

KALL-E: Autoregressive Speech Synthesis with Next-Distribution Prediction

Kangxiang Xia*, Xinfa Zhu*, Jixun Yao, Wenjie Tian, Wenhao Li, Lei Xie†

Audio, Speech and Language Processing Group (ASLP@NPU), Northwestern Polytechnical University, Xi'an
 xkx@mail.nwpu.edu.cn, lxie@nwpu.edu.cn

Abstract

We introduce KALL-E, a novel autoregressive (AR) language model for text-to-speech (TTS) synthesis that operates by predicting the next distribution of continuous speech frames. Unlike existing methods, KALL-E directly models the continuous speech distribution conditioned on text, eliminating the need for any diffusion-based components. Specifically, we utilize a Flow-VAE to extract a continuous latent speech representation from waveforms, instead of relying on discrete speech tokens. A single AR Transformer is then trained to predict these continuous speech distributions from text, optimizing a Kullback–Leibler divergence loss as its objective. Experimental results demonstrate that KALL-E achieves superior speech synthesis quality and can even adapt to a target speaker from just a single sample. Importantly, KALL-E provides a more direct and effective approach for utilizing continuous speech representations in TTS.

Project — <https://github.com/xkx-hub/KALL-E>

Introduction

Large language models (LLMs), known for their impressive zero-shot and in-context learning abilities, have unified the research paradigm in natural language processing. Inspired by the success of text-based LLMs (OpenAI 2023), the next-token prediction framework has been extended to tasks in other modalities. Text-to-speech (TTS), for instance, has recently gained popularity by being framed as a next-token prediction problem (Wang et al. 2023) (Chen et al. 2024). In this approach, speech signals are first tokenized into sequences of discrete units using vector quantization (Défossez et al. 2022) (Zeghidour et al. 2022). A decoder-only language model is then trained on these acoustic tokens. However, unlike naturally discrete text data, speech discretization relies on complex quantization techniques and presents additional challenges.

Despite significant efforts to enhance discrete speech tokenizers (Zhang et al. 2024) (Li et al. 2024) (Ye et al. 2024) (Ji et al. 2024), serious issues remain. First, discrete speech tokens may lose certain information during to-

*These authors contributed equally as first authors.

†Corresponding author.

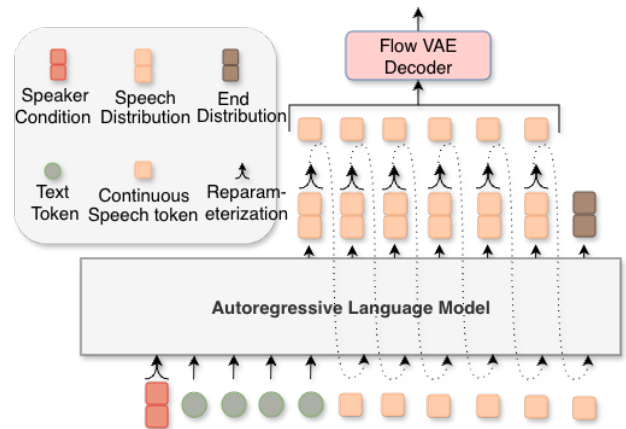


Figure 1: Overview of KALL-E. It predicts the continuous speech distribution frame by frame.

kenization (Meng et al. 2024), even if this loss is imperceptible to humans or specific models. Similar issues have been observed in some vision studies (Tschannen, Eastwood, and Mentzer 2024) (Wang et al. 2025b), where the reconstruction quality of quantized tokens typically falls short of their continuous-valued counterparts. Second, many discrete speech tokens tend to have a high frame rate, resulting in strong similarities between tokens over short periods. This can cause language models to generate long stretches of silence or continuous noise, leading to hallucinations (Song et al. 2025). Some tokenizers use multiple discrete tokens per speech frame to capture richer acoustic details (Kumar et al. 2023), but this greatly increases the training complexity of the language model. On the other hand, recently single-codebook codecs preserve less acoustic information and depend on a more powerful generation module to enhance details (Du et al. 2024a). This multistep process reduces inference efficiency and raises computational costs.

Recent studies (Meng et al. 2024) (Wang et al. 2025a) (Jia et al. 2025) have explored continuous speech representations within autoregressive language modeling frameworks to address the limitations of discrete speech tokenization. However, modeling these continuous representations introduces a range of challenges. Traditional regression-based loss func-

tions used in MELLE (Meng et al. 2024), such as mean absolute error (MAE) and mean squared error (MSE), rely on overly simplified distributional assumptions. These assumptions often fail to capture the multimodal and complex nature of training data, leading to predictions that are ambiguous, oversimplified, or overly averaged. Some studies have introduced additional processing architecture after the language model to enhance the capacity of continuous representations modeling. However, this often leads to a complex model structure. Moreover, relying on post-processing networks to predict continuous representations can undermine the language model’s overall modeling capacity.

A further limitation of existing approaches lies in the failure to fully exploit the ability of continuous representation to carry information from speech almost without loss. Continuous representation avoids information loss caused by quantization, but the frame rate of continuous representation modeled in many works is even several times that of discrete tokens. For LM-based TTS, high frame rate modeling objects will affect modeling stability and limit the maximum modeling time. On the other hand, the increase in sequence length will increase the amount of calculation for autoregressive LM geometrically, reducing the inference speed.

To solve the above problems, we propose KALL-E, a novel AR speech synthesis approach with next-distribution prediction. First, KALL-E employs Kullback-Leibler (KL) divergence loss for next-distribution prediction over traditional cross-entropy. This method models speech features in continuous space, avoiding the information loss from quantization. Second, we explore a novel representation better suited for continuous autoregressive modeling. By introducing flow into the VAE, the extracted continuous features are not constrained to a strict Gaussian prior. This yields more diverse generation results and robust modeling. To leverage the high information density of continuous representations, KALL-E operates at a low 12.5 Hz frame rate, significantly boosting inference speed. Additionally, test-time training (TTT) allows KALL-E to adapt to a target distribution from a single sample, exploiting the high information density of continuous representations to improve model capability. For our experiments, we use the Emilia dataset for primary training. We evaluate KALL-E on Seed TTS eval (Anastassiou et al. 2024), comparing it with popular TTS systems including Seed-TTS (Anastassiou et al. 2024), FireRedTTS (Guo et al. 2024), CosyVoice (Du et al. 2024a), and Llasa (Ye et al. 2025). Experimental results show KALL-E achieves impressive performance in both speed and accuracy. Specifically, it obtains a substantial reduction in word error rate (WER) compared to previous models while maintaining high naturalness. Moreover, KALL-E generates expressive, context-aware speech, effectively handling complex linguistic and emotional features.

Our contributions can be summarized as follows:

- We propose KALL-E, a novel AR speech synthesis approach with next-distribution prediction.
- We explore Flow-VAE, which not only reconstructs speech signals with high quality but also captures a richer acoustic distribution that is more suitable for autoregres-

sive LM modeling.

- Remarkably, KALL-E synthesizes speech at a low frame rate, which improves the inference speed, and achieves the lowest word error rate.
- We open-source our model and code to further advance continuous representational autoregressive modeling.

Preliminaries

Variational Autoencoder (VAE)

Unlike traditional autoencoders, VAE employs latent variables and variational inference to model data distributions explicitly. Specifically, given an observed sequential data \mathbf{X} , VAE introduces continuous latent variables $\mathbf{Z}^c = (\mathbf{z}_t^c \in \mathbb{R}^{d_z^c})_{t=1}^T$ to encode patterns, where d_z^c is the dimension of latent variables chosen as a hyperparameter. The generative model (Decoder) $p_\theta(\mathbf{X} | \mathbf{Z}^c)$ parameterized by θ defines the likelihood of observed data conditioned on latent variables, and the inference model (Encoder) $q_\phi(\mathbf{Z}^c | \mathbf{X})$ parameterized by ϕ approximates the true posterior distribution. The joint distribution of observed and latent variables is given by $p_\theta(\mathbf{X}, \mathbf{Z}^c) = p(\mathbf{Z}^c) p_\theta(\mathbf{X} | \mathbf{Z}^c)$ where the prior distribution is typically assumed as a standard Gaussian distribution $p(\mathbf{Z}^c) = \mathcal{N}(0, \mathbf{I})$. Due to the intractability of directly computing the posterior $p_\theta(\mathbf{Z}^c | \mathbf{X})$, VAE employs the evidence lower bound (ELBO) for optimization:

$$\log p_\theta(\mathbf{X}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}^c | \mathbf{x})} [\log p_\theta(\mathbf{X} | \mathbf{z}^c)] - D_{KL}(q_\phi(\mathbf{Z}^c | \mathbf{X}) || p(\mathbf{Z}^c))}_{\mathcal{O}_{ELBO}}. \quad (1)$$

Maximizing \mathcal{O}_{ELBO} effectively trains the encoder-decoder architecture, encouraging the inferred latent representation \mathbf{Z}^c to summarize the sequential data compactly.

Speech Language Model

Speech language models aim to model speech sequences through latent speech representations, commonly known as speech tokens. Following recent works, a general framework for token-based speech modeling typically consists of three components: a speech tokenizer, an autoregressive model, and a decoder. Specifically, given the raw speech input \mathbf{X} , a speech tokenizer maps \mathbf{X} into a sequence of discrete semantic tokens $\mathbf{Z}^d = (z_t^d \in \mathcal{N}_k)_{t=1}^T$, where $\mathcal{N}_k = \{1, 2, \dots, k\}$ denotes a finite vocabulary of speech units. The implicit distribution learned by the pretrained tokenizer can be represented as $p(\mathbf{Z}^d | \mathbf{X})$. Next, the autoregressive model parameterized by ψ models temporal dependencies of the discrete token sequences as

$$p_\psi(\mathbf{Z}^d) = \prod_{t=1}^T p_\psi(z_t^d | \mathbf{Z}_{1:t-1}^d). \quad (2)$$

Finally, a decoder θ reconstructs the speech \mathbf{X} from the discrete speech tokens \mathbf{Z}^d via $p_\theta(\mathbf{X} | \mathbf{Z}^d)$. However, since discrete tokens primarily capture linguistic information, this approach tends to overlook the rich paralinguistic features inherent in speech signals, and thus, discrete autoregressive models may struggle to integrate paralinguistic nuances.

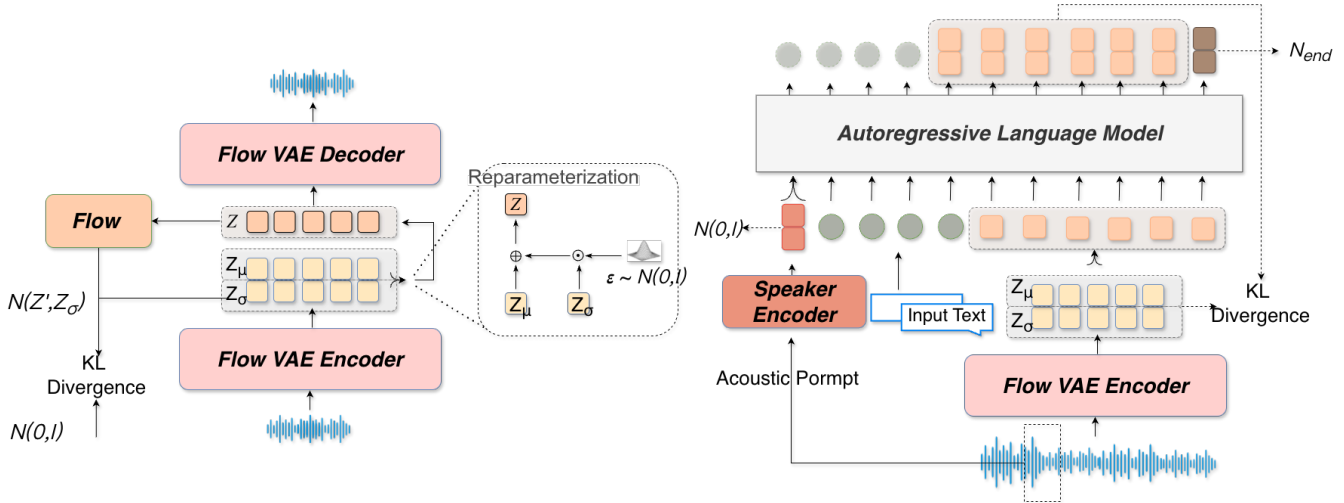


Figure 2: KALL-E Model Architecture: Left: Flow-VAE encoder and decoder for encoding and decoding continuous speech latents. Right: Autoregressive language Model with speaker encoder for text-to-speech generation.

KALL-E

KALL-E conceptualizes text-to-speech (TTS) as a conditional language-modeling task framed as next-distribution prediction. Conventional LLM-based TTS systems rely on a fixed-size token dictionary and optimize a cross-entropy objective to anticipate the next token. This formulation is token-oriented, aiming to approximate the probability distribution of the forthcoming token. In contrast, KALL-E learns to predict continuous speech distributions directly, conditioned on textual input. As illustrated in Figure 2, the model comprises two principal components: (i) a flow variational auto-encoder (Zhang et al. 2025) (Flow-VAE) and (ii) an autoregressive language model. The Flow-VAE encodes raw waveforms into a continuous latent space that captures fine-grained acoustic variation. By integrating a normalizing-flow module into the conventional VAE, we enhance the expressiveness and diversity of the latent distribution. These continuous latents provide well-structured training targets for the language model, which predicts them autoregressively from text. Finally, the Flow-VAE decoder transforms the predicted latents back into the natural-sounding speech.

Flow-VAE

To furnish the autoregressive (AR) language model with continuous speech latents, we first train a variational auto-encoder (VAE) in an unsupervised manner.

We augment the vanilla VAE with a normalising flow (Rezende and Mohamed 2015) f , which establishes a bijective transformation between a simple prior and a more complex posterior. We note that similar approaches using normalizing flows to enhance prior distributions have also been observed in (Kim, Kong, and Son 2021) (Zhang et al. 2025) for text-to-speech. Concretely, we draw $z \sim N(\mu_\phi(x), \sigma_\phi(x))$ and obtain $\tilde{z} = f(z)$. The regularisation objective minimises the Kullback–Leibler divergence

between the transformed posterior and the standard normal prior:

$$L_{kl} = D_{KL}(q_\phi(\tilde{z} | x) || p(\tilde{z})), \quad (3)$$

$$q_\phi(\tilde{z} | x) = N(f(\tilde{z}); \mu_\phi(x), \sigma_\phi(x)) \left| \det \frac{\partial f(\tilde{z})}{\partial \tilde{z}} \right|, \quad (4)$$

$$z \sim N(\mu_\phi(x), \sigma_\phi(x)), \quad (5)$$

$$p(\tilde{z}) = N(\tilde{z}, 0, I). \quad (6)$$

This regularises the encoder to produce Gaussian-distributed latents without forcing them into a unit standard normal, thereby increasing the diversity of the latent space.

As for the detailed architecture of the proposed Flow-VAE, the encoder consists of a stack of down-sampling dilated convolution layers with residual blocks, which effectively capture abstract features in the speech. After encoding, the produced mean and variance are interpreted as the parameters of the learned latent distribution $q_\phi(z | x) = N(\mu_\phi(x), \sigma_\phi(x))$. The decoder mirrors the encoder’s architecture but uses transposed convolution layers with residual blocks to up-sample the latent representation z back into the waveform \hat{x} . Additionally, we integrate advanced techniques, such as the Snake activation function from BigVGAN (Lee et al. 2023), to enhance the decoder’s performance. The Flow-VAE is optimized with a weighted sum of four losses:

$$\begin{aligned} \mathcal{L}_{\text{Flow-VAE}} = & \lambda_{kl} \mathcal{L}_{KL} + \lambda_{\text{recon}} \mathcal{L}_{\text{recon}} \\ & + \lambda_{\text{disc}} \mathcal{L}_{\text{disc}} + \lambda_{\text{fm}} \mathcal{L}_{\text{fm}}. \end{aligned} \quad (7)$$

where $\mathcal{L}_{\text{recon}}$ is an ℓ_1 mel-spectrogram loss, $\mathcal{L}_{\text{disc}}$ combines multi-period and multi-resolution discriminator losses, and \mathcal{L}_{fm} is the feature-matching loss from (Kumar et al. 2019), all balanced by λ hyperparameters.

Autoregressive Language Model

With the assistance of Flow-VAE, KALL-E employs a causal transformer decoder as the language model to autoregressively predict the continuous speech distribution. Specifically, the input text tokens are first converted into embeddings by the text embedding layer. Simultaneously, a linear layer projects the sampled speech distribution z into the dimension of the language model. Additionally, a small segment of the speech is randomly extracted and fed into the speaker encoder, which extracts a speaker embedding S and places it at the front of the sequence. The language model, consisting of blocks of multi-head self-attention and feed-forward layers, takes the concatenated speaker, text, and speech embeddings as input, modeling the dependency between semantic and acoustic information.

At each time step t , the output of the language model, the output hidden h_t , is processed by a linear layer to predict the mean μ_t and variance σ_t of the target speech distribution. These parameters are then used to sample the predicted speech distribution for the subsequent autoregressive (AR) step. This process can be formulated as:

$$p(\mu, \sigma \mid S, text, \theta) = \prod_{t=1}^T p(\mu_t, \sigma_t \mid S, text, Z_{<t}, \theta). \quad (8)$$

$$Z_t \sim N(\mu_t, \sigma_t). \quad (9)$$

where θ is the parameter of the language model.

We use the Kullback-Leibler (KL) divergence loss as the training objective for the AR language model. The KL loss has two components: one arises from the difference between the predicted and real speech distributions, and the other is related to stop prediction. Here, we define $N(1, e)$ as the stop distribution. The KL loss is computed as follows:

$$\begin{aligned} \mathcal{L}_{LM} = & \sum_{t=1}^T D_{KL}(q_\phi(z_t \mid X) \parallel p(Z_t \mid Z_{1:t-1}, text, S, \theta)) \\ & + \lambda_{end} D_{KL}(N_{end} \parallel p(e \mid Z, text, S, \theta)). \end{aligned} \quad (10)$$

where N_{end} is the pre-defined end distribution, and λ_{end} is a hyperparameter that controls its contribution.

Speaker Voice Distribution Modeling

KALL-E without an explicit speaker encoder can handle two extreme cases: (i) reference-guided voice cloning, by prepending one target utterances to the sequence (in-context learning, ICL), the model faithfully mimics that timbre; and (ii) reference-free synthesis, when no audio is supplied, the model hallucinates a plausible but essentially random timbre, which is generally impossible to reproduce later.

To simultaneously achieve high voice diversity and reproducibility, we introduce Speaker Voice Distribution Modeling. As illustrated in Figure 3, we extract a speaker embedding with ECAPA-TDNN (Desplanques, Thienpondt, and Demuynck 2020) and project it through a linear layer to obtain a latent representation. During training, a KL term regularises the latent toward an isotropic Gaussian prior.

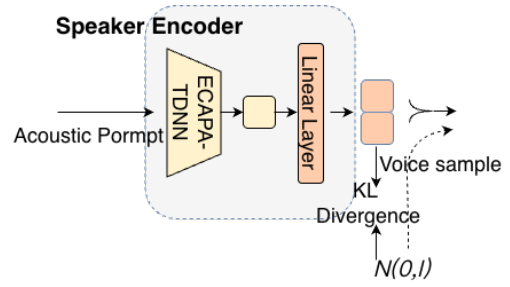


Figure 3: The Architecture of the Speaker Encoder.

During inference, when a reference utterance is available, we extract its speaker latent S and use it as a condition, yielding deterministic voice cloning. If no reference is provided, we instead draw $\tilde{S} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; because the random seed, \tilde{S} , can be stored, the resulting timbre can be reproduced exactly, thereby resolving the “lucky-voice-but-can’t-find-again” problem. Consequently, KALL-E unifies controllable voice cloning with high-diversity, yet repeatable, random voice synthesis within a single latent framework.

Test Time Training

Continuous latents preserve far more fine-grained information than discrete tokens. To exploit this capacity at inference, KALL-E adopts a lightweight test-time training (TTT) (Sun et al. 2020) procedure that adapts the language model to a new speaker from just one utterance. Given the prompt utterance S , we employ the pretrained Flow-VAE to extract the distribution of the prompt speech S . This distribution constitutes the supervision signal for TTT.

We draw N latent sequences by repeating the reparameterisation trick, and collect them into the TTT dataset:

$$z_i \sim N(\mu_\phi(S), \sigma_\phi(S)). \quad (11)$$

$$\mathcal{D}_{TTT} = \{z_1, z_2 \dots z_n\}. \quad (12)$$

During TTT, we freeze the speaker encoder and fine-tune only the language model. The end-of-sequence distribution is excluded from the loss. Given the fixed transcript, which we assume can always be obtained, and a sampled latent prefix, the model predicts the frame-level distribution. The loss is the KL divergence between predicted and target distributions, excluding the end-of-sequence frame.

This approach mitigates overfitting from single-sample adaptation, while enabling the model to better capture speaker-specific details like style, accent, and other paralinguistic features.

Experimental Setup

Dataset

We select the open-source Emilia (He et al. 2024) as the training dataset. Emilia is a multilingual, diverse, in-the-wild speech dataset designed for large-scale speech generation. In this work, we perform unsupervised training on Flow-VAE using Emilia’s speech data. For the autoregressive language model, we utilize both English and Chinese data from Emilia, totaling approximately 96.7k hours.

Model	Latent Dim	Frame Rate	STOI \uparrow	PESQ WB \uparrow	PESQ NB \uparrow	SPK-SMI \uparrow	UTMOS \uparrow
Ground Truth	-	-	1.00	4.64	4.55	1.00	4.09
DAC-12 Layer (Kumar et al. 2023)	-	50	0.95	4.01	4.15	0.95	4.00
DAC-2 Layer (Kumar et al. 2023)	-	50	0.73	1.13	1.40	0.32	1.29
Mimi (Défossez et al. 2024)	-	12.5	0.91	2.25	2.80	0.73	3.56
X-codec2 (Ye et al. 2025)	-	50	0.92	2.43	3.04	0.82	4.13
Stable Audio VAE (Evans et al. 2025)	64	12.5	0.96	3.11	3.73	0.93	4.06
	64	100	0.96	3.56	3.95	0.94	4.07
	64	12.5	0.87	2.01	2.67	0.59	3.66
Flow-VAE (Proposed)	256	12.5	0.94	2.91	3.52	0.87	3.97
	256	25	0.96	3.40	3.87	0.93	4.00
	512	12.5	0.96	3.26	3.80	0.92	3.97
	1024	12.5	0.97	3.71	4.05	0.95	3.99

Table 1: Audio reconstruction results for discrete tokenisers and continuous VAEs on test-clean subset of LibriSpeech.

Given that the pre-training data may suffer from poor audio quality and inaccurate transcripts, we select a subset with higher audio quality and perform a second round of transcription and cleaning using a different automatic speech recognition (ASR) tool than the one used in the original paper. The cleaned output, combined with our internal data, forms our second-stage fine-tuning dataset, comprising about 3,000 hours, which we also use for small-scale experiments. Additionally, we leverage the ESD (Zhou et al. 2022) dataset to train the model’s capability for emotion-aware speech generation. For evaluation, we follow the procedure of Llasa (Ye et al. 2025) and use the test-clean subset of LibriSpeech (Panayotov et al. 2015) to assess the performance of Flow-VAE. The test-clean subset contains 2,620 utterances sampled at 16 kHz. For the review of TTS capabilities, we use the test-zh and test-en subsets of the Seed-TTS (Anastassiou et al. 2024) test sets.

Implementation and Hyperparameters

For Flow-VAE, the encoder and the flow follow the architecture of Glow-WaveGAN (Cong et al. 2021), while the decoder is based on the settings of BigVGAN (Lee et al. 2023). The latent dimension of the speech distribution z is set to 512, with a frame rate of 12.5 Hz. For the autoregressive language model, we utilize the open-source LLama3.2-1B-Instruct (Touvron et al. 2023) model as our backbone. For the speaker encoder, we randomly initialize an ECAPA-TDNN (Desplanques, Thienpondt, and Demuynck 2020) module and randomly select 3-second audio segments from the target speech as input, optimizing it together with the autoregressive language model. We complete the training of Flow-VAE and the autoregressive language model on 8 NVIDIA A100 GPUs. Regarding training hyperparameters, the values of λ_{kl} , λ_{recon} , λ_{disc} , and λ_{fm} are set to 32, 1, 1, and 1, respectively. The λ_{end} is set to 0.02.

Comparison Models

We compare Flow-VAE with several open-source speech tokenizers, including DAC (Kumar et al. 2023), Mimi (Défossez et al. 2024), Xcodec-2 (Ye et al. 2025), and Stable Audio VAE (Evans et al. 2025). Except for Stable

Audio VAE, we use the officially released checkpoints. For Stable Audio VAE, we use the same training data to train a 12.5 Hz model with a dimension size of 64 for comparison. For TTS capabilities, we compare KALL-E with several popular zero-shot TTS models, including Seed-TTS (Anastassiou et al. 2024), FireRedTTS (Guo et al. 2024), CosyVoice (Du et al. 2024a), CosyVoice 2 (Du et al. 2024b), and Llasa (Ye et al. 2025). Note that the training data used by these models may differ. The model most comparable to KALL-E is Llasa-1B-160k, which has similar parameters and training data, both of which are trained from Llama3.2-1B-Instruct (Dubey et al. 2024). For inference efficiency comparison, we also evaluate it against the non-autoregressive language synthesis model F5TTS (Chen et al. 2025). In terms of context awareness, we compare it with Llasa-8B (Ye et al. 2025).

Evaluation Metrics

For Flow-VAE evaluation, we adopt the following metrics: (1) Short-Time Objective Intelligibility (STOI); (2) Perceptual Evaluation of Speech Quality (PESQ); (3) a WavLM-based speaker verification model for speaker similarity (SPK SIM) (Chen et al. 2022); and (4) UTMOS (Saeki et al. 2022). For TTS evaluation, we employ both subjective and objective evaluations. Specifically, we use the Mean Opinion Score (MOS) to assess the naturalness of synthetic speech, and the Similarity Mean Opinion Score (SMOS) to evaluate speaker similarity between synthetic and prompt speech. Each evaluation involves at least 10 participants. The rating scale is as follows: 1 (bad), 2 (poor), 3 (fair), 4 (good), and 5 (great), with half-point increments. Besides, we follow Seed-TTS (Anastassiou et al. 2024) and measure Word Error Rate (WER), Character Error Rate (CER), and speaker similarity (SIM). Moreover, to assess inference efficiency, we report giga-floating-point operations per second (GFLOPS), providing a hardware-agnostic metric that helps neutralize variations across test environments. Additionally, we use Emotion2Vec (Ma et al. 2024) to assess the emotional category of the generated speech.

Experimental Results

VAE Evaluation

We first quantify the audio reconstruction quality of Flow-VAE and then analyze how the normalizing-flow module influences the latent distribution.

Audio-reconstruction Quality. As shown in Table 1, discrete tokenizers require high frame rates to minimize information loss. When DAC is compressed from 12 to 2 layers, PESQ-WB plummets from 4.01 to 1.13 and STOI from 0.95 to 0.73. Even Mimi, stacking eight layers at 12.5 Hz, remains far below the 12.5 Hz VAE. In contrast, Flow-VAE can maintain high-quality speech signal reconstruction at low frame rates, outperforming most discrete tokenizers. Under the matching 64-dim / 12.5 Hz setting, Stable-Audio VAE outperforms Flow-VAE because its KL penalty on the latent Gaussian prior is almost zero, nearly reducing the model to a plain auto-encoder. Flow-VAE deliberately keeps high KL weight, preserving a well-shaped latent space at the cost of reconstruction, an investment that will pay off for generation. Though this gap confirms that a stronger Gaussian constraint indeed limits representational capacity, it is important to note that both the frame rate and the latent dimension affect the reconstruction performance of VAE. The reconstruction quality improves as the latent dimension and the frame rate increase, which enlarges the information capacity of latent variables. By increasing the latent dimension, we achieve a high-quality reconstruction while maintaining a low frame rate.

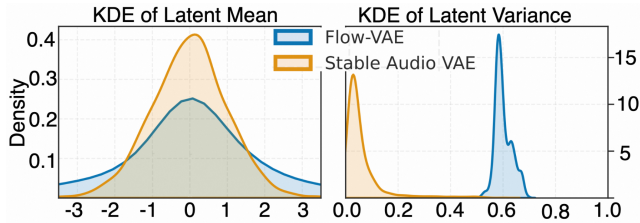


Figure 4: Kernel density estimates of latent representations.

Latent Distribution. We trained both Stable Audio VAE and Flow-VAE with identical data, frame rate, and latent dimension. We then plotted the resulting latent mean and variance distributions. The results are shown in Figure 4.

Flow-VAE’s mean distribution is broader than that of Stable Audio VAE. This suggests Flow-VAE maps audio to a wider range, aiding the model in distinguishing between frames. Additionally, Flow-VAE’s variance distribution differs significantly from Stable Audio VAE’s. Flow-VAE exhibits a larger variance, improving language modeling robustness. This increased variance allows the VAE to tolerate greater prediction deviations from the mean. Moreover, it increases the probability of overlap between distributions. Predicted points in these overlapping regions have a higher likelihood of diverse interpretations. We speculate this explains its ability to generate diverse speech.

Model	test-zh		test-en	
	CER ↓	SIM ↑	WER ↓	SIM ↑
Human	1.26	0.755	2.14	0.734
Seed-TTS	1.12	0.796	2.25	0.762
FireRedTTS	1.51	0.635	3.82	0.460
CosyVoice	3.63	0.723	4.29	0.609
CosyVoice 2	1.45	0.748	2.57	0.652
Llasa-1B-160k	2.22	0.658	3.60	0.563
KALL-E	0.96	0.646	1.94	0.568
KALL-E (TTT)	1.02	0.698	1.90	0.611

Table 2: Recognition error rates and speaker similarity on the Chinese (*test-zh*) and English (*test-en*) sets. Lower CER/WER and higher SIM indicate better performance.

Model	MOS ↑	SMOS ↑
KALL-E	4.17 ± 0.08	3.93 ± 0.15
Llasa 1B	3.92 ± 0.12	3.85 ± 0.08
Llasa 8B	3.97 ± 0.10	3.92 ± 0.11
CosyVoice 2	4.03 ± 0.16	3.92 ± 0.08
F5TTS	3.93 ± 0.07	3.87 ± 0.12

Table 3: Subjective Evaluation on Seed-TTS test sets: Naturalness (MOS) and Speaker Similarity (SMOS).

TTS Evaluation

We compare KALL-E with state-of-the-art LM-based text-to-speech (TTS) systems on the Seed-TTS test set (Anastassiou et al. 2024). We trained KALL-E using a 12.5 Hz, 512-dim Flow-VAE.

Voice Cloning. Table 2 reports the results. Despite being trained with the least data, KALL-E achieves the best decoding accuracy on both languages: 0.96 CER on test-zh and 1.94 WER on test-en. KALL-E (TTT), incorporating test-time training, improves SIM performance while maintaining comparable CER and WER to the base model. Notably, some discrete-token systems report SIM scores higher than KALL-E. Since these systems condition on reference audio when decoding tokens into waveforms, we hypothesize this extra conditioning inflates objective similarity scores and may influence perceived naturalness. We investigate this hypothesis in the subsequent subjective test.

We conducted a subjective evaluation on the Seed-TTS test sets to assess naturalness and speaker similarity. Fifteen native listeners participated in the evaluation via a web interface. Table 3 reports the mean MOS for naturalness and SMOS for speaker similarity. Listeners rated KALL-E highest in naturalness, significantly outperforming Llasa-1B and matching the much larger Llasa-8B. SMOS differences among the top systems were within one interval width, indicating comparable voice fidelity.

Inference Efficiency. To quantify generation efficiency, we compute the floating-point operations (GFLOPs) re-

Model	Parameter	Frame Rate	Flops (Gfpos)↓
KALL-E	1B	12.5	7947.48
Llasa 1B	1B	50	122 170.02
CosyVoice 2	500M	25	12 095.08
F5TTS	300M	93.75	11 546.98

Table 4: Comparison of 10s audio inference speeds for different models. In order to eliminate the impact of the test environment on the final results, we calculated the number of floating-point operations of different models.

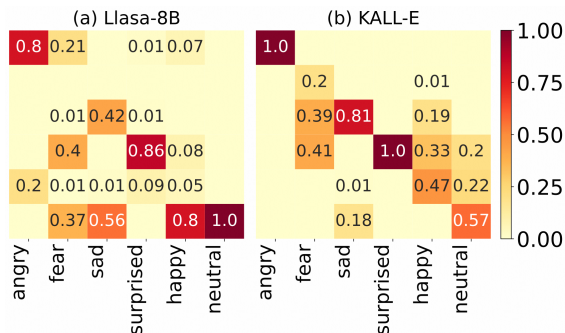


Figure 5: Heatmap of emotion recognition confusion matrices for synthetic speech (LLaSA-8B vs. KALL-E).

quired to synthesise 10s of speech (Table 4). A lower codec frame rate (12.5 Hz) offsets KALL-E’s large parameter count, making it faster than all baselines and showing the high inference efficiency.

Context Awareness. To further assess the context awareness of KALL-E, we leverage ChatGPT to create emotional texts and then synthesize speech only using these texts as input. Finally, we use Emotion2Vev to recognize the emotion and draw the confusion matrix. As shown in Figure 5, compared to Llasa-8B, KALL-E can infer the more appropriate emotion solely from the input text without any emotional speech prompt. We attribute this ability to two aspects. On the one hand, we started training from the text-pretrained LM, and the model itself has a certain understanding of semantics. On the other hand, most of the paralinguistic information, such as emotions, is retained in the continuous representation, which is conducive to the model learning the mapping from text to emotions.

Ablation Study

Flow-VAE Ablation This section evaluates how different Flow-VAE variants impact LM modeling. We also compare Flow-VAE with a same-dimension Stable Audio VAE to assess the flow module’s effect on LM modeling. All experiments are conducted on a small-scale dataset.

Table 5 reveals that reducing the latent dimension affects SIM metrics more than WER. This suggests the VAE prioritizes discarding detailed acoustic features during encoding. Furthermore, replacing Flow-VAE with Stable Audio VAE significantly degrades model performance. This indicates our Flow-VAE is better suited for LM modeling.

Model	Latent Dim	Frame Rate	CER ↓	SIM ↑
Flow-VAE	64	12.5	6.15	0.435
Flow-VAE	256	12.5	4.91	0.438
Flow-VAE	256	25	7.30	0.401
Flow-VAE	512	12.5	2.79	0.495
Flow-VAE	1024	12.5	19.69	0.357
Stable Audio VAE	64	12.5	40.09	0.117

Table 5: Comparison of Flow-VAE and Stable Audio VAE Encoder Configurations: Impact on Language Model Character Error Rate and Similarity

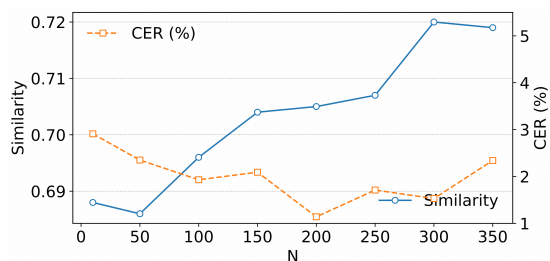


Figure 6: Effect of TTT Training Set Size (N) on speaker similarity and character error rate.

Test Time Training (TTT) Ablation In this section, we verify the ultimate improvement in model performance achieved by TTT. To enhance the comparison, we primarily selected test items from the seed Chinese test set with a non-zero CER index, combined with a few randomly selected items, for a total of 200 items as the test set. During adaptation, we kept the learning rate fixed at 1×10^{-6} and varied the size of the TTT training subset N . Figure 6 plots the resulting speaker similarity (left axis) and CER (right axis). Similarity increases almost monotonically with N , suggesting that most speaker-specific information can be captured with TTT. CER falls sharply around $N = 200$, after which it climbs again. We attribute the rebound to mild over-fitting: the model begins to mimic disfluencies or idiosyncratic pronunciations present in the reference. These findings indicate that TTT is sufficient to enhance both perceived identity and intelligibility.

Conclusion

In this paper, we propose KALL-E, a novel model that predicts speech distribution for speech synthesis. By integrating the flow module into the traditional VAE architecture, KALL-E introduces a continuous representation extracted via Flow-VAE, which is better suited for autoregressive modeling. This continuous representation allows KALL-E to achieve efficient and fast speech synthesis at a low frame rate. Additionally, KALL-E demonstrates strong context awareness, enabling the model to infer emotions from text without any additional emotional speech prompts. The incorporation of TTT further enhances the model’s performance. Experimental results demonstrate the effectiveness of the proposed model and its ability to generate natural, expressive, and context-aware speech.

References

- Anastassiou, P.; Chen, J.; Chen, J.; Chen, Y.; Chen, Z.; Chen, Z.; Cong, J.; Deng, L.; Ding, C.; Gao, L.; Gong, M.; Huang, P.; Huang, Q.; Huang, Z.; Huo, Y.; Jia, D.; Li, C.; Li, F.; Li, H.; Li, J.; Li, X.; Li, X.; Liu, L.; Liu, S.; Liu, S.; Liu, X.; Liu, Y.; Liu, Z.; Lu, L.; Pan, J.; Wang, X.; Wang, Y.; Wang, Y.; Wei, Z.; Wu, J.; Yao, C.; Yang, Y.; Yi, Y.; Zhang, J.; Zhang, Q.; Zhang, S.; Zhang, W.; Zhang, Y.; Zhao, Z.; Zhong, D.; and Zhuang, X. 2024. Seed-TTS: A Family of High-Quality Versatile Speech Generation Models. *CoRR*, abs/2406.02430.
- Chen, S.; Liu, S.; Zhou, L.; Liu, Y.; Tan, X.; Li, J.; Zhao, S.; Qian, Y.; and Wei, F. 2024. VALL-E 2: Neural Codec Language Models are Human Parity Zero-Shot Text to Speech Synthesizers. *CoRR*, abs/2406.05370.
- Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; Wu, J.; Zhou, L.; Ren, S.; Qian, Y.; Qian, Y.; Wu, J.; Zeng, M.; Yu, X.; and Wei, F. 2022. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE J. Sel. Top. Signal Process.*, 16(6): 1505–1518.
- Chen, Y.; Niu, Z.; Ma, Z.; Deng, K.; Wang, C.; Zhao, J.; Yu, K.; and Chen, X. 2025. F5-TTS: A Fairytaler that Fakes Fluent and Faithful Speech with Flow Matching. In *ACL (1)*, 6255–6271. Association for Computational Linguistics.
- Cong, J.; Yang, S.; Xie, L.; and Su, D. 2021. GlowWaveGAN: Learning Speech Representations from GAN-Based Variational Auto-Encoder for High Fidelity Flow-Based Speech Synthesis. In Hermansky, H.; Cernocký, H.; Burget, L.; Lamel, L.; Scharenborg, O.; and Motlíček, P., eds., *22nd Annual Conference of the International Speech Communication Association, Interspeech 2021, Brno, Czechia, August 30 - September 3, 2021*, 2182–2186. ISCA.
- Défosses, A.; Copet, J.; Synnaeve, G.; and Adi, Y. 2022. High Fidelity Neural Audio Compression. *CoRR*, abs/2210.13438.
- Défosses, A.; Mazaré, L.; Orsini, M.; Royer, A.; Pérez, P.; Jégou, H.; Grave, E.; and Zeghidour, N. 2024. Moshi: a speech-text foundation model for real-time dialogue. *CoRR*, abs/2410.00037.
- Desplanques, B.; Thienpondt, J.; and Demuyne, K. 2020. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *INTERSPEECH*, 3830–3834. ISCA.
- Du, Z.; Chen, Q.; Zhang, S.; Hu, K.; Lu, H.; Yang, Y.; Hu, H.; Zheng, S.; Gu, Y.; Ma, Z.; Gao, Z.; and Yan, Z. 2024a. CosyVoice: A Scalable Multilingual Zero-shot Text-to-speech Synthesizer based on Supervised Semantic Tokens. *CoRR*, abs/2407.05407.
- Du, Z.; Wang, Y.; Chen, Q.; Shi, X.; Lv, X.; Zhao, T.; Gao, Z.; Yang, Y.; Gao, C.; Wang, H.; Yu, F.; Liu, H.; Sheng, Z.; Gu, Y.; Deng, C.; Wang, W.; Zhang, S.; Yan, Z.; and Zhou, J. 2024b. CosyVoice 2: Scalable Streaming Speech Synthesis with Large Language Models. *CoRR*, abs/2412.10117.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; Goyal, A.; Hartshorn, A.; Yang, A.; Mitra, A.; Sra-vankumar, A.; Korenev, A.; Hinsvark, A.; Rao, A.; Zhang, A.; Rodriguez, A.; Gregerson, A.; Spataru, A.; Rozière, B.; Biron, B.; Tang, B.; Chern, B.; Caucheteux, C.; Nayak, C.; Bi, C.; Marra, C.; McConnell, C.; Keller, C.; Touret, C.; Wu, C.; Wong, C.; Ferrer, C. C.; Nikolaidis, C.; Allonsius, D.; Song, D.; Pintz, D.; Livshits, D.; Esiobu, D.; Choudhary, D.; Mahajan, D.; Garcia-Olano, D.; Perino, D.; Hupkes, D.; Lakomkin, E.; AlBadawy, E.; Lobanova, E.; Dinan, E.; Smith, E. M.; Radenovic, F.; Zhang, F.; Synnaeve, G.; Lee, G.; Anderson, G. L.; Nail, G.; Mialon, G.; Pang, G.; Cucurell, G.; Nguyen, H.; Korevaar, H.; Xu, H.; Touvron, H.; Zarov, I.; Ibarra, I. A.; Kloumann, I. M.; Misra, I.; Evtimov, I.; Copet, J.; Lee, J.; Geffert, J.; Vranes, J.; Park, J.; Mahadeokar, J.; Shah, J.; van der Linde, J.; Billock, J.; Hong, J.; Lee, J.; Fu, J.; Chi, J.; Huang, J.; Liu, J.; Wang, J.; Yu, J.; Bitton, J.; Spisak, J.; Park, J.; Rocca, J.; Johnstun, J.; Saxe, J.; Jia, J.; Alwala, K. V.; Upasani, K.; Plawiak, K.; Li, K.; Heafield, K.; Stone, K.; and et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.
- Evans, Z.; Parker, J. D.; Carr, C.; Zukowski, Z.; Taylor, J.; and Pons, J. 2025. Stable Audio Open. In *ICASSP*, 1–5. IEEE.
- Guo, H.; Liu, K.; Shen, F.; Wu, Y.; Xie, F.; Xie, K.; and Xu, K. 2024. FireRedTTS: A Foundation Text-To-Speech Framework for Industry-Level Generative Speech Applications. *CoRR*, abs/2409.03283.
- He, H.; Shang, Z.; Wang, C.; Li, X.; Gu, Y.; Hua, H.; Liu, L.; Yang, C.; Li, J.; Shi, P.; Wang, Y.; Chen, K.; Zhang, P.; and Wu, Z. 2024. Emilia: An Extensive, Multilingual, and Diverse Speech Dataset for Large-Scale Speech Generation. *CoRR*, abs/2407.05361.
- Ji, S.; Jiang, Z.; Cheng, X.; Chen, Y.; Fang, M.; Zuo, J.; Yang, Q.; Li, R.; Zhang, Z.; Yang, X.; Huang, R.; Jiang, Y.; Chen, Q.; Zheng, S.; Wang, W.; and Zhao, Z. 2024. WavTokenizer: an Efficient Acoustic Discrete Codec Tokenizer for Audio Language Modeling. *CoRR*, abs/2408.16532.
- Jia, D.; Chen, Z.; Chen, J.; Du, C.; Wu, J.; Cong, J.; Zhuang, X.; Li, C.; Wei, Z.; Wang, Y.; and Wang, Y. 2025. DiTAR: Diffusion Transformer Autoregressive Modeling for Speech Generation. *CoRR*, abs/2502.03930.
- Kim, J.; Kong, J.; and Son, J. 2021. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, 5530–5540. PMLR.
- Kumar, R.; Seetharaman, P.; Luebs, A.; Kumar, I.; and Kumar, K. 2023. High-Fidelity Audio Compression with Improved RVQGAN. In *NeurIPS*.
- Lee, S.; Ping, W.; Ginsburg, B.; Catanzaro, B.; and Yoon, S. 2023. BigVGAN: A Universal Neural Vocoder with Large-Scale Training. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

- Li, H.; Xue, L.; Guo, H.; Zhu, X.; Lv, Y.; Xie, L.; Chen, Y.; Yin, H.; and Li, Z. 2024. Single-Codec: Single-Codebook Speech Codec towards High-Performance Speech Generation. In *Interspeech 2024*, 3390–3394.
- Ma, Z.; Zheng, Z.; Ye, J.; Li, J.; Gao, Z.; Zhang, S.; and Chen, X. 2024. emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. In *ACL (Findings)*, 15747–15760. Association for Computational Linguistics.
- Meng, L.; Zhou, L.; Liu, S.; Chen, S.; Han, B.; Hu, S.; Liu, Y.; Li, J.; Zhao, S.; Wu, X.; Meng, H.; and Wei, F. 2024. Autoregressive Speech Synthesis without Vector Quantization. *CoRR*, abs/2407.08551.
- OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.
- Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 5206–5210. IEEE.
- Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, 1530–1538. JMLR.org.
- Saeki, T.; Xin, D.; Nakata, W.; Koriyama, T.; Takamichi, S.; and Saruwatari, H. 2022. UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022. In *INTERSPEECH*, 4521–4525. ISCA.
- Song, Y.; Chen, Z.; Wang, X.; Ma, Z.; and Chen, X. 2025. ELLA-V: Stable Neural Codec Language Modeling with Alignment-Guided Sequence Reordering. In *AAAI*, 25174–25182. AAAI Press.
- Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A. A.; and Hardt, M. 2020. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, 9229–9248. PMLR.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.
- Tschannen, M.; Eastwood, C.; and Mentzer, F. 2024. GIVT: Generative Infinite-Vocabulary Transformers. In *ECCV (57)*, volume 15115 of *Lecture Notes in Computer Science*, 292–309. Springer.
- Wang, C.; Chen, S.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; He, L.; Zhao, S.; and Wei, F. 2023. Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers. *CoRR*, abs/2301.02111.
- Wang, H.; Liu, S.; Meng, L.; Li, J.; Yang, Y.; Zhao, S.; Sun, H.; Liu, Y.; Sun, H.; Zhou, J.; Lu, Y.; and Qin, Y. 2025a. FELLE: Autoregressive Speech Synthesis with Token-Wise Coarse-to-Fine Flow Matching. *CoRR*, abs/2502.11128.
- Wang, Y.; Lin, Z.; Teng, Y.; Zhu, Y.; Ren, S.; Feng, J.; and Liu, X. 2025b. Bridging Continuous and Discrete Tokens for Autoregressive Visual Generation. *CoRR*, abs/2503.16430.
- Ye, Z.; Sun, P.; Lei, J.; Lin, H.; Tan, X.; Dai, Z.; Kong, Q.; Chen, J.; Pan, J.; Liu, Q.; Guo, Y.; and Xue, W. 2024. Codec Does Matter: Exploring the Semantic Shortcoming of Codec for Audio Language Model. *CoRR*, abs/2408.17175.
- Ye, Z.; Zhu, X.; Chan, C.; Wang, X.; Tan, X.; Lei, J.; Peng, Y.; Liu, H.; Jin, Y.; Dai, Z.; Lin, H.; Chen, J.; Du, X.; Xue, L.; Chen, Y.; Li, Z.; Xie, L.; Kong, Q.; Guo, Y.; and Xue, W. 2025. Llasa: Scaling Train-Time and Inference-Time Compute for Llama-based Speech Synthesis. *CoRR*, abs/2502.04128.
- Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; and Tagliasacchi, M. 2022. SoundStream: An End-to-End Neural Audio Codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30: 495–507.
- Zhang, B.; Guo, C.; Yang, G.; Yu, H.; Zhang, H.; Lei, H.; Mai, J.; Yan, J.; Yang, K.; Yang, M.; Huang, P.; Jin, R.; Jiang, S.; Cheng, W.; Li, Y.; Xiao, Y.; Zhou, Y.; Zhang, Y.; Lu, Y.; and He, Y. 2025. MiniMax-Speech: Intrinsic Zero-Shot Text-to-Speech with a Learnable Speaker Encoder. *CoRR*, abs/2505.07916.
- Zhang, X.; Zhang, D.; Li, S.; Zhou, Y.; and Qiu, X. 2024. SpeechTokenizer: Unified Speech Tokenizer for Speech Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Zhou, K.; Sisman, B.; Liu, R.; and Li, H. 2022. Emotional voice conversion: Theory, databases and ESD. *Speech Commun.*, 137: 1–18.