GC-TTS: Few-shot Speaker Adaptation with Geometric Constraints

Ji-Hoon Kim¹, Sang-Hoon Lee², Ji-Hyun Lee¹, Hong-Gyu Jung², and Seong-Whan Lee¹

Abstract—Few-shot speaker adaptation is a specific Text-to-Speech (TTS) system that aims to reproduce a novel speaker’s voice with a few training data. While numerous attempts have been made to the few-shot speaker adaptation system, there is still a gap in terms of speaker similarity to the target speaker depending on the amount of data. To bridge the gap, we propose GC-TTS which achieves high-quality speaker adaptation with significantly improved speaker similarity. Specifically, we leverage two geometric constraints to learn discriminative speaker representations. Here, a TTS model is pre-trained for base speakers with a sufficient amount of data, and then fine-tuned for novel speakers on a few minutes of data with two geometric constraints. Two geometric constraints enable the model to extract discriminative speaker embeddings from limited data, which leads to the synthesis of intelligible speech. We discuss and verify the effectiveness of GC-TTS by comparing it with popular and essential methods. The experimental results demonstrate that GC-TTS generates high-quality speech from only a few minutes of training data, outperforming standard techniques in terms of speaker similarity to the target speaker.

Index Terms—speech synthesis, few-shot learning, speaker adaptation, fine-tuning, speaker embedding

I. INTRODUCTION

Text-to-Speech (TTS) system aims to convert a normal text into human speech. Conventional TTS pipelines are based on a number of complex stages, each of which requires laborious feature engineering such as text normalization [1], [2]. After the emergency of Tacotron [3] and DeepVoice [4], TTS model can be simplified and generate natural speech from text inputs in an end-to-end manner. While recent TTS models achieve the synthesis of intelligible speech in an end-to-end manner, training these system requires tons of high-quality data collected in controlled settings [5]–[11]. Given this high cost, there have been increasing demands for a few-shot speaker adaptation system which is a technique to function with a few amount of data.

Existing studies for a few-shot speaker adaptation can be categorized into two categories. One is to utilize an independent neural encoder to extract speaker embeddings which are then combined with linguistic features of a pre-trained TTS model [12]–[14]. While these approaches lead to a simple and efficient adaptation, they result in poor adaptation quality, especially in speaker similarity to the target speaker. Another is to fine-tune a pre-trained TTS model to novel speakers with a small amount of adaptation data [15]–[19]. They outperform the former one in terms of speaker similarity. However, the similarity is still lower than when the model is trained with a large amount of training data. Moreover, it is prone to destroy the speaker embedding space well trained for base speakers [20], [21], resulting in degraded generation performance for base speakers.

In this paper, we propose GC-TTS, a few-shot speaker adaptive TTS framework which achieves high-quality speaker adaptation for novel speakers while preserving the ability to synthesize natural speech for base speakers. To achieve this, GC-TTS leverages two geometric constraints [22] which shows impressive results in few-shot image classification. GC-TTS is trained in two stages. In training stage 1, we first train the model for base speakers with a sufficient amount of data (i.e. 10 hours or more). In training stage 2, we continue to train the model for novel speakers on a few minutes of data with two geometric constraints. The first constraint, when used with cross-entropy loss, enables speaker embeddings of utterances from the same speaker to have a small angular distance. The second constraint makes those of utterances from different speakers have large angular distances in the embedding space. Moreover, we only update parts of network parameters in training stage 2 so that we can prevent overfitting and fully utilize the prior knowledge in an effective manner.

While prior studies have only focused on generating intelligible speech for novel speakers, to the best of our knowledge, this is the first attempt to adapt a TTS model to novel speakers without diminishing generation performance for base speakers. The experimental results indicate that GC-TTS is capable of synthesizing improved speech over standard techniques, especially in speaker similarity to the target speaker. In addition, visualization of the speaker embedding space shows GC-TTS constructs discriminative speaker embeddings of novel speakers without contaminating the speaker embedding space well trained for base speakers. The synthesized audio samples are presented at https://7/prml-lab-speech-team.github.io/demo/GC-TTS

II. MODEL ARCHITECTURE

A. Multi-Speaker TTS System

The model architecture of GC-TTS is rooted in Tacotron2 [6] with the extension of a neural speaker encoder and a classification layer, as illustrated in Fig. 1. The speaker
encoder takes a log mel-spectrogram of the target speaker as input and computes the corresponding speaker embedding which is then concatenated with the Tacotron2 encoder output that the decoder will attend to. We follow the structure in [23] for the speaker encoder which comprises six convolution blocks followed by Gated Recurrent Unit (GRU) layer [24], and a Fully Connected (FC) layer. Each convolution block has 2d convolutional layer, batch normalization [25] and relu activation [26]. We also add a classification layer that takes a speaker embedding as an input and predicts the corresponding speaker label. By applying two geometric constraints, speaker embeddings of utterances from the same speaker cluster near the corresponding classification weight, and each classification weight of a novel speaker is separated from the classification weights of the other speakers.

B. Two-Stage Training Procedure

We use a two-stage training procedure. In training stage 1, GC-TTS is pre-trained for base speakers with a large amount of training data. Here, the speaker encoder is trained jointly with the Tacotron2’s reconstruction loss and stop token prediction loss [6]. To train a classifier, we use the softmax layer followed by cross-entropy loss defined as follows:

$$L_{cls} = -\frac{1}{M} \sum_{i=1}^{M} \log \frac{\exp(e_i^T w_{y_i})}{\sum_{j=1}^{n} \exp(e_i^T w_j)}$$  \hspace{1cm} (1)

where $M$ represents a batch size, $n$ refers to the number of training examples, $y_i$ represents the speaker label of the $i$th example, $e_i$ is the speaker embedding computed from the $i$th example, and $w$ represents the weights of classifier.

In training stage 2, we fine-tune the pre-trained network for novel speakers with a few amount of adaptation data. We first freeze the specific modules of the model trained for base speakers in training stage 1. The intuition for this is that the more trainable parameters it has, the more it is vulnerable to overfitting. Following [18], we freeze the weights of the Tacotron2 encoder and the attention module since they are already well trained across diverse texts and multiple speakers with various prosodies. Additionally, we freeze the weights of $K$ bottom convolution blocks in the speaker encoder. This is due to the fact that the bottom convolution blocks tend to learn general features across various speakers [27], [28]. In practice, we find that a $K$ of 4 can lead to the most robust synthesis as we will validate it in Sec V-C.

After freezing the parts of the network, we replicate the other modules to be fine-tuned and build a novel classifier [29]. Based on this network, we employ two geometric constraints in training stage 2, which are described in Section III. The adaptation data of novel speakers is forwarded to the frozen parts of the network, the replicated parts of the network, and the novel classifier. Since the base modules aren’t optimized during the adaptation, the
TABLE I
EVALUATION RESULTS. MOS-NAT. AND MOS-SIM. REFER TO MOS FOR NATURALNESS AND SIMILARITY, RESPECTIVELY. LOWER IS BETTER FOR MCD13 AND RMSE10, AND HIGHER IS BETTER FOR THE OTHER METRICS. MOS RESULTS ARE PRESENTED WITH 95% CONFIDENCE INTERVALS. 1m AND 5m REPRESENT 1 AND 5 MINUTES OF ADAPTATION DATA, RESPECTIVELY. VALUES IN BOLD REPRESENT THE BEST RESULTS FOR EACH METRIC.

| Method         | MOS-Nat.(↑) | MOS-Sim.(↑) | MCD13(↓) | RMSE10(↓) | Cos.-Sim. (↑) | Acc. (↑) |
|----------------|-------------|-------------|----------|-----------|--------------|----------|
| Ground Truth   | 4.05 ± 0.03 | 4.29 ± 0.02 | 2.903    | 35.153    | 0.9723       | 99.75%   |
| Vocoder        | 3.95 ± 0.04 | 4.19 ± 0.04 | 4.869    | 37.630    | 0.8942       | 92.63%   |
| Tacotron2 [6]  | 3.84 ± 0.04 | 3.60 ± 0.04 | 5.766    | 43.068    | 0.7806       | 36.75%   |
| Jia et al. [13]| 3.83 ± 0.04 | 3.13 ± 0.05 | 5.818    | 46.712    | 0.7960       | 32.63%   |
| Cooper et al. [14]| 3.86 ± 0.04 | 3.19 ± 0.03 | 5.345    | 37.858    | 0.8354       | 78.13%   |

C. Vocoder

To convert the predicted mel-spectrograms into a raw waveform [30], we use a Parallel WaveGAN [31] as our vocoder. With Wavenet [32]-based generator and multi-resolution spectrogram training objectives, it achieves high-quality waveform generation in real-time.

III. LOSS FUNCTIONS

Critical to good generalization is to extract discriminative speaker embeddings of novel speakers from limited data. To achieve this, we apply three loss functions in training stage 2: cross-entropy loss and two geometric constraints proposed in [22]. For the two geometric constraints, we first locate the speaker embedding space to a hypersphere by l2-normalizing the speaker embeddings and the classification weights. This eliminates information related to the magnitude and thus makes it possible to geometrically control the speaker embedding space by using only angular information, which is highly beneficial to few-shot learning settings [22]. In addition to these loss functions, Tacotron2’s reconstruction loss and stop token prediction loss are also used as in training stage 1.

A. Weight-Centric Embedding Clustering (WCEC)

If we train the network for novel speakers from scratch using a few data, speaker embeddings of novel speakers may not be well clustered. To address this issue, we employ Weight-Centric Embedding Clustering (WCEC) loss:

\[ \mathcal{L}_{WCEC} = \sum_{i=1}^{n_N} -\log (\cos \theta_{g(e^i)}, \tilde{w}^i_N) \]

where \( n_N \) refers to the number of the novel speakers and \( \cos \theta_{g(e^i)}, \tilde{w}^i_N = g(e^i)^T \cdot (w_N^i/|w_N^i|) \) is the cosine similarity between \( g(e^i) \) and the normalized classification weight of the \( i \)th novel speaker, denoted \( \tilde{w}^i_N \). The function \( g(\cdot) \) is represented as follows:

\[ g(e^i) = \frac{e^i}{|e^i|} \]

Here, \( e^i \) is the arithmetic mean of the normalized speaker embeddings of the \( i \)th novel speaker. We use \( e^i \) as the initial classification weight of the \( i \)th novel speaker, so that the cosine similarity can be initialized with a positive number and this WCEC loss can be defined.

When used with cross-entropy loss, this WCEC loss enables the speaker embeddings of novel speakers to be clustered near the corresponding classification weight.

B. Angular Weight Separation (AWS)

Since the WCEC loss has clustered the speaker embeddings of novel speakers near the corresponding classification weight, now we aim to separate each classification weight apart from the others. To this end, we first clarify the cosine similarity \( u_{ij} \) between the classification weights:

\[ u_{ij} = \begin{cases} 0, & \text{if } \tilde{w}^i \equiv \tilde{w}^j_N \\ \cos \theta_{\tilde{w}^i, \tilde{w}^j_N}, & \text{otherwise} \end{cases} \]

Here, \( \tilde{w}^i \) is the \( i \)th column vector of \( \tilde{W}_B \tilde{W}_N \), which is the normalized weight matrix of base and novel classifier, denoted \( \tilde{W}_B \) and \( \tilde{W}_N \), respectively. Our goal is to reduce \( u_{ij} \) by margin \( m \) to enlarge angular distance \( \theta_{\tilde{w}^i, \tilde{w}^j_N} \) between the classification weights. This AWS loss is defined as follows:

\[ \mathcal{L}_{AWS} = \sum_{i,j} -\log (u_{ij} \cdot 1(u_{ij}) + 1) \sum_{i,j} 1(u_{ij}) \quad (5) \]

where \( 1(u_{ij}) = \begin{cases} 1, & \text{if } \forall i, j, \ u_{ij} > m \\ 0, & \text{otherwise} \end{cases} \)

The logarithm and the factor of 1 are used to align AWS loss with the scale of other loss functions [22]. When each classification weight of a novel speaker is sufficiently separated from that of the other speakers in terms of the angular distance, we stop using the AWS loss. In all our experiments, we use a margin of \( m = 0.5 \) and all the losses have equal weights.

IV. EXPERIMENTS

A. Experimental Settings

We conducted experiments on the VCTK dataset, which contains 108 English native speakers with various accents. The audio was downsampled from 48,000Hz to 22,050Hz.
TABLE II
THE RESULTS OF ABLATION STUDY FOR TWO GEOMETRIC CONSTRAINTS.

| Setting                      | MOS-Nat. (†) | MOS-Sim. (†) | MCD_{13} (↑) | RMSE_{f0} (↑) | Cos.-Sim. (↑) | Acc. (↑) |
|------------------------------|--------------|--------------|--------------|---------------|--------------|----------|
| Proposed                     | 3.76 ± 0.04  | 3.29 ± 0.05  | 5.415        | 38.811        | 0.8211       | 73.50%   |
| \( -L_{WCEC} \)             | 2.28 ± 0.04  | 2.65 ± 0.06  | 6.542        | 41.861        | 0.7024       | 45.00%   |
| \( -L_{AWS} \)              | 3.49 ± 0.03  | 3.18 ± 0.04  | 5.468        | 40.526        | 0.7830       | 66.25%   |
| \( -L_{WCEC} - L_{AWS} \)   | 3.26 ± 0.03  | 3.01 ± 0.05  | 5.980        | 42.517        | 0.7547       | 59.63%   |

TABLE III
THE RESULTS OF ABLATION STUDY FOR FREEZING MODULES. ENC AND ATTN REFER TO TACOTRON2 ENCODER AND ATTENTION MODULE, RESPECTIVELY. SPK-ENC K DENOTES FREEZING K BOTTOM CONVOLUTION BLOCKS IN THE SPEAKER ENCODER.

| Setting                        | MCD_{13} | RMSE_{f0} | Cos.-Sim. | Acc. |
|--------------------------------|----------|-----------|-----------|------|
| Nothing                        | 5.511    | 41.676    | 0.8092    | 68.50% |
| +Enc + Attn                    | 5.447    | 39.483    | 0.8105    | 66.50% |
| +SPK-ENC1                      | 5.498    | 40.536    | 0.8101    | 67.62% |
| +SPK-ENC2                      | 5.465    | 40.820    | 0.8115    | 68.00% |
| +SPK-ENC3                      | 5.479    | 38.294    | 0.8119    | 70.13% |
| +SPK-ENC4                      | 5.415    | 38.811    | 0.8211    | 73.50% |
| +SPK-ENC5                      | 5.469    | 39.156    | 0.8174    | 68.38% |
| +SPK-ENC6                      | 5.436    | 41.767    | 0.8080    | 66.00% |

and the corresponding transcripts were converted from character to phoneme sequence. We randomly chose 85 speakers for the base speakers and 16 speakers for the novel speakers. The remaining 7 speakers were used for the validation set. For each novel speaker, 10 and 50 samples were randomly sampled for the adaptation data (representing less than 1 and 5 minutes, respectively). We constructed the test set for novel speakers that do not appear in any training sets.

We followed the implementation details in the work of [6] with some modifications. In the training stage 1, we used 32 of batch size, and \( 10^{-3} \) of the initial learning rate. The learning rate was decayed by a 0.5 factor in every 50K step with the minimum learning rate of \( 10^{-5} \). In the training stage 2, we used 8 of batch size and \( 10^{-4} \) of fixed learning rate. We utilized the ADAM optimizer [33] with \( \beta_1 = 0.9, \beta_2 = 0.98, \) and \( \epsilon = 10^{-9} \). 80 bands mel-spectrogram was transformed with 1024 of window size, 256 of hop size, and 1024 points of Fourier transform.

B. Evaluation Method

Along with ground truth audio and the vocoded audio from the ground truth mel-spectrogram (Vocoded), we compared our GC-TTS against several baselines. Since they used various experimental setups, the results were from the most similar configuration to ours.

- **Tacotron2** [6] refers to multi-speaker Tacotron2 trained with hours of training data for novel speakers (about 30 minutes for each novel speaker). Since it was trained on a large amount of data, it can be regarded as our upper bound.
- **Jia et al. [13]** transfers well-trained speaker embedding computed from the speaker verification model of [34] into pre-trained multi-speaker Tacotron2. We followed the open-source implementation [35] to implement the model of [34].
- **Cooper et al. [14]** exploits speaker embedding computed from external speaker verification model of [36]. We utilized the official implementation [37] to extract speaker embeddings for multi-speaker Tacotron2.
- **Fine-tuning** is a strong comparable system that finetunes the whole model of Tacotron2 without using two geometric constraints and freezing weights in a similar scheme to [15], [16].

We measured the quality of synthesized speech on two scales. As a subjective test, we performed MOS tests. As an objective assessment, we computed MCD_{13}, RMSE_{f0}, cosine similarity, and speaker classification accuracy.

- **MOS** stands for Mean-Opinion-Score. We performed MOS tests via Amazon MTurk, where at least 20 subjects were asked to rate the naturalness and speaker similarity on a score of 1-5 in 0.5 point increments.
- **MCD_{13}** represents mel-cepstral distortion [38]. It compares mel-frequency cepstral coefficient between ground truth and synthesized audio signals.
- **RMSE_{f0}** computes l2 distance between fundamental frequencies of ground truth and that of synthesized audio [39].
- **Cosine Similarity** measures how similar the voice of the synthesized speech is to that of the target speaker. We extracted x-vectors [40] from generated and the actual speech, then calculated the cosine similarity between them.
- **Classification Accuracy** is another metric for evaluating speaker similarity between synthesized and the actual voice. We trained a test-only speaker classification network using x-vectors [40] extracted from the actual speech of the novel speakers.

V. RESULTS AND ANALYSIS

A. Subjective Evaluation

To evaluate the adaptation performance of GC-TTS in terms of both naturalness and speaker similarity, we conducted the MOS test. For the MOS test, we randomly sampled 80 generated speech from the novel speakers.

Table[II] shows the MOS results compared to our baselines. [13] and [14] achieved better naturalness than GC-TTS because their decoder followed well-trained base speakers; this is not the desired result since we intend to generate natural speech in the voice of novel speakers. In terms of the speaker similarity which is the main metric for the few-shot speaker adaptation problem [16], GC-TTS achieved the
best MOS result. This implies that the perceptual similarity of the speech generated by GC-TTS is better than the others.

B. Objective Evaluation

As objective assessments, we computed MCD$_{13}$ [38], RMSE$_{f0}$ [39], cosine similarity, and speaker classification accuracy using randomly sampled 800 generated speech of the novel speakers.

As demonstrated in Table I, GC-TTS shows the best results in every objective metric. Specifically, the speaker classification accuracy was highly improved, which demonstrates that the proposed model clones a much similar voice to the target speaker than the other models. In addition, when adapted with 50 samples (< 5m), GC-TTS achieves the highest score in RMSE$_{f0}$ with a gap of only 0.228 compared to the Tacotron2 which can be regarded as our upper bound.

C. Ablation Study

We conducted an ablation study of two geometric constraints, i.e. WCEC loss and AWS loss, to verify the effect of them. For brevity, we used 10 samples of each novel speaker as the adaptation data, and each system was trained until convergence.

The results are shown in Table II which demonstrates all the geometric constraints contribute to the adaptation performance in terms of both naturalness and speaker similarity. Especially, removing WCEC loss leads to inconsistent speaker embedding, and thus significantly degrades the generation performance.

Moreover, we performed another ablation study of freezing modules to validate the statement in Sec. II-B. The results are shown in Table III. We can get the most natural synthesis when we freeze Tacotron2 encoder, the attention module, and 4 of bottom convolution blocks. Whereas, freezing no modules show the worst result in MCD$_{13}$.

D. Speaker Embedding Space

We visualize the speaker embedding space using t-Stochastic Neighbor Embedding (t-SNE) visualization. From all 85 base speakers and 16 novel speakers, we randomly chose 12 base speakers and 6 novel speakers and computed their speaker embeddings using 200 utterances of each speaker.

Fig. 2 represents the speaker embedding space of the model fine-tuned with or without two geometric constraints. Fig. 2(a) represents the speaker embedding space computed from the model trained with only the base speakers. The speaker embeddings of the base speakers are well trained but those of the novel speakers are spread out or overlapped with those of base speakers. When the whole network is merely fine-tuned with 10 samples of each novel speaker, the speaker embeddings of the novel speakers somewhat cluster but it damages those of the base speakers as in Fig. 2(b). From Fig. 2(c), it is evident that GC-TTS not only constructs discriminative speaker embeddings of the novel speakers but also preserves the speaker embeddings of the base speakers using only 10 adaptation data from each novel speaker.

VI. CONCLUSION AND FUTURE WORKS

We presented the few-shot speaker adaptive framework with two geometric constraints. The two geometric constraints enabled speaker embedding space to be controlled by only angular information, so that we could extract discriminative speaker embeddings of novel speakers from only a few data while preserving speaker embedding space of base speakers. From the experiments, we demonstrated that GC-TTS significantly improved the speaker similarity of generated speech over standard techniques. The geometric constraints are highly scalable in that they can be applied not only to image classification but also to speaker adaptation. For future work, we will apply the geometric constraints to other tasks such as emotional TTS or cross-lingual problems.

ACKNOWLEDGMENT

The authors would like to thank H.-R. Noh at Netmarble Corp., Seoul, Korea, and H. Chung at Korea Univ., Seoul, Korea, for their valuable discussions and supports.
