# StreamVoice: Streamable Context-Aware Language Modeling for Real-time Zero-Shot Voice Conversion

**Zhichao Wang**[1], **Yuanzhe Chen**[2], **Xinsheng Wang**[1], **Lei Xie**[1*], **Yuping Wang**[2]
[1]Audio, Speech and Language Processing Group (ASLP@NPU)
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Douyin Vision Co., Ltd.

## Abstract

Recent language model (LM) advancements have showcased impressive zero-shot voice conversion (VC) performance. However, existing LM-based VC models usually apply offline conversion from source semantics to acoustic features, demanding the complete source speech and limiting their deployment to real-time applications. In this paper, we introduce StreamVoice, a novel streaming LM-based model for zero-shot VC, facilitating real-time conversion given arbitrary speaker prompts and source speech. Specifically, to enable streaming capability, StreamVoice employs a fully causal context-aware LM with a temporal-independent acoustic predictor, while alternately processing semantic and acoustic features at each time step of autoregression which eliminates the dependence on complete source speech. To address the potential performance degradation from the incomplete context in streaming processing, we enhance the context-awareness of the LM through two strategies: 1) teacher-guided context foresight, using a teacher model to summarize the present and future semantic context during training to guide the model's forecasting for missing context; 2) semantic masking strategy, promoting acoustic prediction from preceding corrupted semantic and acoustic input, enhancing context-learning ability. Notably, StreamVoice is the first LM-based streaming zero-shot VC model without any future look-ahead. Experiments demonstrate StreamVoice's streaming conversion capability while achieving zero-shot performance comparable to non-streaming VC systems.

## 1 Introduction

Voice conversion (VC) aims to transfer a speaker's voice to that of another speaker without changing the linguistic content. This technique has been deployed in many real-world applications, such as movie dubbing, privacy protection, pronunciation correction, etc. With the help of neural semantic features, such as the bottleneck feature (BNF) from an automatic speech recognition (ASR) system, converting source speech from arbitrary speakers in the wild has been successfully achieved (Sun et al., 2016). Meanwhile, converting to an arbitrary target speaker with only one utterance of this speaker, called *zero-shot VC*, has also been researched recently (Qian et al., 2019; Wang et al., 2023c). However, most existing zero-shot VC models are designed for offline systems, which are insufficient to meet the recent growing demands of streaming capability in real-time VC applications, such as live broadcasting and real-time communication (RTC). In this study, we focus on the *streaming zero-shot VC* as illuminated in Fig. 1.

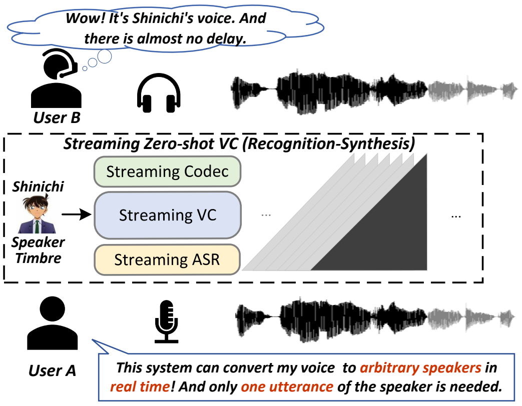
Wow! It's Shinichi's voice. And there is almost no delay.
User B
Streaming Zero-shot VC (Recognition-Synthesis)
Shinichi
Speaker Timbre
Streaming Codec
Streaming VC
Streaming ASR
User A
This system can convert my voice to arbitrary speakers in real time! And only one utterance of the speaker is needed.

Figure 1: The concept of the streaming zero-shot VC employing the widely used recognition-synthesis framework (Sun et al., 2016), where only the encoder of ASR is involved. StreamVoice is built on this popular paradigm.

Disentangling speech into different components, e.g., semantic content and speaker timbre, plays an important role in the zero-shot VC task (Chou and Lee, 2019; Wang et al., 2023d, 2021; Qian et al., 2019). Recently, benefiting from the powerful LM framework and the scaling up of training data, LM-based VC models (Wang et al., 2023c;

[*]Corresponding author

*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7328–7338
August 11-16, 2024 ©2024 Association for Computational Linguistics

Yang et al., 2023; Zhu et al., 2023) with built-in in-context learning ability can learn the context relations between source and target speaker's utterances to capture fine-grained speaker timbre, achieving impressive zero-shot VC performance. However, demanding the complete source speech utterance limits these LM-based VC models to real-time scenarios; thus, they can only be used in offline applications. While several non-LM-based methods (Yang et al., 2022; Wang et al., 2023a) have been proposed for streaming zero-shot VC, the performance fails to generalize well to unseen speakers with high speaker similarity and speech naturalness, mainly due to the limited model capacity to scale up training data, and also the performance degradation caused by the missing future information in streaming scenario.

Inspired by the success of LM-based models in zero-shot VC, we aim to explore the feasibility of LMs for the streaming VC scenario. An intuitive way is to follow the popular recognize-synthesis framework shown in Fig. 1, in which speech is represented in semantic BNF and acoustic features respectively extracted by a streaming ASR and an audio codec. Then, the LM-based VC model undertakes the transformation of semantic information into acoustic features with the target speaker's timbre. However, the development of the LM-based model in streaming zero-shot VC is hampered by two primary challenges.

- **Streamable architecture:** streaming models typically produce immediate output upon receiving current input without reliance on future time steps. Current LM-based VC models perform the conversion only when they get a full-utterance of source speech, which fails to meet the demands of streaming applications. The widely adopted multi-stage language modeling for multi-layer codec prediction introduces complexity to system design, posing a potential risk of cumulative errors. Additionally, the dependency models of the streaming pipeline also impact the design and performance of the VC model.

- **Performance gap:** unlike non-streaming models, streaming models must process frame-wise or chunked input causally on the fly without future information, facing missing context and potential performance degradation. This missing hinders the streaming VC model from achieving high-quality conversion. In addition, as shown in Fig. 1, the VC model relies on the semantic feature BNF from ASR to achieve conversion, which makes semantic features very important. However, streaming ASR exhibits inferior performance compared to its non-streaming counterpart, leading to the BNF carrying low-quality semantic information but more speaker information. In addition to the inherent unavailable future reception, this low-quality semantic input makes achieving high-quality conversion more difficult. The goal of zero-shot VC amplifies the challenges faced by our streaming VC model.

In this work, we propose *StreamVoice*, a streaming LM-based model for high-quality zero-shot VC. Specifically, StreamVoice has a streamable architecture that integrates a single-stage language model that casually generates acoustic codecs with the collaboration of an acoustic predictor. Alternating input of semantic and acoustic features at each time step ensures seamless streaming behavior. Two methods are introduced to enhance the context-awareness of the LM to mitigate the performance gap caused by missing contextual information. 1) We incorporate a teacher-guided context foresight, where the VC model is taught by a teacher non-streaming ASR to infer the present and future semantic information summarized by the teacher, which is then used to enhance the acoustic prediction. 2) To enhance the context learning from the input history, semantic masking encourages acoustic prediction from the preceding acoustic and corrupted semantic input, which also implicitly creates an information bottleneck to reduce the source speaker's information.

Experiments demonstrate StreamVoice's ability to convert speech in a streaming manner with high speaker similarity for both seen and unseen speakers while maintaining performance comparable to non-streaming VC systems. As the first LM-based zero-shot VC model without any future look-ahead, the total pipeline only has 124 ms latency to perform the conversion, 2.4x faster than real-time on a single A100 GPU without engineering optimizations. Converted samples can be found in https://kerwinchao.github.io/StreamVoice/.

## 2 Related Works

**Zero-shot Voice Conversion.** Zero-shot VC imposes stringent demands on speech decoupling and capturing speaker timbre. Many studies specifically design many disentanglement approaches, incorporating intricate structures (Chou and Lee, 2019), loss functions (Wang et al., 2021), and training strategies (Ebbers et al., 2021), to achieve speech decoupling. Rather than embedding explicit disentanglement designs in VC training, some approaches (Gu et al., 2021) leverage a speaker verification (SV) model for speaker representation, while linguistic content is extracted using ASR or self-supervised learning (SSL) models (Sun et al., 2016; Choi et al., 2021). To enhance speaker timbre capturing, some fine-grained speaker modeling methods have also been explored (Yin et al., 2021; Wang et al., 2023d). Recent successes of language models in generative tasks have prompted the exploration of LM-based models in zero-shot VC, yielding impressive results. Using the pre-trained model to decouple speech, the LM-based VC model (Wang et al., 2023c; Yang et al., 2023; Zhu et al., 2023) captures fine-grained speaker timbre from the speaker prompt and then performs the conversion. However, current LM-based VC models are inapplicable to streaming scenarios, constraining their real-world utility. This paper addresses this gap by investigating the zero-shot capabilities of language models specifically tailored for streaming scenarios.

**Streaming Voice Conversion.** Despite the high-quality conversion achieved by non-streaming VC models, their non-streamable structure and reliance on full-utterance input hamper them for real-time streaming applications. For streaming, causal processing and the structure of the streaming pipeline are crucial considerations. Streaming models are compelled to process frame-wise or chunked input on the fly, devoid of access to future information, leading to performance degradation compared to non-streaming counterparts. To address this, a common approach (Hayashi et al., 2022; Kameoka et al., 2021; Ning et al., 2023, 2024) involves the integration of a teacher model to guide the training of the streaming model or the distillation of knowledge from a non-streaming model. Chen et al. (2023b) focus on selecting BNF with minimal semantic information loss through layer-wise analysis, while Chen et al. (2022) incorporate adversarial training to enhance the quality of semantic features. Beyond streaming VC, some efforts have recently been towards streaming zero-shot VC. For instance, VQMIVC (Wang et al., 2021), designed for the non-streaming application, is modified to be streamable by Yang et al. (2022). ALO-VC (2023a) constructs a streaming system using an SV model, a streaming PPG extractor, and a pitch extractor. However, current streaming zero-shot VC, designed for low-resource devices, has limited model capacity with poor generalization to unseen speakers, leading to inferior similarity and naturalness. Motivated by LM's successes in zero-shot VC, we design a streamable LM in streaming scenarios. To tackle distinctive challenges in streaming VC, we enhance the context awareness of the LM to improve conversion quality.

**Language Model-based Speech Generation.** Recent advancements in LMs within natural language processing have showcased potent generation capabilities, influencing the development of LMs in speech generation. By employing codec (Zeghidour et al., 2021) or other SSL models (Chung et al., 2021), speech and audio can be efficiently tokenized into discrete units, facilitating low-bitrate audio representation and semantic extraction. This progress allows speech generation to utilize LM frameworks seamlessly. Taking audio generation as a conditional language modeling task, AudioLM (2023) employs hierarchical language modeling for acoustic prediction from coarse to fine units. VALL-E (2023b) and SpearTTS (2023) extend LMs for zero shot-TTS, which can clone a human's voice with prompt tokens from a short recording. For zero-shot VC, LM-VC (2023c) employs task-oriented optimizations to this task. And some studies (Zhu et al., 2023; Yang et al., 2023) leverage multitask objectives and datasets, achieving high-quality conversion. Despite this progress, existing LM-based VC models usually apply offline processing, demanding complete utterance from the source speech, which hinders their suitability for real-time streaming applications. In contrast to prior studies, we explore the zero-shot capability of the LM-based VC for streaming scenarios. With the enhancement of context awareness, the proposed LM-based VC model achieves results comparable to non-streaming LM-based VC.

## 3 StreamVoice

### 3.1 Overview

As shown in Fig. 2, the development of StreamVoice follows the recognition-synthesis framework. In this framework, speech is first represented as semantic features $\mathbf{s} = \{s_1, s_2, ...s_{T_s}\}$ and acoustic features $\mathbf{a} = \{a_1, a_2, ..., a_{T_a}\}$ by a pre-trained streaming ASR model and a speech codec model respectively. Here, $T_s$ and $T_a$ denote the sequence length. Before inputting to

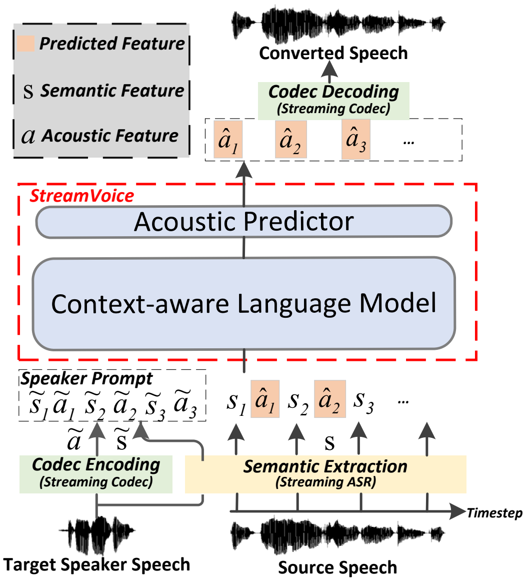

Figure 2: The overall architecture for StreamVoice.

StreamVoice, $\mathbf{s}$ and $\mathbf{a}$ are aligned to the same length $T$. StreamVoice incorporates a context-aware language model and an acoustic predictor to perform a single language modeling process. With the semantic and acoustic features $\{\tilde{\mathbf{s}}, \tilde{\mathbf{a}}\}$ of speech from the target speaker as speaker prompt, the LM leverages the semantic information $\mathbf{s}_{1:t}$ of source speech to autoregressively predict the hidden output ${}^c\mathbf{h}$. In each autoregression time-step of the LM, the acoustic predictor transforms the hidden output ${}^c\mathbf{h}$ to the codec feature $\hat{\mathbf{a}}$ of the converted speech. Finally, the codec model reconstructs the waveform from the predicted codec feature. In the following sections, we will introduce how to build a streamable LM for VC and how to ensure the high-quality conversation of this streaming VC.

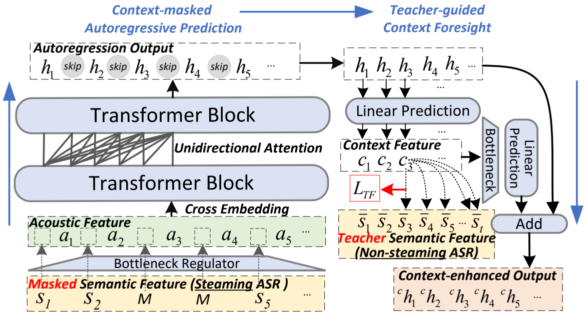
Context-masked
Autoregressive Prediction
Teacher-guided
Context Foresight
Autoregression Output
$h_1$ skip $h_2$ skip $h_3$ skip $h_4$ skip $h_5$ ...
$h_1$ $h_2$ $h_3$ $h_4$ $h_5$ ...
Transformer Block
Unidirectional Attention
Transformer Block
Cross Embedding
Linear Prediction
Context Feature
$c_1$ $c_2$ $c_3$ ...
Bottleneck
Linear Prediction
$L_{TF}$
Acoustic Feature
$a_1$ $a_2$ $a_3$ $a_4$ $a_5$ ...
Bottleneck Regulator
Masked Semantic Feature (Steaming ASR )
$s_1$ $s_2$ M M $s_5$ ...
$\bar{s}_1$ $\bar{s}_2$ $\bar{s}_3$ $\bar{s}_4$ $\bar{s}_5$ ... $\bar{s}_t$
Teacher Semantic Feature
(Non-steaming ASR)
Add
Context-enhanced Output
${}^ch_1$ ${}^ch_2$ ${}^ch_3$ ${}^ch_4$ ${}^ch_5$ ...

Figure 3: The architecture for context-aware LM.

## 3.2 Streamable Architecture

To perform streaming VC, a streamable architecture is necessary. In StreamVoice, the language model is carefully designed to perform full causal processing in the VC task, and the acoustic predictor is designed to achieve frame-wise prediction without dependency on temporal information.

### 3.2.1 Fully Casual Language Model

As shown in Fig. 3, inspired by the success of the LM-based VC model, we intend to achieve streaming zero-shot VC by language models. In previous LM-based VC models (Wang et al., 2023c), the demand of the complete semantic feature $\mathbf{s}$ from source speech to achieve conversion hinders the deployment for real-time application, which can be formulated as $p(a_t|\mathbf{s}_{1:T_s}, \mathbf{a}_{1:t-1})$ for each time step. To achieve streaming, any components of the LM cannot rely on future information. As shown in Fig. 3, decoder-only LM with unidirectional attention can easily fit the requirement of casual generation. To eliminate the dependency of the complete semantic input, semantic and acoustic features $\{\mathbf{s}, \mathbf{a}\}$ are first aligned with each other to the same sequence length $T$ and then they are alternatively inputted to the LM, forming a cross-embedding like $\{s_1, a_1, s_2, a_2, ..., s_T, a_T\}$. With these modifications, the LM can achieve streaming processing, modeling $p(a_t|\mathbf{s}_{1:t}, \mathbf{a}_{1:t-1})$.

To be specific, the semantic feature $\mathbf{s}$ obtained via an ASR model comprises a sequence of embeddings, denoted as $\{s_1, s_2, ..., s_T\}$. On the other hand, the codec tokens obtained from an $L$-layer codec are discrete units represented by $\mathbf{a} \in \mathcal{R}^{T \times L}$. To obtain the acoustic embedding sequence, the codec tokens from each layer undergo separate embedding into the embedding space, and then they are concatenated along the embedding dimension, resulting in the fused acoustic embedding. Both the fused acoustic embedding and semantic features are transformed into the same dimension using linear layers. Subsequently, they are alternately inputted into the language model, forming the cross-embedding.

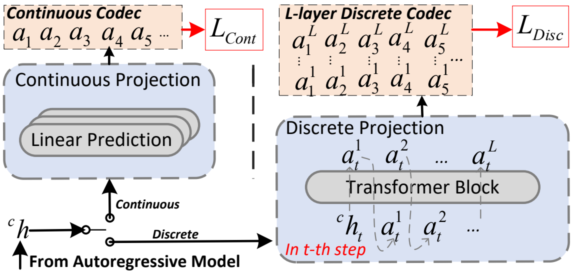
Continuous Codec
$a_1$ $a_2$ $a_3$ $a_4$ $a_5$ ...
$L_{Cont}$
L-layer Discrete Codec
$a_1^L$ $a_2^L$ $a_3^L$ $a_4^L$ $a_5^L$
$a_1^1$ $a_2^1$ $a_3^1$ $a_4^1$ $a_5^1$
$L_{Disc}$
Continuous Projection
Linear Prediction
Discrete Projection
$a_t^1$ $a_t^2$ ... $a_t^L$
Transformer Block
${}^ch_t$ $a_t^1$ $a_t^2$ ...
In t-th step
Continuous
Discrete
${}^ch$
From Autoregressive Model

Figure 4: The architecture for acoustic predictor. Our system can support continuous or discrete projection.

#### 3.2.2 Acoustic Predictor

As the preceding LM has essentially encoded content and speaker into its output ${}^c\mathbf{h}$, the acoustic predictor can be designed in temporal irrelevant to transform ${}^c\mathbf{h}$ into acoustic codec space, which means the predictor can be easily applied in the streaming scenario. Given that the speech can be represented in acoustic features by neural codec in either continuous or discrete forms, we investigate the incorporation of both features in StreamVoice, which are performed by continuous projection and discrete projection, respectively.

**Continuous Projection.** Following Shen et al. (2024), the $D$-dimensional quantized latent vector $\mathbf{a} \in \mathcal{R}^{T\times D}$ encoded by the codec model is used as the continuous acoustic representation. The prediction of the continuous representation involves employing a stack of linear layers, as shown in Fig. 4. The continuous projection loss is calculated as the L2 distance between the predicted acoustic feature $\hat{\mathbf{a}}$ and the ground-truth acoustic feature $\mathbf{a}$, which is defined as:

$$\mathcal{L}_{Cont} = ||\mathbf{a} - \hat{\mathbf{a}}||_2^2. \tag{1}$$

**Discrete Projection.** In general, the codec is designed with multi-layer quantizers to compress original speech into $L$-layer discrete indices $\mathbf{a} \in \mathcal{R}^{T\times L}$ at a low bitrate. Most LM-based work (Wang et al., 2023b,c) stacks multiple LMs to predict discrete features, making the pipeline complicated and unsuitable for the streaming scenario. In contrast, StreamVoice adopts a streamlined multi-layer codec prediction method inspired by MQTTS (Chen et al., 2023a). This method, free from temporal dependencies, can seamlessly integrate into the streaming process of the language model. Specifically, a single-layer transformer is used to model the hierarchical conditional distribution of codecs. As depicted in the right of Fig. 4, at time $t$, the transformer employs the ${}^c\mathbf{h}$ as the starting condition and sequentially generates $a_t^l$ from layer 1 to L. Remarkably, this generation process is independent of the preceding or the future ${}^\mathbf{c}\mathbf{h}$, rendering it well-suited for the demands of a streaming scenario. Notably, in the proposed StreamVoice, we mainly incorporate the discrete projection to achieve acoustic prediction. The discrete projection loss can be described as:

$$\mathcal{L}_{Disc} = -log\prod_{t=1}^{T}\prod_{l=1}^{L} p(a_t^l|\mathbf{a}_{1:t-1}, {}^m\mathbf{s}_{1:t}, t, a_t^{1:l-1}). \tag{2}$$

### 3.3 Context-aware Enhancement

Due to the disadvantage of the causality in the streaming framework, streaming models face missing future reception and potential performance degradation compared to the non-streaming model, while the low-quality semantic input from the streaming ASR, as we mentioned in Section 1, makes achieving high-quality conversion more challenging. To address these issues, a context-aware enhancement method is proposed, which can alleviate incomplete contextual information arising from the semantic input and the absence of future information. Specifically, we introduce context-masked autoregressive prediction in the LM to enhance the capture of *historical* context from the given semantic input. Meanwhile, a teacher-guided context foresight is proposed to ensure the model can imagine the *future* context based on that of its historical context.

**Context-masked Autoregressive Prediction.** As shown in the left of Fig. 3, the LM is achieved by the multi-layer Transformer with unidirectional attention, following the implementation of LLaMA (Touvron et al., 2023). To enhance contextual awareness from the given semantic input, semantic masking is introduced in the LM to encourage acoustic prediction from the corrupted semantic. Specifically, within a sequence of semantic tokens $\mathbf{s} = \{s_1, s_2, ...s_T\}$, we randomly select several indices as start indices at a ratio $r$, and spans of $l$ steps are masked by *[M]*. After masking, LM takes the corrupted semantic feature ${}^m\mathbf{s}$ as input and performs autoregression. With this method, an information bottleneck is also implicitly created in the semantic feature to reduce speaker information. Moreover, during training, we do not explicitly use a speech clip as the speaker prompt. Instead, LM leverages the previous sequence $\{\mathbf{s}_{1:t-1}, \mathbf{a}_{1:t-1}, s_t\}$ as prompts to autoregressively generate hidden representation $h_t$ for further acoustic prediction. Notably, when the current input is $a_t$, the corresponding output is skipped and does not involve further steps.

**Teacher-guided Context Foresight.** As previously discussed, the absence of future information resulting in the loss of contextual information leads to a decline in the conversion performance. Inspired by the effective representation learning exhibited by autoregressive predictive coding (2019) (APC), we introduce teacher-guided context foresight guided by a non-streaming ASR to enhance the autoregression output, as presented in the right of Fig. 3. This allows the model to learn a context vector containing envisioned future information. Specifically, the context representation $\mathbf{c}$ is first derived by linear prediction from the hidden features $\mathbf{h}$, which is generated by the LM through historical context. Subsequently, this $c_t$ is encouraged to discover more general future context information by minimizing the L2 distance not only with $k$ semantic features from future time steps $\overline{s}_{t+1}, ..., \overline{s}_{t+k}$ but also with the current semantic $\overline{s_t}$. This dual minimization approach contributes to precise content delivery and enhances the ability to forecast future context. The loss can be summarized as

$$\mathcal{L}_{TF} = \frac{1}{T-k} \sum_{1}^{T-k} \|c_t - Concat(\overline{s}_t, \overline{s}_{t+1}, ..., \overline{s}_{t+k})\|_2^2 \tag{3}$$

where $Concat(\cdot)$ denotes the concatenation of features along the dimensional axe. Unlike the original APC, which operates between the input and output of an autoregressive model, our approach employs a non-streaming ASR model as a teacher to provide semantic information $\overline{\mathbf{s}}$ for guiding this foresight process. This is done to tackle the inherent challenge of obtaining high-quality semantic features from the streaming ASR. After dimensional transformations, the context representation $\mathbf{c}$ is then combined with $\mathbf{h}$ to form the context-enhanced ${}^{c}\mathbf{h}$, which is then fed into the acoustic predictor.

Furthermore, since the semantic feature $\{\mathbf{s}, \overline{\mathbf{s}}\}$ still may contain speaker-related information. To further ensure the speech decoupling, the bottleneck regulator (Qian et al., 2019), which squeezes out speaker information by reducing dimension size with a linear layer, is applied in $\mathbf{s}$ and $\mathbf{c}$.

### 3.4 Training & Inference Procedure

**Training.** During training, the context-enhanced language model and acoustic predictor are trained together. The total loss can be described as $\mathcal{L}_{total} = \mathcal{L}_{TF} + \mathcal{L}_{Cont}$ for continuous codec or $\mathcal{L}_{total} = \mathcal{L}_{TF} + \mathcal{L}_{Disc}$ for discrete version.

**Streaming Inference.** We use the semantic and acoustic features from a short speech clip of the target speaker as the speaker prompt. Since this clip is randomly selected, which may contain unfinished pronunciation at the end of the clip, we pad a silence clip after the speaker recording before the conversion process to prevent the unexpected continuation. With this prompt, StreamVoice can stream convert the source speech. In discrete projection, we use greedy decoding to choose the codec token with the highest probability. Besides, to ensure the real-time streaming inference of StreamVoice, we employ the key-value cache in LM to reduce redundant calculations. In practice, since the beginning and end of the source speech can be determined by ASR or voice activity detection (VAD), we don't employ techniques, such as window attention or slide attention, to handle the input. StreamVoice's performance decreases when the long input exceeds the maximum training length. Notably, these techniques can be easily integrated into our framework, providing flexibility for future extensions.

## 4 Experiments

### 4.1 Experimental Setup

**Corpus.** A mixed dataset comprising 1,500 hours of Aishell3 (Shi et al., 2021) and an internal Chinese dataset are used to train StreamVoice and Audiodec (Wu et al., 2023). The internal dataset contains recordings from 2679 Chinese speakers, while we use utterances from 200 speakers in Aishell3. To extract semantic features, we incorporate a streaming ASR Fast-U2++ (Liang et al., 2023), which is implemented by WeNet (Yao et al., 2021) and trained on WenetSpeech (Zhang et al., 2022). For zero-shot testing, a set of 400 testing pairs is selected from DIDISpeech (Guo et al., 2021) and EMIME (Wester, 2010), each with a source and target speaker utterance. For evaluation of seen speakers, eight speakers from Aishell3 are selected to form 160 conversion pairs. And 3s speech utterance is used as a speaker prompt in inference. The duration of testing utterances is between 3s and 7s.

**Implement Details.** We use open-sourced code[1] of Audiodec, which has 4 quantizer layers with a 1024 codebook size and 64 codebook dimension, representing a 24kHz waveform in 20ms frame length. The Fast-U2++ uses an 80ms chunk size to perform streaming inference and compresses a 16kHz waveform into a semantic feature with a 40ms frame length. StreamVoice contains 101M parameters. For context-enhanced LM, we employ the variant of Transformer, LLaMA (Touvron et al., 2023), with 6 layers and 8 heads. The hidden and intermediate sizes are 1024 and 4096. We use the officially released code[2] to implement the acoustic predictor, which uses a layer Transformer decoder with a hidden size 256, feed-forward hidden size

[1] https://github.com/facebookresearch/AudioDec

[2] https://github.com/b04901014/MQTTS

1024, and 4 heads. In semantic masking, mask ratio $r$ ranges from 0.01 to 0.02, and span $l$ is set to 10. The foresight step $k$ is set to 4. The bottleneck regulator compresses feature dimensions by 6 times. During training, the max training length is set to 12s. StreamVoice is trained using 8 V100 GPUs with a batch size of 7 utterances per GPU for 700k steps. We use the AdamW optimizer with a learning rate of $5 \times 10^{-4}$. Exponential decay updates the learning rate after each epoch, using a decay ratio of 0.986.

**Evaluation Metrics.** The mean opinion score (MOS) subjectively measures speech naturalness (NMOS) and speaker similarity (SMOS), which are calculated with 95% confidence intervals. We randomly select 120 testing pairs for subjective evaluations involving a group of 15 listeners. For objective evaluations, a neural network-based system with open-source implementation[3] is used to measure speech quality (WV-MOS). Character error rate (CER) measured by an ASR model[4] indicates the speech intelligibility. Speaker similarity (SSIM) is calculated by an SV model (Desplanques et al., 2020) to determine if the converted speech matches the target speaker. Real-time factor (RTF) and latency indicate the streaming performance.

| Method | Quality | | | Similarity | |
|---|---|---|---|---|---|
| | NMOS ↑ | WVMOS ↑ | CER ↓ | SMOS ↑ | SSIM ↑ |
| GT (origin) | - | 3.61 | 6.29 | - | 0.803 |
| *Non-streaming Topline* | | | | | |
| LM-VC | 3.80±0.09 | 3.74 | 8.93 | 3.78±0.08 | 0.742 |
| NS-StreamVoice | 3.87±0.07 | 3.68 | 8.51 | 3.73±0.11 | 0.755 |
| *Streaming Model* | | | | | |
| C-StreamVoice | 3.72±0.10 | 3.49 | 10.2 | 3.67±0.09 | 0.729 |
| StreamVoice | 3.83±0.09 | 3.63 | 9.43 | 3.74±0.08 | 0.740 |

Table 1: Zero-shot performance (unseen speakers)

## 4.2 Experiments Results

### 4.2.1 Zero-shot Evaluation

To evaluate the zero-shot VC performance, one recent LM-based zero-shot VC system, *LM-VC* (Wang et al., 2023c), is selected as the topline system. Besides, a variant of StreamVoice, referred to as *NS-StreamVoice*, using a non-streaming ASR for semantic extraction, is also compared. We implement the proposed system *StreamVoice* integration discrete projection, while *C-StreamVoice* also involves the evaluation since speech can represented in continuous form by codec model. Table. 1 presents both subjective and objective results. Compared with the non-streaming topline LM-VC, our proposed StreamVoice can achieve close results regarding subjective NMOS and SMOS, while a performance gap still exists. Similar results are also observed in objective results. The non-streaming StreamVoice even surpasses the topline model in certain aspects, indicating the effectiveness of our streamable architecture for zero-shot VC. Additionally, C-StreamVoice exhibits inferior performance compared to the discrete version, which can contribute to the over-smoothness in speech generation (Ren et al., 2022) and the mismatch between the ground truth and predicted features.

As illustrated in Table. 2, the RTF of the entire pipeline is below 1, which meets the real-time requirement. Consisting of chunk-waiting latency (80ms) and model inference latency, the overall pipeline latency is 124.3 ms. If using a V100 GPU, StreamVoice can obtain an RTF of 0.56, and the overall latency reaches 137.2 ms. Importantly, unlike previous streaming VC, our VC model is entirely causal without any future look-ahead, highlighting its powerful modeling capability. These results show that StreamVoice can achieve high-quality zero-shot VC in streaming scenarios.

| | RTF | Latency (ms) |
|---|---|---|
| ASR Encoder | 0.13 | 10.4 |
| Codec Decoder | 0.004 | 0.3 |
| StreamVoice | 0.42 | 33.6 |
| Overall | 0.554 | 44.3+80=124.3 |

Table 2: Speed tested on an A100 80G GPU. Latency is obtained by multiplying RTF by 80ms chunk size.

| Method | Quality | | | Similarity | |
|---|---|---|---|---|---|
| | NMOS ↑ | WVMOS ↑ | CER ↓ | SMOS ↑ | SSIM ↑ |
| GT (origin) | - | 3.65 | 6.29 | - | 0.729 |
| *Non-streaming Topline* | | | | | |
| NS-VC | 3.85±0.09 | 3.71 | 8.39 | 3.92±0.08 | 0.744 |
| *Streaming Model* | | | | | |
| IBF-VC | 3.71±0.09 | 3.48 | 9.52 | 3.67±0.10 | 0.687 |
| DualVC2 | 3.80±0.10 | 3.57 | 10.2 | 3.85±0.09 | 0.703 |
| StreamVoice | 3.82±0.09 | 3.50 | 10.0 | 3.82±0.10 | 0.694 |
| + Tuning | 3.78±0.08 | 3.52 | 10.4 | 3.87±0.10 | 0.714 |

Table 3: In-dataset performance (seen speakers)

### 4.2.2 In-dataset Evaluation

To get further insight into StreamVoice, we conducted an in-dataset evaluation on eight seen speak-

[3] https://github.com/AndreevP/wvmos

[4] https://github.com/wenet-e2e/wenet/tree/main/examples/wenetspeech

ers, as shown in Table. 3. A non-streaming VC system (Tian et al., 2020) achieving any2many VC, is selected, referred to as *NS-VC*. Also, *IBF-VC* (Chen et al., 2022) and *DualVC2* (Ning et al., 2024) are recently proposed streaming models for any2many VC. As observed, a performance gap exists between the strong non-streaming topline and streaming models. Among the streaming models, StreamVoice, designed for the zero-shot scenario, delivers similar results to systems designed for in-dataset speakers, even though StreamVoice uses a smaller chunk size of 80ms in streaming ASR, achieving lower ASR performance. In contrast, IBF-VC and DualVC2 employ 160ms chunk size of ASR for streaming VC. It indicates StreamVoice's good conversion ability. With available utterances of target speakers, fine-tuning yields superior performance. This indicates our system can be easily applied to various scenarios with or without the utterances of target speakers.

| Method | WVMOS ↑ | CER ↓ | SSIM ↑ |
|---|---|---|---|
| StreamVoice | 3.63 | 9.43 | 0.740 |
| *w/o Teacher-guided Context Foresight* | | | |
| w/o $\mathcal{L}_{TF}(\overline{s}_t)$ | 2.56 | 76.8 | 0.59 |
| w/o $\mathcal{L}_{TF}(\overline{s}_{t+1:t+k})$ | 3.39 | 13.7 | 0.728 |
| w/o Semantic Masking | 3.47 | 13.0 | 0.715 |
| w/o Bottleneck Regulator | 3.59 | 9.21 | 0.718 |

Table 4: Results of ablation studies.

### 4.3 Ablation Study

As presented in Table 4, we conducted several ablations studies. In *w/o* teacher-guided context foresight, we discard the prediction of current and future semantic information, forming two ablations *w/o* $\mathcal{L}_{TF}(\overline{s}_t)$ and *w/o* $\mathcal{L}_{TF}(\overline{s}_{t+1:t+k})$. As can be seen, a noticeable decrease occurs in all evaluation metrics when the $\mathcal{L}_{TF}(\overline{s}_{t+1:t+k})$ is discarded, especially in WVMOS and CER. This indicates that this foresight improves performance in capturing the linguistic content. But when only integrating context from future semantics, the model *w/o* $\mathcal{L}_{TF}(\overline{s}_t)$ faces severe performance loss. It shows that only using future information interferes with delivering current linguistic content. In *w/o* semantic masking, we observe a performance decrease in all evaluation metrics when the semantic masking is discarded. This indicates that StreamVoice, trained with semantic masking, effectively enhances contextual learning from the preceding input while improving speaker timbre capturing. Furthermore, the results of dropping the bottleneck regulator show that its integration effectively prevents the source speaker information contained in the semantic feature from leaking into the converted speech, with little effect on speech quality.

| Type of ASR | WVMOS ↑ | CER ↓ | SSIM ↑ |
|---|---|---|---|
| Non-streaming ASR | 3.68 | 8.51 | 0.755 |
| *Streaming ASR (Moritz et al., 2019)* | | | |
| + 0ms Future Look-ahead | 3.19 | 91.7 | 0.674 |
| + 160ms Future Look-ahead | 3.48 | 10.6 | 0.727 |
| *Streaming ASR (Fast-U2++ (Liang et al., 2023))* | | | |
| Chunk (80ms) | 3.63 | 9.43 | 0.740 |
| Chunk (160ms) | 3.69 | 9.16 | 0.744 |

Table 5: Analysis of dependency on different ASR.

### 4.4 Discussion: Dependency Analysis

In this section, we will explore the dependency relations between the selection of ASR and codec and the performance of StreamVoice.

**ASR.** To investigate the impact of ASR on StreamVoice, three representative ASR systems, including non-streaming ASR [4], widely used CTC-based streaming ASR (Moritz et al., 2019), and the recently proposed streaming Fast-U2++ (Liang et al., 2023), are selected to perform semantic extraction. As can be seen in Table 5, StreamVoice using semantic features of non-streaming ASR outperforms those using streaming ASR. This discrepancy may be attributed to the inherent performance gap between non-streaming and streaming ASR models, resulting in different semantic extraction abilities. Besides, without future look-ahead in StreamVoice, using semantic features from (Moritz et al., 2019) cannot achieve reasonable conversion, while we introduce a 160ms future look-ahead in StreamVoice, i.e., modeling $p(a_t|\mathbf{a}_{1:t-1}, \mathbf{s}_{1:t+m}, t)$ with $m$ future look-ahead, yield good conversion results. This issue may arise from delayed CTC spike distributions and token emission latency existing in streaming ASR (Liang et al., 2023), leading to semantic information shifting. Benefiting from the low emission latency of Fast-U2++, StreamVoice can perform conversion without future look-ahead. With a longer chunk size employed in Fast-U2++, StreamVoice can obtain better results while reaching a larger latency of 270ms. A trade-off still exists between performance and speed.

**Codec.** In StreamVoice, we employ a low-latency streaming codec Audiodec (Wu et al., 2023). As presented in Table. 6, we validate the

| Type of Audiodec | WVMOS ↑ | CER ↓ | SSIM ↑ | RTF |
|---|---|---|---|---|
| w/ $2kbps$ Audiodec | 3.63 | 9.43 | 0.740 | 0.42 |
| w/ $8kbps$ Audiodec | 3.61 | 9.38 | 0.738 | 0.61 |
| Large w/ $8kbps$ Audiodec | 3.68 | 9.12 | 0.751 | 0.90 |

Table 6: Analysis of dependency on Audiodec with various bitrate.

performance of StreamVoice using codecs with different bitrates, including 2kbps and 8kbps, where higher bitrate codecs achieve superior reconstruction quality to lower bitrate ones. The 2kbps Audiodec utilizes 4 layers of quantization and represents audio with a frame length of 20ms, while the 8kbps Audiodec employs 8 layers with a frame length of 10ms. Using the configuration of StreamVoice mentioned in Section 4.1, the results in different bitrates of codec models show no obvious differences. When increasing the number of transformer layers in the codec predictor, forming *Large w/ 8kbps Audiodec*, conversion performance using 8kbps codec improves noticeably, but resulting in slower inference. This result shows that the design of StreamVoice depends on the codec configuration, affecting both conversion quality and inference speed.

## 5 Conclusions

This paper introduces StreamVoice, a novel LM-based zero-shot VC system designed for streaming scenarios. Specifically, StreamVoice employs a single-stage framework encompassing a context-aware LM and an acoustic predictor. The casual design of the model's input and structure ensures compliance with streaming behavior. To address performance degradation caused by missing complete contextual information in streaming scenarios, context-aware LM adopts teacher-guided context foresight to make the model able to forecast the current and future information given by a teacher. Besides, semantic masking is introduced in LM to enhance context learning from historical input and facilitate better disentanglement. Finally, an acoustic predictor collaborates with the LM to generate the target speech. Experiments demonstrate that StreamVoice achieves streaming zero-shot VC while maintaining performance comparable to non-streaming VC systems.

## 6 Limitations

We have to point out that StreamVoice still has limitations. In our configuration, StreamVoice needs a GPU, such as V100 and A100, to achieve real-time streaming inference. The design of streaming VC heavily relies on the ASR and the speech codec as mentioned in Section 4.4. Besides, StreamVoice also faces the out-of-domain problem, which causes performance degradation for utterances with accents, strong emotions, or unseen recording environments. Our future work will first use more training data with diversity coverage to explore StreamVoice's modeling ability. Also, we will focus on optimizing our streaming pipeline, such as high-fidelity codec with low bitrate and unified streaming model.

## 7 Ethics Statement

Since StreamVoice can convert source speech to desired speakers, it may carry potential risks of misuse for various purposes, such as spreading fake information or phone fraud. To prevent the abuse of the VC technology, many studies have focused on synthetic speech detection (Yi et al., 2022). Meanwhile, we also encourage the public to report the illegal usage of VC to the appropriate authorities.

## References

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2023. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533.

Li-Wei Chen, Shinji Watanabe, and Alexander Rudnicky. 2023a. A vector quantized approach for text to speech synthesis on real-world spontaneous speech. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Yuanzhe Chen, Ming Tu, Tang Li, Xin Li, Qiuqiang Kong, Jiaxin Li, Zhichao Wang, Qiao Tian, Yuping Wang, and Yuxuan Wang. 2023b. Streaming voice conversion via intermediate bottleneck features and non-streaming teacher guidance. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Ziyi Chen, Haoran Miao, and Pengyuan Zhang. 2022. Streaming non-autoregressive model for any-to-many voice conversion. *Arxiv*.

Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. 2021. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. In *Neural Information Processing Systems(NeurIPS)*, pages 16251–16265.

Juchieh Chou and Hungyi Lee. 2019. One-shot voice conversion by separating speaker and content representations with instance normalization. In *International Speech Communication Association (Interspeech)*, pages 664–668.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. 2019. An Unsupervised Autoregressive Model for Speech Representation Learning. In *International Speech Communication Association (Interspeech)*, pages 146–150.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. w2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *International Speech Communication Association (Interspeech)*, pages 3830–3834.

Janek Ebbers, Michael Kuhlmann, Tobias Cord-Landwehr, and Reinhold Haeb-Umbach. 2021. Contrastive predictive coding supported factorized variational autoencoder for unsupervised learning of disentangled speech representations. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3860–3864.

Yewei Gu, Zhenyu Zhang, Xiaowei Yi, and Xianfeng Zhao. 2021. MediumVC: Any-to-any voice conversion using synthetic specific-speaker speeches as intermedium features. *Arxiv*.

Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, and Xiangang Li. 2021. Didispeech: A large scale mandarin speech corpus. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6968–6972.

Tomoki Hayashi, Kazuhiro Kobayashi, and Tomoki Toda. 2022. An investigation of streaming non-autoregressive sequence-to-sequence voice conversion. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6802–6806.

Hirokazu Kameoka, Kou Tanaka, and Takuhiro Kaneko. 2021. Fasts2s-vc: Streaming non-autoregressive sequence-to-sequence voice conversion. *Arxiv*.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. 2023. Speak, Read and Prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718.

Chengdong Liang, Xiao-Lei Zhang, BinBin Zhang, Di Wu, Shengqiang Li, Xingchen Song, Zhendong Peng, and Fuping Pan. 2023. Fast-u2++: Fast and accurate end-to-end speech recognition in joint ctc/attention frames. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2019. Streaming end-to-end speech recognition with joint ctc-attention based models. In *Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 936–943.

Ziqian Ning, Yuepeng Jiang, Pengcheng Zhu, Shuai Wang, Jixun Yao, Lei Xie, and Mengxiao Bi. 2024. Dualvc 2: Dynamic masked convolution for unified streaming and non-streaming voice conversion. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11106–11110.

Ziqian Ning, Yuepeng Jiang, Pengcheng Zhu, Jixun Yao, Shuai Wang, Lei Xie, and Mengxiao Bi. 2023. Dualvc: Dual-mode voice conversion using intra-model knowledge distillation and hybrid predictive coding. In *International Speech Communication Association (Interspeech)*, pages 2063–2067.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. 2019. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning (ICML)*, pages 5210–5219.

Yi Ren, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2022. Revisiting over-smoothness in text to speech. In *Association for Computational Linguistics(ACL)*, pages 8197–8213.

Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, sheng zhao, and Jiang Bian. 2024. NaturalSpeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *International Conference on Learning Representations (ICLR)*.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2021. AISHELL-3: A multi-speaker mandarin tts corpus. In *International Speech Communication Association (Interspeech)*, pages 2756–2760.

Lifa Sun, K. Li, Hao Wang, Shiyin Kang, and H. Meng. 2016. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Xiaohai Tian, Zhichao Wang, Shan Yang, Xinyong Zhou, Hongqiang Du, Yi Zhou, Mingyang Zhang, Kun Zhou, Berrak Sisman, Lei Xie, and Haizhou Li. 2020. The NUS & NWPU system for Voice Conversion Challenge 2020. In *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, pages 170–174.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *Arxiv*.

Bohan Wang, Damien Ronssin, and Milos Cernak. 2023a. Alo-vc: Any-to-any low-latency one-shot voice conversion. In *International Speech Communication Association (Interspeech)*, pages 2073–2077.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023b. Neural codec language models are zero-shot text to speech synthesizers. *Arxiv*.

Disong Wang, Liqun Deng, Yu Ting Yeung, Xiao Chen, Xunying Liu, and Helen Meng. 2021. Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion. In *International Speech Communication Association (Interspeech)*, pages 1344–1348.

Zhichao Wang, Yuanzhe Chen, Lei Xie, Qiao Tian, and Yuping Wang. 2023c. Lm-vc: Zero-shot voice conversion via speech generation based on language models. *IEEE Signal Processing Letters*, pages 1157–1161.

Zhichao Wang, Liumeng Xue, Qiuqiang Kong, Lei Xie, Yuanzhe Chen, Qiao Tian, and Yuping Wang. 2023d. Multi-level temporal-channel speaker retrieval for robust zero-shot voice conversion. *Arxiv*.

Mirjam Wester. 2010. The EMIME bilingual database. Technical report, The University of Edinburgh.

Yi-Chiao Wu, Israel D Gebru, Dejan Marković, and Alexander Richard. 2023. Audiodec: An open-source streaming high-fidelity neural audio codec. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. 2023. Uniaudio: An audio foundation model toward universal audio generation. *Arxiv*.

Haoquan Yang, Liqun Deng, Yu Ting Yeung, Nianzu Zheng, and Yong Xu. 2022. Streamable speech representation disentanglement and multi-level prosody modeling for live one-shot voice conversion. In *International Speech Communication Association (Interspeech)*, pages 2578–2582.

Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. 2021. WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. In *International Speech Communication Association (Interspeech)*, pages 4054–4058.

Jiangyan Yi, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Ye Bai, Cunhan Fan, Shan Liang, Shiming Wang, Shuai Zhang, Xinrui Yan, Le Xu, Zhengqi Wen, and Haizhou Li. 2022. Add 2022: the first audio deep synthesis detection challenge. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Dacheng Yin, Xuanchi Ren, Chong Luo, Yuwang Wang, Zhiwei Xiong, and Wenjun Zeng. 2021. Retriever: Learning content-style representation as a token-level bipartite graph. In *International Conference on Learning Representations (ICLR)*.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. SoundStream: An end-to-end neural audio codec. *Transactions on Audio, Speech, and Language Processing*, 30:495–507.

Binbin Zhang, Hang Lv, Pengcheng Guo, Qijie Shao, Chao Yang, Lei Xie, Xin Xu, Hui Bu, Xiaoyu Chen, Chenchen Zeng, et al. 2022. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6182–6186.

Xinfa Zhu, Yuanjun Lv, Yi Lei, Tao Li, Wendi He, Hongbin Zhou, Heng Lu, and Lei Xie. 2023. Vec-tok speech: speech vectorization and tokenization for neural speech generation. *Arxiv*.