# LLFM-Voice: Emotionally Expressive Speech and Singing Voice Synthesis with Large Language Models via Flow Matching

Yanze Wang *, Xuming Han, Shuai Lv, Ting Zhou, Yali Chu

Posted Date: 22 April 2025

doi: 10.20944/preprints202504.1831.v1

Keywords: emotional speech synthesis; singing voice synthesis; large language models; flow matching; fine-grained emotional generator

 doi:10.20944/preprints202504.1831.v1



*Article*

# LLFM-Voice: Emotionally Expressive Speech and Singing Voice Synthesis with Large Language Models via Flow Matching

**Yanze Wang [1,*], Xuming Han [1], Shuai Lv [1], Ting Zhou [1] and Yali Chu [2]**

[1] School of Information Science and Technology, Jinan University, Guangzhou 510632, China
[2] School of Mathematics and Statistics, Changchun University of Technology, Changchun 130000, China
* Correspondence: hanxibird@gmail.com

**Abstract:** Although emotional speech synthesis has seen significant progress, existing methods often struggle to generate naturally fluent emotional expression and face trade-offs between emotional richness and overall speech quality. We propose a Emotionally Expressive Speech and Singing Voice Synthesis with **L**arge **L**anguage Models via **F**low **M**atching, LLFM-Voice, a unified framework that enhances emotional expressiveness in both speech and singing voice synthesis. Our method leverages the contextual modeling capabilities of large language models and incorporates emotional information through an autoregressive mechanism. To further capture musical nuances in singing, we design a fine-grained emotional generator that integrates vocal techniques, tension, and pitch for precise control of expressive singing. In addition, we introduce a flow matching-based acoustic model that models the temporal evolution of mel spectrograms instead of predicting them directly, thereby mitigating artifacts introduced by conventional spectral modeling. Experiments show that LLFM-Voice outperforms baseline systems across multiple emotional expressiveness metrics, producing speech with richer emotional content and singing voices with more natural melodic expression.

**Keywords:** emotional speech synthesis; singing voice synthesis; large language models; flow matching; fine-grained emotional generator

## 1. Introduction

Text-to-Speech (TTS), also known as speech synthesis, aims to convert input text into high-quality speech with natural prosody and emotional expressiveness. It has been widely adopted in applications such as virtual assistants, audiobooks, and smart navigation systems. Singing Voice Synthesis (SVS), built upon TTS, introduces fine-grained control over pitch, rhythm, and musical prosody, enabling the generation of stylistically diverse singing voices from lyrics and musical scores. This technology plays a crucial role in areas such as digital music production and virtual vocal performers.

From a technological perspective, speech synthesis has evolved from traditional methods such as concatenative synthesis and Hidden Markov Models (HMMs) [1–3] to advanced generative deep neural network architectures [4–8]. A seminal milestone was Google's WaveNet, which employed an autoregressive structure to directly generate speech waveforms, laying the groundwork for neural vocoders. This was followed by the Tacotron series, which adopted end-to-end sequence-to-sequence architectures and significantly improved the naturalness of synthetic speech. Microsoft's FastSpeech series addressed inference efficiency and alignment stability by leveraging non-autoregressive modeling. Baidu's DeepVoice series adopted a modular design that facilitated scalability and multi-speaker modeling, while Transformer-TTS introduced a fully parallelized architecture to enhance the robustness of long-form text synthesis. These advancements have led to substantial improvements in speech quality, timbre control, and emotional expression, forming the foundation of modern TTS and SVS systems.

 doi:10.20944/preprints202504.1831.v1



Building upon these high-fidelity synthesis frameworks, Emotional TTS has emerged as a prominent research direction. Existing approaches incorporate emotion tags (e.g., anger, joy, sadness) to enable basic emotion control [9–12]. However, significant challenges remain in dynamic emotion modeling and expressive singing voice synthesis. First, most current methods adopt static emotion representations, which fail to capture the temporal evolution of emotions and their contextual dependencies, resulting in unnatural and discontinuous emotional transitions. Second, compared to speech, emotional expression in singing is inherently more complex and influenced by multiple interacting factors, including pitch, rhythm, melody, lyrics, and singing style. Current models still struggle to achieve accurate and expressive emotional control in this domain.

To address these limitations, we propose a novel framework—**LLFM-Voice: Emotionally Expressive Speech and Singing Voice Synthesis with Large Language Models via Flow Matching**. Our approach aims to enhance the emotional naturalness of synthesized speech and singing voices while maintaining high synthesis quality. The main contributions of this work are as follows:

- **Emotion prompting via Large Language Models.** Leveraging the powerful contextual understanding and generative capabilities of large language model (LLM), the system utilizes emotional speech prompts to generate speech with dynamic emotional expressiveness, achieving smoother emotional transitions and more human-like emotional speech synthesis.
- **Fine-grained emotional generator for singing voice synthesis.** Based on music theory, we design a fine-grained emotional generator for SVS by integrating singing-specific characteristics such as vocal techniques, expressive tension, and pitch. This enables precise control over emotional intensity, emotional transitions, and variations in singing style throughout the song.
- **Flow matching-based acoustic model.** We propose a flow matching-based acoustic model that does not directly predict the Mel-spectrogram. Instead, it models the dynamic transformation from an initial distribution to the Mel-spectrogram using flow matching, aiming to improve the stability and quality of the synthesized speech.

## 2. Related Work

In the field of TTS and SVS, a unified coding and decoding architecture is generally used. As shown in Figure 1, the architecture mainly consists of three core modules: content front-end, acoustic model and vocoder. The input text is first converted into linguistic features by the content front end, then mapped to the mel spectrogram by the acoustic model, and finally the target speech is synthesized by the vocoder.

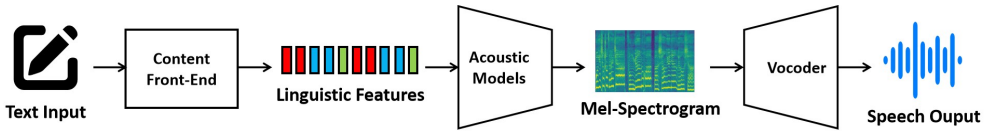



**Figure 1.** The architecture of TTS and SVS

In recent years, the development of TTS technology has primarily followed two representative pathways: models based on Variational Autoencoders[13–16] and those utilizing diffusion-based generative frameworks[17–19]. Building on these advancements, researchers have extensively explored emotion modeling and expressive SVS, aiming to improve the naturalness and emotional richness of synthesized speech.

### *2.1. Emotionally Expressive TTS*

In the domain of explicit emotion modeling, EmoTTS [20] enhances expressiveness by incorporating emotion vectors, enabling the generation of speech that reflects various emotional states such as anger, joy, and sadness. EmoQ-TTS [21] further proposes a fine-grained emotion control mechanism, allowing dynamic modulation of emotional intensity over time to produce more subtle and natural emotional expressions. MixEmotion [22] focuses on emotion blending, fusing multiple

 doi:10.20944/preprints202504.1831.v1



emotional categories during synthesis to create speech with multi-layered emotional dynamics, thereby improving its realism and variability. Additionally, EmoDiff [23] introduces diffusion-based generation to model emotions in a stepwise denoising process, enhancing both the controllability and naturalness of emotional expression, while addressing the ambiguity issues common in conventional TTS models. Despite these advancements, existing emotional TTS approaches still have notable limitations. Most systems rely on static emotion labels and fail to capture the dynamic evolution of emotions within a given context. As a result, emotional transitions in synthesized speech—especially in long-form text or dialogue scenarios—often appear abrupt and lack coherence.

### *2.2. Emotionally Expressive SVS*

Compared to speech synthesis, singing voice synthesis (SVS) demands a higher level of emotional expressiveness. Singing conveys not only emotional information but is also influenced by various musical factors such as melody, rhythm, and lyrical semantics. To address this complexity, researchers have explored a range of emotion control techniques in SVS. XiaoiceSing [24], a high-quality and emotionally rich AI singing system, models emotional trajectories with high precision, significantly enhancing the expressiveness of the generated vocals. ViSinger [25], built upon the VITS framework, optimizes duration modeling and prosody control, improving the adaptability of the system to various singing styles. DiffSinger [26], leveraging diffusion models, applies progressive denoising to improve naturalness across timbre, rhythm, and emotional depth, advancing the state of high-quality SVS. However, most existing SVS models still struggle with fine-grained emotion control. Accurately simulating emotional transitions across diverse melodies remains a challenge, and the expressive quality of synthesized singing often falls short compared to real human performances.

Unlike existing approaches, our framework eliminates the need for explicit emotion labels. Instead, the model utilizes the contextual reasoning capabilities of large language models (LLM) [27], which are guided by emotionally informative prompt speech to extract emotion cues from surrounding context. In addition, to address the complexity of emotional expression in singing, we propose a music theory-guided modeling approach. By leveraging flow matching, the system generates fine-grained, multi-dimensional expressive features for singing voice, enabling deeper understanding and more precise control of musical emotion. Finally, we design a flow matching-based acoustic model that achieves high-quality expressive speech synthesis while maintaining excellent audio fidelity.

## 3. Background of Flow Matching

In this study, we employ a flow matching approach for two purposes: generating expressive singing parameters and synthesizing mel spectrograms within the speech synthesis model. In the field of image generation, Stable Diffusion 3 [28] introduces a novel generative framework called flow matching, which models data generation via an ordinary differential equation (ODE) solver. Compared to traditional diffusion probabilistic models [8], Flow Matching eliminates the need for designing a reverse process and avoids complex mathematical formulations. Instead, it constructs a straight-line path in probability space to achieve generative capabilities similar to those of diffusion models. This approach not only offers higher generation efficiency and a simpler gradient formulation, but also demonstrates significant improvements in training speed and overall performance.

As shown in Figure 2, In this approach, the data evolves from an initial distribution $x_0 \sim p_0$ to a target distribution $x_1 \sim p_1$, where $p_0$ denotes the prior distribution and $x_1$ represents the desired data distribution. The evolution is defined over a continuous time interval $t \in [0, 1]$. Unlike feature-dependent modeling strategies, this method initializes $p_0(x)$ as a standard Gaussian distribution $N(0, 1)$, enabling the model to sample from unbiased noise and thereby reduce potential entanglement among data features.

 doi:10.20944/preprints202504.1831.v1



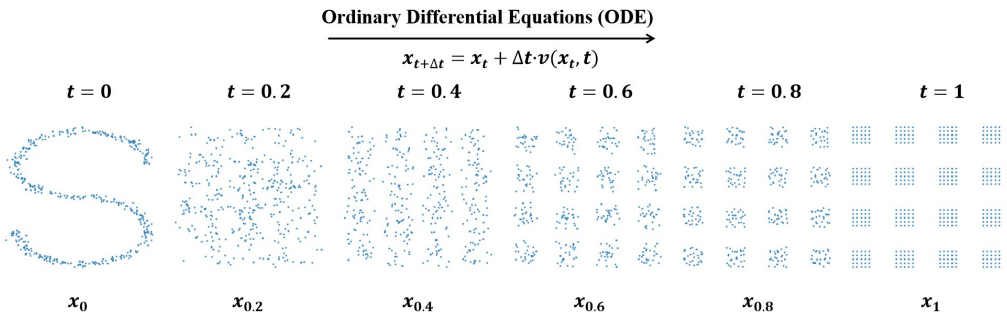



**Figure 2.** The process of flow matching.

To construct a continuous density trajectory between $p_0$ and $p_1$, a neural network is typically employed to predict a time-dependent vector field $v(x_t, t)$, which characterizes the direction and magnitude of data movement in the latent space over time. Most existing implementations adopt a U-Net architecture for this purpose. Given the predicted vector field, the evolution path can be recovered by numerically solving the associated ordinary differential equation (ODE). The forward dynamics can be discretized using the Euler method, as shown in Equation 1:

$$x_{t+\Delta t} = x_t + \Delta t \cdot v(x_t, t) \tag{1}$$

Here, $\Delta t = 1/N$ denotes the integration step size, where $N$ is the total number of sampling steps, and $x_t$ represents the system state at time $t$.

During training, the objective is to guide the neural network to learn a vector field $v_t(x)$ that closely approximates the optimal vector field $u_t(x)$, which corresponds to a linear interpolation between the initial and target distributions. To achieve this, we define the following flow-matching loss function:

$$L_{\text{FM}}(\theta) = \|v_t(x) - u_t(x)\|^2 \tag{2}$$

Here, $\theta$ denotes the parameters of the neural network, $u_t(x) = x_1 - x_0$ represents the target directional vector, and $v_t(x)$ is the estimated vector field generated by the model.

However, conventional flow matching methods typically treat all time steps uniformly during training, without accounting for variations in learning difficulty across different time intervals. In practice, when $t$ approaches 0 or 1, the optimal vector field tends to align closely with the mean of the corresponding distribution, making the prediction task relatively easier. In contrast, the intermediate time steps often involve more complex dynamics and are therefore harder to model accurately. To address this issue, we introduce a time-weighted training strategy based on a log-normal distribution, which emphasizes learning in the mid-range of the temporal domain. The corresponding weighted loss function is defined as follows:

$$L_{\text{FMLog}}(\theta) = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{t(1-t)} \exp\left(-\left(\log \frac{t}{1-t}\right)^2\right) \cdot \|v(x(t), t) - (x_1 - x_0)\|^2 \tag{3}$$

This weighting scheme assigns smaller penalties to intermediate time steps, thereby enhancing the model's ability to fit the more ambiguous regions of the trajectory. Conversely, higher weights are applied near the boundaries (e.g., as $t \to 0$ or $t \to 1$), which facilitates faster convergence toward an optimal solution.

## 4. LLFM-Voice

This section presents our proposed system LLFM-Voice, and elaborates on its core components. As shown in Figure 3, the system adopts a modular training strategy and comprises four key modules: the LLM-driven emotional content front-end, the fine-grained emotional generator,the flow matching-based acoustic model and the vocoder.

 doi:10.20944/preprints202504.1831.v1



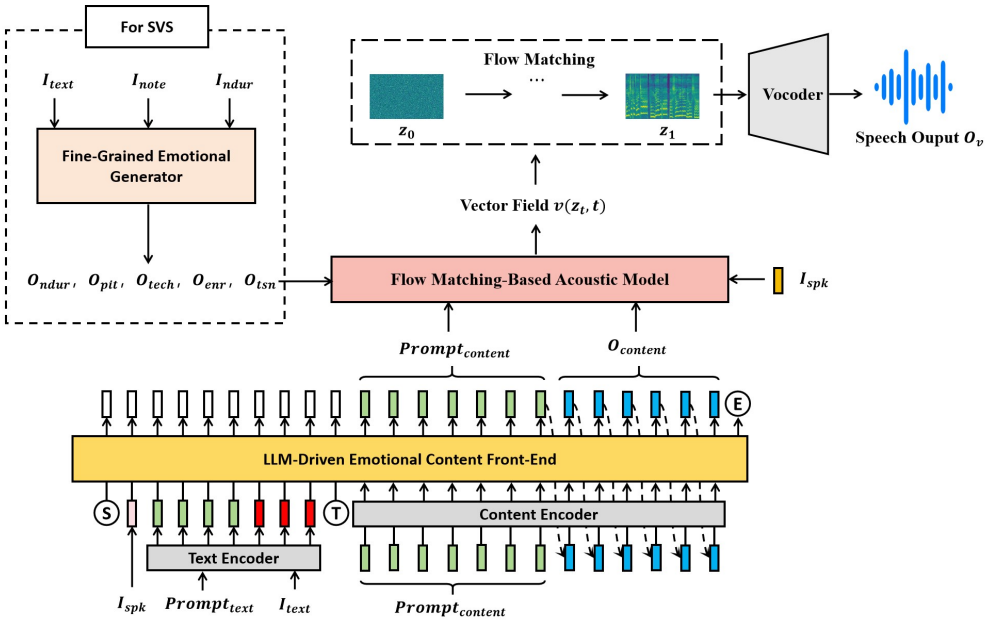



**Figure 3.** The overall framework of LLFM-Voice.

- TTS mode: Directly uses the content representation $O_{\text{content}}$ generated by the large language model front-end as input to the acoustic model, without invoking the fine-grained emotional generator.
- SVS mode: Utilizes the singing expressiveness generator to produce five-dimensional expressive parameters— phoneme duration($O_{pdur}$), pitch ($O_{pit}$), singing technique ($O_{tech}$), energy ($O_{enr}$), and tension ($O_{\text{tsn}}$). These parameters are then combined with $O_{content}$ as input to the acoustic model, enhancing emotional and stylistic expressiveness.

### *4.1. LLM-Driven Emotional Content Front-End*

This paper proposes an LLM-driven emotional content front-end, which models the content front-end as an autoregressive speech token prediction task based on LLM. Compared with traditional rule-based speech modeling methods, LLM demonstrate significant advantages in contextual understanding, emotional expressiveness, and long-range sequence modeling, owing to their powerful sequence modeling capabilities.

As shown in Figure 4, the proposed method takes the input text $I_{text}$, speaker voiceprint $I_{spk}$, and content representation $I_{content}$ as inputs, and uses an LLM to autoregressively generate the output content representation $O_{content}$ for acoustic modeling. The construction of the input sequence is the core component of speech content modeling with LLM. The LLM must establish cross-modal associations between text and speech features while maintaining semantic consistency over long temporal spans. To address this, we design the input sequence structure as follows:

$$[\text{Ⓢ}, I_{spk}, I_{text} = \{x_i\}_{i\in[1:U]}, \text{Ⓣ}, I_{content} = \{y_j\}_{j\in[1:L]}, \text{Ⓔ}] \tag{4}$$

Here, Ⓢ and Ⓔ denote the start and end of the sequence, respectively. Ⓣ is a special token used to guide the LLM in recognizing the transition point between different content representations. $I_{text}$ represents the input text sequence, $I_{spk}$ denotes the speaker voiceprint, and $I_{content}$ is the input content representation.

 doi:10.20944/preprints202504.1831.v1



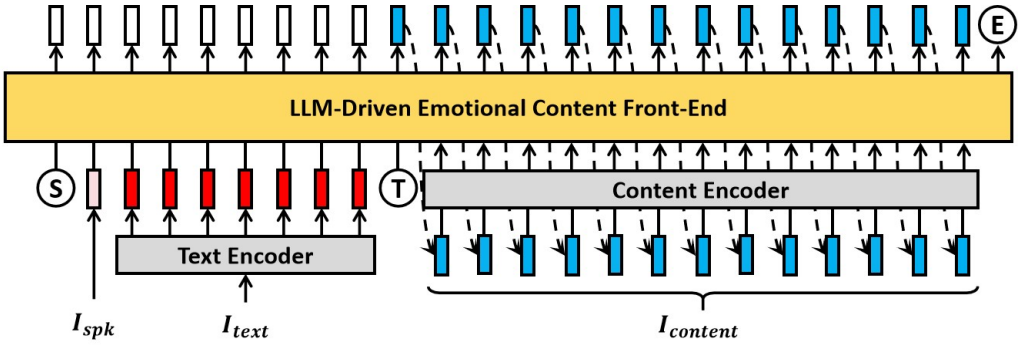



**Figure 4.** Architecture of LLM-driven autoregressive emotional content front-end

In the network architecture, the content encoder adopts the embedding layers of a Transformer [7] to learn contextual representations. The text encoder integrates a BPE tokenizer [29] with a Transformer-based architecture to enhance modeling flexibility and generalization. The autoregressive emotional content front-end consists of multiple Transformer encoder blocks, followed by a final linear layer with a Softmax activation to predict the probability distribution of the output content representation $O_{content}$.

During training, the content front-end learns to generate accurate speech content representations based on known input sequences using an autoregressive prediction strategy. A special token Ⓣ is inserted between $I_{text}$ and $I_{content}$ to signal the context-switching point, thereby assisting the LLM in aligning text and acoustic features. A masked multi-head attention mechanism is employed to ensure that, at each time step $t$, the model can only attend to current and preceding tokens, preventing information leakage and preserving causality. As a result, the loss function focuses solely on the input content representation $I_{content}$ and the predicted representation $O_{content}$, and is optimized using a cross-entropy loss.

$$L_{LLM} = -\frac{1}{L+1}\sum_{l=1}^{L+1}\log q(y_l) \tag{5}$$

Here, $L$ denotes the length of the content representation sequence, and $q(y_l)$ represents the posterior probability of $y_l$, obtained from the final Softmax layer.

During inference, leveraging the contextual learning capabilities of the LLM, we propose an emotional embedding method that does not require an additional emotion recognition model. A speech sample with emotional expression is provided as a prompt, from which both the text encoding and content representation are extracted and used as the prompt text encoding $Prompt_{text}$ and the prompt content representation $Prompt_{content}$, respectively. These components are then used to construct the input sequence as follows:

$$[Ⓢ, I_{spk}, Prompt_{text}, I_{text} = \{x_i\}_{i\in[1:U]}, Ⓣ, Prompt_{content}] \tag{6}$$

Under this mechanism, the prediction of the content representation $I_{content}$ is conditioned on the implicit emotional features embedded in the prompt content representation $Prompt_{content}$. Due to the autoregressive nature of the LLM, the generated content representation can progressively inherit the emotional characteristics of the prompt speech, enabling smooth emotional transitions and natural emotional migration.

The core advantages of this method are as follows: (1) It eliminates the need for an external emotion recognition model by allowing the LLM to internally model emotional features. (2) It adopts a prompt-based learning mechanism, enabling the LLM to infer emotionally aligned content representations from example speech prompts. (3) It supports dynamic emotional control, allowing the emotional expression of synthesized speech to be adjusted flexibly by providing different emotional speech prompts.

 doi:10.20944/preprints202504.1831.v1



The specific training and inference process is as follows:

**Algorithm 1** The traning proces of LLM-driven emotional content front-end

**Require:** Content front-end $LLM(\theta)$; Dataset $D_{\text{train}} = \{(I_{text}, I_{spk}, I_{content})\}_{m=1}^{M}$; Number of training rounds $N_{\text{iter}}$; Learning rate $\eta$;

**for** $i = 1, 2, \ldots, N_{\text{iter}}$ **do**
  From $D_{\text{train}}$ sample $(I_{text}, I_{spk}, I_{content})$
  Forward propagation:
  $O_{content} = LLM([\text{Ⓢ}, I_{spk}, I_{text} = \{x_i\}_{i\in[1:U]}, \text{Ⓣ}, I_{content} = \{y_j\}_{j\in[1:L]}, \text{Ⓔ}])$
  Calculates the loss function:
  $L_{LLM} = -\frac{1}{L+1}\sum_{l=1}^{L+1}\log q(y_l)$
  Backward propagation: Calculate the gradient $\nabla_\theta L_{LLM}$
  Take gradient descent step on $\theta \leftarrow \theta - \eta \cdot \nabla_\theta L_{LLM}$
**end for**
**return** $LLM(\theta)$

**Algorithm 2** The inference proces of LLM-driven emotional content front-end

**Require:** $LLM(\theta)$, Input text $I_{text}$, Target speaker voiceprint $I_{spk}$, Prompt text $Prompt_{text}$, Prompt content $Prompt_{content}$, Maximum sequence length $N_{max}$

Set the time step $t = 1$, Initializing the output $O_{content} = \emptyset$
**for** $t = 1, 2, \ldots, N_{max}$ **do**
  $y_t = LLM([\text{Ⓢ}, I_{spk}, Prompt_{text}, I_{text}, \text{Ⓣ}, Prompt_{content}, O_{content}])$
  Update $O_{content} = O_{content} \cup \{y_t\}$
  If $y_t = \text{Ⓔ}$, break
**end for**
**return** $O_{content}$

### *4.2. Fine-Grained Emotional Generator*

To enable more fine-grained emotional expressiveness in singing voice synthesis, we propose a fine-grained emotional generator. This module takes the input lyrics and MIDI as inputs and generates key expressive parameters for singing. It works in coordination with the content front-end and acoustic model to achieve high-quality singing voice synthesis. As illustrated in Figure 5, the generator receives the text sequence $I_{text}$, the note sequence $I_{note}$, and the note duration sequence $I_{ndur}$ as inputs, and outputs a set of multi-dimensional expressive parameters, including phoneme duration $O_{odur}$, pitch $O_{pit}$, singing technique $O_{tech}$, energy $O_{enr}$, and vocal tension $O_{tsn}$.

 doi:10.20944/preprints202504.1831.v1



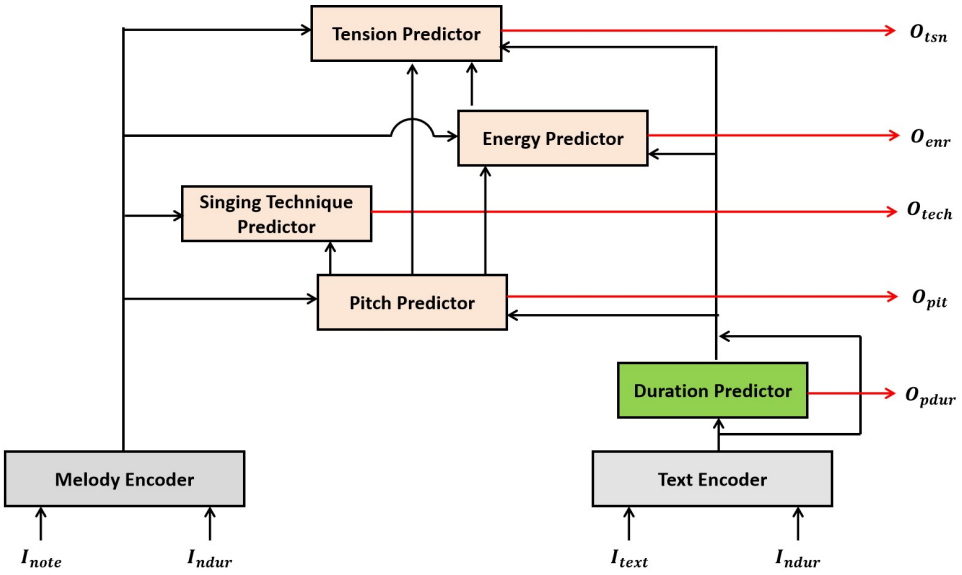



**Figure 5.** The architecture of fine-grained emotional generator

Specifically, a melody encoder is first used to extract musical features such as rhythm, style, and vocal range by processing the note sequence $I_{note}$ and note duration sequence $I_{ndur}$. Meanwhile, a text encoder processes the text sequence $I_{text}$ along with the note duration sequence $I_{ndur}$ to extract semantic information from the lyrics. Both encoders are based on Transformer architectures and produce the melody encoding $E_{midi}$ and text encoding $E_{text}$, respectively. Then based on principles of music theory, we further design a hierarchical expressiveness predictor composed of the following components:

(1) Duration Predictor: Phoneme duration $O_{pdur}$, as the most fundamental feature, is predicted directly from the text encoding $E_{text}$. A residual connection is applied to obtain the content encoding $E_{content}$ for subsequent modeling stages.

(2) Pitch Predictor: Pitch $O_{pit}$ serves as the core parameter of vocal expressiveness. It is predicted by the pitch predictor using both the melody encoding $E_{midi}$ and the content encoding $E_{content}$, and serves as the basis for modeling downstream expressive features.

(3) Singing Technique Predictor: The singing technique $O_{tech}$ is primarily influenced by pitch $O_{pit}$ and $E_{midi}$.

(4) Energy Predictor: The energy parameter $O_{enr}$ is predicted based on the combination of melody encoding $E_{midi}$, content encoding $E_{content}$, and pitch $O_{pit}$.

(5) Tension Predictor: As the most implicit expressive parameter, tension $O_{tsn}$ is modeled by integrating the melody encoding $E_{midi}$, content embedding $E_{content}$, pitch $O_{pit}$, and energy $O_{enr}$.

As a relatively simple feature, phoneme duration is predicted directly by a fully connected layer in the duration predictor. In contrast, the pitch, singing technique, energy, and tension predictors adopt a flow matching method to model vector fields, and incorporate dilated convolutional networks to enhance the modeling capacity. As illustrated in Figure 6, the input to this network includes the flow matching distribution $x_t$ at time step $t$, the text encoding $E_{text}$, the melody encoding $E_{midi}$, the time step index $t$, and relevant expressivity parameters such as pitch $O_{pit}$ and energy $O_{enr}$, depending on the specific prediction task. The time step $t$ is converted into a temporal encoding $E_t$ using sinusoidal positional encoding from the Transformer and is projected to match the channel dimension of other inputs.

 doi:10.20944/preprints202504.1831.v1



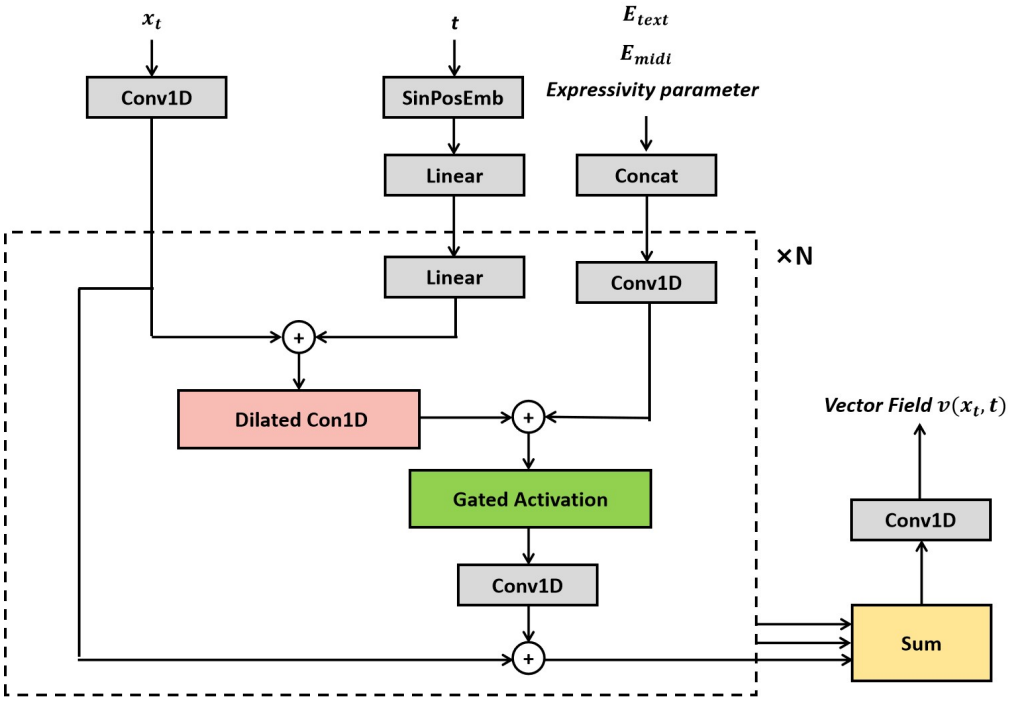



**Figure 6.** The architecture of dilated convolutional networks which is used to predict vector field $v(x_t, t)$ of flow matching

In the dilated convolutional network, dilated convolutions expand the receptive field by inserting gaps within the convolutional kernel, thereby improving the model's ability to capture long-range temporal dependencies without increasing computational cost. The mathematical formulation is as follows:

$$y[i] = \sum_{k=0}^{K-1} x[i + 2^{i*rate} \cdot k] \cdot w[k] \tag{7}$$

Here, $x[i]$ denotes the input signal, $w[k]$ represents the convolution kernel weights, $K$ is the kernel size, $i$ indicates the $i$-th layer among the $N$ dilated convolutional layers, and $rate$ refers to the predefined dilation factor.

Furthermore, at the output layer of the dilated convolution network, a gated activation unit is introduced using a Tanh-Sigmoid mechanism to enhance nonlinear feature modeling, as illustrated in the equation. This gating mechanism helps suppress gradient vanishing and improves the model's adaptability to various contextual conditions, thereby enhancing the stability of the generated expressive parameters.

$$y = \sigma(\text{gate}) \cdot \tanh(\text{filter}) \tag{8}$$

In this mechanism, the gate and filter correspond to the first and second halves of the input feature channels, respectively. The gate component $\sigma(gate)$ is computed using a Sigmoid function to generate gating coefficients, which dynamically regulate the information flow. The filter component $\tanh(filter)$ applies a Tanh activation to capture nonlinear features, ensuring that the activation values remain within the range $(-1, 1)$.

During training, each expressivity parameter is optimized individually. The overall loss function is defined as follows:

$$L_{total} = L_{pdur} + L_{FMpit} + L_{FMenr} + L_{FMtech} + L_{FMtsn} \tag{9}$$

Among them, $L_{pdur}$ denotes the mean squared error between the predicted phoneme duration $O_{dur}$ and the target phoneme duration $T_{pdur}$. The remaining four losses are flow matching losses $L_{FMlog}$, consistent with those used in the multi-dimensional perceptual flow matching acoustic model, as

 doi:10.20944/preprints202504.1831.v1



defined in Equation 3. This loss function ensures that the predicted vector field $v(x_t, t)$ aligns with the evolutionary trajectory of the target expressive parameters, thereby enhancing the naturalness and expressiveness of the generated results.

The detailed training procedure is as follows:

**Algorithm 3** The training process of the fine-grained emotional generator

**Require:** Dataset $D_{\text{train}} = \{(I_{text}, I_{note}, I_{ndur}, T_{pdur}, T_{pit}, T_{tech}, T_{enr}, T_{tsn})\}_{m=1}^{M}$; Fine-grained emotional generator $G(\theta)$; Number of training rounds $N_{\text{iter}}$; Learning rate $\eta$;

**for** $i = 1, 2, \ldots, N_{\text{iter}}$ **do**
  From $D_{\text{train}}$ sample $(I_{text}, I_{note}, I_{ndur}, T_{pdur}, T_{pit}, T_{tech}, T_{enr}, T_{tsn})$
  Randomly sample $t$ from $[0, 1]$
  Samples $x_0^{pit}, x_0^{tech}, x_0^{enr}, x_0^{tsn}$ from a random normal distribution $N(0, 1)$
  Forward propagation:

$$O_{pdur}, v(x_t^{pit}), v(x_t^{tech}), v(x_t^{enr}), v(x_t^{tsn}) = G(x_t^{pit}, x_t^{tech}, x_t^{enr}, x_t^{tsn}, t, I_{text}, I_{note}, I_{ndur})$$

  Calculates the loss function: $L_{total} = L_{pdur} + L_{FMpit} + L_{FMenr} + L_{FMtech} + L_{FMtsn}$
  Backward propagation: Calculate the gradient $\nabla_\theta L_{total}$
  Take gradient descent step on $\theta \leftarrow \theta - \eta \cdot \nabla_\theta L_{total}$
**end for**
**return** $G(\theta)$

### 4.3. Flow Matching-Based Acoustic Model

In this section, we present the proposed flow matching-based acoustic model in detail. The model employs a conditional flow matching method under optimal transport to learn the distribution of mel spectrograms and generates samples from this distribution conditioned on a set of acoustic features. As illustrated in Figure 7, we design a series of flow matching blocks to predict the vector fields required for solving the associated ordinary differential equation (ODE).

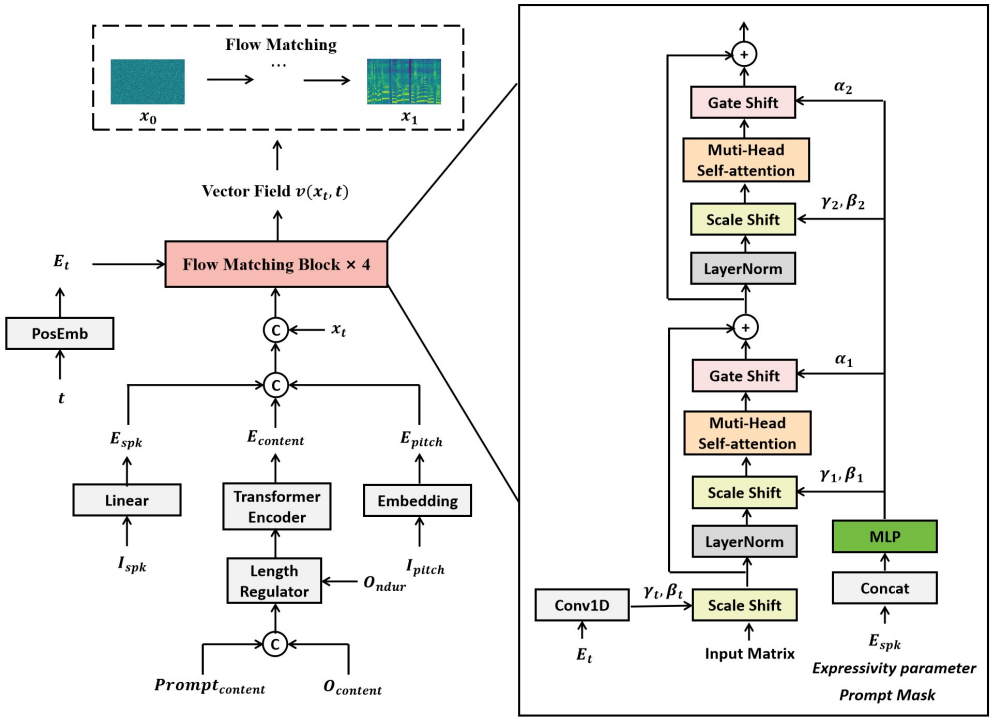



**Figure 7.** The architecture of flow matching-based acoustic model.

 doi:10.20944/preprints202504.1831.v1



Before feeding features into the proposed flow matching module, various modal inputs are first encoded into a unified representation. Specifically, a sinusoidal positional encoding mechanism based on the Transformer is used to generate temporal encodings $E_t$ for each time step. Speaker voiceprint $I_{spk}$ are linearly projected into speaker encodings $E_{spk}$. For the complex content representation $I_{content}$, a Transformer encoder is applied to capture contextual dependencies, yielding content encoding $E_{content}$. Meanwhile, the pitch sequence $I_{pitch}$ is embedded using a Transformer embedding layer to obtain pitch encoding $E_{pitch}$. Additionally, the flow state distribution at each time step $t$ is constructed using a linear interpolation formula: $x_t = x_0 \cdot t + x_1 \cdot (1 - t)$. All encoded features are then concatenated along the feature dimension and fed into the flow matching block of the acoustic model to enhance the expressiveness of vector field prediction.

Existing flow matching models typically employ U-Net [30] architectures for vector field estimation. To better adapt to the specific requirements of speech modeling, we design a hierarchical flow matching block. In this module, the input to the MLP is composed of three components: the speaker encoding $E_{spk}$, expressive parameters, and a prompt mask. The expressive parameters include four dimensions: pitch $O_{pit}$, singing technique $O_{tech}$, energy $O_{enr}$, and vocal tension $O_{tsn}$. In speech synthesis tasks, these parameters are not required and are thus input as zero matrices.The prompt mask is a mechanism proposed in this paper to incorporate the prompt content generated by the content front-end. It is constructed as a binary 0–1 vector with a length matching the content features: the first segment (equal in length to the Prompt content) is filled with zeros, and the remaining part (aligned with the output content length) is filled with ones. This mask guides the model to learn the positional relationship between the prompt and the output, enabling selective embedding of emotional and semantic information from the prompt into the generated features. Then each flow matching block employs two mechanisms for conditional information injection:

(1) Scale Shift, as shown in Equation 10, uses learnable scaling and bias parameters $(\gamma, \beta)$ to modulate the feature distribution:

$$\text{ScaleShift}(x) = x * \gamma + \beta \tag{10}$$

(2) Gate Shift, as shown in Equation 11, introduces a gated activation to dynamically regulate information flow:

$$\text{GateShift}(x) = x(1 + \alpha) + \alpha \tag{11}$$

Here, $\alpha$, $\gamma$, and $\beta$ are learnable parameters, and $x$ represents the input feature matrix.

More specifically, the temporal encoding $E_t$ is passed through a 1D convolution to obtain two learnable parameters $(\gamma_t, \beta_t)$, which are used to modulate features along the time dimension. The conditional matrix is passed through a multilayer perceptron (MLP) to produce six adaptive parameters $(\alpha_1, \gamma_1, \beta_1, \alpha_2, \gamma_2, \beta_2)$, enabling conditional feature encoding.Each flow matching block sequentially performs the following operations:1) Layer normalization and conditional feature modulation on the input; 2) Cross-dimensional interaction through a multi-head self-attention mechanism; 3) Dynamic gating via the Gate Shift mechanism; 4) Residual connections to preserve original features and improve training stability; 5) Feature integration via 1D convolution to produce the final predicted vector field $v(z_t, t)$.

The core design principle of this architecture is that the closer a block is to the output layer, the stronger its sensitivity to conditional information should be. In acoustic modeling tasks, timbre-related features are assigned greater importance, while features such as energy exert relatively weaker influence. The perceptual feature block adaptively learns to adjust the weights of different types of features, allowing the model to dynamically optimize the fusion strategy according to task-specific needs. The specific training process is as follows:

 doi:10.20944/preprints202504.1831.v1



**Algorithm 4** The training process of flow matching-based acoustic model.

**Require:** $FMAC(\theta)$; Dataset $D_{train} = \{(I_{content}, I_{spk}, I_{pit}, I_{tech}, I_{enr}, I_{tsn}, T_{mel})\}_{m=1}^{M}$; Number of training rounds $N_{iter}$; Learning rate $\eta$;

**for** $i = 1, 2, \ldots, N_{iter}$ **do**
  From $D_{train}$ sample $(I_{content}, I_{spk}, I_{pit}, I_{tech}, I_{enr}, I_{tsn}, T_{mel})$
  Randomly sample $t$ from $[0, 1]$
  Samples $x_0$ from a random normal distribution $N(0, 1)$
  Calculate $x_t = x_0 + (x_1 - x_0) * t$
  $mask\ prompt[i] = \begin{cases} 0, & i < \lfloor p \cdot L \rfloor \\ 1, & i \geq \lfloor p \cdot L \rfloor \end{cases}$, where $p \sim \mathcal{U}(0, 0.5), L = length\ of\ I_{content}$
  Forward propagation:
  $v(x_t, t) = FMAC(x_t, t, prompt\ mask, I_{content}, I_{spk}, I_{pit}, I_{tech}, I_{enr}, I_{tsn})$
  Calculates the loss function:
  $L_{FMLog} = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{t(1-t)} \exp\left(-\left(\log \frac{t}{1-t}\right)^2\right) \cdot \|v(x(t), t) - (x_1 - x_0)\|^2$
  Backward propagation: Calculate the gradient $\nabla_\theta L_{FMlog}$
  Take gradient descent step on $\theta \leftarrow \theta - \eta \cdot \nabla_\theta L_{FMlog}$
**end for**
**return** $FMAC(\theta)$

## 5. Results

To evaluate the effectiveness of the proposed method in both speech and singing voice conversion tasks, we conduct experiments on a Mandarin multi-speaker speech dataset and a Mandarin single-singer singing dataset. The two primary datasets used in this study are AISHELL-3 [31] and Opencpop [32].

**(1) AISHELL-3**

Aishell-3 is a high-quality, multi-speaker Mandarin speech synthesis dataset publicly released by the AISHELL Foundation. It comprises 218 Mandarin Chinese speakers with a balanced gender distribution, making it suitable for multi-speaker modeling and voice conversion tasks. The dataset includes over 85,000 utterances with a total duration of approximately 85 hours. Recorded at a sampling rate of 44.1 kHz, 16-bit depth, in a professional studio environment, it ensures exceptional audio clarity. To construct the speech test set, we excluded 10 randomly selected speakers from the training process and randomly sampled 10 utterances per speaker, resulting in a total of 100 speech samples.

**(2) Opencpop**

Opencpop is a dataset specifically designed for Chinese singing voice synthesis (SVS) and singing voice conversion (SVC) tasks, developed and released by the Pop Song Research Group. This dataset features the following characteristics: The recordings were captured in a professional, noise-free environment, making it highly suitable for singing research. The dataset was performed by a professional female singer to ensure clear articulation and consistent timbre. It encompasses 100 Chinese pop songs with a total duration of approximately 5.2 hours. The audio was sampled at 44.1 kHz with 16-bit PCM encoding. Word-by-word aligned lyrics annotations are provided to ensure rhythmic consistency during singing transitions. Additionally, precise pitch curves are included, facilitating accurate modeling of pitch variations. To construct the singing voice test set, we randomly selected 10 scores from the dataset that were not used in training and extracted 10 vocal segments from each score, resulting in a total of 100 singing samples.

### *5.1. Data Processing*

In this paper, we extract a variety of key information from raw speech data, including text phonemes, content representation, voiceprint information, pitch, energy and singing voice, for subsequent analysis and modeling.The content representation $I_{content}$ is extracted using a pre-trained automatic speech recognition model, SenseVoice[33], which provides high-precision linguistic fea-

 doi:10.20944/preprints202504.1831.v1



tures. Speaker identity features $I_{spk}$ are obtained using the pre-trained speaker verification model Camplus[34], which captures speaker-dependent characteristics. Pitch information $I_{pitch}$ is extracted using the pre-trained neural pitch estimation model RMVPE[35], which directly derives pitch features from raw audio, ensuring high accuracy and robustness. Energy $I_{energy}$ is calculated as the root mean square (RMS) energy of each frame in the speech signal. The singing technique are derived from the publicly available Opencpop dataset provided by SinTechSVS [36]. Text features are converted to the phoneme level using the Seq2Seq method [37]. For tension extraction, the WORLD harmonic analysis method [38] is employed to extract the full harmonic $H_{full}$ and half harmonic $H_{bass}$ in singing voices. The formula for calculating tension is defined as follows:

$$T = \log \frac{r}{l - r} \tag{12}$$

$$r = \frac{\text{RMS}(H_{full} - H_{bass})}{\text{RMS}(H_{full})} \tag{13}$$

Here, $H_{full}$ represents the full harmonic, $H_{bass}$ represents the half harmonic, and RMS denotes the root mean square of the data.

### *5.2. Model Configuration*

In the LLM-driven emotional content front-end, the input text $I_{text}$ has a vocabulary size of 51,866 and is first mapped to a 512-dimensional embedding. This embedding is then processed by a Transformer encoder consisting of 6 layers, 16 attention heads, a hidden dimension of 4096, and an output dimensionality of 1024, resulting in the text encoding $E_{text}$. The input content representation $I_{content}$ is projected from 4096 to 1024 dimensions, yielding the content encoding $E_{content}$. The input speaker embedding, originally 192-dimensional, is linearly projected to 1024 dimensions. The LLM itself adopts a Transformer encoder architecture with 14 blocks, 16 attention heads, a hidden size of 4096, and an output size of 1024. The final layer is a linear projection to a 4097-dimensional vector for predicting the one-hot encoding of the output content representation $O_{content}$.

In the fine-grained singing expressiveness generator, the input text $I_{text}$ shares the same vocabulary size of 51,866 and is embedded into 256-dimensional vectors. The note input $I_{note}$ has a vocabulary size of 128 and is also embedded into 256-dimensional vectors. The note duration input $I_{ndur}$ is projected via a linear layer into 256 dimensions. Both the melody encoder and the text encoder adopt the same Transformer encoder structure, each consisting of 4 layers, 2 attention heads, and a hidden size of 256. The duration predictor is implemented as a single fully connected layer. The pitch predictor, singing technique predictor, energy predictor, and tension predictor are all implemented using flow matching. The underlying dilated convolutional network consists of 20 dilated convolution layers. The time step $t$ and the flow matching distribution $z_t$ are first fused and projected into a 256-dimensional representation, which is then processed through the dilated convolutional layers to produce a 512-dimensional output. Expressive parameters, along with the melody encoding $E_{midi}$ and the text encoding $E_{text}$, are stacked and passed through a 1D convolutional layer to obtain a 512-dimensional hidden representation. A final convolutional layer outputs a 1-dimensional vector field $v(x_t, t)$.

In the flow matching-based acoustic model, the time encoding is first passed through a one-dimensional convolutional layer with an output channel size of $512 \times 2$ to produce two parameters, $\gamma_t$ and $\beta_t$ required for feature conditioning. The multi-head self-attention mechanism employs 4 attention heads with a hidden dimension of 400. The multilayer perceptron (MLP) consists of two linear layers, with a hidden size of 400 and an output size of $400 \times 6$, generating the conditional parameters $\alpha_1, \gamma_1, \beta_1, \alpha_2, \gamma_2, \beta_2$Finally, the entire multi-dimensional perception network outputs a 100-dimensional predicted vector field $v(x_t, t)$ through a one-dimensional convolutional layer.

 doi:10.20944/preprints202504.1831.v1



### 5.3. Training Setup

In this study, we independently trained three core modules—namely the content front-end, the fine-grained emotional generator, and the flow matching-based acoustic model—on both the speech and singing voice datasets. The Adam optimizer was employed for parameter optimization, and each model was trained for 100 epochs to ensure full convergence. A learning rate schedule with a warm-up mechanism was adopted, defined as:

$$lr = lr_i * warmup^{0.5} * min(step^{-0.5}, step * warmup^{-1.5}) \tag{14}$$

where $lr_i$ denotes the initial learning rate and $warmup$ represents the number of warm-up steps. The initial learning rates and warm-up steps were set to $(0.002, 3000)$ for the content front-end, $(0.001, 2500)$ for the acoustic model, and $(0.004, 3000)$ for the fine-grained emotional generator.

In addition, a dynamic batch size strategy was adopted, with the batch size adjusted according to the total frame length of the input content representation $I_{\text{content}}$. Specifically, the maximum number of frames was set to 20,000 for the content front-end, 10,000 for the acoustic model, and 30,000 for the fine-grained emotional generator, balancing training efficiency and memory usage.

### 5.4. Baseline Models and Evaluation Metrics

This work selects both emotional speech synthesis models and singing voice synthesis models as baselines to evaluate performance across speech and singing tasks. These baseline systems cover mainstream approaches based on variational autoencoders and diffusion models, ensuring comprehensive and scientifically rigorous comparisons.

**(1) Baseline Models for Speech Synthesis:**

1) VITS [13] (ICML 2021): A classic speech synthesis model based on the variational autoencoder (VAE) architecture. It is widely adopted in TTS tasks for its efficient feature modeling capabilities.

2) Mixed Emotion TTS [22] (IEEE Transactions on Affective Computing, 2022): A TTS model incorporating mixed emotional embeddings to improve emotional expressiveness.

3) EmoDiff [23] (ICASSP 2023): An emotional speech synthesis method based on diffusion models, enabling expressive generation through emotion-aware denoising processes.

**(2) Baseline Models for Singing Voice Synthesis:**

1) Visinger [25] (ICASSP 2022): A singing voice synthesis model based on the VITS architecture, using a variational autoencoder framework for high-quality singing waveform generation.

2) Diffsinger [26] (AAAI 2022): A diffusion-based singing synthesis model that progressively denoises latent representations to generate high-fidelity singing audio.

3) ComoSpeech [19] (Proceedings of the 31st ACM International Conference on Multimedia, 2023): A state-of-the-art diffusion-based singing synthesis approach. It preserves high-frequency details and formant features during the generation process through a carefully designed denoising schedule, resulting in more natural and perceptually clearer singing output.

**(3) Evaluation index:**

1) Mean Opinion Score (MOS): A subjective evaluation metric that measures the naturalness of synthesized speech. Scores are provided by 10 native Mandarin speakers with good pitch perception.

2) Mel Cepstral Distortion (MCD): An objective metric that calculates the spectral distance between the converted speech and the target speech. Lower values indicate better conversion quality.

3) Word Error Rate (WER): Evaluates speech intelligibility by using an automatic speech recognition (ASR) system. Lower WER values indicate better intelligibility.

4) Speaker Mean Opinion Score (SMOS): A subjective measure of speaker similarity between converted and target speech. Evaluated by the same 10 Mandarin-speaking listeners, it reflects perceived voice timbre similarity.

5) Speaker Embedding Cosine Similarity (SECS): An objective metric based on cosine similarity between speaker embeddings of original and converted speech. Higher scores indicate better speaker identity preservation.

 doi:10.20944/preprints202504.1831.v1



6) Variance Mean Opinion Score (VMOS): A subjective evaluation metric that assesses the emotional expressiveness of synthesized speech. Scores are provided by 10 listeners who are native Mandarin speakers with good pitch perception, aiming to evaluate how well the emotional content is conveyed in the synthesized speech.

7) Emotion Embedding Cosine Similarity (EECS): An objective metric based on the emo2vec [39] emotional embedding model. It computes the cosine similarity between the emotion vectors of the synthesized speech and the target speech. Higher values indicate better emotional consistency and preservation.

8) F0 Pearson Correlation (FPC): This metric calculates the Pearson correlation coefficient between the pitch contours of the target and synthesized speech, measuring the pitch retention capability of the system. It is defined as:

$$FPC = \frac{\sum(F0_{\text{target}} - \bar{F0}_{\text{target}})(F0_{\text{converted}} - \bar{F0}_{\text{converted}})}{\sqrt{\sum(F0_{\text{target}} - \bar{F0}_{\text{target}})^2} \cdot \sqrt{\sum(F0_{\text{converted}} - \bar{F0}_{\text{converted}})^2}} \tag{15}$$

where $F0_{\text{target}}$ and $F0_{\text{converted}}$ denote the pitch trajectories of the target and synthesized speech, respectively. A correlation coefficient $r = 1$ indicates perfect pitch alignment (melodic consistency), $r = 0$ indicates no correlation, and $r = -1$ indicates an inverse correlation, implying extreme pitch mismatch.

### *5.5. Experimental Results*

**(1) Speech Synthesis**

**1) Speech Quality Evaluation**

The primary objective of speech synthesis is to generate speech that is natural, clear, and expressive, achieving high-quality standards in terms of naturalness, timbre consistency, and intelligibility. To comprehensively evaluate the performance of the proposed highly expressive speech synthesis system, we conducted experiments on a speech test set and assessed the results using MOS, MCD, and WER metrics.

As shown in Table 1, the proposed LLFM-Voice significantly outperforms baseline models in speech quality, achieving the best results in both MOS (4.12) and WER (3.86%). Compared to VITS, LLFM-Voice leverages the contextual learning capability of large language models to more accurately model the relationship between text semantics and speech content, resulting in smoother and more natural synthesized speech. Additionally, the autoregressive emotional embedding employed in LLFM-Voice enables smoother emotional transitions across the speech sequence, further enhancing the naturalness of emotional expression. Compared to Mixed Emotion TTS, LLFM-Voice shows clear improvements in both MCD and WER. This demonstrates that autoregressive modeling with large language models can more precisely predict phonetic features, ensuring speech clarity and avoiding the phoneme blurring issues often caused by static emotional embeddings. Compared with EmoDiff, LLFM-Voice also yields superior results in MCD and WER. Although EmoDiff utilizes diffusion models to enhance emotional expressiveness, its denoising process may cause phoneme oversmoothing, which compromises clarity and fine-grained control of emotional variation. In contrast, LLFM-Voice, driven by a large language model-based content front-end, more effectively models prosody, rhythm, and dynamic emotional changes in speech, thereby significantly enhancing the expressiveness of the synthesized output.

 doi:10.20944/preprints202504.1831.v1



**Table 1.** Speech Quality Evaluation

| Evaluated Models | MOS(↑) | MCD(↓) | WER(↓) |
|---|---|---|---|
| GT | $4.34 \pm 0.05$ | - | 1.79% |
| VITS | $3.92 \pm 0.04$ | 5.52 | 7.48% |
| Mixed Emotion | $3.22 \pm 0.05$ | 7.21 | 26.97% |
| EmoDiff | $3.88 \pm 0.04$ | 5.94 | 13.31% |
| **LLFM-Voice (Ours)** | $\mathbf{4.12 \pm 0.05}$ | **5.12** | **3.86**% |

**2) Speech Timbre Similarity Evaluation**

To comprehensively evaluate the performance of different speech synthesis methods in terms of speaker timbre preservation, we utilize two metrics: Speaker Mean Opinion Score (SMOS) and Speaker Embedding Cosine Similarity (SECS).

As shown in Table 2 and Figure 8, LLFM-Voice demonstrates outstanding performance in both SMOS (4.03) and SECS (0.85), significantly outperforming VITS and Mixed Emotion TTS, and achieving speaker similarity scores comparable to EmoDiff. These results indicate that LLFM-Voice, through autoregressive modeling powered by a large language model, can more accurately predict speaker timbre features, producing synthetic speech that closely resembles the target speaker's voice. Compared to VITS, LLFM-Voice shows substantial improvements in both SMOS and SECS, suggesting that VITS, which relies solely on a variational autoencoder to model timbre, may suffer from timbre blurring or loss of speaker individuality. In contrast, LLFM-Voice benefits from the contextual representation learned by the large language model, which effectively enhances speaker identity consistency. Relative to Mixed Emotion TTS, LLFM-Voice achieves a significant advantage in the SECS metric. This implies that traditional methods based on static emotional embeddings have limitations in timbre control and struggle to preserve speaker-specific characteristics. LLFM-Voice, by employing an autoregressive mechanism, allows the synthesized speech to gradually inherit the target speaker's timbre features, thus improving stability and speaker fidelity. Compared to EmoDiff, LLFM-Voice yields a higher SMOS and slightly better SECS. Although EmoDiff, as a diffusion-based model, can improve speaker similarity to some extent, its denoising process may result in the loss of fine-grained timbre details, thereby affecting overall clarity. LLFM-Voice, on the other hand, leverages the semantic understanding and emotional embedding capabilities of large language models to maintain more natural and consistent timbre, while also delivering better subjective perceptual quality.

**Table 2.** Speech Timbre Similarity Evaluation

| Evaluated Models | SMOS(↑) | SECS(↑) |
|---|---|---|
| VITS | $3.58 \pm 0.04$ | 0.75 |
| Mixed Emotion | $3.23 \pm 0.06$ | 0.63 |
| EmoDiff | $3.71 \pm 0.05$ | 0.82 |
| **LLFM-Voice (Ours)** | $\mathbf{4.03 \pm 0.04}$ | **0.85** |

 doi:10.20944/preprints202504.1831.v1



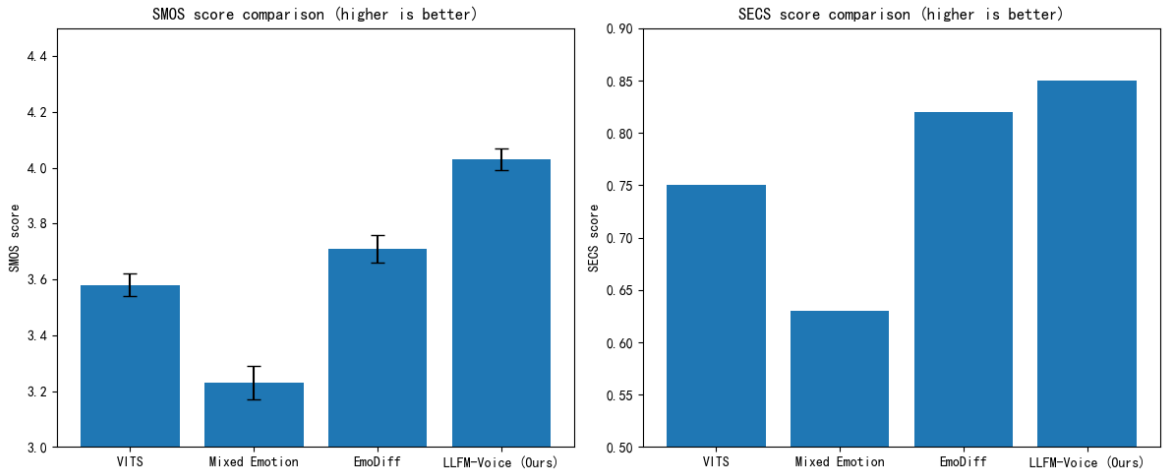



**Figure 8.** Speech Timbre Similarity Evaluation

**3) Emotional Similarity Evaluation**

To evaluate the emotional expressiveness of different speech synthesis methods, we generated and assessed 100 synthesized utterances for each of four primary emotion categories: happiness, sadness, anger, and surprise. The textual content of each utterance was carefully designed to align with the target emotion. The Emotion Embedding Cosine Similarity (EECS) metric was used to measure emotional accuracy by calculating the cosine similarity between the emotional embedding of the synthesized speech and that of the target emotion. A higher EECS value indicates better emotional alignment.

As shown in Table 3 and Figure 9, LLFM-Voice outperforms both Mix-Emotion and Diff-EMO across all emotion categories, achieving the highest EECS scores. This demonstrates that LLFM-Voice possesses more accurate emotional modeling capabilities and can generate speech that more closely reflects the intended emotional state. Compared to Mix-Emotion, LLFM-Voice shows substantial improvements in all categories. This suggests that traditional methods relying on mixed emotional embeddings face limitations in controlling emotional intensity and dynamic variation, often resulting in flat or less expressive synthesized speech. In contrast, LLFM-Voice benefits from the contextual modeling strength of large language models, enabling more precise prediction of emotional cues and producing speech with more natural intensity and expressive detail. When compared to Diff-EMO, LLFM-Voice performs better in the categories of happiness, sadness, and surprise, while slightly underperforming in the anger category. This implies that diffusion models may hold certain advantages in modeling extreme or highly intense emotions. However, the denoising process inherent to diffusion models may lead to the loss of subtle emotional details, affecting the overall stability of emotional expression. LLFM-Voice, driven by the autoregressive learning capacity of large language models, is able to capture the dynamic evolution of emotion within context, resulting in smoother and more natural emotional transitions in the synthesized speech.

**Table 3.** Emotional Similarity Evaluation

| Evaluated Models | happiness | sadness | anger | surprise |
|---|---|---|---|---|
| Mix-Emotion | 0.62 | 0.55 | 0.59 | 0.67 |
| Diff-EMO | 0.79 | 0.77 | 0.89 | 0.78 |
| **LLFM-Voice (Ours)** | **0.92** | **0.83** | **0.82** | **0.82** |

 doi:10.20944/preprints202504.1831.v1



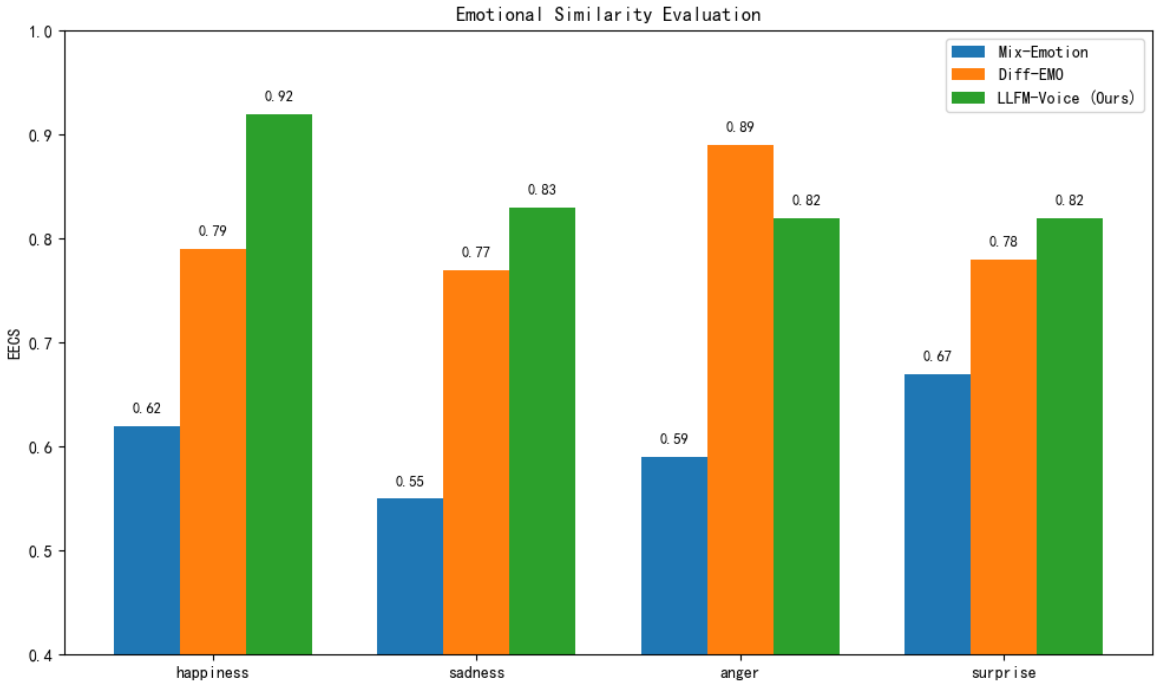



**Figure 9.** Emotional Similarity Evaluation

**(2) Singing Voice Synthesis**

**1) Singing Voice Quality Evaluation**

To comprehensively evaluate the quality of synthesized singing voices, we adopt three metrics: MOS, MCD, and WER, which respectively assess perceptual quality, timbre fidelity, and intelligibility.

As shown in Table 4, LLFM-Voice achieves the best performance across all evaluation metrics. It obtains the highest MOS (4.18), and also surpasses all baseline models in MCD (5.23) and WER (5.87%), demonstrating its ability to generate singing voices with higher audio fidelity, clarity, and naturalness, while improving overall intelligibility. Compared to Visinger, LLFM-Voice shows significant improvements on all metrics. This indicates that although the VITS-based Visinger can effectively learn the mapping between phonemes and acoustic features, it still struggles with capturing expressive nuances, maintaining timbre stability, and handling detailed control. In contrast, LLFM-Voice benefits from a large language model-driven autoregressive content front-end, which enhances semantic-acoustic alignment and results in more naturally expressive and fluent singing voices. Compared to Diffsinger, LLFM-Voice also outperforms in MOS, MCD, and WER. While diffusion models like Diffsinger improve perceptual quality through progressive denoising, they may introduce artifacts or timbre distortion, especially in melodic segments with significant pitch variation. LLFM-Voice addresses this limitation by employing hierarchical modeling of expressive features, allowing dynamic adjustment of energy, pitch, and vocal techniques in response to melodic changes, thereby improving both quality and intelligibility. When compared to ComoSpeech, LLFM-Voice achieves better MOS and MCD scores, though it shows a slightly higher WER. This suggests that ComoSpeech, with its diffusion probabilistic modeling, has advantages in modeling high-frequency details and formant structures, contributing to clearer audio quality. However, its longer iterative inference process may introduce instability, which can negatively affect lyric intelligibility. LLFM-Voice, by integrating the semantic modeling capability of large language models with fine-grained control of singing expressiveness, ensures high-quality synthesis while improving the coherence and emotional layering of the singing voice, thus offering a superior overall listening experience.

 doi:10.20944/preprints202504.1831.v1



**Table 4.** Singing Voice Quality Evaluation

| Evaluated Models | MOS (↑) | MCD (↓) | WER (↓) |
|---|---|---|---|
| GT | $4.52 \pm 0.04$ | - | 3.12% |
| Visinger | $3.62 \pm 0.05$ | 6.45 | 7.21% |
| Diffsinger | $3.89 \pm 0.04$ | 6.02 | 6.78% |
| ComoSpeech | $3.97 \pm 0.06$ | 6.13 | 6.32% |
| **LLFM-Voice (Ours)** | $\mathbf{4.18 \pm 0.05}$ | **5.23** | **5.87%** |

**2) Singing Voice Timbre Similarity Evaluation**

To evaluate the timbre consistency of different singing voice synthesis methods, we adopt two core metrics: Speaker Mean Opinion Score (SMOS) and Speaker Embedding Cosine Similarity (SECS).

As shown in Table 5 and Figure 10, LLFM-Voice achieves the best performance in both SMOS (4.13) and SECS (0.84), demonstrating superior timbre preservation capabilities compared to all baseline methods. Compared to Visinger, LLFM-Voice shows substantial improvements in both SMOS and SECS. This suggests that while Visinger, built on the VITS architecture, can capture acoustic features, it struggles with timbre stability, especially over long singing sequences, where timbre drift may occur. In contrast, LLFM-Voice leverages an autoregressive content front-end combined with fine-grained emotional control, allowing the synthesized timbre to remain consistent throughout extended singing passages and across variations in pitch and rhythm. When compared to Diffsinger, LLFM-Voice outperforms in SECS, indicating that diffusion models may suffer from timbre degradation during the denoising process. The stochastic nature of diffusion sampling may also introduce timbre variance, resulting in less stable vocal output. LLFM-Voice, on the other hand, benefits from semantically grounded modeling powered by large language models, which strengthens the alignment between text, melody, and vocal timbre, leading to improved consistency. Relative to ComoSpeech, LLFM-Voice achieves a higher SMOS score, though the improvement in SECS is marginal. This suggests that ComoSpeech, based on DDPM-style progressive denoising, has certain advantages in preserving fine-grained timbre details. However, its diffusion process may introduce nonlinear distortions, leading to slightly reduced subjective timbre consistency. By contrast, LLFM-Voice employs hierarchical modeling of expressive features, allowing the synthesized timbre to adapt more naturally to pitch and prosodic changes while better matching the target voice, thus enhancing the perceived timbre coherence.

**Table 5.** Singing Voice Timbre Similarity Evaluation

| Evaluated Models | SMOS (↑) | SECS (↑) |
|---|---|---|
| Visinger | $3.62 \pm 0.05$ | 0.73 |
| Diffsinger | $3.89 \pm 0.04$ | 0.79 |
| ComoSpeech | $3.86 \pm 0.06$ | 0.81 |
| **LLFM-Voice (Ours)** | $\mathbf{4.13 \pm 0.05}$ | **0.84** |

 doi:10.20944/preprints202504.1831.v1



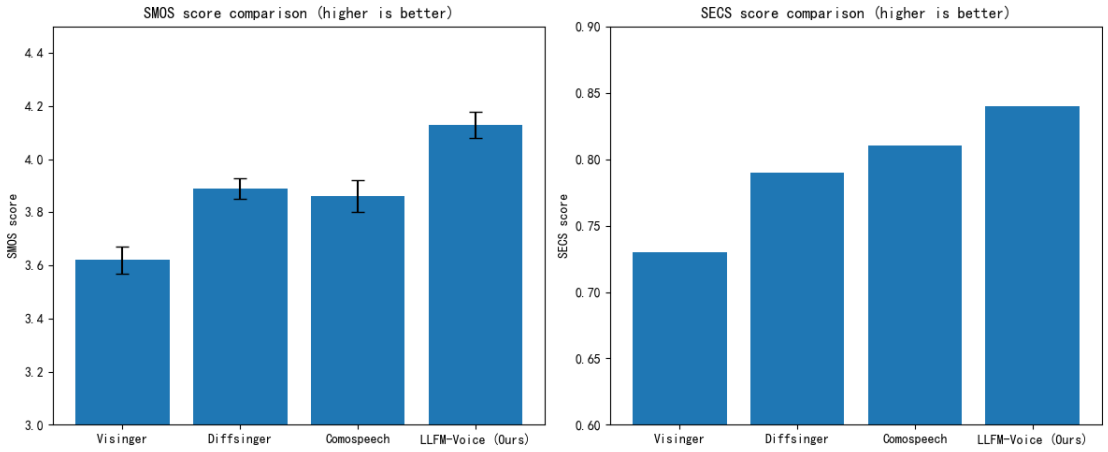



**Figure 10.** Singing Voice Timbre Similarity Evaluation

**3) Evaluation of Emotional Expressiveness in Singing Voice Synthesis**

To assess the emotional expressiveness of different singing voice synthesis methods, we adopt two key metrics: Variance Mean Opinion Score (VMOS) and F0 Pearson Correlation (FPC).

As shown in Table 6 and Figure 11, LLFM-Voice achieves the best performance on both VMOS (4.03) and FPC (0.79), demonstrating its ability to generate singing voices that are not only natural but also rich in emotional expressiveness. Compared to Visinger, LLFM-Voice shows significant improvements in both metrics. While Visinger, based on the VITS architecture, can generate clear singing audio, it exhibits limitations in expressive control, particularly when handling highly dynamic emotional transitions. In contrast, LLFM-Voice leverages an autoregressive content front-end, enabling dynamic adjustments of pitch, rhythm, and energy throughout long-form singing sequences, resulting in more accurate emotional portrayal. Relative to Diffsinger, LLFM-Voice also shows marked improvements in both VMOS and FPC. Although diffusion models can retain some emotional cues during the denoising process, their non-autoregressive architecture often leads to discontinuities in emotional progression, resulting in rigid transitions. LLFM-Voice, with its context-aware modeling capability powered by a large language model, ensures smoother emotional flow and more natural expressiveness in synthesized singing. When compared to ComoSpeech, LLFM-Voice demonstrates a more noticeable gain in VMOS, while the improvement in FPC is relatively modest. This suggests that ComoSpeech, using DDPM-based progressive denoising, excels in preserving detailed acoustic features and prosodic stability. However, its pitch variation tends to be relatively flat, limiting its ability to express dynamic emotional changes. In contrast, LLFM-Voice employs hierarchical modeling of expressive singing features, allowing finer control over pitch, vocal techniques, energy, and tension, thereby enhancing emotional depth and dynamic variation in synthesized singing.

**Table 6.** Evaluation of Emotional Expressiveness in Singing Voice Synthesis

| **Evaluated Models** | **VMOS(↑)** | **FPC(↑)** |
|---|---|---|
| GT | $4.32 \pm 0.04$ | 1.00 |
| Visinger | $3.12 \pm 0.06$ | 0.69 |
| Diffsinger | $3.25 \pm 0.05$ | 0.74 |
| ComoSpeech | $3.32 \pm 0.04$ | 0.76 |
| **LLFM-Voice (Ours)** | $\mathbf{4.03 \pm 0.03}$ | **0.79** |

 doi:10.20944/preprints202504.1831.v1



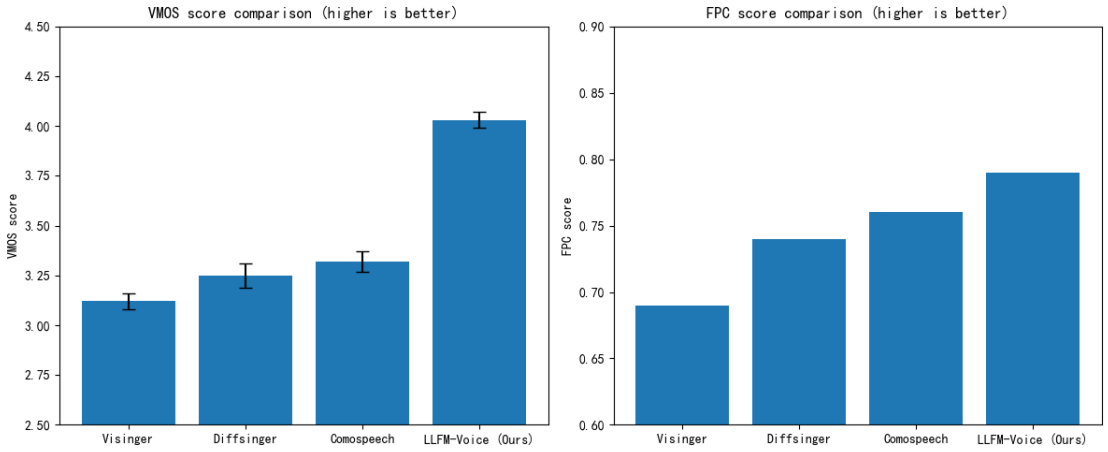
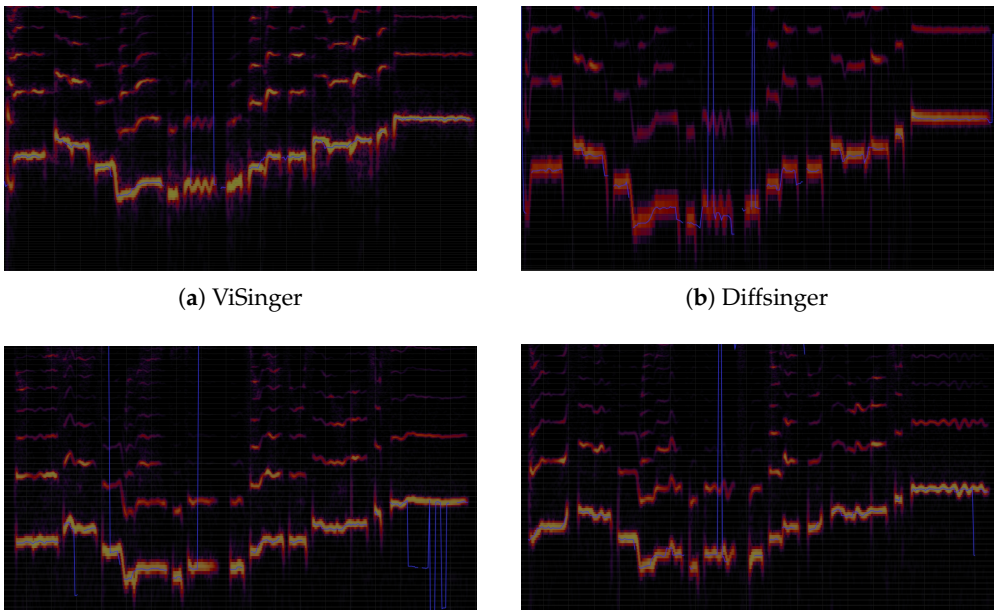



**Figure 11.** Evaluation of Emotional Expressiveness in Singing Voice Synthesis

As illustrated in Figure 12, we further analyzed the pitch contour of generated singing voices. LLFM-Voice is capable of producing vibrato with natural tremolo characteristics during sustained notes, as well as appropriate melodic ornaments such as pitch slides and turns, based on emotional semantics. Thanks to the contextual modeling of the large language model, the system achieves smooth pitch transitions between phonetic units. In contrast, Visinger and ComoSpeech, despite offering some level of pitch control, lack musical expressiveness in the form of ornamental techniques like vibrato or pitch transitions. Diffsinger can synthesize vibrato in certain segments, but its lack of strong contextual continuity often leads to abrupt transitions.



(**a**) ViSinger (**b**) Diffsinger

(**c**) Comospeech (**d**) LLFM-Voice

**Figure 12.** The pitch contour of generated singing voices

## 6. Conclusion

In this paper, we propose LLFM-Voice, a unified framework designed to enhance emotional expressiveness in both speech and singing voice synthesis. Leveraging the contextual learning capability of LLM, our method generates emotionally rich speech guided by a prompt utterance. Additionally, we introduce a fine-grained emotional generator for singing, built upon a flow matching model, which integrates multimodal features such as melody, rhythm, and prosody. This enables precise control over emotional transitions and vocal style diversity, forming a highly expressive SVS system. Furthermore,

 doi:10.20944/preprints202504.1831.v1



we propose a flow matching-based acoustic model that maintains high audio quality while producing emotionally expressive speech. The proposed LLFM-Voice framework achieves superior performance in terms of timbre consistency, emotional naturalness, and expressive richness, outperforming existing methods. Experimental results demonstrate that LLFM-Voice excels across key emotional expressiveness metrics including EECS, VMOS, and FPC, producing speech and singing voices that are both emotionally richer and melodically more natural.

## 7. Future Work

In the current implementation of the LLM-driven emotional content front-end, emotional embedding is primarily guided by acoustic and textual features from prompt speech through autoregressive modeling. However, the full potential of prompt-based reasoning in large language models remains underutilized. Future work will explore converting the prompt speech into semantic guidance cues for the LLM and incorporating natural language instructions (e.g., "generate a cheerful sentence in Chinese") to directly generate emotionally expressive speech in different dialects. This direction aims to further enhance the generalization and controllability of large language model-based speech synthesis systems.

**Author Contributions:** Y.W.; methodology, Y.W.; software, Y.W.; validation, Y.W.; formal analysis, Y.W.; investigation, Y.W.; resources, Y.W.; data curation, Y.W. and X.H.; writing—original draft preparation, Y.W., X.H ,S.L., T.Z., and Y.C.; writing—review and editing, Y.W.; visualization, X.H.; supervision, X.H.; project administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** The APC was funded by the National Social Science Foundation of China (24BTJ041).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in this study are contained in this paper. Further inquiries (including but not limited to trained models and code) can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| TTS | text to speech |
| SVS | singing voice synthesis |
| ODE | ordinary differential equation |
| MOS | Mean Opinion Score |
| MCD | Mel Cepstral Distortion |
| WER | Word Error Rate |
| SMOS | Similarity Mean Opinion Score |
| SECS | Speaker Embedding Cosine Similarity |
| EECS | Emotion Embedding Cosine Similarity |
| FPC | F0 Pearson Correlation |
| ICML | International Conference on Machine Learning |
| ICASSP | IEEE International Conference on Acoustics, Speech and Signal Processing |
| AAAI | Association for the Advancement of Artificial Intelligence Conference |

## References

1. YoshimuraÝ, T.; TokudaÝ, K.; MasukoÝÝ, T.; KobayashiÝÝ, T.; KitamuraÝ, T.. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. Eurospeech*, Location, 1999; 2347–2350.

 doi:10.20944/preprints202504.1831.v1




2. Toda, T.; Black, A.; W.; Tokuda, K.. Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model. *Speech communication* **2008**, *50*, 215–227.
3. Kayte, S.; Mundada, M.; Gujrathi, J.. Hidden Markov model based speech synthesis: A review. *International Journal of Computer Applications* **2015**, *130*, 35–39.
4. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
5. Goodfellow, I.; J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y.. Generative adversarial nets. *Advances in neural information processing systems* **2014**, *27*, .
6. Rezende, D.; J.; Mohamed, S.; Wierstra, D.. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, Location, 2014; 1278–1286.
7. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; N.; Kaiser, L.; Polosukhin, I.. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*, .
8. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S.. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, Location, 2015; 2256–2265.
9. An, X.; Wang, Y.; Yang, S.; Ma, Z.; Xie, L.. Learning hierarchical representations for expressive speaking style in end-to-end speech synthesis. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Location, 2019; 184–191.
10. Choi, H.; Park, S.; Park, J.; Hahn, M.. Multi-speaker emotional acoustic modeling for cnn-based speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Location, 2019; 6950–6954.
11. Shankar, R.; Sager, J.; Venkataraman, A.. A Multi-Speaker Emotion Morphing Model Using Highway Networks and Maximum Likelihood Objective. In *INTERSPEECH*, Location, 2019; 2848–2852.
12. Du, Z.; Sisman, B.; Zhou, K.; Li, H.. Expressive voice conversion: A joint framework for speaker identity and emotional style transfer. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Location, 2021; 594–601.
13. Kim, J.; Kong, J.; Son, J.. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, Location, 2021; 5530–5540.
14. Cong, J.; Yang, S.; Xie, L.; Su, D.. Glow-wavegan: Learning speech representations from gan-based variational auto-encoder for high fidelity flow-based speech synthesis. *arXiv* 2021, preprint arXiv:2106.10831.
15. Kong, J.; Park, J.; Kim, B.; Kim, J.; Kong, D.; Kim, S.. Vits2: Improving quality and efficiency of single-stage text-to-speech with adversarial learning and architecture design. *arXiv* 2023, preprint arXiv:2307.16430.
16. Casanova, E.; Weber, J.; Shulby, C.; D.; Junior, A.; C.; Gölge, E.; Ponti, M.; A.. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International conference on machine learning*, Location, 2022; 2709–2720.
17. Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; Catanzaro, B.. Diffwave: A versatile diffusion model for audio synthesis. *arXiv* 2020, preprint arXiv:2009.09761.
18. Popov, V.; Vovk, I.; Gogoryan, V.; Sadekova, T.; Kudinov, M.. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International conference on machine learning*, Location, 2021; 8599–8608.
19. Ye, Z.; Xue, W.; Tan, X.; Chen, J.; Liu, Q.; Guo, Y.. Comospeech: One-step speech and singing voice synthesis via consistency model. In *Proceedings of the 31st ACM International Conference on Multimedia*, Location, 2023; 1831–1839.
20. Osman, M.. Emo-tts: Parallel transformer-based text-to-speech model with emotional awareness. In *2022 5th International Conference on Computing and Informatics (ICCI)*, Location, 2022; 169–174.
21. Im, C.; Lee, S.; Kim, S.; Lee, S.. Emoq-tts: Emotion intensity quantization for fine-grained controllable emotional text-to-speech. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Location, 2022; 6317–6321.
22. Zhou, K.; Sisman, B.; Rana, R.; Schuller, B.; W.; Li, H.. Speech synthesis with mixed emotions. *IEEE Transactions on Affective Computing* **2022**, , .
23. Guo, Y.; Du, C.; Chen, X.; Yu, K.. Emodiff: Intensity controllable emotional text-to-speech with soft-label guidance. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Location, 2023; 1–5.
24. Lu, P.; Wu, J.; Luan, J.; Tan, X.; Zhou, L.. Xiaoicesing: A high-quality and integrated singing voice synthesis system. *arXiv* 2020, preprint arXiv:2006.06261.

 doi:10.20944/preprints202504.1831.v1




25. Zhang, Y.; Cong, J.; Xue, H.; Xie, L.; Zhu, P.; Bi, M.. Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Location, 2022; 7237–7241.
26. Liu, J.; Li, C.; Ren, Y.; Chen, F.; Zhao, Z.. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, Location, 2022; 11020–11028.
27. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; others, . Llama: Open and efficient foundation language models. *arXiv* 2023, preprint arXiv:2302.13971.
28. Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; others, . Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, Location, 2024; .
29. Sennrich, R.; Haddow, B.; Birch, A.. Neural machine translation of rare words with subword units. *arXiv* 2015, preprint arXiv:1508.07909.
30. Ronneberger, O.; Fischer, P.; Brox, T.. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Location, 2015; 234–241.
31. Shi, Y.; Bu, H.; Xu, X.; Zhang, S.; Li, M.. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. *arXiv* 2020, preprint arXiv:2010.11567.
32. Wang, Y.; Wang, X.; Zhu, P.; Wu, J.; Li, H.; Xue, H.; Zhang, Y.; Xie, L.; Bi, M.. Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis. *arXiv* 2022, preprint arXiv:2201.07429.
33. An, K.; Chen, Q.; Deng, C.; Du, Z.; Gao, C.; Gao, Z.; Gu, Y.; He, T.; Hu, H.; Hu, K.; others, . Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms. *arXiv* 2024, preprint arXiv:2407.04051.
34. Wang, H.; Zheng, S.; Chen, Y.; Cheng, L.; Chen, Q.. Cam++: A fast and efficient network for speaker verification using context-aware masking. *arXiv* 2023, preprint arXiv:2303.00332.
35. Wei, H.; Cao, X.; Dan, T.; Chen, Y.. RMVPE: A robust model for vocal pitch estimation in polyphonic music. *arXiv* 2023, preprint arXiv:2306.15412.
36. Zhao, J.; Chetwin, L.; Q.; H.; Wang, Y.. Sintechsvs: A singing technique controllable singing voice synthesis system. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2024**, , .
37. Sutskever, I.; Vinyals, O.; Le, Q.; V.. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* **2014**, *27*, .
38. Morise, M.; Yokomori, F.; Ozawa, K.. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems* **2016**, *99*, 1877–1884.
39. Ma, Z.; Zheng, Z.; Ye, J.; Li, J.; Gao, Z.; Zhang, S.; Chen, X.. Emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. *arXiv* 2023, preprint arXiv:2312.15185.