcing procedure by providing a ground truth mel-spectrogram to the autoregressive decoder instead of using the predicted one. As it is shown in Fig. 3 (top), the learning objectives are output mel-spectrogram Y and Y' compared to the ground truth mel-spectrogram Y^{gt}, and stop token Z compared to the ground truth stop token Z^{gt}. The model is optimized by minimizing the frame-level reconstruction loss using the summed mean squared error (MSE) and binary cross entropy (BCE) loss functions as follows:

$$Loss_{frame} = MSE\left(Y^{gt}, Y\right) + MSE\left(Y^{gt}, Y'\right) + BCE\left(Z^{gt}, Z\right). \tag{11}$$

2) SPEAKER ENCODER FOR ASV MODEL TRAINING ON HIGH-RESOURCE LANGUAGE

In Stage 2, as shown in Fig. 3 (bottom), we train speaker encoder for the ASV on a high-resource domain. The speaker encoder model is trained using GE2E loss with the concept of making a speaker embedding vector as close as possible to the speaker's centroid and as far as possible from other speakers'centroid, especially the closest one.

A batch of mel-spectrograms created from $N_{spk} \times N_{utt}$ utterances of N_{spk} speakers (N_{utt} utterances for each speaker) is used to calculate similarity matrix $S_{ij,k}$, as follows:

$$\boldsymbol{S}_{ij,k} = \begin{cases} f_{\cos}\left(\boldsymbol{q}_{ij}, \boldsymbol{c}_i^{(-j)}\right) & \text{if } k = i \\ f_{\cos}\left(\boldsymbol{q}_{ij}, \boldsymbol{c}_k\right) & \text{otherwise,} \end{cases} \tag{12}$$

where f_{cos} is the cosine similarity function with learnable weight and bias, $\boldsymbol{q}_{ij}$ is the d-vector embedding of mel-spectrogram M_{ij} extracted from speaker i utterance j, $\boldsymbol{c}_i^{(-j)}$ is the centroid of speaker i without utterance j, and $\boldsymbol{c}_k$ is the centroid of the speaker k, $k \neq i$.

The model is trained using GE2E loss which is the sum of all vector embedding losses against the similarity matrix $\boldsymbol{S}$ which is formulated as follows:

$$GE2Eloss\left(\boldsymbol{S}\right) = \sum\nolimits_{i,j} L\left(\boldsymbol{q}_{ij}\right), \tag{13}$$

where $L(\boldsymbol{q}_{ij})$ is the loss in the embedding vector $\boldsymbol{q}_{ij}$, $1 \leq i \leq N_{spk}$ and $1 \leq j \leq N_{utt}$, which is formulated as follows:

$$L\left(\boldsymbol{q}_{ij}\right) = -S_{ij,i} + \log \sum\nolimits_{k=1}^{N_{spk}} \exp\left(S_{ij,k}\right). \tag{14}$$

This loss function pushes each embedding vector toward its corresponding centroid and away from all other centroids.

3) ZERO-SHOT MULTILINGUAL MULTI-SPEAKER TTS MODEL TRAINING ON LOW-RESOURCE LANGUAGE

In Stage 3 the zero-shot multilingual multi-speaker TTS model is trained using our proposed transfer learning scheme from pre-trained models on the high-resource domain and then fine-tuned on the low-resource target domain.

First, some sub-network parameters in the Tacotron2-based TTS trained in Stage 1 are transferred as initial weights of our multilingual multi-speaker TTS network parameters. Some sub-networks of the source model are transferred using network-based DTL since they have the same dimensions as in the target model, such as text encoder and post-net. Meanwhile, to transfer the mode parameters of attention, autoregressive decoder, and pre-net networks from the source model to the target model, a partial network-based DTL is applied. These networks have different neural input dimensions due to new information added to the networks. Language, style, and speaker embedding concatenated with encoded text become the new input for attention and autoregressive decoder network. Meanwhile, the speaker embedding is also concatenated with mel-spectrogram as the new input for pre-net. Second, the speaker encoder network parameters trained for the ASV task in Stage 2 are transferred from the source model to the target model using network-based DTL.

TABLE 1. The model hyper-parameters of zero-shot multilingual multi-speaker TTS.

Character Embedding	1 layer FCNN: FC-512-ReLU
Text Encoder	3 layer CNN: 512 filters with shape 5x1; 1 layer bi-LSTM: 512 units
Language Encoder	1 layer FCNN: FC-8-ReLU
Style Encoder	6 layer CNN: 32, 32, 64, 64, 128, 128 filters with shape 3x3 and stride 2x2; 1 layer GRU: 128 units Multi-head Attention: dim = 256; head= 8; token=10
Speaker Encoder	3 layer LSTM: 384 units 1 layer FCNN: FC-128
Attention	1 layer LSTM: 1024 units Q,K,V layer FCNN: FC-128-tanh 1 CNN: 32 filters with shape 31x1 1 layer FCNN: FC-128-tanh
PreNet	2 layer FCNN: FC-256-ReLU (Dropout=0.1)
ARDecoder	2 layer LSTM: 1024 units
PostNet	5 layer CNN: 512 filters with shape 5x1

The next process is fine-tuning the entire zero-shot multilingual multi-speaker TTS model on the low-resource target domain. In this training process, all network parameters, except the speaker encoder's, are updated using a new loss function. We freeze the speaker encoder network weights generated in Stage 2. As shown in Fig. 3 (middle), to train the initialized target model we apply the learning objective that integrates frame-level reconstruction loss and utterance-level speaker loss as formulated below:

$$Loss_{total} = Loss_{frame} + Loss_{spea\,\mathrm{ker}}, \tag{15}$$

where $Loss_{total}$ is the total loss function, $Loss_{frame}$ is the frame-level reconstruction loss formulated in Eq. (11), and $Loss_{speaker}$ is the utterance-level perceptual loss for speaker reconstruction as formulated below:

$$Loss_{spea\,\mathrm{ker}} = MSE\left(\boldsymbol{q}, \boldsymbol{q}'\right), \tag{16}$$

where $\boldsymbol{q}$ and $\boldsymbol{q}'$ are the d-vector speaker representations of target speaker mel-spectrogram Y^{GT} and predicted mel-spectrogam after post-net Y', respectively.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

We train the source model using English, the highest-resourced language, as the source domain. To train the monolingual single-speaker TTS as the first source model we use LJSpeech for English [68], a 24 hours English transcribed speech corpus consisting of clips from a female with a sampling rate of 22050 Hz. This is a practical choice as we have a pre-trained TTS on LJSpeech from our previous study [28]. No gender bias is found in our experiments caused by this choice. Meanwhile, to train the speaker encoder as the second source model we use LibriSpeech-360 dataset [69]. This English speech corpus is a multi-speaker speech dataset recorded with a sampling rate of 16000 Hz and consisting of 921 speakers (482 males and 439 females) with a total duration of about 363.5 hours.

To train zero-shot multilingual multi-speaker TTS as the target model we combine three datasets as the target domain: 1) TITML-IDN for Indonesian [70], a 14.5 hours Indonesian speech corpus recorded at a sample rate of 16000 Hz from 20 Indonesian speakers (11 males and 9 females); 2) OpenSLR jv-ID for Javanese [71], a 7 hours Javanese speech corpus recorded at a sample rate of 48000 Hz from 41 Javanese speakers (21 males and 20 females); OpenSLR su-ID for Sundanese [71], a 5.5 hours Sundanese speech corpus recorded at a sample rate of 48000 Hz from 39 Sundanese speakers (20 males and 19 females).

We divide the data into two categories, seen and unseen speaker set. Seen speaker set is speech data from speakers included in the training data. It consists of 30 speakers with 5 males and 5 females for each language. Our seen speaker data is divided again, 90% for training and 10 % for test. As for unseen speaker set, which is speech data from speakers excluded from the training set, we use 24 speakers with 4 males and 4 females for each language.

An audio pre-processing is applied to change the sample rate into 16000 Hz. For Javanese and Sundanese datasets, some transcriptional improvements are also made. For each utterance, we extract mel-spectrogram as the acoustic feature with 80 mel channels, Hann typed window size 1024, hop size = 256, and 1024-point FFT.

B. NETWORK DETAILS

Our model is implemented in Python and uses the PyTorch library. We leverage open source Tacotron-2 [72] to build our models by adding new networks and adjusting some hyper-parameters. The more detailed TTS setting model can be seen in Table 1. For the vocoder, we use open source NVIDIA WaveGlow [73] with some hyper-parameters adjustments. The speaker encoder utilizes open source [74] with some changes according to our research.

The models are trained in a single NVIDIA DGX-1 GPU using batch size 32, ADAM optimization [75] with default parameters, learning rate from 1e-3 and weight decay 1e-6.

C. BASELINES

We use two models as baseline: multilingual multi-speaker TTS with simple neural speaker embedding, Tdv_mne, and multilingual multi-speaker TTS with pre-trained speaker encoder, Tdv_mse.

1) TDV_MNE

Tdv-mne is an extension of Tacotron-2 [7] by adding a language and speaker encoder to handle multilingual and multi-speaker as in our previous work [19]. The speaker encoder in Tdv_mne is a simple neural embedding to produce speaker representation that is jointly trained with the entire TTS model. This scheme requires an explicit speaker label on the training data. During inference the speaker encoder functions like an embedding lookup table with a speaker label input to produce a speaker representation. Tdv_mne is only able to synthesize speech from speakers seen in the training data and is used as the performance comparison of our proposed system for the seen speaker test.

2) TDV_MSE

Tdv-mse is an extension of the Tacotron-2-based zero-shot multi-speaker TTS [21] by adding a language encoder to handle the multilingual model. The speaker encoder in Tdv_mse is a pre-trained d-vector speaker encoder trained on the ASV. This speaker encoder does not require an explicit speaker label in the training data because it extracts the speaker representation directly from the target speaker mel-spectrogram during model training. Tdv_mse is able to synthesize utterances from speakers that are not in the training data during inference and is used as the performance comparison of our model for both the seen and unseen speaker tests.

The differences between Tdv_mse and our proposed model are: 1) We use d-vector speaker representation of the speaker encoder as input to the pre-net concatenated with the previous mel-spectrum frame; 2) We add a style encoder to the architecture; 3) We add speaker reconstruction loss at training process. Ablation studies on the effects of modifying the model architecture and adding a speaker loss function are discussed in Section V.B.

D. MODEL EVALUATION

To evaluate the TTS models we use subjective assessments involving respondents and objective assessments by measuring the acoustic and speaker features of the synthesized speech signal. Two types of subjective evaluation are used: mean opinion score (MOS) to measure the quality of sound synthesis [76] and differential mean opinion score (DMOS) to measure the speaker similarity [20]. MOS and DMOS were accessed by 15-30 respondents by presenting the same set of speech clips for all models and recorded real human voice. MOS evaluation uses an absolute scale of 1-5 with 0.5 point increase, 1: bad, 2: poor, 3: fair, 4: good, 5: excellent. DMOS evaluation uses an absolute scale of 1-4 with 1 point increase, 1: different-sure, 2:different-not sure, 3:same-not sure, 4:same-sure [77].

Two metrics for objective evaluation are used: mel-cepstral distortion (MCD_K) to measure synthetic speech distortion in the spectrum level [78] and cosine similarity to measure the similarity between the synthesized speech and the target speaker voice. To calculate MCD_K the acoustic feature mel-frequency cepstrum coefficients (MFCCs) are extracted from the synthetic and reference speech signals. Then, we calculate the sum of the first K roots of the coefficients of the MFCCs (without involving the overall energy, $c_{t,0}$) as follows:

$$MCD_k = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum\nolimits_{k=1}^{K}\left(c_{t,k} - c'_{t,k}\right)^2}, \tag{17}$$

where T is the number of frames of both prediction and reference audio, K is the number of MFCC dimensions, $c_{t,k}$ is the k^{th} MFCC coefficient value on the t^{th} frame of the reference audio, $c'_{t,k}$ is the value of the k^{th} MFCC coefficient on the t^{th} frame of the predicted audio.

Speaker space cosine similarity as used in [20], [22] is calculated from the utterance-level d-vector speaker representation of the synthetic sound $\boldsymbol{d}_u$ with the target speaker model $\boldsymbol{d}_{spk}$ with the following formula:

$$\begin{aligned} Cos\left(\boldsymbol{d}_u, \boldsymbol{d}_{spk}\right) &= \frac{\boldsymbol{d}_u \cdot \boldsymbol{d}_{spk}}{\|\boldsymbol{d}_u\|\|\boldsymbol{d}_{spk}\|} \\ &= \frac{\sum_{i=1}^{N}\boldsymbol{d}_{u,i}\cdot\boldsymbol{d}_{spk,i}}{\sqrt{\sum_{i=1}^{N}\boldsymbol{d}_{u,i}^2}\sqrt{\sum_{i=1}^{N}\boldsymbol{d}_{spk,i}^2}}, \end{aligned} \tag{18}$$

where $\boldsymbol{d}_u$ and $\boldsymbol{d}_{spk}$ are N-dimensional vector representations.

V. RESULTS AND ANALYSES

A. COMPARATIVE STUDY ON NETWORK-BASED DEEP TRANSFER LEARNING

Our previous study [28] concluded that alignment learning is an important part in TTS using attention-based encoder decoder. If the model fails to produce a reasonable alignment map between the encoder step and decoder step then it also fails to synthesize intelligible speech. In our current study, we compare the learning process of zero-shot multi-speaker TTS model using three different training schemes: training the model from scratch (FS), training the model using network-based DTL (NB-DTL), and training the model using partial NB-DTL.

The difference between NB-DTL and partial NB-DTL that we use in this comparative study refers to the transferring type of sub-networks from the source model to the target model. NB-DTL scheme only transfers sub-networks of the source models that have the same dimensions with the target model: text encoder and post-net from pre-trained monolingual single-speaker TTS source model and speaker encoder from pre-trained speaker encoder source model (see Fig. 3). Meanwhile, our proposed partial NB-DTL scheme transfers sub-networks of the source models with the same or different dimensions as the target model. Thus, apart from transferring text encoder, speaker encoder, and post-net parameters, the proposed scheme also partially transfers attention, autoregressive decoder, and pre-net parameters as illustrated in Fig. 1(b) and formulated by Algorithm 1.

Fig. 4 shows the learning process comparison of the zero-shot multilingual multi-speaker TTS using the three training schemes. The graph of training and validation loss shows that the more the parameters are transferred from the pre-trained source model to the target model, the lower (better) the loss value and the faster the model achieves convergence.

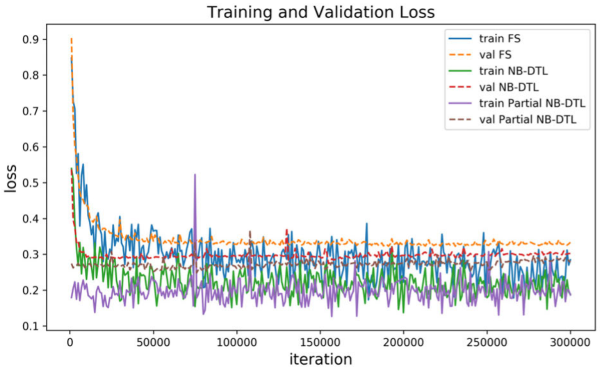


FIGURE 4. Training and validation Loss for Tdv_mse_gst model using three different training methods: training from scratch (FS), network-based deep transfer learning (NB-DTL), and partial network-based deep transfer learning (partial NB-DTL).

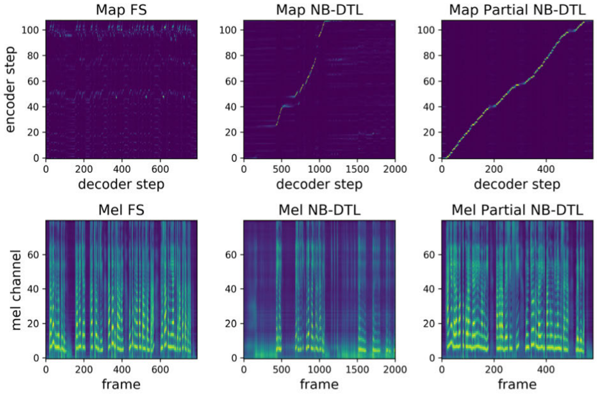


FIGURE 5. Alignment map and mel-spectrogram for Tdv_mse_gst model using three training methods: training from scratch (FS), network-based deep transfer learning (NB-DTL), and partial network-based deep transfer learning (partial NB-DTL).

Fig. 5 shows the alignment map generated by the model trained using each training scheme. The standard training from scratch FS fails to learn the mapping between encoder step and decoder step. NB-DTL scheme is better than FS scheme, but it still has not succeeded in producing a reasonable alignment map. Both schemes are unable to produce an intelligible synthesized speech. Meanwhile, our proposed partial NB-DTL scheme successfully generates a diagonal alignment map and is able to synthesize intelligible speech sound.

B. COMPARATIVE STUDY ON THE PROPOSED ARCHITECTURE AND LOSS FUNCTION

We study the effect of adding style encoder to the TTS model architecture and speaker reconstruction loss in increasing speaker space similarity. We compare the performance of our

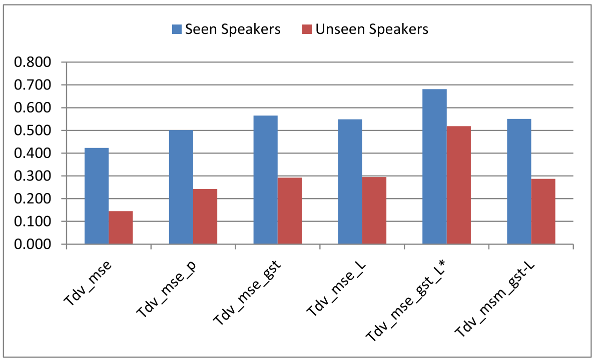


FIGURE 6. **Speaker cosine similarity for various TTS models using D-vector speaker representation.**

proposed model to baseline model Tdv_mse on both seen and unseen speaker test sets. All models are trained using partial NB-DTL scheme on the same training and test sets.

As explained in Section IV.C, our proposed model differs from the baseline model Tdv_mse in two things: the model architecture and the loss function. As for the architecture, a style encoder is added and d-vector speaker embedding is fed into attention- decoder and pre-net. As for the loss function, an additional speaker reconstruction loss is applied together with frame-level reconstruction loss. To find the contribution of each modification to increase speaker similarity, we conduct experiments with the results reported in Fig. 6.

There are six different models: 1) Tdv_mse, a baseline model trained using frame-level reconstruction loss; 2) Tdv_mse_p, a baseline model with added speaker embedding before pre-net and trained using frame-level reconstruction loss; 3) Tdv_mse_gst, our proposed model trained using frame-level reconstruction loss; 4) Tdv_mse_L, a baseline model trained using frame-level reconstruction loss and utterance-level speaker reconstruction loss; 5) Tdv_mse_gst_L, our proposed model trained using frame-level reconstruction loss and utterance-level speaker reconstruction loss; 6) Tdv_msm_gst_L, similar to our proposed model except it uses speaker model lookup to obtain the speaker representation as in recent work [20] instead of using utterance-level speaker embedding.

Fig. 6 shows that applying additional networks increases the speaker space similarity in both seen and unseen speaker test sets. Each modification contributes some level of improvement. Comparing Tdv_mse and Tdv_mse_p shows that adding speaker embedding into pre-net slightly increases the speaker similarity. From comparing Tdv_mse_gst and Tdv_mse_p we can see that adding a style encoder to obtain speaking style representation of target speaker for conditioning mel-spectrogram decoder also increases the speaker similarity. Meanwhile, from the comparison between Tdv_mse and Tdv_mse_L as well as between Tdv_mse_gst and Tdv_mse_gst_L we can see that applying speaker reconstruction loss contributes even more to increasing the speaker similarity. Applying the same proposed loss function in both Tdv_mse_L and Tdv_mse_gst_L, we can see that adding style encoder and speaker embedding into pre-net plays the most significant role in the speaker similarity improvement.

Similar to recent work [20], we also experiment using a speaker model as formulated in Eq.(3) that is fed to mel-spectrogram decoder network during the training process. However, Tdv_msm_gst_L that uses the speaker model lookup gives a worse performance than Tdv_mse_gst_L that uses utterance-level d-vector speaker encoder as formulated in Eq.(2).

Overall, our proposed model and loss function, Tdv_mse_gst_L, can significantly increase the speaker cosine similarity compared to the baseline model Tdv_mse. For the rest of the paper, we denote Tdv_mse_gst_L as the proposed TTS model for zero-shot speaker adaptation.

C. SPEECH QUALITY EVALUATION

In this section, we present a more detailed speech quality evaluation using MCD and MOS for our proposed model as well as the baseline models Tdv_mne and Tdv_mse. All models are trained using partial network-based DTL. In addition to dividing the test categories into seen and unseen speaker test, we also divide them into native and foreign language test. Native language test refers to target speakers of a language used to synthesize text in the same language, while foreign language test refers to target speakers of a language used to synthesize text in different languages. An example of the latter is when target Indonesian speakers are used to synthesize texts in Javanese and Sundanese.

1) MEL-CEPSTRUM DISTORTION

We use MCD_{13} with $K = 13$ features of MFCC to measure the distortion between the synthesis and the reference signal which includes the pronunciation of vowels and consonants. For native language we use the same speaker for both signals. For foreign language we compare the synthetic speech from a foreign speaker with the reference speech of a native speaker on the same text. We use dynamic time warping (DTW) to match the lengths between both signals.

Table 2 presents the MCD evaluation for the baseline models and the proposed model trained using our proposed strategy. Overall, the models give a low distortion rate. These results show the effectiveness of our partial network-based DTL. Using this training scheme all models can successfully synthesize intelligible speech using a small amount of training data of the target languages. We do not report the models trained using the standard training since they fail to produce understandable speech.

The MCD evaluation shows that there is no significant difference between baseline models and our proposed model, between seen and unseen speakers, as well as between native and foreign languages. Different from most cases for other languages, the syntheses of Sundanese speakers in native language have the average MCD value of seen test that is slightly bigger than that of unseen test. However, from the confidence interval presented using a 95% confidence level

TABLE 2. MCD evaluation.

LANG	MODEL	SEEN SPEAKERS				UNSEEN SPEAKERS			
		ID	JV	SU	ALL	ID	JV	SU	ALL
Native	Tdv_mne	2.508±1.088	8.114±1.303	2.801±0.788	4.474±0.689	-	-	-	-
	Tdv_mse	2.212±0.957	9.736±1.497	3.085±1.132	5.011±0.771	4.511±1.926	13.172±1.195	1.916±1.101	6.533±0.856
	Proposed	1.900±0.815	8.064±1.384	2.564±0.785	4.176±0.671	4.512±1.938	12.007±1.104	1.862±0.778	6.127±0.829
Foreign	Tdv_mne	9.435±0.854	3.836±1.453	8.243±1.415	7.171±0.753	-	-	-	-
	Tdv_mse	9.403±0.902	3.789±1.484	7.579±1.225	6.924±0.724	8.826±0.964	3.500±0.182	9.861±1.800	7.396±0.713
	Proposed	9.222±0.817	3.873±1.489	7.820±1.337	6.972±0.739	8.422±0.997	3.697±0.191	9.569±1.742	7.230±0.695

TABLE 3. MOS for speech naturalness.

LANG	MODEL	SEEN SPEAKERS				UNSEEN SPEAKERS			
		ID	JV	SU	ALL	ID	JV	SU	ALL
Native	GT	4.247±0.052	4.684±0.057	4.409±0.046	4.447±0.031	4.379±0.052	4.674±0.048	4.387±0.048	4.480±0.029
	Tdv_mne	4.164±0.096	4.527±0.089	4.316±0.088	4.335±0.055	-	-	-	-
	Tdv_mse	4.025±0.102	4.440±0.117	4.294±0.087	4.224±0.062	4.061±0.080	4.394±0.098	4.346±0.071	4.267±0.050
	Proposed	4.182±0.102	4.383±0.114	4.397±0.087	4.321±0.060	4.197±0.073	4.442±0.089	4.317±0.067	4.318±0.044
Foreign	Tdv_mne	4.288±0.087	4.150±0.086	4.171±0.094	4.203±0.052	-	-	-	-
	Tdv_mse	4.372±0.081	4.286±0.080	4.323±0.088	4.327±0.048	4.210±0.072	4.288±0.060	4.369±0.070	4.289±0.038
	Proposed	4.238±0.096	4.264±0.081	4.246±0.094	4.249±0.053	4.273±0.070	4.303±0.063	4.378±0.066	4.318±0.038

we can say that there is no difference between both MCD values with a 0.05 level of significance.

The MCD evaluation shows that the syntheses on Javanese have a worse distortion than that of Indonesian or Sundanese. This can be seen from native language for Javanese speakers with MCD value of 8.064 and 12.007 for seen and unseen speakers, respectively. Foreign language test containing Javanese syntheses by Indonesian/Sundanese speakers also gives a worse distortion than those on the other languages by Javanese speakers. These results show that the language factor contributes more to the spectral distortion than the speaker factor. The fact that Javanese has slightly different grapheme-to-phoneme rules compared to the other languages may cause the worse distortion. For example, the vowel "a" in Indonesian/Sundanese is pronounced as the phoneme /a/, but in Javanese it can be pronounced as /ɔ/ or /a/ depending on the words. All models seem unable to capture this well.

2) MOS FOR NATURALNESS

For the MOS test, respondents were asked to listen to a set of speech utterances generated by the baseline models, the proposed model, and the original human voice (ground truth). The speech utterances were presented in random order without any label indicating the source of the sounds. Each respondent assessed the quality of the audio recordings from the four different sources. This indirect comparison test is an adequate tool to determine the relative quality of the speech audio among the models and the real human voice from the respondent's perspective.

The MOS evaluation results are presented in Table 3. In general there is no significant difference between the proposed model and the baseline models from respondents' subjective assessments. Overall, all models are able to synthesize speech with a good quality that is not far from the real human voice. These results prove that all models trained using the proposed partial network-based DTL successfully synthesize intelligible natural speeches.

D. SPEAKER SIMILARITY EVALUATION

The speaker similarity evaluation between the synthesized speech and the real target speaker speech includes objective evaluation cosine similarity and subjective evaluation DMOS. We present the value with 95% confidence level.

1) SPEAKER COSINE SIMILARITY

To measure the speaker similarity of the synthesized speech sound and the target speaker's speech sound, we compare the utterance-level d-vector speaker representation of the synthesized sound to the d-vector speaker model of the same target speaker using cosine similarity. A higher value indicates a better system performance.

Table 4 presents the evaluation of cosine similarity for the baseline models and the proposed one. Our modification on the model architecture and loss function is proven to significantly improve speaker similarity, especially in the unseen speaker test where the baseline Tdv_mse model has poor performance. Our model is able to increase unseen speaker similarity by 0.468 in native language (from 0.196 in the baseline model to 0.664 in our model) and 0.279 in foreign language (from 0.094 in the baseline model to 0.373 in our model). Overall seen speaker test also shows that our model has the best cosine similarity compared to both baseline models. Baseline model Tdv_mse which uses a pre-trained speaker encoder is worse than Tdv_mne which uses a simple neural speaker embedding that is trained together with the TTS system. Meanwhile, our model with additional style encoder and

TABLE 4. Speaker cosine similarity.

LANG	MODEL	SEEN SPEAKERS				UNSEEN SPEAKERS			
		ID	JV	SU	ALL	ID	JV	SU	ALL
Native	GT	0.981±0.004	0.964±0.009	0.963±0.008	0.970±0.004	0.985±0.003	0.972±0.005	0.966±0.008	0.975±0.002
	Tdv_mne	0.626±0.054	0.792±0.040	0.703±0.046	0.707±0.028	-	-	-	-
	Tdv_mse	0.699±0.052	0.741±0.052	0.744±0.048	0.731±0.030	0.249±0.043	0.196±0.041	0.144±0.036	0.196±0.025
	Proposed	**0.977±0.006**	**0.865±0.037**	**0.837±0.051**	**0.893±0.022**	**0.891±0.027**	**0.548±0.034**	**0.554±0.047**	**0.664±0.016**
Foreign	Tdv_mne	0.238±0.053	**0.503±0.073**	0.403±0.058	0.381±0.038	-	-	-	-
	Tdv_mse	0.033±0.027	0.218±0.051	0.093±0.026	0.115±0.023	0. 059±0.029	0.162±0.040	0.061±0.034	0.094±0.021
	Proposed	**0.593±0.055**	0.412±0.048	0.405±0.051	**0.470±0.031**	**0.371±0.055**	**0.415±0.041**	**0.333±0.063**	**0.373±0.031**

TABLE 5. DMOS for speaker similiraty.

LANG	MODEL	SEEN SPEAKERS				UNSEEN SPEAKERS			
		ID	JV	SU	ALL	ID	JV	SU	ALL
Native	Tdv_mne	3.235±0.088	3.500±0.077	3.373±0.098	3.369±0.051	-	-	-	-
	Tdv_mse	3.285±0.077	3.350±0.093	3.567±0.077	3.401±0.049	2.527±0.090	2.466±0.091	2.933±0.094	2.642±0.054
	Proposed	**3.360±0.083**	**3.660±0.066**	3.587±0.076	**3.536±0.044**	**2.827±0.083**	**3.125±0.086**	**3.089±0.088**	**3.014±0.050**
Foreign	Tdv_mne	2.688±0.080	2.868±0.075	3.033±0.069	2.863±0.043	-	-	-	-
	Tdv_mse	2.341±0.082	2.530±0.084	2.645±0.075	2.505±0.047	1.992±0.067	2.484±0.067	2.685±0.061	2.387±0.039
	Proposed	**2.783±0.086**	2.865±0.078	3.035±0.072	**2.894±0.045**	**2.353±0.068**	**2.737±0.068**	**2.941±0.058**	**2.677±0.038**

speaker reconstruction loss outperforms Tdv_mne. Overall, our model provides the best speaker similarity for both the seen and unseen speaker tests as well as the native and foreign language test categories.

2) DMOS FOR SPEAKER SIMILARITY

For the DMOS test, respondents were asked to compare speaker similarity of a pair of speech sounds. One is generated by the TTS models and the other is the real target speaker voice. No label of speech source is presented. Similar to the MOS test, the speech pair sequences for DMOS test were also presented in random order. Thus, this evaluation can be considered as an indirect comparison evaluation between models to determine the relative speaker similarity among the models compared to the real target speaker speech from the respondent's perspective.

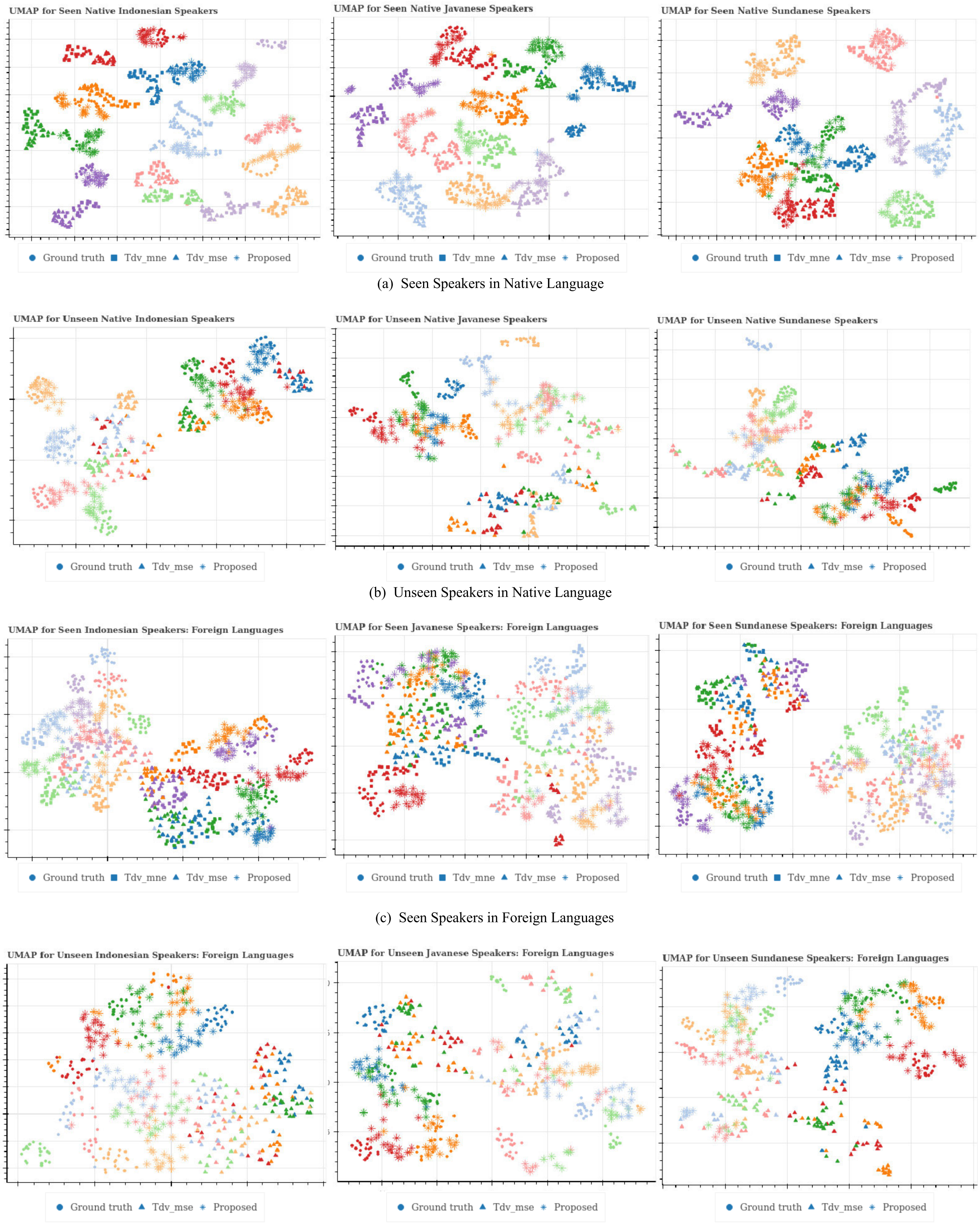
The DMOS results are presented in Table 5. Overall, the proposed model received the highest speaker similarity rating from respondents on both seen and unseen speaker test as well as on both native and foreign languages test. The foreign language test has a lower speaker similarity rating than the native language test. The respondents' assessments are in line with the objective cosine similarity that our proposed model is able to increase speaker similarity. However, the DMOS values in foreign language are still low (below 3.00). It seems that the respondents tend to be indecisive for synthesized speech with low cosine similarity (below 0.600).

E. SPEAKER SPACE VISUALISATION

To visualize d-vector speaker representation we use uniform manifold approximation and projection (UMAP) [79] to project high-dimensional speaker space into two-dimensional space. UMAP is a dimensional reduction technique that can be used for visualization similar to t-SNE [80]. UMAP is faster than t-SNE and scaling well in terms of both dataset size and dimensionality.

Fig. 7 visualizes the speaker space generated by our proposed model, baseline models, and the target speaker's voice. Fig. 7 (a) and (b) present native language results for seen and unseen speakers respectively for each languages, Indonesian, Javanese, and Sundanese. These figures show that our proposed model can synthesize speech close to the real target speaker space for both the seen and the unseen speakers test. Meanwhile, speaker speech space synthesized by the baseline models is far from the target speaker space. Even worse, some speeches of unseen target female speakers generated by the baseline model (triangle plots) are found to be close to the target male speaker space, and vice versa. This is not the case for our proposed model that is able to generate speech sounds (asterisk plots) quite close to the target speaker space for unseen speakers (circle plots), although not as close as for seen speakers.

Fig. 7 (c) and (d) visualize speaker space of synthetic speech in foreign language for both seen and unseen speakers respectively. However, since our dataset all speakers are not polyglot, i.e. each speaker only speaks in one native language, we use target speakers' speech in their native language as the ground truth for comparison. From this figure the speaker space similarity of all models in foreign language is lower than the one in the native language. The distance of unseen speaker space of the baseline model Tdv_mse in native language, which is already far from the target speaker space, is even farther in foreign languages. Some triangle plots of unseen female speakers are very far from the ground truth speaker space. They are closer to the unseen male speaker



(a) Seen Speakers in Native Language

(b) Unseen Speakers in Native Language

(c) Seen Speakers in Foreign Languages

(d) Unseen Speakers in Foreign Languages

FIGURE 7. D-vector visualisation for both seen and unseen speakers in both native and foreign languages for each language, Indonesian (left), Javanese (middle), and Sundanese (right). Different colors indicate different speakers. Bold colors indicate female speakers while soft colors indicate male speakers.

space, as seen in the red and blue plots in Fig. 7 (d). This is not the case for our proposed model.

Overall, our proposed model gives a closer speaker space (asterisk plots) to the target speaker space (circle plots) than that of the baseline models (triangle/square plots) for both seen and unseen speakers in both native and foreign languages.

VI. CONCLUSION

This study develops a zero-shot multilingual multi-speaker Tacotron-2-based TTS model for low resource languages. Given a text sequence and a few seconds of target speaker's speech sample, our model is able to synthesize speech sound for both seen and unseen target speakers in both native and foreign languages. We use a pre-trained d-vector speaker encoder to produce a d-vector speaker embedding used for the speaker adaptation process.

We address two main problems in this study: 1) handling low resources problem; 2) improving zero-shot speaker adaptation, especially for unseen speakers. To solve low-resource problem we propose partial network-based DTL from a pre-trained monolingual single-speaker TTS in high-resource languages as the first source model to our target model. We also use a pre-trained d-vector speaker encoder for ASV task in a high resource language as the second source model. To improve the performance of zero-shot speaker adaptation, we add a style encoder to get style embedding from the target speaker speech sample to condition the mel-spectrogram decoder. Moreover, we combine utterance-level speaker loss with frame-level mel-spectrogram reconstruction loss during fine-tuning the model in low-resource target languages.

Experiments in Indonesian, Javanese, and Sundanese show that our proposed partial network-based DTL strategy is an effective strategy for training the models in low-resource languages. This strategy opens the opportunity for low-resource languages to be researched and developed using the recent advanced deep learning technology which generally requires a large amount of training data. This paper presents an example of using this opportunity to enhance Tacotron-2 architecture for zero-shot multilingual multi-speaker adaption. Our experiments show that our proposed model and loss function significantly improve the performance of zero-shot speaker adaptation for both seen and unseen speakers in both native and foreign languages.

The objective evaluation of mel-spectrum distortion shows that synthesized speech in Javanese gives the worst distortion. This may be related to the fact that the grapheme-to-phoneme rules in Javanese have a slightly lower similarity compared to that of Indonesian and Sundanese. Therefore, the benefit of using joint multilingual dataset for Javanese is also lower than for Indonesian and Sundanese. To address the kind of language related problem, for future work apart from leveraging transfer learning we plan to combine machine learning with linguistic knowledge. We are interested in using language-specific information, such as a phonetic dictionary, combined with the latest TTS technology for further development of more than 700 low-resource ethnic languages in Indonesia. The use of a dataset containing multilingual speakers where a speaker speaks several languages can also be explored further to improve zero-shot speaker adaptation in foreign languages.

ACKNOWLEDGMENT

The authors would like to thank Mirna Adriani for her support and useful insight in the topic, also would like to thank Tokopedia-UI AI Center team and Lab 1231 Fasilkom UI team, especially Machmud Roby Alhamidi and Grafika Jati, for their helpful discussion and feedback, and also would like to thank appreciate Ken Sasongko for his contribution during survey process.

REFERENCES

[1] W. B. Kleijn, F. S. C. Lim, A. Luebs, J. Skoglund, F. Stimberg, Q. Wang, and T. C. Walters, "WaveNet based low rate speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 676–680.

[2] A. Van Den Oord *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 9, 2018, pp. 6270–6278.

[3] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2017, pp. 264–273.

[4] S. O. Arik, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 2963–2971.

[5] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[6] 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. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPECH)*, Aug. 2017, pp. 4006–4010.

[7] 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)*, Apr. 2018, pp. 4779–4783.

[8] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6905–6909.

[9] Y. Zheng, J. Tao, Z. Wen, and J. Yi, "Forward–backward decoding sequence for regularizing end-to-end TTS," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2067–2079, Dec. 2019.

[10] Y. Liu and J. Zheng, "Es-Tacotron2: Multi-task Tacotron 2 with pre-trained estimated network for reducing the over-smoothness problem," *Information*, vol. 10, no. 4, p. 131, Apr. 2019.

[11] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Appl. Sci.*, vol. 9, no. 19, p. 4050, Sep. 2019.

[12] E. Nachmani, A. Polyak, Y. Taigman, and L. Wolf, "Fitting new speakers based on a short untranscribed sample," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, vol. 8, 2018, pp. 5932–5940.

[13] J. Park, K. Zhao, K. Peng, and W. Ping, "Multi-speaker end-to-end speech synthesis," 2019, *arXiv:1907.04462*.

[14] 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. 7th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–16.

[15] Y. Deng, L. He, and F. Soong, "Modeling multi-speaker latent space to improve neural TTS: Quick enrolling new speaker and enhancing premium voice," 2019, *arXiv:1812.05253v4*.

[16] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain with one-shot speaker adaptation," 2018, *arXiv:1803.10525*.

[17] H.-T. Luong and J. Yamagishi, "Scaling and bias codes for modeling speaker-adaptive DNN-based speech synthesis systems," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 610–617.

[18] Q. Hu, E. Marchi, D. Winarsky, Y. Stylianou, D. Naik, and S. Kajarekar, "Neural text-to-speech adaptation from low quality public recordings," in *Proc. 10th ISCA Workshop Speech Synth. (SSW)*, Sep. 2019, pp. 24–28.

[19] K. Azizah, M. Adriani, and W. Jatmiko, "Hierarchical transfer learning for multilingual, multi-speaker, and style transfer DNN-based TTS on low-resource languages," *IEEE Access*, vol. 8, pp. 179798–179812, 2020.

[20] E. Cooper, C.-I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6184–6188.

[21] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, and F. Ren, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 4485–4495.

[22] Z. Liu and B. Mak, "Cross-lingual multi-speaker text-to-speech synthesis for voice cloning without using parallel corpus for unseen speakers," 2019, *arXiv:1911.11601*.

[23] Z. Cai, C. Zhang, and M. Li, "From speaker verification to multispeaker speech synthesis, deep transfer with feedback constraint," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Oct. 2020, pp. 3974–3978.

[24] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A survey on recent approaches for natural language processing in low-resource scenarios," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT)*, 2021, pp. 2545–2568.

[25] A. K. Singh, "Natural language processing for less privileged languages: Where do we come from? Where are we going?" in *Proc. IJCNLP Workshop NLP Less Privileged Lang.*, Jan. 2008, pp. 7–12.

[26] C. Cieri, M. Maxwell, S. Strassel, and J. Tracey, "Selection criteria for low resource language programs," in *Proc. 10th Int. Conf. Lang. Resour. Eval.*, 2016, pp. 4543–4549.

[27] S. Ruder. (2019). *The 4 Biggest Open Problems in NLP*. [Online]. Available: https://ruder.io/4-biggest-open-problems-in-nlp/

[28] K. Azizah and M. Adriani, "Hierarchical transfer learning for text-to-speech in Indonesian, Javanese, and Sundanese languages," in *Proc. 12th Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, Oct. 2020, pp. 421–428.

[29] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4879–4883.

[30] E. Bender. (2019). The #BenderRule: On naming the languages we study and why it matters. The Gradient. [Online]. Available: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/%0AHigh

[31] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, and D. Warde-farley, "Generative adversarial nets," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 2672–2680.

[32] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, 2017, pp. 1126–1135.

[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[34] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, 2016.

[35] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proc. 27th Int. Conf. Artif. Neural Netw. (ICANN)*, 2018, pp. 2672–2680.

[36] A. Magueresse, V. Carles, and E. Heetderks, "Low-resource languages: A review of past work and future challenges," 2020, *arXiv:2006.07264*.

[37] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. J. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6940–6944.

[38] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 301–308.

[39] D.-R. Liu, C.-Y. Yang, S.-L. Wu, and H.-Y. Lee, "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 640–647.

[40] S. Novitasari, A. Tjandra, S. Sakti, and S. Nakamura, "Cross-lingual machine speech chain for Javanese, Sundanese, Balinese, and Bataks speech recognition and synthesis," in *Proc. 1st Joint Spoken Lang. Technol. Under-Resour. Lang. (SLTU)*, May 2020, pp. 131–138.

[41] Y.-J. Chen, T. Tu, C.-C. Yeh, and H.-Y. Lee, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 2075–2079.

[42] R. V. Pawar, R. M. Jalnekar, and J. S. Chitode, "Review of various stages in speaker recognition system, performance measures and recognition toolkits," *Anal. Integr. Circuits Signal Process.*, vol. 94, no. 2, pp. 247–257, Feb. 2018.

[43] S. P. Todkar, S. S. Babar, R. U. Ambike, P. B. Suryakar, and J. R. Prasad, "Speaker recognition techniques: A review," in *Proc. 3rd Int. Conf. for Converg. Technol. (I2CT)*, Apr. 2018, pp. 1–5.

[44] D. Sztahó, G. Szaszák, and A. Beke, "Deep learning methods in speaker recognition: A review," *Periodica Polytechnica Elect. Eng. Comput. Sci.*, vol. 65, no. 4, pp. 310–328, 2021, doi: 10.3311/PPee.17024.

[45] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, L. P. García-Perera, D. Povey, P. A. Torres-Carrasquillo, S. Khudanpur, and N. Dehak, "State-of-the-art speaker recognition for telephone and video speech: The JHU-MIT submission for NIST SRE18," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 1488–1492.

[46] S. Sreedharan and C. Eswaran, "A review on speaker verification: Challenges and issues," *Int. J. Sci. Technol. Res.*, vol. 8, no. 8, pp. 956–960, 2019.

[47] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4052–4056.

[48] L. Li, D. Wang, Z. Zhang, and T. F. Zheng, "Deep speaker vectors for semi text-independent speaker verification," *J. Latex Class Files*, vol. 13, no. 9, pp. 1–5, 2015.

[49] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: An end-to-end neural speaker embedding system," 2017, *arXiv:1705.02304*.

[50] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Aug. 2017, pp. 999–1003.

[51] J. Jung, H. Heo, I. Yang, S. Yoon, H. Shim, and H. Yu, "D-vector based speaker verification system using raw waveform CNN," in *Proc. Int. Seminar Artif. Intell., Netw. Inf. Technol. (ANIT)*, vol. 150, 2018, pp. 126–131.

[52] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5329–5333.

[53] D. Garcia-Romero, D. Snyder, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "X-vector DNN refinement with full-length recordings for speaker recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 1493–1496.

[54] A. Kanagasundaram, S. Sridharan, G. Sriram, S. Prachi, and C. Fookes, "A study of x-vector based speaker recognition on short utterances," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 2943–2947.

[55] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2019, pp. 1652–1656.

[56] N. Dehak, P. J. Kenny, R. Dehak, D. Pierre, and O. Pierre, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[57] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2015, pp. 92–97.

[58] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2018, pp. 3623–3627.

[59] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. 35th Int. Conf. Mach. Learn. (PMLR)*, vol. 80, 2018, pp. 5180–5189.

[60] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9906, 2016, pp. 694–711.

[61] F. G. Germain, Q. Chen, and V. Koltun, "Speech denoising with deep feature losses," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 2723–2727.

[62] S. Kataria, P. S. Nidadavolu, J. Villalba, N. Chen, P. Garcia-Perera, and N. Dehak, "Feature enhancement with deep feature losses for speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7584–7588.

[63] M. Kawanaka, Y. Koizumi, R. Miyazaki, and K. Yatabe, "Stable training of DNN for speech enhancement based on perceptually-motivated black-box cost function," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 7524–7528.

[64] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep learning-based objective assessment for voice conversion," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Sep. 2019, pp. 1541–1545.

[65] K.-S. Lee, "Voice conversion using a perceptual criterion," *Appl. Sci.*, vol. 10, no. 8, p. 2884, Apr. 2020.

[66] R. Liu, B. Sisman, G. Gao, and H. Li, "Expressive TTS training with frame and style reconstruction loss," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1806–1818, 2021.

[67] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3617–3621.

[68] K. Ito. (2017). *The LJ Speech Dataset 2017*. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[69] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[70] D. P. Lestari, K. Iwano, and S. Furui, "A large vocabulary continuous speech recognition system for Indonesian language," in *Proc. 15th Indonesian Sci. Conf. Jpn.*, 2006, pp. 17–22.

[71] K. Sodimana, P. De Silva, S. Sarin, O. Kjartansson, M. Jansche, K. Pipatsrisawat, and L. Ha, "A step-by-step process for building TTS voices using open source data and frameworks for Bangla, Javanese, Khmer, Nepali, Sinhala, and Sundanese," in *Proc. 6th Workshop Spoken Lang. Technol. Under-Resour. Lang. (SLTU)*, Aug. 2018, pp. 66–70.

[72] NVIDIA. (2018). *Tacotron 2 Without WaveNet*. [Online]. Available: https://github.com/NVIDIA/tacotron2

[73] NVIDIA. (2018). *WaveGlow*. [Online]. Available: https://github.com/NVIDIA/WaveGlow

[74] Mozilla. (2020). *Speaker Encoder*. [Online]. Available: https://github.com/mozilla/TTS/tree/master/TTS/speaker_encoder

[75] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[76] Y. Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Nov./Dec. 2003, pp. 577–582.

[77] 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," in *Proc. Speaker Lang. Recognit. Workshop (Odyssey)*, Jun. 2018, pp. 195–202.

[78] R. F. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (CCSP)*, vol. 1, May 1993, pp. 125–128.

[79] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

[80] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE Laurens," *J. Mach. Learn. Res.*, vol. 9, no. 1, pp. 2579–2605, 2008.

KURNIAWATI AZIZAH (Member, IEEE) received the B.S. degree in informatics engineering from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 1997, and the M.Phil. degree in computer speech, text, and internet technology (CSTIT) from the University of Cambridge, Cambridge, U.K., in 2006. She is currently pursuing the Ph.D. degree in computer science with Universitas Indonesia, Jakarta.

She was an IT Consultant with MINCOM Indoservices, Jakarta, Indonesia, from 1998 to 2000, and with Switchlab, London, U.K., from 2000 to 2017. Since 2008, she has been a Lecturer with the Faculty of Computer Science, Universitas Indonesia. Her research interests include deep learning, natural language processing (NLP), speech processing, and computer vision.

WISNU JATMIKO (Senior Member, IEEE) received the B.S. degree in electrical engineering and the M.Sc. degree in computer science from Universitas Indonesia, Jakarta, Indonesia, in 1997 and 2000, respectively, and the Dr. Eng. degree from Nagoya University, Japan, in 2007.

He works as a Full Professor with the Faculty of Computer Science, Universitas Indonesia. His research interests include autonomous robot, optimization, real-time traffic monitoring systems, machine learning, and artificial intelligence. He was the Chair of IEEE Indonesia Section, in 2019 and 2020.