A Fast and High-quality Text-to-Speech Method with Compressed Auxiliary Corpus and Limited Target Speaker Corpus

Ye Tao[*1], **Chaofeng Lu**[1], **Meng Liu**[2],
Kai Xu[3], **Tianyu Liu**[1], **Yunlong Tian**[4], **Yongjie Du**[4]
[1]School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao
[2]School of Reliability and Systems Engineering, Beihang University, Beijing
[3]BYD Automotive Co., Ltd., Xi'an
[4]National Engineering Research Center of Digital Home Networking, Qingdao
ye.tao@qust.edu.cn, 2579356425@qq.com, liumeng518@buaa.edu.cn,
1393474236@qq.com, 1152206571@qq.com, tianyl@haier.com, tcldu@163.com

Abstract

With an auxiliary corpus (non-target speaker corpus) for model pre-training, Text-to-Speech (TTS) methods can generate high-quality speech with a limited target speaker corpus. However, this approach comes with expensive training costs. To overcome the challenge, a high-quality TTS method is proposed, significantly reducing training costs while maintaining the naturalness of synthesized speech. In this paper, we propose an auxiliary corpus compression algorithm that reduces the training cost while the naturalness of the synthesized speech is not significantly degraded. We then use the compressed corpus to pre-train the proposed TTS model CMDTTS, which fuses phoneme and word multi-level prosody modeling components and denoises the generated mel-spectrograms using denoising diffusion probabilistic models (DDPMs). In addition, a fine-tuning step that the conditional generative adversarial network (cGAN) is introduced to embed the target speaker feature and improve speech quality using the target speaker corpus. Experiments are conducted on Chinese and English single speaker's corpora, and the results show that the method effectively balances the model training speed and the synthesized speech quality and outperforms the current models.

Keywords: Text-to-Speech, auxiliary corpus, reducing training costs, fine-tuning

1. Introduction

Recently, there have been notable advancements in Text-to-Speech (TTS) systems based on deep learning (Liu et al., 2024) concerning the generation of high-fidelity speech (Skerry-Ryan et al., 2018; Ren et al., 2020). Nonetheless, achieving high-quality models like Tacotron2 (Shen et al., 2018) and FastSpeech2 (Ren et al., 2020) demands extensive training data. Given the expense of collecting such a sizable corpus, researchers have explored various strategies for synthesizing speech using a limited target speaker corpus. Some studies have concentrated on enlarging the corpus through the data augmentation methods (Xu et al., 2020). Meanwhile, several investigations seek to mitigate the constraints imposed by a limited target speaker corpus through the multi-speaker modeling techniques (Cooper et al., 2020) and knowledge transfer from non-target speakers.

In the quest to enhance synthesized speech quality, researchers have embraced various techniques. These include prosody modeling with a GMM-based mixture density network (Du and Yu, 2021), multi-speaker modeling (Cooper et al., 2020) and acoustic feature post-processing (Bollepalli et al., 2019). The main goal of such methods is to enrich the naturalness of the synthesized speech. The advancement of denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020; Zhang et al., 2023) has led to the emergence of methods such as DiffWave (Kong et al., 2020b) and Prodiff (Huang et al., 2022), aimed at enhancing the fidelity of synthesized speech. However, little attention has been paid to improving the quality of synthesized speech with compressed auxiliary corpus. More data can provide more learning opportunities and help the model better capture the characteristics of the speech. Therefore, improving the quality of speech synthesis while reducing the training cost by reducing the training data is a contradiction. We analyze the corpus characteristics to address this problem and propose a novel speech synthesis method with limited target speaker corpus.

An auxiliary corpus compression algorithm is proposed to reduce the corpus size while maintaining its representativeness and diversity. The compressed auxiliary corpus is used to train the proposed TTS model CMDTTS. We use a neural network-based reference encoder to extract the prosody information better from the real mel-spectrograms. The phonemes are embedded and encoded, and the words in the input text and the context information are extracted using a BiLSTM. To improve the naturalness of the synthesized speech, the improved DDPMs is used to fine-tune

LREC-COLING 2024, pages 525–535
20-25 May, 2024.

the generated mel-spectrograms. In addition, we introduce conditional generative adversarial networks (cGAN), a fine-tuning process using the target speaker corpus while embedding the target speaker's style, resulting in personalized speech synthesis. Experiments show that the auxiliary corpus compression algorithm works well in Chinese and English corpora. Compared to state-of-the-art methods, the proposed method completes model training faster and with less quality degradation.

Our contributions are as follows: **1**) A novel algorithm is proposed for compressing auxiliary corpora, which effectively mitigates the negative impact of corpus compression on speech quality while reducing model training costs; **2**) We introduce a non-autoregressive model CMDTTS that combines a multi-level prosody modeling component and DDPMs fine-tuning mel-spectrograms. The reference encoder captures phoneme-level and word-level features from the real mel-spectrogram, while the addition of DDPMs helps in generating mel-spectrograms that closely resemble the real data; **3**) We propose a fine-tuning strategy that uses cGAN to fine-tune CMDTTS to synthesize speech with a higher degree of naturalness and speaker similarity.

2. Related Work

2.1. Text-to-Speech with Limited Target Speaker Corpus

In recent years, speech synthesis models based on a large non-target speaker corpus and a limited target speaker corpus have made remarkable progress in the quality of speech synthesis. Currently, transfer learning (TL) (Xing et al., 2022) has become one of the important techniques for improving the performance of speech synthesis, especially when dealing with limited target speaker data. The main idea of TL is to learn knowledge from non-target speaker corpora and apply it to the target speaker. However, training such high-quality TTS models requires a large amount of high-quality multi-speaker speech corpora. This significantly increases the training cost of the model.

In cases where the target speaker data is limited, researchers have proposed data augmentation methods based on Voice Conversion (VC) (Walczyna and Piotrowski, 2023). Huybrechts et al. (Huybrechts et al., 2021) proposed a method for synthesizing speech with a limited target speaker corpus by training a VC model to generate speech with the style of the target speaker. This synthesized speech and the target corpus are used to jointly train a TTS model, followed by fine-tuning using the target speaker corpus. Building upon this, Shah et al. (Shah et al., 2021) enhanced the naturalness and similarity to the target speaker by replacing the autoregressive model used in (Huybrechts et al., 2021) with a non-autoregressive model and subsequently fine-tuning it using cGAN (Mirza and Osindero, 2014). However, both of these methods heavily rely on the capability of the VC model, significantly increasing the training cost of the TTS model.

2.2. Denoising Diffusion Probabilistic Models

The denoising diffusion probability models is a probabilistic denoising method. Specifically, this model first establishes the joint probability distribution between noise and speech signals by observing their statistical characteristics. Then, by maximizing this joint probability distribution, the noise parameters are estimated. Finally, using these parameters, the elimination of noise is achieved.

Recently, the DDPMs have been developed rapidly in various applications such as text-to-image (Zhang et al., 2023) and text-to-speech (Huang et al., 2022). Grad-TTS(Popov et al., 2021) employs a framework based on stochastic differential equations to model the noise and various parameters of the reconstructed data. A diffusion-based decoder converts the parametric Gaussian noise output by the encoder into a mel-spectrogram. However, the diffusion process requires multiple iterations, resulting in a slower sampling speed. Researchers have proposed adversarial learning methods to reduce iterations and learn adaptive noise schedules to address this issue. Salimans et al. (Salimans and Ho, 2022) proposed a method that utilizes knowledge distillation (Gou et al., 2021) to accelerate sampling and demonstrated its strong performance. These methods primarily focused on the image domain. Huang et al. (Huang et al., 2022) investigated a progressive fast diffusion model for speech synthesis and demonstrated improved sampling speed and high-quality synthesized speech. Nevertheless, the abovementioned method must be performed on a large target speaker corpus.

2.3. Generative Adversarial Networks

Generative adversarial networks (GANs) consist of two parts: a generator and a discriminator. The goal of the generator is to generate data that are as realistic as possible, while the goal of the discriminator is to differentiate between the generated data and real data. GANs perform well in acoustic models and vocoders for speech synthesis. Glow-WaveGAN (Cong et al., 2021) generates high-quality speech by combining variational autoencoder and GANs to learn latent representations and model them directly. HiFi-GAN (Kong et al., 2020a) improves sample quality by modeling periodic patterns in audio. Jets (Lim et al., 2022) enhances the expressiveness of the trained models by training FastSpeech2 with HiFi-GAN and eliminates the reliance on external text-speech alignment tools by aligning the learning objectives. Yuan et al. (Yuan et al., 2022) achieved speech synthesis for a small number of target corpora by pretraining with a public corpus on two GANs-based vocoders and fine-tuning with a small amount of adaptation data. However, all these methods trained directly by GANs are costly.

3. Proposed Method

The approach for speech synthesis with a limited corpus of target speakers includes three primary stages, as shown in Figure 1. Initially, redundancy data in the auxiliary corpus is eliminated following the proposed auxiliary corpus compression algorithm, aiming at reducing training costs. Subsequently, the CMDTTS is trained utilizing the compressed corpus. Finally, the CMDTTS model is improved by fine-tuning with the target speaker corpus employing a cGAN to enhance the quality of speech signals.

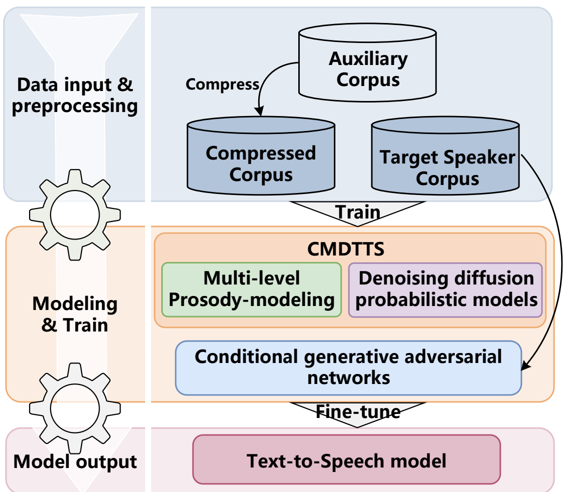


Figure 1: The input consists of an auxiliary corpus and a target speaker corpus. Through auxiliary corpus compression, model training, and cGAN fine-tuning, a TTS model is obtained as the output.

3.1. Auxiliary Corpus Compression Algorithm

To reduce the word error rate (WER) of synthesized speech, it is necessary to minimize the differences in the types of phonemes and function words before and after compression of the auxiliary corpus. Meanwhile, the distribution of phonemes and function words in the compressed corpus is homogenized to achieve more natural synthesized speech. The difference between the target domain data of the corpus before and after compression is minimized when training domain-specific speech synthesis models.

We use N_p and N_{fw} to denote the number of phonemes and function words data in the auxiliary corpus, and N'_p and N'_{fw} denote the number of phonemes and function words in the compressed corpus, respectively. N'_{ps} and N'_{fws} represent the number of phonemes in the compressed corpus phoneme set and the number of function words in the function word set respectively. The probability function is denoted by $P()$ and the weight of z_n is denoted by λ_n. The smaller Z is, the better the performance of the compressed corpus.

$$\begin{cases} z_1 = \min\left(\frac{1}{N'_p}\sum_{i=1}^{N'_p}\left(P(p_i) - \frac{1}{N'_p}\right)^2\right), \\ z_2 = \min\left(\frac{1}{N'_{fw}}\sum_{i=1}^{N'_{fw}}\left(P(fw_i) - \frac{1}{N'_{fw}}\right)^2\right), \\ z_3 = \max\left(N'_{ps}\right), \\ z_4 = \max\left(N'_{fws}\right), \\ Z = \sum_{n=1}^{4}\lambda_n z_n, \sum_{n=1}^{4}\lambda_n = 1 \end{cases} \tag{1}$$

Algorithm 1 is proposed based on Equation 1 for compressing a single-speaker corpus. N_{ps} and N_{fws} represent the number of phonemes in the auxiliary corpus phoneme set and the number of function words in the function word set, respectively. S_i represents the redundancy score of utterance U_i, and C and C', respectively, denote the number of utterances in the corpus before and after compression. Algorithm 1 requires the user to provide the compression ratio of the corpus.

Algorithm 1: Compressing auxiliary corpus
Requires: auxiliary corpus C, compression ratio r

Initialize $C' = C$
for each utterance U_i with index i in C **do**
    $S_i = \sum \frac{\alpha_1 N_{ps}}{N_p} + \sum \frac{\alpha_2 N_{fw}}{N_{fw}} + \frac{\alpha_3}{len(U_i)}$
endfor
Sort (C') based on S_i
for each utterance U'_j with index j in C' **do**
    if $|C'| > (1-r)|C|$ **do**
        if $N_{ps} == N'_{ps}$ & $N_{fws} == N'_{fws}$ **do**
            Remove U'_j from C'
endfor

3.2. CMDTTS

3.2.1. Architecture

The architecture of CMDTTS is shown in Figure 2. The encoder, duration predictor, and decoder followed FastSpeech2. A prosody modeling component, which consists of a self-attention module and a reference encoder, is used to improve the extraction and prediction of prosody information. We introduce DDPMs to fine-tune the mel-spectrograms obtained from decoding to improve the quality of synthesized speech.

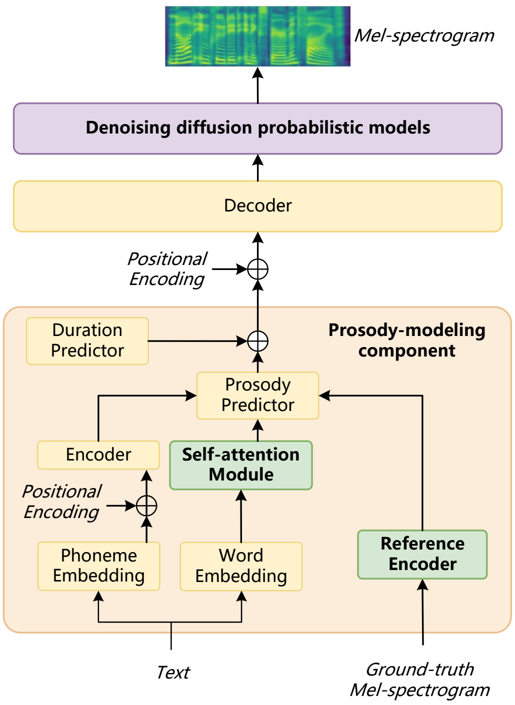


Figure 2: The overall architecture for CMDTTS. Both the auxiliary corpus and the compressed corpus can be used as input to the TTS model.

3.2.2. Prosody Modeling Component

The multi-level prosody modeling component is shown in Figure 2. The inputs of this component are the text and the ground-truth mel-spectrogram. The encoder encodes the phoneme embedding and generates a phoneme hidden sequence. Word embeddings are applied to the input text, and a self-attention module is used to capture the dependencies of adjacent words. We use a reference encoder based on a neural network to extract prosody information.

The self-attention module aims to capture dependency between adjacent words in the text by using the attention weights, as shown in Figure 3. It consists of identical self-attention blocks. A BiLSTM is used to enhance the sequence modeling. The word embedding sequence is the input to the module; two LSTMs process the sequence in opposite directions to compute two final hidden states, and the input text sequence is computed by a summation operation. The details of the forward LSTM are as follows:

$$f_t = \text{sigmoid}\left(W_{fv}V_t + W_{fh}H_{t-1} + b_f\right), \quad (2)$$

$$i_t = \text{sigmoid}\left(W_{iv}V_t + W_{ih}H_{t-1} + b_i\right), \quad (3)$$

$$o_t = \text{sigmoid}\left(W_{ov}V_t + W_{oh}H_{t-1} + b_o\right), \quad (4)$$

$$\tilde{C}_t = \tanh\left(W_{cv}V_t + W_{ch}H_{t-1} + b_c\right), \quad (5)$$

where V_t and H_t indicate the input vector and the hidden unit vector, respectively. W_{fv}, W_{ix}, W_{ox}, W_{cv} denote the different weight matrices for V_t; W_{fh}, W_{ih}, W_{oh}, W_{ch} are the different weight matrices for h_t; and b_f, b_i, b_o, b_c denote the bias vectors.

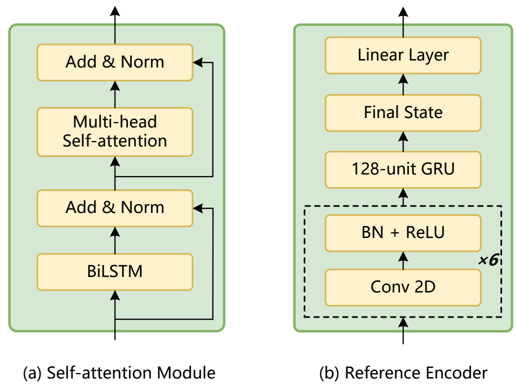


Figure 3: The left subfigure is the self-attention module, and the right subfigure is the reference encoder.

Taking inspiration from Skerry et al.'s work (Skerry-Ryan et al., 2018), we employ a reference encoder to extract prosody information. The architecture is shown in Figure 3, starting with a 6-layer 2D convolutional network. After each convolutional layer, a ReLU activation function is applied to zero out all negative values, which allows the network to learn more complex, nonlinear mappings. A 128-width GRU layer compresses the sequence into a fixed-length vector. The output of 128 dimensions is summed up and finally projected onto the desired dimension through a linear layer.

3.2.3. Denoising Diffusion Probabilistic Models

As shown in Figure 4, the input of DDPMs is mel-spectrograms x_t which takes noisy, diffusion time index t and variance v, and the output is denoised mel-spectrograms x_0. The Linear Layer, ReLU, and Swish denote the fully connected layer and activation function. The number of residual layers is M.

Inspired by (Jeong et al., 2021), the clean data are predicted directly in DDPMs to improve the

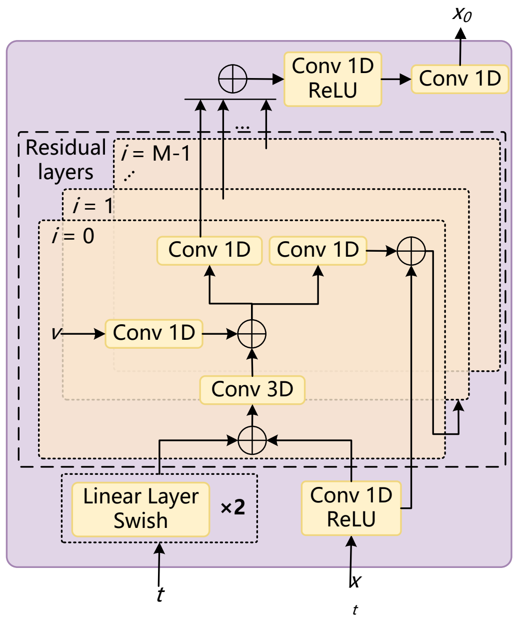


Figure 4: The network of denoising diffusion probabilistic models.

quality of mel-spectrograms. Knowledge distillation technology reduces the order of magnitude of sampling time. v_p, v_e, and v_d denote the pitch, energy, and duration respectively. $\hat{v}_p$, $\hat{v}_e$, and $\hat{v}_d$ are used to denote the corresponding predicted values respectively. The sample reconstruction loss L_θ, the variance reconstruction loss L_v and the loss of DDPMs L_{DDPMs} are calculated as follows:

$$L_\theta = \left\| x_\theta \left(\alpha_t x_0 + \sqrt{1-\alpha_t^2 \epsilon} \right) - \hat{x}_0 \right\|_2^2, \tag{6}$$

$$L_v = \|v_p - \hat{v}_p\|_2^2 + \|v_e - \hat{v}_e\|_2^2 + \|v_d - \hat{v}_d\|_2^2, \tag{7}$$

$$L_{DDPMs} = L_\theta + L_v, \tag{8}$$

where ϵ denotes the standard Gaussian noise.

3.3. Fine-tuning with cGAN

The architecture of cGAN is shown in Figure 5. We use the trained CMDTTS model as the generator. s is the ground-truth mel-spectrogram, $G(s)$ is the generated mel-spectrogram, and c is the conditional information (target domain and target speaker). Fine-tuning is performed by feeding c as an additional input layer to the generator and discriminator. The generator's priori input noise and condition are combined in a joint hidden representation, and the inputs c and s of the discriminator are passed through a discriminant function to determine the authenticity of $G(s)$.

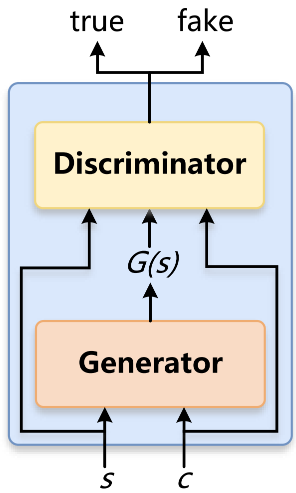


Figure 5: The network structure of cGAN.

The generator and discriminator follow the two-player min-max game in model training, and the loss function is as follows:

$$\begin{aligned} L_{cGAN} = & \mathbb{E}_{s\sim d_{data}(s)} \left[\log D\left(s\,|c\right)\right] + \\ & \mathbb{E}_{z\sim d_z(z)} \left[\log\left(1 - D\left(G\left(s\,|c\right)\right)\right)\right], \end{aligned} \tag{9}$$

where s is a sample from the actual data distribution d_{data} and z is a sample from the noise distribution d_z, $D(s\,|c)$ denotes the prediction result of the discriminator for the actual sample s given the condition c, and $G(s\,|c)$ denotes the fake sample generated by the generator based on the noise z given the condition c.

4. Experiment

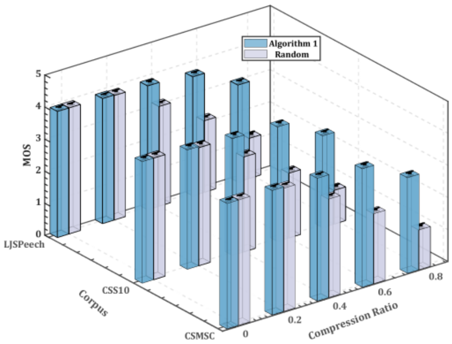
To evaluate the method, the CSMSC (Baker, 2017), the Chinese part of CSS10 (Park and Mulc, 2019), and the LJSpeech (Ito and Johnson, 2017) are treated as the auxiliary corpora. A single speaker's data in VCTK(Veaux et al., 2016) is the English target speaker corpus called MHTO. A Chinese target speaker corpus LQDE is also collected, containing 6-minute recordings from a voice-over specialist. All the trainings are conducted on a single GeForce RTX 2080Ti GPU. After the acoustic model inference, we use a well-trained HiFi-GAN(Kong et al., 2020a) as the vocoder to generate speech.[1].

For the subjective evaluation, we invited 20 Chinese speakers and 20 English speakers to evaluate the Mean Opinion Scores (MOS) of synthesized speech. The speech quality on a scale of 0 to 5, with 5 being the best. In each test, scores are given for 20 test utterances synthesized by each experimental model and are reported with a 95% confidence interval.

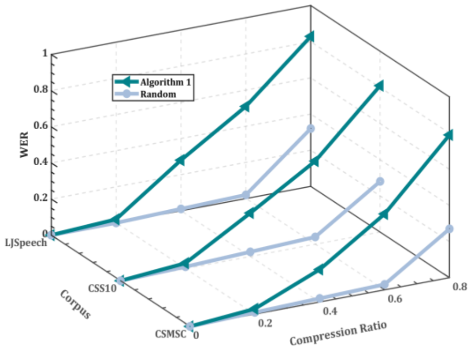
[1]Synthesized speech samples are available at: https://2579356425.github.io/CMDTTS/

4.1. Performance of Auxiliary Corpus Compression Algorithm

In this section, the CSMSC, CSS10, and LJSpeech are compressed by a random method and the algorithm 1, respectively. Then we use the compressed corpora to train FastSpeech2. We evaluate the speech quality with MOS, speech intelligibility with WER, and training speed with model training time. The results are shown in Figure 6 and Figure 7.



(a) MOS of synthesized speech



(b) WER of synthesized speech

Figure 6: This figure shows the performance of the algorithm 1. Subfigure (a) shows the MOS of the synthesized speech and subfigure (b) shows the objective evaluation WER.

Figure 6 shows that within the compression ratio of 0 to 0.2, both the random compression approach and algorithm 1 exhibit negligible influence on the naturalness and intelligibility of synthesized speech. However, as compression ratios elevate to 0.2 to 0.4 and 0.4 to 0.6, the random compression method results in a significant decrease in MOS and a surge in WER. Conversely, using algorithm 1 for corpus compression reduces speech quality less. Both compression methods lead to substantial degradation of model performance during the process of compression ratio from 0.6 to 0.8.

This is mainly due to the large volume of data in the auxiliary corpus and the redundancy of

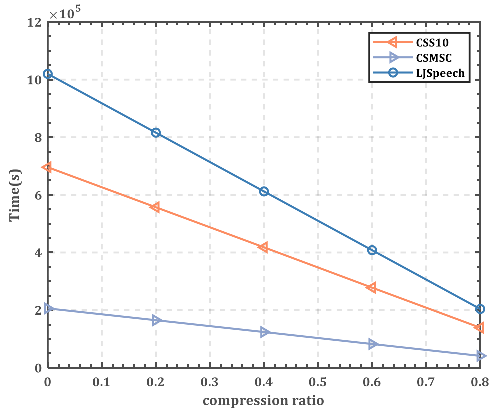


Figure 7: The figure of model training time with the change of compression ratio.

phonemes and function words. Both compression methods can remove redundant data within the compression ratio threshold range of 0 to 0.2. When the compression ratio is within 0.2 to 0.6, algorithm 1 can continuously remove redundant data, while the random compression method removes rare phonemes and function words. In contrast, the random compression method eliminates phonemes and function words, which are already relatively scarce. As the compression ratio increases from 0.6 to 0.8, algorithm 1 continues to remove redundant utterances from the corpus. When the ratio reaches a critical point, there is no more redundancy in the corpus. Further compression of the corpus reduces the variety of phonemes and function words, resulting in a rapid decline in speech quality.

As seen from Figure 7, model training time decreases proportionally with the increase of corpus compression ratio. The results show that algorithm 1 effectively improves the speed of model training and reduces the degradation of speech quality.

4.2. Parameters Studies of Algorithm 1

In this section, we studied the impact of the parameter μ_n in algorithm 1 on synthesized speech during corpus compression. Due to $\sum_{n=1}^{3} \mu_n = 1$, we conducted the study by varying μ_1 and μ_2 while keeping other variables constant. We trained FastSpeech2 (Ren et al., 2020) on three different corpora and evaluated the quality of synthesized speech using the average of MOS.

As shown in Figure 8, it can be seen that the quality of synthesized speech is different for different corpora using the same parameters to compress the corpus. When μ_1 is in the interval $[0.4, 0.5]$

Table 1: The results of ablation studies. The best MOS, second MOS, and worst training time are in red, orange, and brown colors, respectively.

Model	CSMSC & LQDE		CSS10 & LQDE		LJSpeech & MHTO	
	MOS	Time (s)	MOS	Time (s)	MOS	Time (s)
Base	3.80 ± 0.08	6.94×10^5	3.78 ± 0.07	2.05×10^5	3.82 ± 0.08	1.01×10^6
Base+MD	4.00 ± 0.07	8.60×10^5	3.93 ± 0.09	2.53×10^5	4.00 ± 0.07	1.26×10^6
Base+cGAN	3.85 ± 0.08	7.04×10^5	3.82 ± 0.07	2.16×10^5	3.87 ± 0.07	1.02×10^6
Base+MD+cGAN	4.05 ± 0.05	8.72×10^5	4.00 ± 0.08	2.65×10^5	4.03 ± 0.09	1.27×10^6
Base+C	3.78 ± 0.09	2.05×10^5	3.74 ± 0.06	6.14×10^4	3.80 ± 0.07	3.04×10^5
Base+C+MD	3.96 ± 0.07	2.56×10^5	3.90 ± 0.08	7.65×10^4	3.96 ± 0.08	3.79×10^5
Base+C+cGAN	3.83 ± 0.08	2.20×10^5	3.80 ± 0.06	7.43×10^4	3.85 ± 0.06	3.17×10^5
Base+C+MD+cGAN	4.04 ± 0.07	2.71×10^5	4.00 ± 0.06	8.94×10^4	4.02 ± 0.07	3.92×10^5

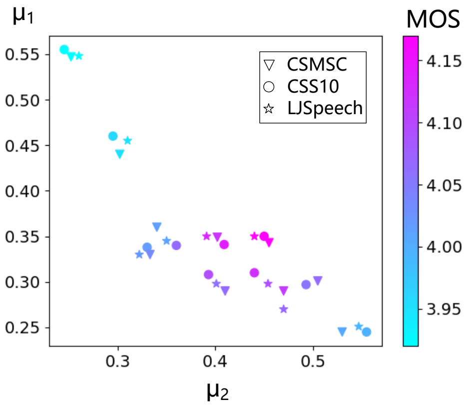


Figure 8: PESQ follows the parameters μ_n.

and μ_2 is in the interval $[0.3, 0.4]$, the algorithm 1 shows good performance on CSMSC, CSS10 and LJSpeech. The reason is that the type and number of phonemes and function words have a greater impact on the synthesized speech than the length of a single utterance.

4.3. Ablation Studies for Proposed Method

In this section, we validated the effectiveness of each module by conducting ablation studies. These modules include using algorithm 1 to compress the corpus, the proposed speech synthesis model CMDTTS which fuse multi-level prosody modeling component and DDPMs, and fine-tuning the model using cGAN. CSMSC and CSS10 are Chinese auxiliary corpora, while LQDE is the Chinese target speaker corpus. LJSpeech and MHTO are respectively auxiliary and target speaker corpora for English. The batch size for training and fine-tuning all models in this section is set to 4. We evaluate the following 8 models:

(1). (Base) FastSpeech2 was trained using an auxiliary corpus and then fine-tuned with the target speaker corpus in the same language.

(2). (Base+MD) CMDTTS was trained utilizing an auxiliary corpus for initial training and then fine-tuning with the target speaker corpus.

(3). (Base+cGAN) CGAN was introduced into the fine-tuning step towards (Base), keeping other steps unchanged.

(4). (Base+MD+cGAN) CGAN was introduced into the fine-tuning step towards (Base+MD), keeping other steps unchanged.

(5). (Base+C) Auxiliary corpus was compressed using algorithm 1 with a compression ratio 0.7. FastSpeech2 was trained using the compressed corpus and then fine-tuned with the target speaker corpus in the same language.

(6). (Base+C+MD) The model architecture was replaced by CMDTTS while keeping other settings the same as (Base+C).

(7). (Base+C+cGAN) The model training corpus was compressed by algorithm 1. Other settings are identical to (Base+cGAN).

(8). (Base+C+MD+cGAN) The model architecture was replaced by CMDTTS while keeping other settings the same as (Base+C+cGAN).

The results are shown in Table 1. Comparing the results of three auxiliary corpora, the MOS of model (Base+C) decreased by 0.02 to 0.04 compared to (B), yet the training time of (Base+C) is only one-third of (Base). Similarly, the MOS of (Base+C+MD) decreased by 0.03 to 0.04 compared to (Base+MD), with the training time of (Base+C+MD) being only one-third of (Base+MD). This indicates that the proposed algorithm 1 significantly reduces the training cost of the model with a compression ratio of 0.7 while maintaining the naturalness of synthesized speech without significant degradation.

Comparing (Base+MD) with (Base), it is evident that the model's training time increased by approximately one-fourth. Yet, there was a significant improvement in speech quality, with an average MOS increase of 0.18 across the three auxiliary

Table 2: Our method is matched with (Base+C+MD+cGAN) in ablation studies. The best MOS, second MOS, and best training time are in red, orange and blue colors, respectively.

Model	CSMSC & LQDE		CSS10 & LQDE		LJSpeech & MHTO	
	MOS	Time (s)	MOS	Time (s)	MOS	Time (s)
Tacotron2	3.92 ± 0.07	1.37×10^6	3.87 ± 0.06	4.01×10^5	3.91 ± 0.07	1.98×10^6
FastSpeech2	3.93 ± 0.08	1.19×10^6	3.77 ± 0.07	3.48×10^5	3.92 ± 0.07	1.72×10^6
JETS	4.04 ± 0.07	1.24×10^6	3.94 ± 0.08	3.63×10^5	4.02 ± 0.06	1.79×10^6
VITS	4.05 ± 0.07	1.43×10^6	3.96 ± 0.08	4.19×10^5	4.03 ± 0.07	1.76×10^6
ProDiff	4.08 ± 0.05	1.65×10^6	4.01 ± 0.06	4.84×10^5	4.05 ± 0.08	2.02×10^6
Our Method	4.05 ± 0.05	4.15×10^5	4.00 ± 0.07	1.22×10^5	4.03 ± 0.06	5.14×10^5

corpora. In comparison between (Base+C+MD) and (Base+C), the average MOS increased by 0.17, with the additional model training time decreasing by an order of magnitude compared to the original time. Furthermore, (Base+C+MD+cGAN) exhibited an average MOS increase of approximately 0.19 over (Base+C+cGAN), while the cost of increased model training time decreased by an order of magnitude compared to before. The CMDTTS architecture demonstrated exceptionally high performance, showcasing the effectiveness of integrating multi-level prosody modeling components and denoising diffusion probability models in enhancing speech quality.

Compared to (Base), (Base+cGAN) resulted in an increase in MOS for synthesized speech by 0.05 to 0.07, while the average model training time increased by only 1.1×10^4 seconds. Similarly, (Base+C+cGAN) exhibited an MOS improvement of 0.05 to 0.06 over (Base+C), with the fine-tuning time increasing by one order of magnitude less than the training of (Base+C). Moreover, (Base+C+MD+cGAN) showed an MOS improvement of approximately 0.09 over (Base+C+MD), with a little additional fine-tuning time. This indicates that cGAN fine-tuning can effectively enhance speech quality relatively cheaply.

In conclusion, algorithm 1, multi-level prosody modeling, DDPMs, and cGAN fine-tuning can be combined to increase the model training speed without significantly degrading speech quality.

4.4. Performance Comparison with Other Methods

In this section, a performance comparison between our method and other speech synthesis methods is given in Table 2. The methods include Tacotron2 (Shen et al., 2018), FastSpeech2 (Ren et al., 2020), JETS (Lim et al., 2022), VITS (Kim et al., 2021), and ProDiff (Huang et al., 2022). These five models are trained using an auxiliary corpus and fine-tuned using the target speaker corpus, which has the same language. Our method follows the (Base+C+MD+cGAN) in ablation studies.

Compared with the traditional autoregressive model Tacotron2, our method not only improves the Mean Opinion Score (MOS) by 0.13 but also reduces the training time of the model by an order of magnitude. In addition, a comparison study with the baseline FastSpeech2 shows a 0.11 increase in MOS and a 1.85 times increase in model training speed. This indicates that successfully combining multi-level prosodic modeling components, DDPMs, and cGAN fine-tuning techniques significantly improves speech quality.

Our approach shows remarkable performance through meticulous comparisons with other state-of-the-art methods, closely rivaling the leading model, ProDiff, regarding speech quality while surpassing both VITS and JETS, ranking in the second-best position. Our method consistently outperforms VITS, JETS, and ProDiff in model training time by an order of magnitude across corpora such as CSMSC and LJSpeech. Furthermore, on the CSS10, our approach demonstrates superiority over these three methods by a significant margin, ranging from two to three times faster. When considering both model training speed and speech quality jointly, our proposed method outperforms traditional TTS models and the current approaches.

5. Conclusion

In this work, we propose a method for fast-training speech synthesis models with a limited target speaker corpus. An algorithm is designed to compress the auxiliary corpus, which removes redundant utterances and significantly reduces the model training cost. The CMDTTS is proposed, which fuses multi-level prosody modeling and DDPMs, using a neural network-based reference encoder to extract prosody information from mel-spectrograms and DDPMs as a post-processing network to fine-tune the generated mel-spectrograms. CGAN was introduced to fine-tune the model with the target speaker feature. Experimental results on Chinese and English corpora show that our proposed method performs better than all baseline methods regarding combined model training speed and naturalness of synthesized speech.

6. Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0612102, and in part by Key technology research and industrialization demonstration projects in Qingdao 24-1-2-qljh-19-gx.

7. Bibliographical References

Data Baker. 2017. Chinese standard mandarin speech corpus.

Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. 2017. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754.

Bajibabu Bollepalli, Lauri Juvela, and Paavo Alku. 2019. Generative adversarial network-based glottal waveform model for statistical parametric speech synthesis. *arXiv preprint arXiv:1903.05955*.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. 2023. Hydra attention: Efficient attention with many heads. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 35–49. Springer.

Zehua Chen, Yihan Wu, Yichong Leng, Jiawei Chen, Haohe Liu, Xu Tan, Yang Cui, Ke Wang, Lei He, Sheng Zhao, et al. 2022. Resgrad: Residual denoising diffusion probabilistic models for text to speech. *arXiv preprint arXiv:2212.14518*.

Jian Cong, Shan Yang, Lei Xie, and Dan Su. 2021. Glow-wavegan: Learning speech representations from gan-based variational auto-encoder for high fidelity flow-based speech synthesis. *arXiv preprint arXiv:2106.10831*.

Erica Cooper, Cheng-I Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. 2020. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6184–6188. IEEE.

Tobias Cornille, Fengna Wang, and Jessa Bekker. 2022. Interactive multi-level prosody control for expressive speech synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8312–8316. IEEE.

Ryunosuke Daido and Yuji Hisaminato. 2016. A fast and accurate fundamental frequency estimator using recursive moving average filters. In *INTERSPEECH*, pages 2160–2164.

Chenpeng Du and Kai Yu. 2021. Phone-level prosody modelling with gmm-based mdn for diverse and controllable speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:190–201.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y Bengio. 2014. Generative adversarial nets. In *Neural Information Processing Systems*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819.

Daniel Griffin and Jae Lim. 1984. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2595–2605.

Goeric Huybrechts, Thomas Merritt, Giulia Comini, Bartek Perz, Raahil Shah, and Jaime Lorenzo-Trueba. 2021. Low-resource expressive text-to-speech using data augmentation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6593–6597. IEEE.

Keith Ito and Linda Johnson. 2017. The lj speech dataset.

Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. 2021. Diff-tts: A denoising diffusion model for text-to-speech. *arXiv preprint arXiv:2104.01409*.

Sri Karlapati, Alexis Moinet, Arnaud Joly, Viacheslav Klimkov, Daniel Sáez-Trigueros, and Thomas Drugman. 2020. Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech. *arXiv preprint arXiv:2004.14617*.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020a. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020b. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Dan Lim, Sunghee Jung, and Eesung Kim. 2022. Jets: Jointly training fastspeech2 and hifi-gan for end to end text to speech. *arXiv preprint arXiv:2203.16852*.

Peng Liu, Chuanxu Wang, and Min Zhao. 2024. Modal consensus and contextual separation for weakly supervised temporal action localization. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4220–4224. IEEE.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Kyubyong Park and Thomas Mulc. 2019. Css10: A collection of single speaker speech datasets for 10 languages. *arXiv preprint arXiv:1903.11269*.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR.

Guo-Jun Qi. 2020. Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal of Computer Vision*, 128(5):1118–1140.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. 2001. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE.

Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*.

Raahil Shah, Kamil Pokora, Abdelhamid Ezzerg, Viacheslav Klimkov, Goeric Huybrechts, Bartosz Putrycz, Daniel Korzekwa, and Thomas Merritt. 2021. Non-autoregressive tts with explicit duration modelling for low-resource highly expressive speech. *arXiv preprint arXiv:2106.12896*.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Leyuan Sheng, Dong-Yan Huang, and Evgeniy N Pavlovskiy. 2019. High-quality speech synthesis using super-resolution mel-spectrogram. *arXiv preprint arXiv:1912.01167*.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *2010 IEEE international conference on acoustics, speech and signal processing*, pages 4214–4217. IEEE.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. 2016. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

Tomasz Walczyna and Zbigniew Piotrowski. 2023. Overview of voice conversion methods based on deep learning. *Applied Sciences*, 13(5):3100.

Xiaolin Xing, Yu Hong, Minhan Xu, Jianmin Yao, and Guodong Zhou. 2022. Taking actions separately: A bidirectionally-adaptive transfer learning method for low-resource neural machine translation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4481–4491.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812.

Xin Yuan, Yongbing Feng, Mingming Ye, Cheng Tuo, and Minghang Zhang. 2022. Adavocoder: Adaptive vocoder for custom voice. *arXiv preprint arXiv:2203.09825*.

Chenshuang Zhang, Chaoning Zhang, Mengchun Zhang, and In So Kweon. 2023. Text-to-image diffusion model in generative ai: A survey. *arXiv preprint arXiv:2303.07909*.

Yi Zhao, Shinji Takaki, Hieu-Thi Luong, Junichi Yamagishi, Daisuke Saito, and Nobuaki Minematsu. 2018. Wasserstein gan and waveform loss-based acoustic model training for multi-speaker text-to-speech synthesis systems using a wavenet vocoder. *IEEE access*, 6:60478–60488.