Chen *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2024) 2024:28
https://doi.org/10.1186/s13636-024-00351-9

EURASIP Journal on Audio,
Speech, and Music Processing

METHODOLOGY

Open Access

Optimizing feature fusion for improved zero-shot adaptation in text-to-speech synthesis

Zhiyong Chen[1†], Zhiqi Ai[1†], Youxuan Ma[1], Xinnuo Li[1] and Shugong Xu[1*]

Abstract

In the era of advanced text-to-speech (TTS) systems capable of generating high-fidelity, human-like speech by referring a reference speech, voice cloning (VC), or zero-shot TTS (ZS-TTS), stands out as an important subtask. A primary challenge in VC is maintaining speech quality and speaker similarity with limited reference data for a specific speaker. However, existing VC systems often rely on naive combinations of embedded speaker vectors for speaker control, which compromises the capture of speaking style, voice print, and semantic accuracy. To overcome this, we introduce the Two-branch Speaker Control Module (TSCM), a novel and highly adaptable voice cloning module designed to precisely processing speaker or style control for a target speaker. Our method uses an advanced fusion of local-level features from a Gated Convolutional Network (GCN) and utterance-level features from a gated recurrent unit (GRU) to enhance speaker control. We demonstrate the effectiveness of TSCM by integrating it into advanced TTS systems like FastSpeech 2 and VITS architectures, significantly optimizing their performance. Experimental results show that TSCM enables accurate voice cloning for a target speaker with minimal data through both zero-shot or few-shot fine-tuning of pretrained TTS models. Furthermore, our TSCM-based VITS (TSCM-VITS) showcases superior performance in zero-shot scenarios compared to existing state-of-the-art VC systems, even with basic dataset configurations. Our method's superiority is validated through comprehensive subjective and objective evaluations. A demonstration of our system is available at https://great-research.github.io/tsct-tts-demo/, providing practical insights into its application and effectiveness.

Keywords Voice cloning, Text to speech, Speech synthesis, Zero-shot learning, Few-shot learning

1 Introduction

Text-to-speech (TTS) technology, pivotal in human-computer interaction, aims to generate human-like speech from text and finds extensive application across various domains. Traditionally, TTS systems necessitate substantial training data from target speakers, limiting their flexibility and applicability in low-resource scenarios [1, 2]. Recent advancements have shifted focus towards voice cloning (VC) or zero-shot TTS (ZS-TTS) [3–5], which enables synthesizing speech for any speaker with minimal data, enhancing the system's adaptability for custom voice generation. This capability is especially beneficial for creating personalized media content, custom chatbots, and enhancing multimodal interactions between humans and computers, including interactions with large language models (LLMs). Despite strides towards artificial general intelligence (AGI) with advancements like GPT-4 showcasing zero-shot generalization in text processing, the demand for specialized models tailored to excel in distinct tasks such as voice cloning in TTS synthesis continues to grow, underscoring the importance of dedicated expert models for achieving high-quality, reliable performance [6].

Voice cloning using deep learning was initially proposed in the work of extending the Deep Voice 3 model

[†] Zhiyong Chen and Zhiqi Ai contributed equally to this work.

*Correspondence:
Shugong Xu
shugong@shu.edu.cn
[1] School of Communication and Information Engineering, Shanghai University, Shanghai, China

[3, 7] for creating personalized speech interfaces. Subsequently, [8] employed Tacotron 2 [9] alongside a speaker encoder for adapting to new speakers. More recent efforts have focused on capturing the voice print of the target speaker directly from audio samples, eliminating the need for corresponding transcripts and labels [10–12]. These TTS systems go beyond mere text input; they are designed to synthesize speech conditioned on a latent vector extracted from the speech of a specific speaker. Methods based on speaker control vectors offer a promising solution and have achieved notable results in the voice cloning task.

Numerous researchers have highlighted the effectiveness of current advanced TTS architectures in voice cloning. The FastSpeech 2 model stands out due to its rapid training and inference, attributed to its non-autoregressive nature. Its transformer-based encoder-decoder architecture offers flexibility for voice cloning. AdaSpeech, as introduced by [13, 14], enhances FastSpeech 2 by incorporating an acoustic information module and integrating conditional layer normalization in the mel-spectrogram decoder, leading to markedly superior adaptation quality over traditional methods. Concurrently, VITS-based TTS systems [15, 16], harnessing the conditional variational auto-encoder (CVAE), are gaining attention, yielding more natural speech and with superior quality that cannot be distinguish from real human speech. A notable advancement in voice cloning technology is proposed by the YourTTS and OpenVoice systems [6, 17]. These system augments the VITS framework with a speaker encoder, thereby enhancing its zero-shot voice cloning capabilities.

In addition to zero-shot adaptation for speakers, zero-shot style adaptation in TTS has emerged as a highly relevant task. Works such as PromptStyle and PromptTTS [18, 19] focus on style control for speech, specifically targeting speech emotion or accent. These approaches introduce separate style encoders and control latent vectors into TTS systems, employing mechanisms similar to those used for speaker voice cloning, thereby broadening the scope of customization and expressiveness in synthesized speech.

A prevalent challenge in voice cloning technology is the optimal utilization of style-control vectors or speaker embeddings within the TTS pipeline. Traditional voice cloning methods often rely on simplistic affine additions of speaker vectors at designated system locations, which can overlook the sequential nature of sequence features inherent to TTS models. It compromises the precise control of style, voice print, and semantic information from reference speech, leading to suboptimal outcomes. To address this, we introduce the Two-branch Speaker Control Module-based TTS (TSCM-TTS), a novel approach to zero-shot TTS that enhances speaker control in advanced TTS systems with our innovative Two-branch Speaker Control Module (TSCM). We further integrate this module within transformer encoder blocks, resulting in the TSCM-Transformer (TSCT).

Designed as an easily adaptable method, TSCM can be incorporated into any TTS system for enhanced speaker and style control. Our further integration of TSCM with robust frameworks like FastSpeech 2 and VITS demonstrates superior performance compared to many advanced ZS-TTS models. The key innovations of our work include the following:

- Our TSCM-TTS method advances speaker control by optimizing the integration of the speaker's vector into the ZS-TTS/VC models, outperforming current style-control techniques. This innovation enables effective voice personalization with minimal sample sentences, achieving enhanced zero-shot speaker adaptation. Notably, it ensures voice print accuracy and semantic naturalness, enabling precise voice customization.
- The proposed TSCM-TTS method is highly adaptable. We outline the specialized integration of TSCM within the FastSpeech 2 and VITS frameworks, which significantly enhances the capabilities of both few-shot and zero-shot TTS systems.
- Through extensive evaluations against multiple benchmarks in voice cloning tasks across various languages, our TSCM-enhanced VITS framework (TSCM-VITS) proves to be highly effective, outperforming numerous state-of-the-art systems even with fundamental training datasets and settings. Its exceptional performance is confirmed by a broad spectrum of subjective and objective assessments, underscoring its practicality and efficacy in real-world applications.

The remainder of this paper is organized as follows: Section 2 discusses related work in the field. Section 3 delves into the preliminary concepts underpinning our research. Section 4 describes the methodology behind our Two-branch Speaker Control Module (TSCM). Section 5 focuses on the optimized integration of TSCM with state-of-the-art text-to-speech (TTS) models. The experimental results and analysis from our experiments are detailed in Sections 6 and 7. Finally, Section 8 concludes the paper with our findings.

2 Related work

2.1 Text to speech (TTS)

TTS technology has evolved significantly over time. Early speech synthesis methods included articulatory, formant, concatenative, and statistical parametric speech synthesis (SPSS) [20–23]. The introduction of deep learning has led to major improvements in speech synthesis using neural networks, significantly outperforming older techniques [9, 24, 25].

WaveNet marked a transformative shift in speech synthesis, generating speech directly from linguistic features using a neural network [24]. This led to subsequent innovations such as Tacotron 2 [9] and FastSpeech 2 [25], resulting in significant improvements in TTS quality. FastSpeech 2, in particular, advanced its predecessor by integrating explicit pitch and energy control [26] and by generating the mel spectrogram in parallel, which notably accelerated the synthesis process.

Recent developments have introduced end-to-end TTS synthesis models like VITS and VITS2, employing conditional VAE architectures to achieve state-of-the-art results [15, 16]. Another notable innovation is TorToise or XTTS, which employs denoising diffusion probabilistic models, similar to those utilized in image generation, to produce high-quality speech [27].

2.2 Voice cloning (zero-shot TTS)

Voice cloning stands as a critical task within text-to-speech (TTS) technologies, striving to produce speech that closely mimics any target speaker from minimal additional training process. Our study delves into the challenges of both few-shot (w/ fast tuning) and zero-shot (w/o tuning) adaptations, leveraging a limited dataset and its corresponding transcriptions.

In the realm of *few-shot TTS* adaptation, works like [13] have fine-tuned specific parameters within the FastSpeech 2 framework. *Zero-shot TTS* advancements include utilizing style tokens in the Tacotron model [10] and extending FastSpeech 2 to achieve zero-shot capabilities [12, 14]. The ECAPA-TDNN model, originally developed for speaker recognition, has also been adapted for TTS to enhance speech synthesis for new speakers [28]. A notable example is YourTTS, which extends the VITS model for zero-shot voice cloning [17]. Microsoft's VALL-E presents an innovative approach to zero-shot TTS by framing it as a conditional language modeling task [29], using auto-regressive (AR) text encoder. Recent advancements include optimizing the VALL-E architecture with the GPT-based AR decoder, as seen in LauraTTS [30], trying to achieve unified large language model (LLM) for all AI tasks. Additionally, methods extending approach beyond VITS, like OpenVoice, propose a dual-stage inference process for improved synthesis [6]. These contributions are summarized in Table 1. Despite the development of the LLM, the demand for expert models excel in distinct tasks such as voice cloning continues to grow, since dedicated expert models normally achieve high-quality, reliable performance.

Table 1 State-of-the-art zero-shot TTS systems

Methods	Timeline	Multilingual	General type
AdaSpeech [13]	2022.12	No	Expert
YourTTS [17]	2023.1	Yes	Expert
VALL-E-X [29]	2023.3	Yes	Unified-LLM
Coqui XTTS [27]	2023.10	Yes	Expert
LauraTTS [30]	2024.1	N/A	Unified-LLM
OpenVoice [6]	2024.1	N/A	Expert

Notwithstanding the progress, existing TTS systems often overlook the optimization needed for adapting speech for unseen speakers, primarily due to simplistic speaker control mechanisms—a limitation we address later in this study. Our work introduces a novel method for speaker and style control that enhances the conditioning of control signals, aiming to achieve more optimal performance in voice cloning.

3 Preliminary concepts

3.1 The architecture of FastSpeech 2

Our system is based on the advanced TTS model FastSpeech 2, which synthesizes the corresponding mel spectrogram from a given phoneme sequence. The model architecture of FastSpeech 2 is shown in Fig. 1. Phoneme sequences are obtained from normalized text sequences using the grapheme-to-phoneme tool[1]. Then, generated mel spectrogram is transferred to the speech waveform by using a pretrained HiFi-GAN vocoder[2].

The FastSpeech 2 model contains three sub-modules, namely encoder, variance adaptor, and mel-spectrogram decoder, respectively. The encoder is designed for encoding the input phoneme sequence into the hidden sequence, followed by the variance adaptor, which adds different variance information (such as duration, pitch, and energy) into the hidden sequence. Finally, the mel-spectrogram decoder decodes the processed hidden sequence to get the corresponding mel spectrogram. The feed-forward transformer block, which contains a multi-head self-attention layer and two 1D convolution layers, is the basic structure for the encoder and mel-spectrogram decoder. The pitch and energy predictors in the variance predictor are introduced to provide more variance information for the mel-spectrogram decoder, to ease the one-to-many mapping problem in TTS, while the phoneme sequence is expanded by the length regulator

[1] https://github.com/Kyubyong/g2p

[2] https://github.com/jik876/hifi-gan

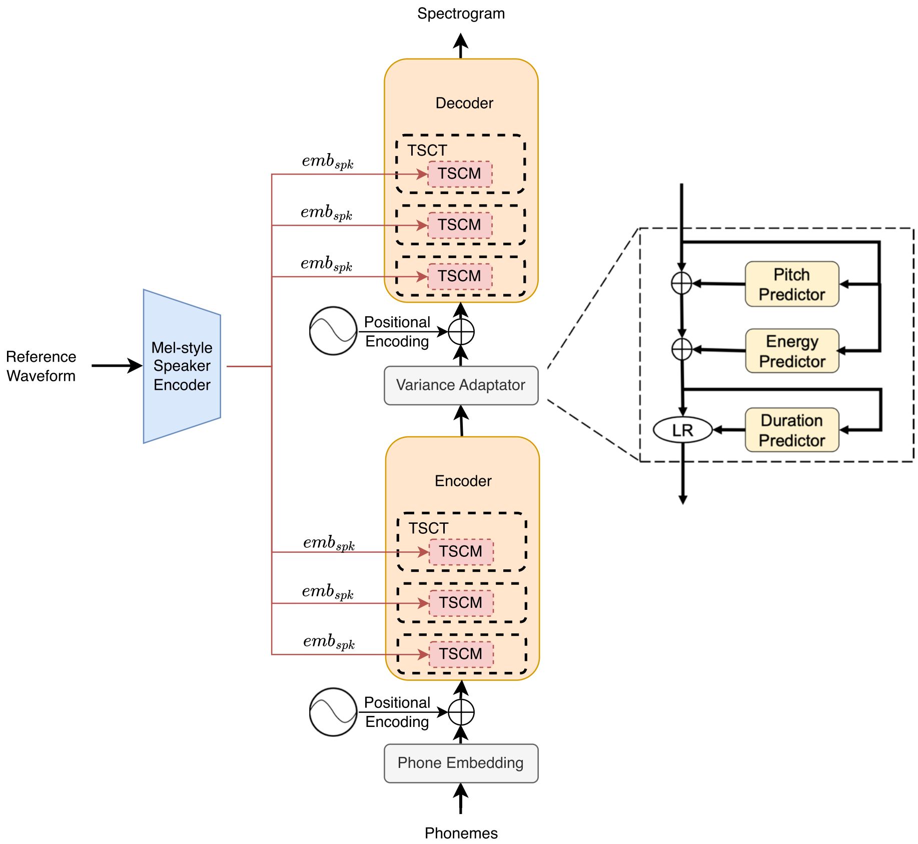

Fig. 1 Integration of our TSCM method into the FastSpeech 2 architecture. The TSCM-FastSpeech2 model utilizes a mel-style encoder to extract the latent speaker vector from the reference mel spectrogram of the target speaker and the TSCM-Transformer (TSCT) serving as an advanced control for speaker identity

module to match the length of the mel-spectrogram sequence. Figure 1 offers an overview of the baseline FastSpeech 2 architecture, augmented with our innovations for the sake of compactness. Detailed discussion of our modifications will follow in subsequent sections.

3.2 The architecture of VITS

The VITS model is described as a conditional variational autoencoder (CVAE) with adversarial learning for end-to-end text-to-speech (TTS) synthesis. It employs variational inference augmented with normalizing flows and an adversarial training process to enhance the expressive power of generative modeling. The model also proposes a stochastic duration predictor to handle the one-to-many relationship in speech synthesis, allowing for the generation of speech with diverse rhythms from text input. Through these mechanisms, the VITS model aims to improve the naturalness and efficiency of TTS systems, striving to generate more natural-sounding audio than conventional two-stage models. In Fig. 5, we show the basic VITS training architecture, which also includes our modifications for conciseness. Further details on these modifications will be provided in later sections

3.3 Basic methods for speaker control in zero-shot TTS

Consistent with established methods such as those in AdaSpeech [13] and YourTTS [17], incorporating a speaker identity encoding module is essential for a TTS model to generate speech in various desired voices. Specialized speaker encoders, like the mel-style encoder from [14] or the ECAPA-TDNN [28], are typically employed.

The conventional method for integrating speaker embeddings into the model's architecture typically includes either a direct addition or a combination of addition and multiplication with the model's hidden sequence output, as discussed in AdaSpeech [13] and supplementary literature [14]. Shown in Fig. 2, this naive approach can be mathematically described as follows:

$$h^{i}_{\text{next}} = f_{\theta}(\text{emb}_{\text{spk}}) * h^{i}_{\text{current}} + f_{\theta}(\text{emb}_{\text{spk}}) \tag{1}$$

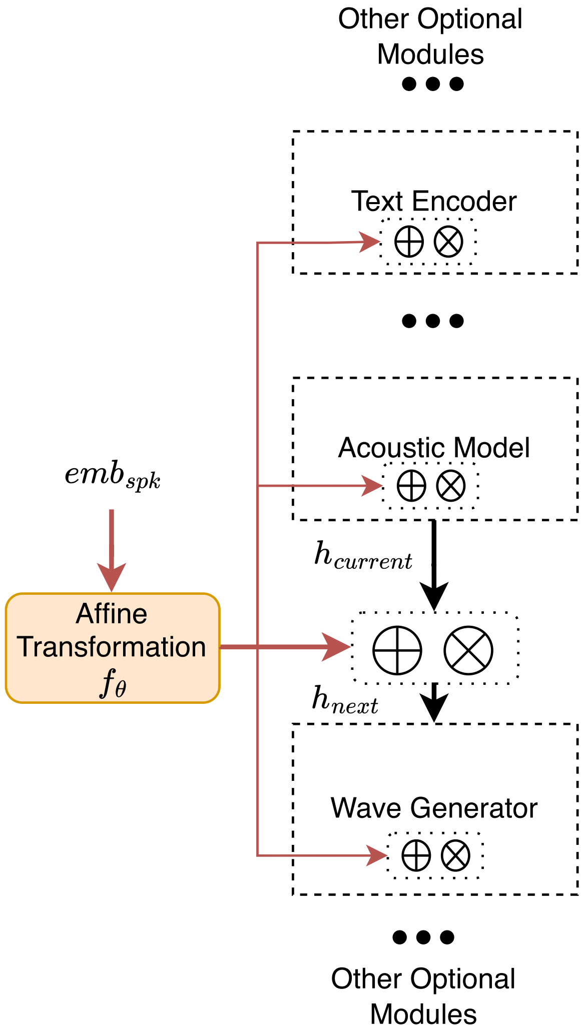

Fig. 2 A conceptual overview of the baseline style-control method for current state-of-the-art zero-shot TTS and voice cloning models

Here, f_θ serves as a transformation function to affine the speaker embedding $\mathrm{emb}_{\mathrm{spk}}$ to align with the dimensions of the current hidden state h_{current}. This operation is conducted frame by frame, ranging from $i = 1$ to T, where T denotes the total number of frames. The outcome h_{next} is a frame-level feature that is conditioned upon the identity of the speaker.

4 Two-branch Speaker Control Module (TSCM)

To enhance feature fusion beyond naive approaches described in the preliminaries, we aim for more efficient control of style, voice print, and semantic information from reference speech. This is critical for improving the effectiveness of sequence generation, resulting in better semantic accuracy, voice quality, and speaker similarity in generated speech. Our methodology leverages three fundamental and established sequence processing blocks: the Gated Convolutional Network (GCN) [31], GRU-based (RNN) modeling, and self-attention mechanisms. The integration of these components through the TSCM and TSCM-Transformer enables rapid and efficient zero-shot speaker and style control. Importantly, the TSCM is designed as an on-the-shelf module that can be seamlessly integrated into a variety of advanced TTS systems, thereby broadening its application for improved speaker and style adaptation. The design, implementation, and evaluation of these innovations are detailed in the subsequent sections.

4.1 Methodology for TSCM

To adapt popular text-to-speech models for voice cloning tasks, we utilize the mel-style speaker encoder module as mentioned in [14] to obtain the latent speaker vector emb_{spk}. This encoder improves adaptation performance for unseen speakers. As illustrated in Fig. 1, the mel-style encoder extracts the latent speaker vector emb_{spk} from the mel spectrogram of reference speech.

To efficiently control the speaker identity in the generated speech's mel spectrogram using the reference audio, we propose the advanced Two-branch Speaker Control Module (TSCM).

First, inspired by the Gated Convolutional Network (GCN) [31], we introduce a connection to the convolutional layers (*conv*1 and *conv*2), which is conditioned on the speaker vector as a Soft Gate, as illustrated in Fig. 3. This connection comprises a convolution layer (*conv*3) and a Sigmoid activation function layer. The enhanced CNN layers constitute the *CNN branch* of the TSCM block. The input sequential feature h_{input} is combined with the time-expanded latent speaker vector emb_{spk}. This combined input is then processed through the *conv*3 layer and Sigmoid function to generate a ranged-limited Soft Gate control signal. The output of the CNN branch, h_{cnn}, is represented as follows:

$$h_{cnn} = Sigmoid(conv(h_{input} + emb_{spk})) * conv(h_{input}) \tag{2}$$

where *Sigmoid* and *conv* represent the Sigmoid activation function and convolution operation, respectively. The CNN branch effectively models local frame-to-frame information in speech, ensuring that the output features are controlled by the speaker vector.

Additionally, we incorporate a gated recurrent unit (GRU) module, termed the *RNN branch*, to manage the overall speaker style of the generated sentence. The GRU module decodes each frame h_{input} with the speaker vector emb_{spk} as the initial state, described as follows:

$$h_{rnn}(t) = \begin{cases} GRU(emb_{spk}, h_{input}(t)) & \text{if } t = 1 \\ GRU(h_{rnn}(t-1), h_{input}(t)) & \text{else} \end{cases} \tag{3}$$

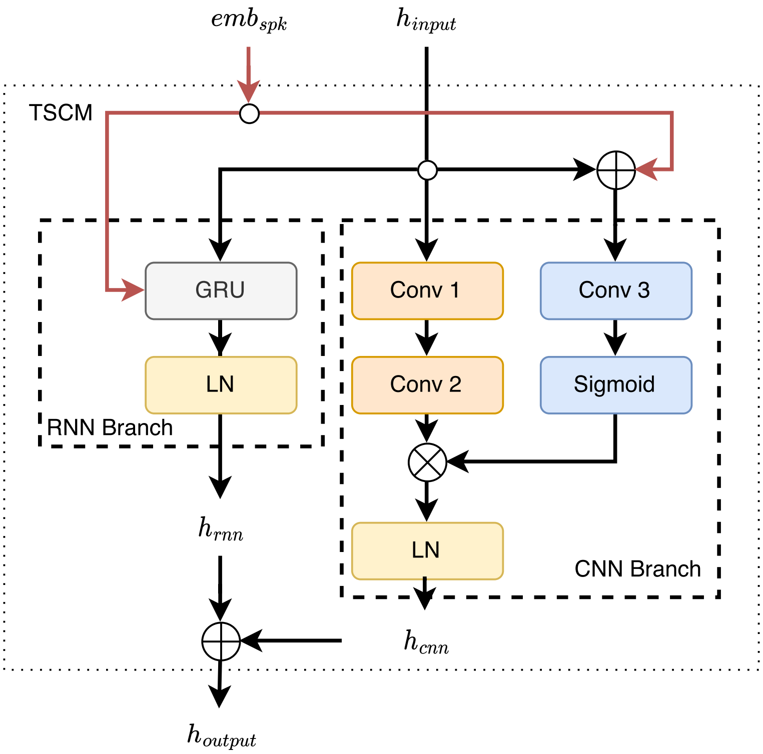

Fig. 3 The details of the TSCM block are shown in the figure, where the emb_{spk} is used for constraining the hidden state by introducing both the recurrent and a convolution branch. The addition and multiplication operations are represented by ⊕ and ⊗ respectively

where $h_{input}(t)$ and $h_{rnn}(t)$ represent the GRU's input and output frames at each step. The GRU outputs across all decoding steps are combined to form the final feature h_{rnn} of the RNN branch, enabling utterance-wide speaker style control.

Finally, we apply layer normalization (LN) [32] to both CNN and RNN branch outputs (h_{cnn} and h_{rnn}) and then combine them as the final output of each TSCM block. This module effectively controls the speaker identity at both utterance-wide and local frame levels.

4.2 Methodology for TSCM-Transformer (TSCT)

The Two-branch Speaker Control Module (TSCM) is naturally designed to be combined with the transformer encoder block, a component widely utilized in state-of-the-art TTS systems [15, 25]. Within the transformer encoder, the hidden state for the attention module, h_{attn}, exists at the phoneme and frame level and is typically used as the input for the feed-forward network (FFN), as depicted in Fig. 4. TSCM effectively replaces the FFN, enabling precise control of the speaker identity. This integration gives rise to what we designate as the TSCM-Transformer (TSCT). The output of TSCT, h_{output}, is processed from the block input h_{input} as follows:

$$h_{attn} = MultiHeadAttn(h_{input}) \tag{4}$$

$$h_{output} = TSCM(h_{attn}, emb_{spk}) \tag{5}$$

where $MultiHeadAttn(\cdot)$ represents the transformer's multi-head attention network. This seamless integration of TSCM with the transformer encoder not only allows for effective speaker identity control but also facilitates smoother incorporation with current transformer-based state-of-the-art TTS systems.

5 Integration of TSCM within the state-of-the-art TTS systems

5.1 Optimized integration of TSCM within the FastSpeech 2 framework

We have augmented the FastSpeech 2 architecture by incorporating the Two-branch Speaker Control Module (TSCM). This process replaces the speaker-control components of the AdaSpeech framework [13], which combined speaker embedding with the phoneme encoder's output hidden features using naive addition and utilized Speaker-Adaptive Layer Normalization in the mel decoder. Our method integrates the TSCM-Transformer (TSCT) into both the phoneme encoder

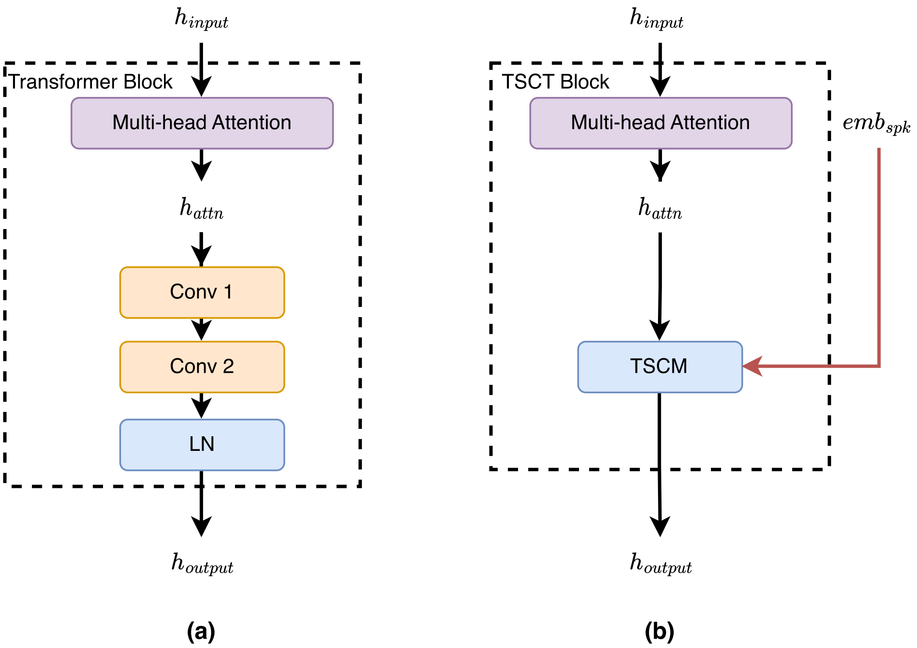

Fig. 4 The original feed-forward transformer block and the details of the TSCM block integration to the transformer models, resulting in the TSCT block

and mel-spectrogram decoder, as depicted in Fig. 4. The mel-style speaker encoder is employed for extracting speaker embeddings. The overall architecture, shown in Fig. 1, is referred to as TSCM-FastSpeech 2.

5.2 Optimized integration of TSCM within the VITS framework

The incorporation of the Two-branch Speaker Control Module (TSCM) into the VITS architecture aims to enable better speaker control across its three main modules: the text encoder, duration predictor, and generator. Figures 5 and 6 illustrate the overall architecture of this model for training and inference, respectively. The following sections outline the mathematical formulations and discussions related to this integration.

- **Duration predictor**: We apply TSCM to consider both temporal patterns and speaker-specific information. With text features w_{text} and speaker embedding emb_{spk}, the enhanced output, w_{dur}, is determined as follows:

$$w_{\text{dur}} = \text{TSCM}(w_{\text{text}}, \text{emb}_{\text{spk}}) \tag{6}$$

- **Generator**: The generator takes flow output features w_{flow} and combines them with the speaker embedding using TSCT. The modified output w_{gen} is then given by the following:

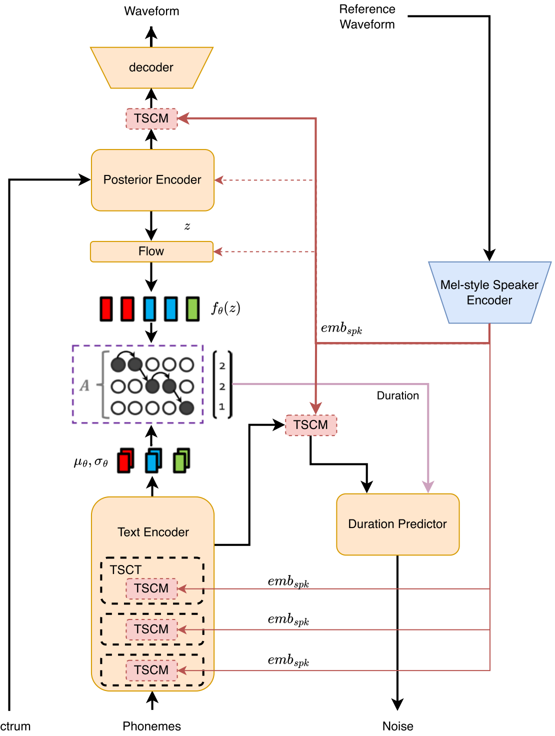

Fig. 5 Overview of our proposed TSCM-VITS model during the training procedure, adapted from [15]. This figure highlights the TSCM integration we introduced to the VITS framework, aiming to optimize feature fusion for zero-shot speaker adaptation in text-to-speech synthesis

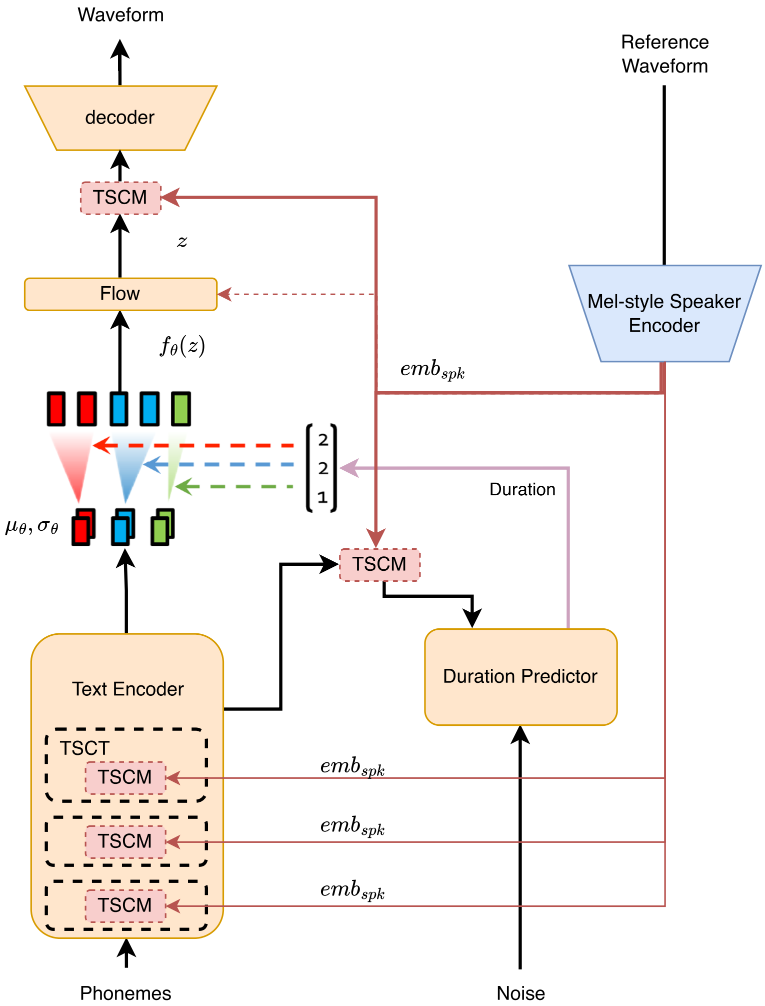

Fig. 6 Overview of our proposed TSCM-VITS model during the inference procedure, adapted from [15], with TSCM, our optimized feature fusion for zero-shot speaker adaptation in text-to-speech synthesis

$$w_{\mathrm{gen}} = \mathrm{TSCM}(w_{\mathrm{flow}}, \mathrm{emb}_{\mathrm{spk}}) \tag{7}$$

- **Text encoder**: TSCT is utilized in the text encoder to process the output from the multi-head attention modules in each transformer block. If w_{h} is the multi-head attention output, the resulting w_{out} is given by the following:

$$w_{\mathrm{h}} = \mathrm{MultiHeadAttention}(w_{\mathrm{in}}) \tag{8}$$

$$w_{\mathrm{out}} = \mathrm{TSCM}(w_{\mathrm{h}}, \mathrm{emb}_{\mathrm{spk}}) \tag{9}$$

Incorporating TSCM into the VITS framework significantly improves its core modules. The duration predictor gains enhanced contextual awareness, allowing for more accurate reflection of temporal patterns and speaker characteristics. The generator, with its refined ability to capture speaker-specific attributes, contributes to the naturalness and expressivity of the synthesized speech. Furthermore, the text encoder, through TSCM-Transformer (TSCT), more effectively contextualizes textual inputs with the speaker's unique tones and rhythms, leading to a robust text representation for subsequent processing. For the other modules in the VITS framework, we maintain the speaker-control method as utilized in YourTTS [17], as indicated by the dashed thin arrows in Figures 5 and 6. The integration of TSCM across these components is expected to yield highly expressive and personalized speech synthesis, and we denote this enhanced system as TSCM-VITS.

6 Experiments

6.1 Training setup

Our TSCM-TTS system and the corresponding FastSpeech 2-based and VITS-based baselines were trained on the *train-clean-360* subset of the LibriTTS multi-speaker TTS corpus [33]. For the purpose of few-shot fine-tuning and testing, we randomly selected 12 speakers, comprising an equal number of males and females. The rest of the data, encompassing utterances from 892 different speakers, was used for the initial training phase. A similar training and evaluation approach was adopted for the Chinese AiShell dataset [34], where 12 speakers were reserved for evaluation, and the remaining data was utilized to train the Chinese Mandarin versions of our proposed TSCM-TTS system and baseline models.

In our experiments, all utterances are resampled to a frequency of 22,050 Hz. We extract 80-dimensional mel spectrograms from these re-sampled waveforms. Each model is trained for 250,000 steps using a batch size of 50. Regarding other training considerations, such as losses and data processing strategies, we adhere to the protocols described in the original works. Specifically, for VITS models, we adopt the training settings from [15]. For FastSpeech 2, we adhere to the configurations outlined in [25]. Its inference process involves converting generated mel spectrograms to audio waveforms via a pretrained HiFi-GAN vocoder [35].

6.2 Evaluation details and metrics

6.2.1 Metrics

We employ an array of metrics to thoroughly evaluate the quality of generated speech and the comparative performance of different models. For subjective evaluation, we conduct human evaluations using the MOS (mean opinion score) for naturalness and the SMOS (similarity MOS) for speech similarity.

For objective assessments, we utilize a third-party pretrained speaker verification model[3] to extract speaker embeddings from the provided speech. We compute the speaker encoder cosine similarity (SECS) by comparing speaker embeddings from both synthesized and ground-truth speeches. Additionally, we measure mel-cepstral distortion (MCD) to gauge structural disparities [36]. Dynamic time warping (DTW) is employed to synchronize unequal-length speech sequences prior to their comparison [37]. Word error rate (WER) is determined using a CNN-Transformer ASR pretrained model from SpeechBrain[4]. These metrics align closely with the evaluation methodologies detailed in prominent studies such as [17, 30], thereby ensuring our assessment approach is consistent with current best practices in the field.

6.2.2 Zero-shot and few-shot setups

For the few-shot setup, we select approximately 15 text-audio pairs per speaker and fine-tune them for about 1k steps using the test set. In the zero-shot scenario, a single utterance serves as the reference audio, and speech is generated in line with the provided text for each speaker. We select 15 text samples for each speaker from the testing set, ensuring they are distinct from those used during training or fine-tuning, to form the test list. Subsequently, the generated speeches are evaluated using both subjective and objective metrics, with the objective evaluations encompassing all generated speech files.

6.2.3 Subjective evaluation

We established a web-based system for subjective assessments. A set of 16 utterances was randomly chosen from the test list. For each utterance, outputs from all systems, along with the reference audio, were randomly presented on a single webpage. In the MOS assessment, listeners rated the synthesized utterance quality. In the SMOS assessment, listeners gauged the similarity between actual and synthesized utterances for the same speaker. Both MOS and SMOS used a 1 to 5 rating scale. For each test, over 15 judges participated, each rating the 16 randomly chosen utterances from all participated systems.

6.3 Other implementation details on model configurations

For TSCM-TTS systems, the mel-style speaker encoder architecture is based on [14]. The foundational modules of TTS systems align with FastSpeech 2 [25] and VITS [15]. Within each feed-forward transformer block, the multi-head attention design adheres to FastSpeech 2, but the feed-forward layers are replaced with the TSCM. The CNN branch of TSCM employs dilated convolution, with kernel sizes for *Conv*2 and *Conv*3 set to 1 and *Conv*1 set to 9. The GRU unit's hidden size in the RNN branch is 256. A dropout [38] with a rate of 0.2 precedes the LN in each TSCM branch. For baseline systems, we adopted *speaker control modules* from AdaSpeech [13] and YourTTS [17] for their respective FastSpeech 2 and VITS architectures. All other configurations, particularly the speaker encoder and training setup, are kept consistent with TSCM-TTS to guarantee a fair comparison.

7 Results and ablation studies

7.1 Comparative results for TSCM on FastSpeech 2

In our comparative experiments, we evaluated the performance of several models. The basic *GT mel + HiFi-GAN* model utilizes the ground-truth mel spectrogram,

[3] https://github.com/resemble-ai/Resemblyzer

[4] https://huggingface.co/speechbrain/asr-transformer-transformerlm-librispeech

Table 2 Comparative metrics of FastSpeech 2-based models on LibriTTS

Model	Type	WER (%) ↓	MCD (dB) ↓	SECS ↑	MOS ↑	SMOS ↑
GT	-	-	-	-	4.56	4.48
GT mel + HiFi-GAN	-	-	2.99	0.97	4.16	4.04
FS2 w/o TSCM control [12]	Zero-shot	7.05	7.49	0.76	2.80	2.79
TSCM-FastSpeech 2 (ours)	Zero-shot	**6.73**	**7.29**	**0.77**	**2.91**	**2.85**
FS2 w/o TSCM control [12]	Few-shot	6.84	7.28	0.86	3.38	3.00
TSCM-FastSpeech 2 (ours)	Few-shot	**6.50**	**7.03**	**0.87**	**3.81**	**3.69**

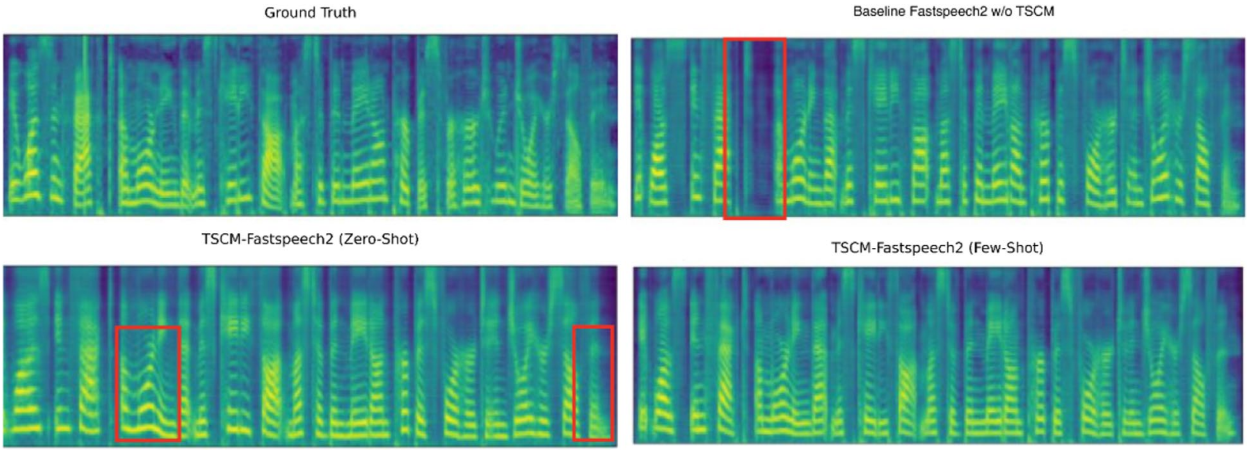

Fig. 7 Comparison of mel spectrograms synthesized by baseline models and TSCM-FastSpeech 2

which is directly converted into audio using a pretrained HiFi-GAN vocoder. The baseline model, *FS2 without TSCM control*, is the integration of FastSpeech 2 with a latent speaker vector obtained via the mel-style encoder, and it employs style control based on the methods and positions outlined in *AdaSpeech* [12], as detailed in Section 3.3. Lastly, our proposed *TSCM-FastSpeech 2* builds upon FastSpeech 2 by incorporating the TSCM-Transformer (TSCT) block.

Referring to Table 2, our TSCM-FastSpeech 2 model, tested on the LibriTTS dataset, demonstrates superior performance. The MCD metric for our model improves by approximately 0.20 and 0.25 compared to the zero-shot and few-shot variants of the baseline model, respectively. Similarly, the SMOS score for our method registers an increase of about 0.13 and 0.69 over the zero-shot and few-shot versions of *FS2 without TSCM control*.

In a closer examination of the synthesis quality, the mel spectrograms further substantiate our findings. Referring to Fig. 7, the mel spectrograms generated by different systems are compared directly against the ground-truth (GT) spectrogram. The TSCM-FastSpeech 2 model clearly yields superior synthesis quality. When set side by side with the GT spectrogram, the TSCM-FastSpeech 2 version reveals fewer spectral inconsistencies and captures more granular details. In contrast, the baseline model exhibits certain spectral gaps and mismatches. The regions marked with red rectangles highlight these spectral disparities. Such spectral visualizations reaffirm that the TSCM-FastSpeech 2 system, even in a zero-shot setup, closely matches the fidelity of the GT spectrogram more than its counterparts.

These metrics suggest that our synthesized speech more closely resembles authentic speech, offering a more natural listening experience and improved similarity to the speaker's unique voice print.

7.2 Comparative results for TSCM on VITS

Table 3 presents the performance metrics of various VITS-based models on the LibriTTS dataset, with *ground truth (GT)* serving as the benchmark. The *VITS without TSCM control* model is our baseline for zero-shot voice cloning. This enhanced the VITS architecture by integrating *YourTTS* [17] features, applying speaker control techniques as detailed in Section 3.3. Our *TSCM-VITS* model outperforms the baseline and other adaptations, showing superior performance in all evaluated subjective and objective metrics. Notably, despite the baseline's

Table 3 Comparative metrics of VITS-based TSCM models on LibriTTS

Model	Type	WER (%) ↓	MCD (dB) ↓	SECS ↑	MOS ↑	SMOS ↑
GT	-	-	-	-	4.56	4.48
VITS w/o TSCM control [15, 17]	Zero-shot	6.44	7.00	0.78	3.76	3.68
TSCM-VITS (ours)	Zero-shot	**6.03**	**6.87**	**0.80**	**4.12**	**3.79**
VITS w/o TSCM control [15, 17]	Few-shot	6.65	6.70	0.85	3.88	3.81
TSCM-VITS (ours)	Few-shot	**5.68**	**6.57**	**0.87**	**4.30**	**4.12**

Table 4 Comparative metrics of VITS-based TSCM models on AiShell (Chinese)

Model	Type	WER (%) ↓	MCD (dB) ↓	SECS ↑	MOS ↑	SMOS ↑
GT CN	-	-	-	-	4.60	4.61
VITS w/o TSCM control [15, 17]	Zero-shot	12.21	6.95	0.80	4.32	4.30
TSCM-VITS (ours)	Zero-shot	**10.96**	**6.83**	**0.81**	**4.44**	**4.43**

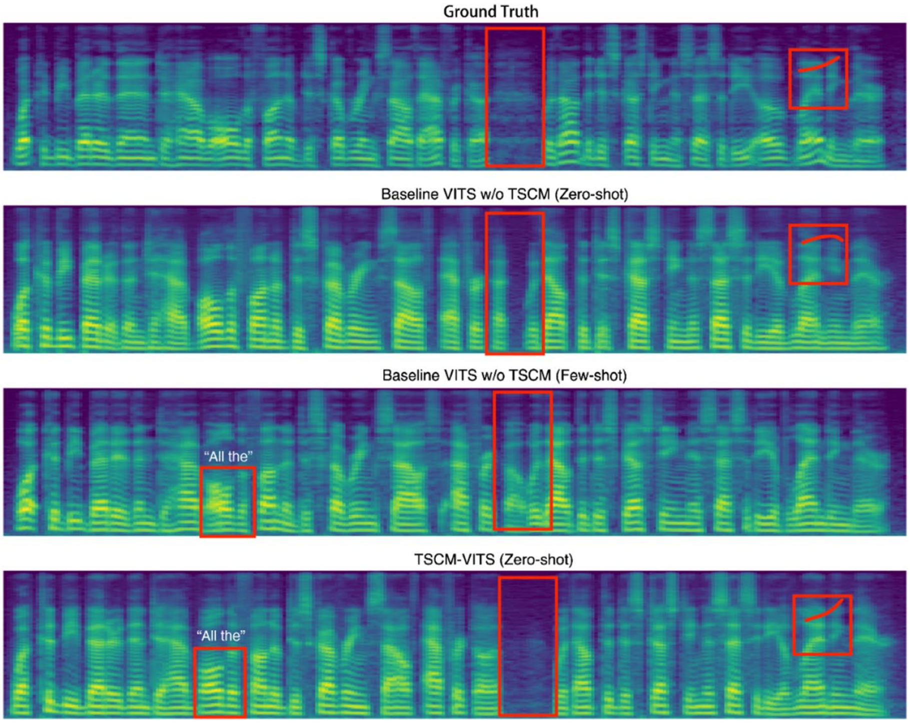

Fig. 8 Comparison of mel spectrograms synthesized by baseline models and TSCM-VITS

improved performance in few-shot scenarios, TSCM-VITS consistently excels, underlining its effectiveness.

In Table 4, we extend our analysis to models trained on the Chinese AiShell3 dataset, focusing on zero-shot TTS performance. This comparison reaffirms the TSCM-VITS model's consistent superiority over baseline systems in both Chinese AiShell and English LibriTTS datasets across every evaluated metric, showcasing its robust applicability and effectiveness in voice cloning across languages.

Upon closely examining the mel spectrograms in Fig. 8, it is clear that the TSCM-VITS system performs better than the others. When we compare the outputs to the ground-truth (GT) spectrogram, the TSCM-VITS system

Table 5 Performance comparative metrics of our TSCM-TTS with other advanced zero-shot TTS systems on English speech synthesis

Model	**Subjective**		**Objective**	
	MOS ↑	**SMOS ↑**	**MCD (dB) ↓**	**SECS ↑**
GT	4.56	4.48	-	-
Microsoft VALL-E-X (2023 DEMO) [29, 39]	3.80	3.65	9.04	0.74
Coqui AI XTTS v2 (2023 DEMO) [27, 40]	4.05	3.60	9.35	0.72
OpenVoice (2024 DEMO) [6, 41]	4.04	3.95	8.37	0.79
LauraTTS (2024 DEMO) [30, 42]	4.08	3.90	8.40	**0.83**
TSCM-VITS zero-shot (ours)	**4.30**	**4.12**	**6.79**	0.81

has fewer incorrect parts, and its details are clearer. Additionally, the tone of the TSCM-VITS is closer to what is in the GT. An example of this is in the area highlighted by the red rectangle at the end of the spectrogram. In the GT, the tone goes up, while the baseline model's tone goes down. However, the TSCM-VITS correctly follows the upward tone seen in the GT. This shows that the TSCM-VITS system can produce outputs that are much closer to the original audio, making it a reliable choice for speech synthesis.

These results clearly highlight that in both zero-shot and few-shot scenarios, the TSCM-VITS consistently outperforms the baselines with all metrics. Such findings emphasize the benefits of integrating TSCM into the VITS architecture.

7.3 Comparative voice cloning performance of TSCM-TTS with state-of-the-art systems

Table 5 presents a comparative analysis of our *TSCM-VITS zero-shot* model against leading *zero-shot TTS* systems, evaluated using the latest online demos as referenced in the table. Among these, the *Microsoft VALL-E-X* system emerges as a notable voice cloning solution from 2023, alongside *CoquiAI's XTTS*, which leverages the TorToise framework for high-quality voice cloning. Our model, *TSCM-VITS zero-shot*, builds upon the VITS architecture by incorporating the TSCM module, enhancing voice quality and speaker similarity as detailed in Section 7.2.

Additionally, we benchmark against recent advancements such as *OpenVoice*, a cutting-edge non-AR method, and *LauraTTS*, which utilizes a GPT-based AR approach. Despite these systems employing more complex multistage training and inference processes, our TSCM-VITS zero-shot model demonstrates competitive performance across key metrics, underscoring its effectiveness in voice quality and speaker similarity. Our approach distinguishes itself by being a direct, lightweight enhancement to the VITS architecture, offering state-of-the-art performance without the complexity of multistage methods, thus presenting a promising avenue for future research integration.

Expanding our evaluation to the AiShell3 Chinese dataset, as documented in Table 6, *TSCM-VITS* consistently outperforms these state-of-the-art systems in multilingual contexts, including *Microsoft VALL-E-X* and *Coqui AI's XTTS*. This reaffirms the *TSCM-VITS* module's superiority in delivering high-quality voice cloning across both English and Chinese datasets. Fig. 9 further demonstrate the mel-spectrum comparison of these advanced systems, indicating the effectiveness of our proposed method.

Additionally, Table 7 outlines the datasets utilized by the state-of-the-art (SOTA) systems featured in Table 5. Employing the standard LibriTTS dataset settings, our proposed TSCM-TTS system outperforms other SOTA systems, affirming its effectiveness.

As shown in Fig. 10, we present a t-SNE visualization of speaker embeddings for enrolled utterances and synthesized utterances, extracted using the Resemblyzer package. The XTTS system, while generating audios with superior auditory effects, not only falls short in terms of speaker control but also exhibits speaker confusion, as indicated by the dispersed positioning of the enrollment utterances relative to the synthesized audios. The VALL-E-X system demonstrates improved speaker control, with the enrollment utterances positioned more centrally within the cluster of synthesized audio embeddings.

Table 6 Performance comparative metrics of our TSCM-TTS with other advanced zero-shot TTS systems on Chinese speech synthesis

Model	**Subjective**		**Objective**	
	MOS ↑	**SMOS ↑**	**MCD (dB) ↓**	**SECS ↑**
GT CN	4.60	4.61	-	-
Microsoft VALL-E-X CN [29, 39]	3.72	3.73	8.51	0.80
Coqui AI XTTS v2 CN [27, 40]	4.11	4.10	9.35	0.72
TSCM-VITS zero-shot CN (ours)	**4.44**	**4.43**	**6.83**	**0.81**

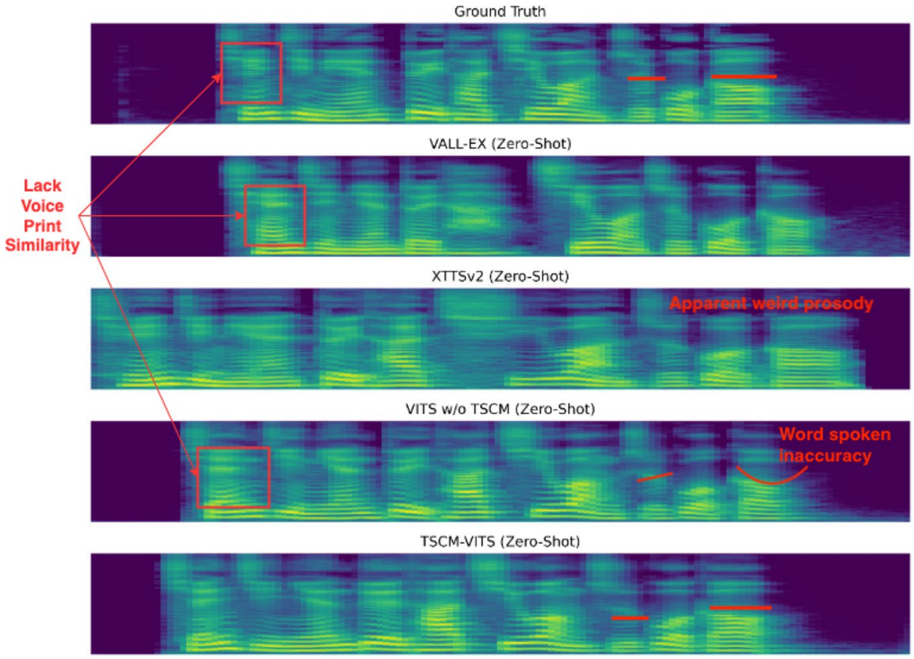

Fig. 9 Comparison of mel spectrograms synthesized by several advanced ZS-TTS systems and TSCM-VITS in Chinese zero-shot scenarios. The TSCM-VITS shows better voice print similarity and better word spoken accuracy. More samples can be found in our online demo

Table 7 A concise overview of the datasets employed to train various state-of-the-art TTS models (English version), as reflected in the datasets underlying their publicly available demo versions

Model	Training datasets
VALL-E-X	LibriLight (60k h), etc. [29]
XTTS v2	LibriTTS, VTCK, LJSpeech, etc. [43]
OpenVoice	LibriTTS
LauraTTS	LibriTTS, etc. [30]
TSCM-TTS (ours)	LibriTTS (360 h)

However, it is worth noting that the audio quality of VALL-E-X does not match that of other systems when evaluated by additional metrics. Our system, TSCM-TTS, exhibits superior speaker control, shown by the dense clustering of synthesized audio embeddings around the enrollments. Furthermore, according to various metrics, our system also delivers improved audio quality.

This comparison highlights the TSCM-TTS's capability in achieving superior zero-shot voice cloning with high-quality speech synthesis.

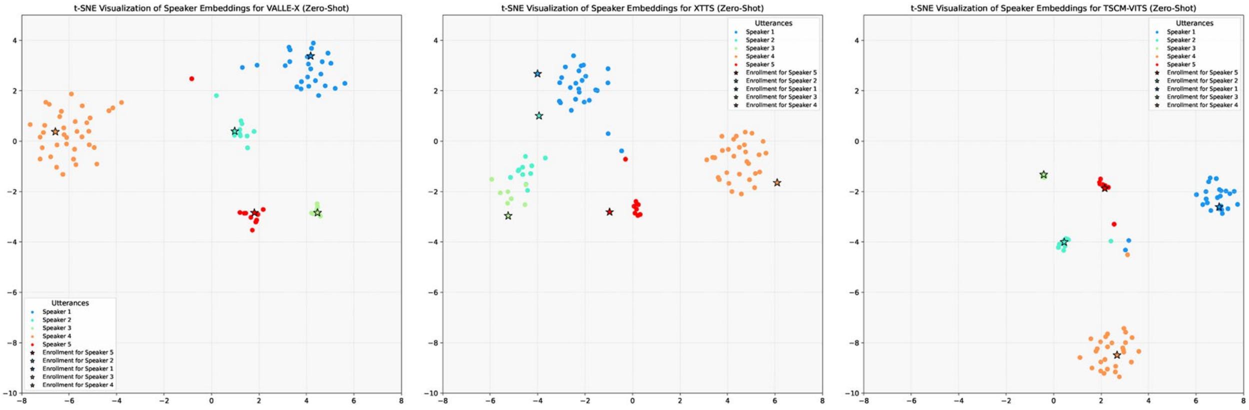

Fig. 10 Comparison of speaker embeddings: TSCM-VITS versus state-of-the-art zero-shot TTS systems

Table 8 Ablation study on TSCM's effectiveness in speaker control on TSCM-FastSpeech 2

Model	MCD (dB) ↓	SECS ↑
TSCM CNN only	7.16	0.858
TSCM RNN only	7.24	0.843
TSCM parallel (proposed)	**7.03**	**0.866**

Table 9 Comparative analysis of *in-domain multi-speaker TTS (MS-TTS)* systems: proposed TSCM-VITS versus VITS+ baseline

Model	MCD (dB) ↓	SECS ↑
VITS+ MS-TTS [15, 16]	6.11	0.875
TSCM-VITS MS-TTS (ours)	**5.45**	**0.883**

7.4 Ablation study and auxiliary analysis

7.4.1 Ablation study on TSCM's effectiveness in speaker control

We conducted an ablation study to assess the contributions of the CNN and RNN branches within the TSCM for speaker control in voice cloning. For this purpose, we compared the primary architecture, *TSCM parallel* with two other variants: *TSCM CNN* which employs only the CNN branch within the TSCM block and *TSCM RNN* which utilizes solely the RNN branch within the TSCM block.

Table 8 showcases the outcomes. The results clearly show that using both the CNN and RNN branches in *TSCM parallel* gives better performance than using them individually, achieving a lower MCD score. This combined approach, using both local feature control from the CNN and utterance-level control from the RNN, provides more effective speaker-specific modeling. Our findings underscore the importance of the TSCM module with both branches for enhanced speaker control in voice cloning systems.

7.4.2 Auxiliary analysis of in-domain speakers: a comparative study of TSCM-VITS and VITS+ in multi-speaker TTS systems

In addition to voice cloning, we also assessed the performance of our proposed TSCM-VITS system in the context of *in-domain multi-speaker text to speech (MS-TTS).* For baseline comparison, we employed an advanced version from the VITS series, termed *VITS+*. This system features an enhanced speaker control branch integrated into its text encoder, as described in [16], a leading work in the field. Table 9 presents our findings. Notably, the TSCM-VITS demonstrates superior speaker control, which is reflected in its improved objective results.

7.4.3 Comparison of few-shot learning capability

We investigate the influence of few-shot fine-tuning steps on the efficacy of our proposed system, setting the step range from 0 to 10,000. The experimental setups remain consistent with prior descriptions. The zero-shot scenario corresponds to a step count of 0, where the model, as trained, is employed directly for testing.

From Fig. 11, a significant enhancement in MCD is observed as the training steps increase. The SECS value shows a subtle upward trend throughout the fine-tuning process. Meanwhile, the WER exhibits a gentle decrease with increased steps, hinting at improved recognition accuracy. Distinctly, the TSCM-VITS model outperforms the TSCM-FastSpeech 2 consistently across the metrics. It is noteworthy that the most prominent performance boost is observed between 0k and 1k steps, underscoring the effectiveness of the fine-tuning phase. After 5,000 steps, the enhancements in SECS and WER metrics stabilize. Given these observations, we determine an optimal fine-tuning range of 1 to 5k steps for our few-shot experiments, as evidenced in prior sections.

We then delve into the implications of varying fine-tuning sample sizes on the performance of our proposed system. The fine-tuning sample size is varied from 5 to 30, with increments set at intervals of 5.

As illustrated in Fig. 12, a notable enhancement in MCD, WER, and SECS is observed up to the first 15 samples. The TSCM-VITS model consistently outperforms its counterpart, TSCM-FastSpeech 2, across all sample sizes. Notably, post the 15-sample mark, the progression in the evaluation metrics begins to stabilize, with only marginal improvements discerned. Taking into account the balance between achieving optimal performance and the availability of training data, a fine-tuning sample size of 15 emerges as the optimal choice for the TSCM-TTS systems. This selection has been employed in the few-shot experiments showcased in the preceding Sections 7.1 and 7.2.

In few-shot scenarios, our TSCM-TTS system efficiently achieves improved voice cloning with minimal data, enabling fast adaptation with reduced computational resources. This underscores the system's practicality and effectiveness in rapidly adjusting to new speakers or styles.

7.4.4 Comparison of TSCM using word-level spoken quality

We evaluate the semantic-level spoken correlation through a word-level spoken correlation score. This metric, detailed in Table 10, alongside other subjective evaluations, collectively underscores the TSCM-VITS system's enhanced ability to capture semantic information. This is reflected in the improved naturalness and contextual coherence of word pronunciation within sentences.

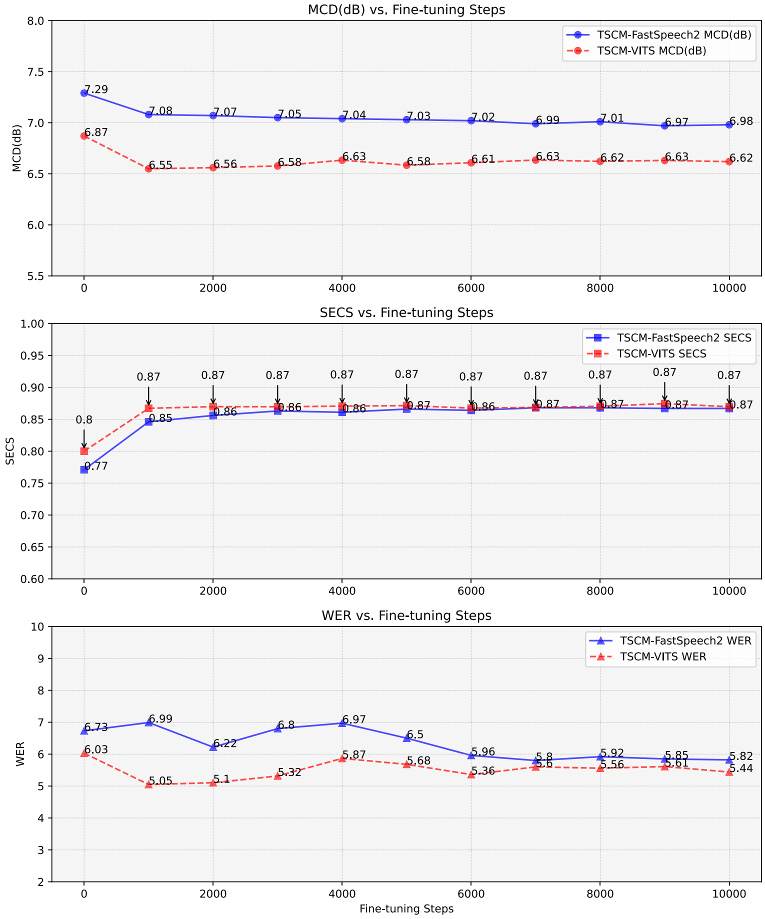

Fig. 11 The impact of *fine-tuning steps* on the performance of our proposed systems in few-shot scenarios

To quantify the improvement of the TSCM systems using an objective metric that correlates with subjective perceptions, we introduce a semantic-related auxiliary metric: the word alignment score. This score is defined as follows:

$$score_{word} = \text{mean}\left(\left|\frac{d_{predict} - d_{word}}{d_{word}}\right| \times 100\right). \quad (10)$$

Here, d_{word} represents the duration of each word in the ground truth, measured using the Montreal Forced Aligner (MFA) algorithm, and $d_{predict}$ denotes the predicted word duration by the TTS system. A lower score indicates better alignment with the ground truth, signifying more accurate word-level spoken timing.

Our TSCM-VITS system demonstrates superior performance compared to both baseline and state-of-the-art systems in this objective evaluation. This underscores our system's enhanced ability for zero-shot voice cloning, producing utterances that more closely mimic the ground truth in terms of spoken word alignment and overall sound quality.

7.4.5 *Computational speed analysis*

In this section, we conduct a computational cost analysis to evaluate the efficiency of the proposed TSCM module, focusing on training and inference speeds. This analysis is carried out on Nvidia 30 series GPUs, typical of server-side GPU setups. It is important to note that both the baseline systems and our TSCM-VITS framework are configured for server-side execution. None of these systems, including ours, have been optimized specifically for the trade-off between computational efficiency and performance. Therefore, our comparative evaluation of training and inference speeds is conducted solely in the context of server-side performance.

The results, detailed in Table 11, demonstrate that incorporating the TSCM module into the VITS framework marginally decreases computational speed. Training speed sees a 13% reduction, while inference speed decreases by 15% compared to the baseline VITS model without the TSCM module. Despite these speed

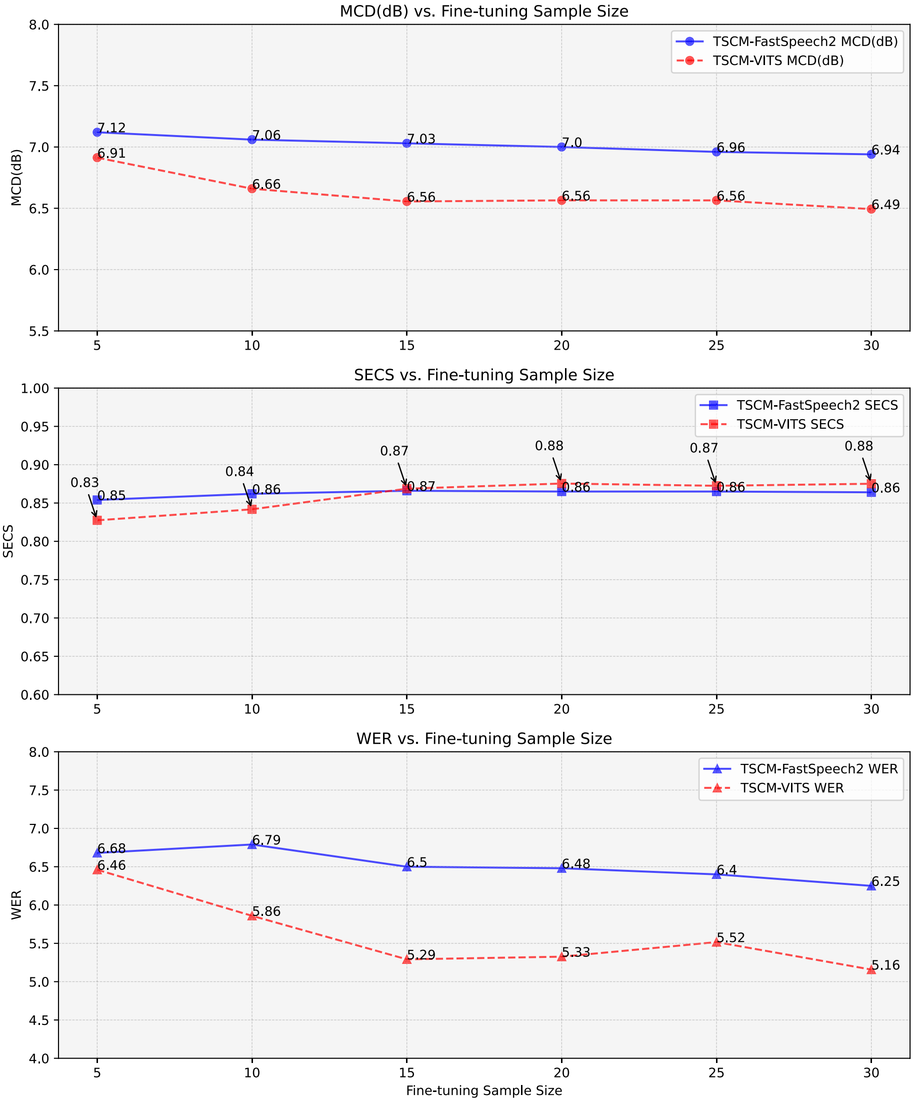

Fig. 12 The impact of *fine-tuning samples* on the performance of our proposed systems in few-shot scenarios

Table 10 Auxiliary analysis for TSCM-TTS: word-level spoken alignment

Model	Word alignment score (%)↓
VALL-E-X	20.04
XTTS v2	15.22
OpenVoice	18.93
LauraTTS	25.08
VITS w/o TSCM	15.33
TSCM-VITS (ours)	**14.23**

reductions, the overall performance impact remains minimal, suggesting that the addition of the TSCM module introduces a negligible effect on user experience regarding computational efficiency.

Table 11 Computational speed analysis for TSCM-TTS

Methods	Training speed (it/s)↑	Inference speed (sentence/s)↑
VITS w/o TSCM	1.578	5.284
TSCM-VITS (ours)	1.374	4.471

8 Conclusion and future works

In the realm of voice cloning (VC) and zero-shot TTS (ZS-TTS) technologies, a significant challenge lies in optimally utilizing style-control vectors or speaker embeddings within the TTS pipeline. This often results in diminished control over style, voice print, and semantic information from reference speech, leading to suboptimal outcomes. Our study introduces the Two-branch Speaker Control Module-based TTS

(TSCM-TTS), a novel approach aimed at improving voice customization capabilities in voice cloning systems. This method facilitates enhanced zero-shot and few-shot speaker adaptation using minimal reference audio samples within the current voice cloning architecture.

The adaptability and versatility of TSCM are noteworthy, enabling its straightforward integration into existing leading TTS models. We have detailed the incorporation of TSCM with FastSpeech 2 and VITS, two advanced TTS frameworks. This integration significantly boosts their performance, especially in zero-shot TTS scenarios for TSCM-VITS, even with basic training settings, surpassing many state-of-the-art systems in multi-language evaluations. Our comprehensive comparative analyses demonstrate the benefits of the TSCM-centric approach, with both objective and subjective evaluations confirming its superiority.

Looking ahead, several promising directions have emerged for further enhancing our work. The TSCM could be seamlessly extended for speech style control, potentially being applied in emerging multimodal speech style editing tasks to modulate speech emotion, language, and accent. Such advancements aim to expand its utility across various speech synthesis challenges. Additionally, with the growing interest in artificial general intelligence (AGI) and large language models (LLMs), TSCM's integration into LLM-based ZS-TTS models for improved control presents an interesting direction for research. This will require more extensive study to understand how its integration with TSCM-based style and speaker control modules can be effectively realized.

Abbreviations

ZS-TTS	Zero-shot text-to-speech synthesis
VC	(Zero-shot) voice cloning (=ZS-TTS)
TSCM	Two-branch Speaker Control Module
VITS	An advanced CVAE-based TTS model [15]
FS2	FastSpeech 2, a fast TTS model
TSCT	TSCM-based transformer
ZS/FS	Zero-shot learning/few-shot learning

Acknowledgements

We would like to express our gratitude to the various funding agencies that have supported this work.

Authors' contributions

ZC and ZA, as the first authors with equal contributions, designed the study, conducted the experiments, and were involved in the analysis and interpretation of the data. XL also took part in performing the experiments. YM played a role in conceptualizing the study. Prof. SX, the corresponding author, was instrumental in the conception and overall design of the study and provided supervision for the research project. All authors were actively involved in drafting and revising the manuscript. Furthermore, every author has read and approved the final version of the manuscript.

Authors' information

• Zhiyong Chen received his M.Eng. degree in Communication Engineering from Shanghai University, China, in 2021. He is currently pursuing a Ph.D. in Information and Communication Engineering at the same institution. His research interests include speaker recognition, speech recognition, acoustics, and emerging machine learning paradigms.
• Zhiqi Ai received his B.Eng. degree in Communication Engineering from Shanghai University, China, in 2021, where he continued to pursue the M.Eng. degree in Signal and Information Processing. His research interests include speaker recognition, keyword spotting, speech synthesis, and talking face generation.
• Youxuan Ma received his B.Eng. degree in Communication Engineering from Shanghai University, China, in 2020, where he continued to receive the M.Eng. degree in Communication and Information System, in 2023. His research interests include speech synthesis and voice anti-spoofing.
• Xinnuo Li is currently a senior undergraduate student in Communication Engineering in Shanghai University. His research interests include speech synthesis, generative models, and deep learning.
• Shugong Xu (Fellow, IEEE) received his master's degree in pattern recognition and intelligent control and a Ph.D. degree in EE from Huazhong University of Science and Technology, China. He is a Professor at Shanghai University, where he was the Founding Head of the Shanghai Institute for Advanced Communication and Data Science. Before joining Shanghai University, he held positions at Intel Labs, Huawei Technologies, Sharp Laboratories of America, and conducted research at the City College of New York, Michigan State University, and Tsinghua University.
Prof. Xu has published over 160 peer-reviewed research papers, holds more than 60 US and China patents, and has made significant contributions to international standards such as IEEE 802.11, 3GPP LTE, and DLNA. His research interests include 6G wireless communication systems, machine learning, pattern recognition, and AI-enabled embedded systems.
In recognition of his work, Prof. Xu was awarded the "National Innovation Leadership Talent" by the China government in 2013 and elevated to IEEE Fellow in 2015. He also received the 2017 Award for Advances in Communication from the IEEE Communications Society.

Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61871262, 61901251, and 62071284; the Innovation Program of Shanghai Municipal Science and Technology Commission under Grants 21ZR1422400, 20JC1416400, and 20511106603; Pudong New Area Science & Technology Development Fund, Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101130012; and Foshan Science and Technology Innovation Team Project under Grant FS0AA-KJ919-4402-0060.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the authors upon reasonable request. Any additional materials, such as code or supplementary information, can also be provided by the authors upon request. The relevant audio data and DEMO systems that support the findings of this study are also available at https://great-research.github.io/tsct-tts-demo/ following the date of publication.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 17 November 2023 Accepted: 8 May 2024
Published online: 28 May 2024

References

1. Q. Xie, X. Tian, G. Liu, K. Song, L. Xie, Z. Wu, H. Li, S. Shi, H. Li, F. Hong, H. Bu, X. Xu, in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. The multi-speaker multi-style voice cloning challenge 2021 (IEEE, 2021), p. 8613–8617
2. X. Tan, T. Qin, F. Soong, T.Y. Liu, A survey on neural speech synthesis. (2021). arXiv preprint arXiv:2106.15561
3. S. Arik, J. Chen, K. Peng, W. Ping, Y. Zhou, Neural voice cloning with a few samples. Adv. Neural Inf. Process. Syst. **31** (2018)

4. B.J. Choi, M. Jeong, J.Y. Lee, N.S. Kim, Snac: Speaker-normalized affine coupling layer in flow-based architecture for zero-shot multi-speaker text-to-speech. IEEE Signal Process. Lett. **29**, 2502–2506 (2022)
5. S.J. Cheon, B.J. Choi, M. Kim, H. Lee, N.S. Kim, A controllable multi-lingual multi-speaker multi-style text-to-speech synthesis with multivariate information minimization. IEEE Signal Process. Lett. **29**, 55–59 (2022)
6. Z. Qin, W. Zhao, X. Yu, X. Sun, OpenVoice: versatile instant voice cloning. (2023). arXiv preprint arXiv:2312.01479
7. W. Ping, K. Peng, A. Gibiansky, S.Ö. Arik, A. Kannan, S. Narang, J. Raiman, J. Miller, in *ICLR*. Deep Voice 3: scaling text-to-speech with convolutional sequence learning (ICLR, 2018)
8. Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. Lopez Moreno, Y. Wu, et al., Transfer learning from speaker verification to multispeaker text-to-speech synthesis. Adv. Neural Inf. Process. Syst. **31** (2018)
9. J. Shen, R. Pang, R.J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, et al., in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions (IEEE, 2018), p. 4779–4783
10. Y. Wang, D. Stanton, Y. Zhang, R.S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, R.A. Saurous, in *International Conference on Machine Learning*. Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis (PMLR, 2018), p. 5180–5189
11. E. Cooper, C.I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, J. Yamagishi, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings (IEEE, 2020), p. 6184–6188
12. Y. Wu, X. Tan, B. Li, L. He, S. Zhao, R. Song, T. Qin, T.Y. Liu, in *Proc. Interspeech 2022*. AdaSpeech 4: adaptive text to speech in zero-shot scenarios (ISCA, 2022), p. 2568–2572
13. M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, T.Y. Liu, et al., in *International Conference on Learning Representations*. AdaSpeech: adaptive text to speech for custom voice (ICLR, 2020)
14. D. Min, D.B. Lee, E. Yang, S.J. Hwang, in *International Conference on Machine Learning*. Meta-StyleSpeech: multi-speaker adaptive text-to-speech generation (PMLR, 2021), p. 7748–7759
15. J. Kim, J. Kong, J. Son, in *International Conference on Machine Learning*. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech (PMLR, 2021), p. 5530–5540
16. J. Kong, J. Park, B. Kim, J. Kim, D. Kong, S. Kim, VITS2: improving quality and efficiency of single-stage text-to-speech with adversarial learning and architecture design. (2023). arXiv preprint arXiv:2307.16430
17. E. Casanova, J. Weber, C.D. Shulby, A.C. Junior, E. Gölge, M.A. Ponti, in *International Conference on Machine Learning*. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone (PMLR, 2022), p. 2709–2720
18. G. Liu, Y. Zhang, Y. Lei, Y. Chen, R. Wang, Z. Li, L. Xie, PromptStyle: controllable style transfer for text-to-speech with natural language descriptions. (2023). arXiv preprint arXiv:2305.19522
19. Z. Guo, Y. Leng, Y. Wu, S. Zhao, X. Tan, in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. PromptTTS: controllable text-to-speech with text descriptions (IEEE, 2023), pp. 1–5
20. C.H. Coker, A model of articulatory dynamics and control. Proc. IEEE **64**(4), 452–460 (1976)
21. D.H. Klatt, Software for a cascade/parallel formant synthesizer. J. Acoust. Soc. Am. **67**(3), 971–995 (1980)
22. J. Olive, in *ICASSP'77. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Rule synthesis of speech from dyadic units, vol. 2 (IEEE, 1977), p. 568–570
23. K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, K. Oura, Speech synthesis based on hidden Markov models. Proc. IEEE **101**(5), 1234–1252 (2013)
24. A. Van Den Oord, et al., Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499. (2016)
25. Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, T.Y. Liu, in *International Conference on Learning Representations*. FastSpeech 2: fast and high-quality end-to-end text to speech (ICLR, 2020)
26. Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, T.Y. Liu, FastSpeech: fast, robust and controllable text to speech. Adv. Neural Inf. Process. Syst. **32** (2019)
27. J. Betker, Better speech synthesis through scaling. (2023). arXiv preprint arXiv:2305.07243
28. J. Xue, Y. Deng, Y. Han, Y. Li, J. Sun, J. Liang, in *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. ECAPA-TDNN for multi-speaker text-to-speech synthesis (IEEE, 2022), p. 230–234
29. C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, et al., Neural codec language models are zero-shot text to speech synthesizers. (2023). arXiv preprint arXiv:2301.02111
30. J. Wang, Z. Du, Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, et al., LauraGPT: listen, attend, understand, and regenerate audio with GPT. (2023). arXiv e-prints pp. arXiv–2310
31. Y.N. Dauphin, A. Fan, M. Auli, D. Grangier, in *International conference on machine learning*. Language modeling with gated convolutional networks (PMLR, 2017), p. 933–941
32. J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization. (2016). arXiv preprint arXiv:1607.06450
33. H. Zen, V. Dang, R. Clark, Y. Zhang, R.J. Weiss, Y. Jia, Z. Chen, Y. Wu, in *Proc. Interspeech 2019*. LibriTTS: a corpus derived from LibriSpeech for text-to-speech (ISCA, 2019), p. 1526–1530
34. Y. Shi, H. Bu, X. Xu, S. Zhang, M. Li, AiShell-3: a multi-speaker Mandarin TTS corpus and the baselines. (2020). arXiv preprint arXiv:2010.11567
35. J. Kong, J. Kim, J. Bae, HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv. Neural Inf. Process. Syst. **33**, 17022–17033 (2020)
36. R. Kubichek, in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*. Mel-cepstral distance measure for objective speech quality assessment, vol. 1 (IEEE, 1993), p. 125–128
37. S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. **11**(5), 561–580 (2007)
38. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
39. Microsoft. VALLE-X online demo (2023). https://huggingface.co/spaces/Plachta/VALL-E-X. Accessed 20 May 2024
40. CoquiTTS. Coqui TTS online demo (2023). https://github.com/coqui-ai/TTS. Accessed 20 May 2024
41. OpenVoice. OpenVoice online demo (2024). https://huggingface.co/spaces/myshell-ai/OpenVoice. Accessed 20 May 2024
42. LauraTTS. LauraTTS online demo (2024). https://modelscope.cn/models/iic/speech_synthesizer-laura-en-libritts-16k-codec_nq2-pytorch/summary. Accessed 20 May 2024
43. CoquiTTS. Coqui TTS datasets (2023). https://docs.coqui.ai/en/dev/tts_datasets.html. Accessed 20 May 2024

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.