Computer Speech and Language 96 (2026) 101892

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

A speech prediction model based on codec modeling and transformer decoding

Heming Wang [a], Yufeng Yang [a,*], DeLiang Wang [b]

[a] Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, United States
[b] School of Data Science, The Chinese University of Hong Kong, Shenzhen, Guangdong, China

ARTICLE INFO

Keywords:
Speech prediction
Packet loss concealment
Speech codec
Speech inpainting
Transformer decoder

ABSTRACT

Speech prediction is essential for tasks like packet loss concealment and algorithmic delay compensation. This paper proposes a novel prediction algorithm that leverages a speech codec and transformer decoder to autoregressively predict missing frames. Unlike text-guided methods requiring auxiliary information, the proposed approach operates solely on speech for prediction. A comparative study is conducted to evaluate and compare the proposed and existing speech prediction methods on packet loss concealment (PLC) and frame-wise speech prediction tasks. Comprehensive experiments demonstrate that the proposed model achieves superior prediction results, which are substantially better than other state-of-the-art baselines, including on a recent PLC challenge. We also systematically examine factors influencing prediction performance, including context window lengths, prediction lengths, and training and inference strategies.

1. Introduction

Speech signals exhibit considerable correlations between consecutive samples and frames both acoustically and phonetically. In a classical study, Miller and Licklider (1950) demonstrate that the intelligibility of interrupted speech degrades little if the interruptions occur frequently and the uninterrupted fraction of the speech signal is relatively high (e.g. 75%). Therefore, it is feasible to perform speech prediction by leveraging past speech signals. The significance of speech prediction stems from its diverse applications, including recovering lost speech packets during speech transmission (Yang et al., 2023b), compensating for algorithmic delay in speech enhancement and activate noise control (Tan and Wang, 2018; Zhang and Wang, 2021; Luo et al., 2023, 2024), inpainting lost speech segments (Adler et al., 2011; Soni et al., 2018; Kegler et al., 2020; Miotello et al., 2023), and conditioned speech synthesis (Oord et al., 2016; Kalchbrenner et al., 2018; Prenger et al., 2019).

Conventional approaches predict speech signal by exploiting short-time correlations of speech, including interpolation (Janssen et al., 1986; Merazka, 2013), hidden Markov models (Rodbro et al., 2006), linear prediction (LP) (Yong et al., 1988; Gunduzhan and Momtahan, 2001; Kondo and Nakagawa, 2004), non-negative matrix factorization (Mokrỳ et al., 2023), sinusoidal modeling (Lindblom and Hedelin, 2002; Lagrange et al., 2005), compressed sensing (Haneche et al., 2020) and similarity graphs (Perraudin et al., 2018). While effective for short speech gaps, these methods struggle with restoring longer missing intervals as speech is decidedly a nonstationary signal. Furthermore, the complex nature of speech utterances makes accurately predicting their correlations challenging for these methods.

To address these limitations, deep neural network (DNN) based methods have been proposed. Representative techniques include using recurrent neural networks (RNNs) to capture speech sample dependencies and perform packet loss concealment

* Corresponding author.
E-mail address: yang.5662@osu.edu (Y. Yang).

https://doi.org/10.1016/j.csl.2025.101892
Received 27 February 2025; Received in revised form 5 August 2025; Accepted 22 September 2025
Available online 26 September 2025

(PLC) (Lotfidereshgi and Gournay, 2018), convolutional networks to treat spectrograms as images for inpainting (Kegler et al., 2020), encoder–decoder architectures for magnitude and complex spectrogram inpainting (Marafioti et al., 2019), and generative adversarial networks (GANs) to generate missing segments from neighboring regions (Ebner and Eltelt, 2020; Pascual et al., 2021). In addition, low bit-rate codecs are also introduced to reconstruct long-term packet loss (Andersen et al., 2002; Skoglund et al., 2008; Stimberg et al., 2020; Jiang et al., 2022). Recently, diffusion-based methods have also shown promise due to improved training stability and generative capacity (Moliner et al., 2023).

Another line of work focuses on self-supervised speech prediction and reconstruction. Autoregressive networks are used to model speech signals by predicting future frame spectra (Chung et al., 2019; Chung and Glass, 2020). Other techniques mask and reconstruct speech segments to learn short- and long-term dynamics (Ravanelli et al., 2020; Chi et al., 2021). However, these methods prioritize learning speech representations for downstream tasks, such as automatic speech recognition rather than signal reconstruction, and they do not perform well for speech prediction as a result.

Audio synthesis techniques are also employed to generate audio based on past information. Classic works in this field include the autoregressive speech synthesizer of WaveNet (Oord et al., 2016) which leverages dilated convolution to enlarge the receptive field to support longer-range dependency, and WaveRNN (Kalchbrenner et al., 2018) which utilizes RNN to model long-term dependencies. Other speech synthesizers generate the intermediate representations of mel-spectrogram, which are then converted to audio (Donahue et al., 2019; Prenger et al., 2019). Missing speech is also predicted by utilizing information from other modalities, for instance, auxiliary text information (Prablanc et al., 2016; Marafioti et al., 2019; Borsos et al., 2022), and visual information (Zhou et al., 2019; Morrone et al., 2021; Montesinos et al., 2023). In addition, text-to-speech systems (Kong et al., 2020; Tan et al., 2024) can potentially help to recover missing speech segments.

Recently, neural audio codec models have been applied to code continuous audio into discrete representations (Wu et al., 2024), facilitating the development of audio language models (LMs). Specifically, neural audio codec models are used to convert continuous audio to discrete codes, which can then be used to develop audio LMs. High-performance neural audio codecs and codec-based LMs have been developed to perform various tasks like text-to-speech synthesis, music generation, speech-to-speech translation, and speech enhancement by conditioning on textual or acoustic inputs (Agostinelli et al., 2023; Wang et al., 2024; Xue et al., 2024). Despite these efforts, successful studies are in text-to-speech tasks by relying on text embeddings or prefix-based prior information.

Previous research has developed a variety of techniques for speech prediction and reconstruction. There is, however, a lack of research that focuses on and comprehensively examines the speech prediction task itself. This article partly aims to fill this void by systematically investigating speech prediction methodologies and comprehensively evaluating them in a task-driven manner. We also propose a novel approach that employs speech codecs and transformer decoders to autoregressively predict speech frames based solely on past speech signals. The adoption of speech codecs provides significant advantages, as discrete speech representations facilitate predictive mapping in the embedding space. Our speech prediction model operates on frames of waveform samples, not spectrogram frames. To evaluate speech prediction performance, we focus on two important tasks: PLC and frame-wise autoregressive prediction. Our experiments on both tasks demonstrate that the proposed codec-based model achieves better prediction quality, higher objective scores like DNSMOS, and longer-range prediction, compared to existing approaches. Additionally, we systematically examine factors influencing prediction performance, including prediction lengths and context lengths.

The contribution of this study is three-fold. First, we propose a codec-based speech prediction approach that effectively leverages acoustic tokens and embeddings extracted from speech codecs. Second, we systematically evaluate the speech prediction performance of the proposed and other state-of-the-art baselines on two speech-prediction tasks. Third, we investigate the factors that impact speech prediction performance and examine different training and inference strategies.

The rest of the paper is organized as follows. The formulation of the speech prediction problem is given in Section 2. Section 3 describes the proposed model. Section 4 presents the training and test datasets, and experimental setup. Evaluation results and comparisons with other baselines are provided in Section 5. Finally, concluding remarks are given in Section 6.

2. Problem formulation

We consider two common approaches to speech prediction: a masking-based approach and an autoregressive approach, which are illustrated in Fig. 1. For the masking-based approach, we take a speech signal $\mathbf{x} \in \mathbb{R}^T$ of T samples and L frames, where each frame contains N samples. The nth frame of $\mathbf{x}$ can be represented as:

$$\mathbf{x}_n = x(t_n), \ldots, x(t_n + N - 1). \tag{1}$$

We use a binary frame-level mask $\mathbf{m}_n \in 0, 1$, $n = 0, \ldots, L-1$, to denote the missing frames, setting the masked frames to zero. During training, the model takes the masked speech samples $\mathbf{x} \odot \mathbf{m}$ as input, where $\odot$ is the Hadamard product, and predicts $\mathbf{x}$ utilizing the information from unmasked frames. This can be viewed as a speech inpainting task, where the model learns to fill in missing segments based on the available context. Let f be a DNN model with parameters θ. The masking-based speech prediction can be expressed as:

$$\mathbf{x} = f(\theta, \mathbf{x} \odot \mathbf{m}). \tag{2}$$

During inference, the model skips unmasked frames and predicts the missing frames based on the given frames. To assess prediction capability, we can simulate different mask patterns during training, such as random masks, burst masks, or their combinations.

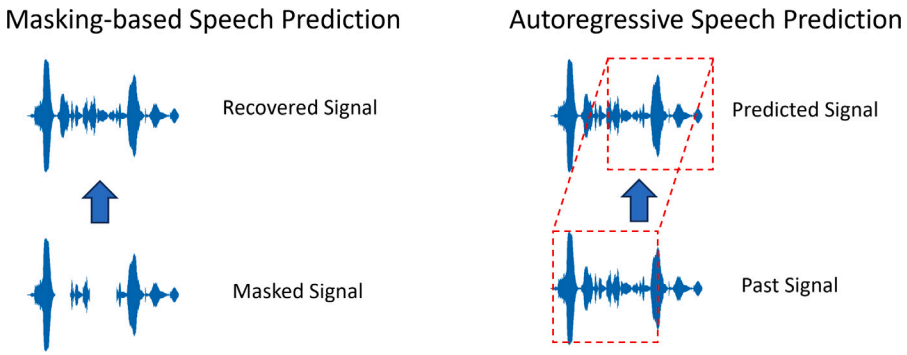


Fig. 1. Illustration of masking-based and autoregressive speech prediction.

In the autoregressive approach, given a sequence of L past frames $\mathbf{x}_{past} = \mathbf{x}_0, \dots, \mathbf{x}_{L-1}$ as input, the model aims to predict future J frames $\mathbf{x}_{future} = \mathbf{x}_L, \dots, \mathbf{x}_{L+J-1}$ in an autoregressive manner. This can be formulated as:

$$\mathbf{x}_{future} = f(\theta, \mathbf{x}_{past}). \tag{3}$$

During training, the model learns to predict future speech frames based on past information in a sequential manner, leveraging the temporal dependencies within speech signals. During inference, it can flexibly predict an arbitrary number of future frames by feeding the previously predicted frames as input. It is worth noting that the masking-based approach is more suitable for tasks like audio inpainting, where speech segments are missing or corrupted and low algorithmic delay is not demanding, while the autoregressive approach is better aligned with applications like real-time speech synthesis or delay-compensated training, the latter referring to model training to predict future frames based on past frames (Zhang and Wang, 2021).

In this study, we focus on the autoregressive approach since algorithm delay is a crucial issue in real-world applications. In addition, the autoregressive approach represents a more challenging case of the masking-based approach where only past information is utilized.

3. Proposed model

To predict speech frames, we propose to leverage pretrained speech codecs, which offer several advantages for this task. First, during pre-training, speech codecs learn to encode speech information, enabling them to handle significant information loss in speech signals. Second, tokenized representations compress the embedding space, making prediction and recovery in the embedding space easier relative to operating directly on speech signals. Third, existing phonetic information can be leveraged to aid faithful resynthesis of original speech, enhancing fidelity.

The proposed architecture combines a speech codec and an acoustic embedding prediction model. We employ a transformer decoder to perform autoregressive speech prediction, taking acoustic features as input. The predicted speech tokens are then used to generate missing speech frames.

3.1. Speech codec

Speech codecs play a pivotal role in encoding and decoding audio signals, facilitating transmission and storage. Recent speech codecs like EnCodec (Défossez et al., 2023), SoundStream (Zeghidour et al., 2021), and HiFiCodec (Yang et al., 2023a) leverage encoder–decoder networks and residual vector quantization (RVQ) for efficient speech compression and reconstruction. These codecs typically comprise three components: a feature encoder, a vector quantizer, and a speech decoder. The feature encoder converts input speech into lower-dimensional feature maps, which are then discretized into codes by the vector quantizer using codebooks. The speech decoder reconstructs clean speech from these code tokens and codebooks.

We employ a pretrained Factorized Codec (FACodec) (Ju et al., 2024) for speech token prediction, which is a recently proposed for speech and decomposes the speech generation process into two stages: a compression stage and a diffusion stage. The compression stage encodes raw waveform samples into a compact latent representation. This latent representation captures the essential characteristics of speech signal while significantly reducing its dimensionality. At the diffusion stage, latent representations are generated using a diffusion model trained on the compressed latent space, and are then converted to high-quality speech samples using a speech decoder. Furthermore, FACodec disentangles the complex speech waveform into distinct subspaces. It consists of a speech encoder, a timbre extractor, three factorized vector quantizers (FVQs) for content, prosody, and acoustic details, and a speech decoder. The speech encoder extracts a latent representation which is fed into the timbre extractor to obtain a global timbre vector. Separate factorized vector quantizers quantize the latent representations into discrete tokens capturing the content, prosody, and acoustic detail attributes. The decoder then reconstructs the waveform from these disentangled representations. This factorization allows for efficient and high-quality speech generation, achieving state-of-the-art performance in terms of quality,

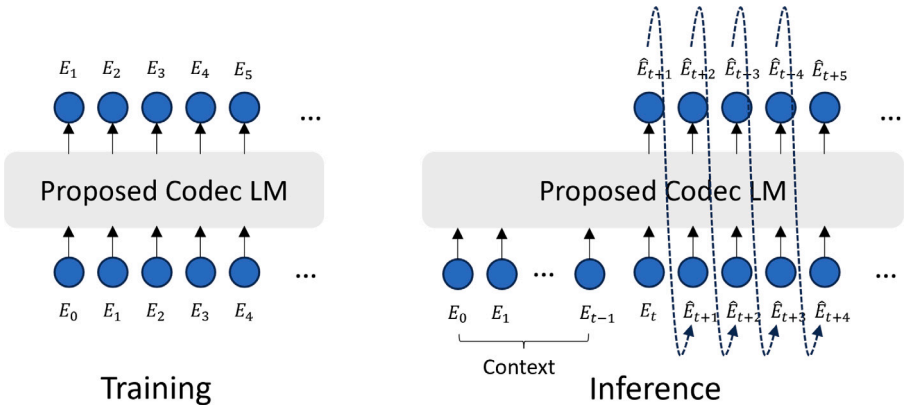


Fig. 2. Proposed pipeline for speech prediction during training and inference. Acoustic embeddings are quantized into tokens, which are then synthesized into speech waveforms using a codec decoder.

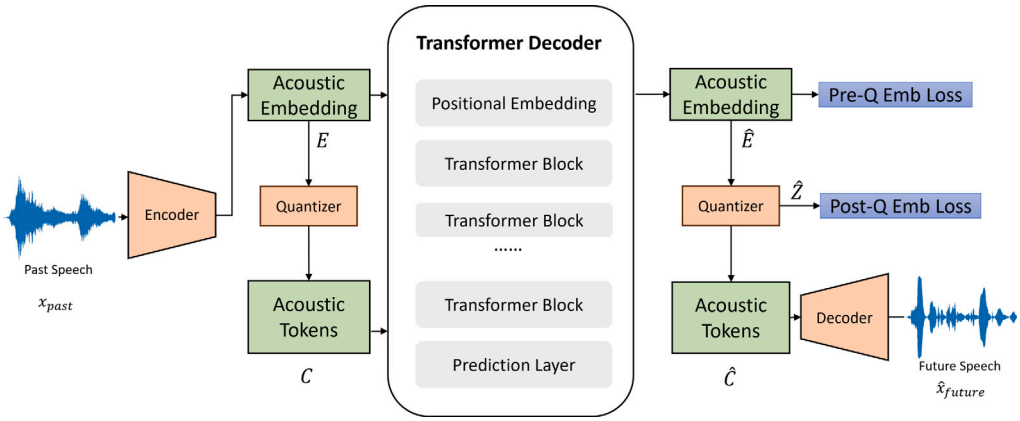


Fig. 3. Illustration of the proposed network architecture. A transformer decoder is employed to predict pre-quantizer acoustic embeddings, and all codec related components (encoder, quantizer and decoder) are frozen during training and inference.

similarity, prosody, and intelligibility. Specifically, we adopt pretrained checkpoints[1] that support 16 kHz audio, and have a frame shift of 200 samples. For each frame, 6 codec tokens are generated, and 6 FVQs are used for quantizing content, prosody, and acoustic details. Each FVQ has a codebook size of 1024.

3.2. Acoustic embedding prediction

Fig. 2 illustrates the proposed pipeline for acoustic embedding prediction, for training and inference. During training, the primary input of acoustic embedding is extracted from past speech frames $E_0, E_1, \ldots, E_{L-1}$, and trained to predict one frame ahead, i.e., $E_1, E_2, \ldots, E_L$. During inference, the model takes a fixed length of past speech frames as input (from $E_0, E_1, \ldots, E_{t-1}$), and autoregressively predicts future speech frames in a similar manner, taking the previously predicted frames into account for subsequent predictions if desired.

Specifically, as illustrated in Fig. 3, we use a pretrained speech codec to extract input features: acoustic embeddings $E \in \mathbb{R}^{(L+J)\times D}$ where D is the embedding dimension of the codec encoder, and acoustic tokens $C \in \mathbb{R}^{(L+J)\times P}$ where P is the number of speech codes per frame. Acoustic embeddings are obtained by passing input speech to the frozen encoder of the pretrained codec, also denoted as pre-quantizer embedding. Acoustic tokens are extracted by querying the codebooks in the quantizers using the acoustic embeddings. These features are first transformed to the same dimension of $(L + J) \times K$ with embedding layers and then summed with sinusoidal positional embeddings added (Vaswani et al., 2017) before feeding to the transformer decoder, which predicts the next-frame acoustic embedding $\hat{E}$ based on the previous speech. Speech tokens are derived from predicted embeddings, enabling future speech synthesis using the codec decoder.

[1] https://huggingface.co/amphion/naturalspeech3_facodec.

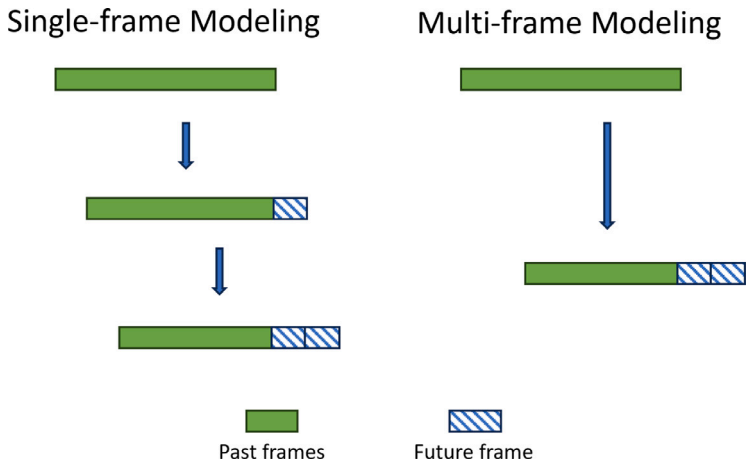


Fig. 4. Illustration of inference processes for single-frame and multi-frame models. When predicting two frames ahead, the single-frame approach infers twice, and using the first predicted frame as known speech information. The multi-frame inference approach directly maps the past speech two frames ahead.

The transformer decoder consists of 12 sequentially stacked blocks, each of which is designed to capture complex dependencies in the input data. Each block utilizes an embedding dimension K of 512, which defines the size of the vector space used to represent tokens. The multi-head attention mechanism within each block is configured with 8 attention heads, enabling the model to focus on different parts of the input simultaneously. The model incorporates 256-dimensional acoustic embeddings as input, encoding the audio features into a compact representation. The feed-forward layers, which follow the self-attention mechanism in each block, are configured with a hidden dimension of 2048, facilitating the learning of rich transformations and feature representations. To prevent overfitting, dropout is applied during training with a dropout rate of 0.1. At the final stage of the decoder, a prediction layer is employed to map the 512-dimensional outputs of the last decoder block into a 1024-dimensional space, which corresponds to the size of the codebook vocabulary. This projection aligns the model output with the discrete codebook entries used for downstream tasks.

3.3. Single-frame versus multi-frame modeling for multi-frame prediction

Fig. 4 illustrates two ways for multi-frame speech prediction: single-frame and multi-frame modeling.

In the conventional autoregressive approach, the model is trained to predict only one future frame at a time. During inference, it predicts multiple frames recursively by using the previously predicted frame as input for predicting the next frame, as illustrated in Fig. 2. While single-frame modeling is simple and flexible, this approach may accumulate prediction errors as recursion proceeds through the sequence, potentially degrading the quality of multi-frame prediction.

Multi-frame modeling, on the other hand, trains the model to predict multiple frames at once. By learning the dependencies within a longer sequence during training, the model may better exploit contextual information for more accurate predictions of longer speech segments. This approach is also more efficient during inference as it avoids repeated predictions. However, it is less flexible, as the number of predicted frames needs to be fixed during training. We investigate both single-frame and multi-frame modeling in this paper.

3.4. Training objective

Our preliminary experiments showed suboptimal speech prediction performance when training solely on acoustic tokens, potentially due to information loss from discretization and low acoustic token prediction accuracy. Therefore, instead of a conventional classification-based loss, our training objective is designed to optimize acoustic embeddings.

Specifically, we compute the mean absolute error (MAE) on the pre-quantized embeddings E and post-quantized embeddings Z. $\mathcal{L}^{pre}$ optimizes direct embedding prediction, while $\mathcal{L}^{post}$ computes the distance between ground-truth codebook entry and the derived entries from $\hat{E}$, ensuring that the prediction can better represent the predicted acoustic token C. We derive codebook entries $Z \in \mathbb{R}^{P\times(L+J)\times D}$ from embeddings E in a differentiable manner, where P is the number of code tokens and D is the feature dimension of each entry. The overall loss is:

$$\mathcal{L}^{codec} = \mathcal{L}^{pre} + \lambda \mathcal{L}^{post}$$

$$\mathcal{L}^{pre} = \frac{1}{(L+J)D} \sum_{t=1}^{(L+J)} \sum_{d=1}^{D} |E(t,d) - \hat{E}(t,d)|$$

$$\mathcal{L}^{post} = \frac{1}{P(L+J)D} \sum_{i=1}^{P} \sum_{t=1}^{(L+J)} \sum_{d=1}^{D} |Z_i(t,d) - \hat{Z}_i(t,d)|, \tag{4}$$

where Z_i represents the post-quantizer embeddings that correspond to the ith token. The coefficient λ is empirically set to 0.2 based on the validation performance.

4. Datasets and experimental setup

To evaluate the efficacy of the proposed approach, we conduct a comprehensive assessment through a series of experiments and systematic comparisons on PLC and frame-wise speech prediction tasks.

For the PLC task, we utilize the dataset presented in the INTERSPEECH 2022 Audio Deep Packet Loss Concealment Challenge (Diener et al., 2022). Compiled from a public-domain podcast dataset, it includes 23,184 training pairs of clean and lossy utterances, 966 utterances for validation, and another 966 utterances for testing. Each utterance approximately has a duration of 10 s. To mimic real-world packet loss, the dataset introduces audio gaps by zero-masking segments based on actual network loss patterns, mimicking network-induced audio losses.

For the frame-wise autoregressive speech prediction task, we conduct experiments on the WSJCAM0 dataset (Garofolo et al., 1993). This dataset consists of recorded utterances from the Wall Street Journal at 16 kHz sampling rate using a close-talk Sennheiser HMD414 microphone. The dataset comprises a total of 30 h of speech from 119 speakers. From this dataset, we select the `SI_TR_S` subset as the training data, which contains approximately 25 h of speech. Additionally, we set aside the `SI_DT_05` subset for validation, which consists of 1.5 h of utterances. The evaluation is conducted on the `SI_ET_05` subset, which has 651 utterances that have no overlap with the training or validation set.

During training, we adopt a frame length of 20 ms and a frame shift of 12.5 ms for generating speech features to be consistent with the FACodec. Each utterance is preprocessed with unit normalization to fit the value range of [−1, 1]. Our training procedure employs the Adam optimizer (Kingma and Ba, 2015), with batches of 32 utterances and an initial learning rate of 6e−4 for 200 epochs. A learning rate scheduler is applied, which halves the learning rate after three epochs without validation loss improvement. To maintain training stability, gradient clipping is set at a 3.0 threshold. During training, we randomly cut a 4-s segment from each utterance, and for utterances of shorter duration within each batch, zero-padding is applied at the end to match the dimensions of their longer counterparts. All models are evenly distributed across two NVIDIA Volta V100 32 GB GPUs for training, and the training process is expedited using automatic mixed precision (Micikevicius et al., 2017).

We assess speech prediction performance using three standard objective metrics. The perceptual evaluation of speech quality (PESQ) (Rix et al., 2001) is a widely-used metric that measures the quality of speech signals based on auditory perception. Additionally, we employ DNSMOS (Reddy et al., 2022), a non-intrusive metric that estimates speech quality without requiring a reference signal. Both PESQ and DNSMOS scores range from −0.5 to 4.5, with higher values indicating better speech quality. For the PLC task, we use the PLCMOS metric[2] officially provided by the deep PLC challenge organizers, with higher scores indicating better performance.

5. Results and comparisons

5.1. Packet loss concealment

We first present experimental results on the PLC challenge testing dataset and compare our results with existing baselines in Table 1; the results of the baseline methods are copied from the leaderboard in Diener et al. (2022). Our model achieves a PLCMOS score of 4.29. This score is slightly better than the top score by KuaiShou (Li et al., 2022), which is significantly higher than those of other methods documented in the PLC challenge (Diener et al., 2022). Our model also produces the best DNSMOS score among the strong baselines, and a large PESQ improvement over the zero-filling baseline. Unlike other approaches relying on lossmap simulations and masking-based training, our proposed method only needs clean speech information and conducts autoregressive training. It is worth noting the proposed method uses a 12.5 ms frame shift, but for the PLC challenge task requiring a 10 ms frame shift, we only consider the initial 10 ms of each recursive prediction during inference. In addition, our system does not require frame lookahead, using only past information. Specifically, during PLC inference, given the loss traces (either 1 or 0), if no packet loss is detected, maintain the model output as is. In case of a detected loss, the model uses past information within a fixed-length context window of 80 frames to reconstruct the lost packet. For a lost packet occurring within the first 80 frames, its prediction is based on the shorter context window from the first frame to the frame immediately before the lost packet. In addition, a newly reconstructed frame is used in predicting subsequent lost packets. Compared to the zero-filling baseline, our method provides large improvements in speech quality, achieving 1.38 PLCMOS, 0.44 DNSMOS, and 1.15 PESQ improvements. On a 2.30 GHz Intel Xeon Gold 5218 CPU, the proposed model has a latency of around 10.50 ms, with 0.42 ms for token prediction and 0.08 ms for codec processing, incorporating a 10 ms frame shift.

[2] https://github.com/microsoft/PLC-Challenge/tree/main/PLCMOS.

Table 1
The objective scores of different models on the PLC challenge test set.

Model	PLCMOS	DNSMOS	PESQ
Zero-filling baseline	2.90	3.44	2.19
Amazon (Valin et al., 2022)	3.74	3.79	–
Aibaba Inc. (Liu et al., 2022)	3.83	3.68	–
Oldenburg University (Westhausen and Meyer, 2022)	3.98	3.69	–
KuaiShou Inc. (Li et al., 2022)	4.28	3.80	–
Proposed	**4.29**	**3.83**	**3.34**

Table 2
Single-frame speech prediction performance of different models on the WSJ0 dataset.

	STOI	PESQ	DNSMOS
LPC (Kondo and Nakagawa, 2004)	0.531 ± 0.012	1.102 ± 0.072	2.460 ± 0.131
LSTM (Lotfidereshgi and Gournay, 2018)	0.650 ± 0.032	1.693 ± 0.113	2.546 ± 0.165
WaveNet (Oord et al., 2016)	0.686 ± 0.012	1.278 ± 0.092	2.669 ± 0.164
TFGAN (Wang et al., 2021)	0.720 ± 0.025	2.494 ± 0.202	3.490 ± 0.218
TF-CrossNet (Kalkhorani and Wang, 2024)	0.752 ± 0.029	2.448 ± 0.178	2.696 ± 0.176
Proposed	**0.849 ± 0.035**	**3.221 ± 0.198**	**3.558 ± 0.207**

5.2. Frame-wise speech prediction

Unlike one-pass prediction where the model takes an entire utterance as input and generates the complete sentence, we perform frame-wise speech prediction to satisfy real-time processing requirements. This can also be applied to compensate for algorithm delay. For this task, we systematically evaluate the prediction performance using a frame size of 200 samples (12.5 ms). We employ a fixed-length context window for predicting future frames. When predicting consecutive frames, we incorporate the previously predicted frame from the DNN into the context window to predict the next frame. Otherwise, we shift the context window. All predicted frames are concatenated and compared to the corresponding ground-truth speech frames, discarding the frames in the initial context window during comparison.

5.2.1. Predicting one frame ahead

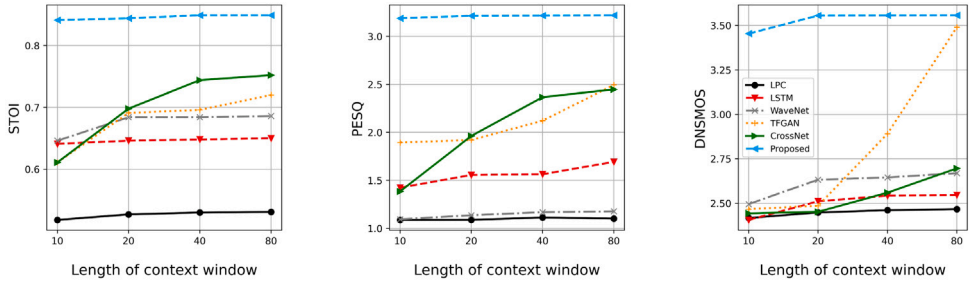
We compare the speech frame prediction performance of different models on the WSJ0 dataset and present the results in Table 2. Specifically, we use a context window of 80 frames and predict one future frame for all models. For each metric, we provide the mean and standard deviation. For traditional approaches, we evaluate the classic LPC method (Gold and Morgan, 2000; Kondo and Nakagawa, 2004), which is computationally efficient. We also compare with neural network based methods, including adaptive LSTM (Lotfidereshgi and Gournay, 2018), an RNN capable of capturing temporal dependencies, and two generative models for speech prediction: the autoregressive waveform mapping model WaveNet (Oord et al., 2016) and the generative adversarial network TFGAN (Wang et al., 2021). Note that WaveNet operates on waveform samples, whereas our model operates on waveform frames which should be more efficient at runtime. The recently proposed TF-CrossNet (Kalkhorani and Wang, 2024) is also included for its powerful modeling capability. For LPC prediction, we use an LPC order of 64 for forward prediction only, due to causality constraint, and we also compute with Levinson–Durbin recursion for acceleration. We employ autoregressive training for LSTM and TF-CrossNet to predict future speech frames based on past ones and compute loss across all predicted frames. WaveNet compresses samples using μ-law compression, and is trained on past compressed samples to predict future ones. TFGAN's generator and discriminator are trained similarly in an autoregressive manner to produce and differentiate legitimate predicted frames.

As evident from Table 2, our method outperforms the baseline methods in terms of DNSMOS. TFGAN also achieves a high DNSMOS score due to adversarial training, which improves perceptual speech quality with the help of its discriminator. In addition, the proposed method achieves substantially higher STOI and PESQ scores than the baselines. Our codec-based model demonstrates superior ability in capturing speech characteristics for high-quality frame-wise prediction. This is likely because speech information is effectively compressed by the codec, which utilizes a pre-trained codebook that reduces ambiguity and uncertainty in the recovery mapping, facilitating DNN-based prediction.

Our informal listening to predicted speech utterances indicates that the proposed method yields more intelligible speech than the comparison baselines. In terms of speech quality, our method also produces better sounding signals. Among the baselines, TFGAN predicted utterances sound the best, followed by those predicted by TF-CrossNet. These observations are consistent with the objective scores in Table 2. We have created a demo page to provide the predicted utterances from the methods in Table 2 at https://whmrtm.github.io/uploads/ARSP_demo.html.

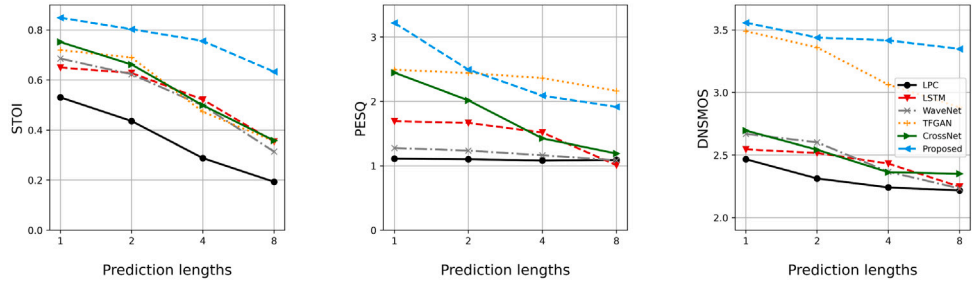
5.2.2. Comparison of different context windows

Determining the optimal context window length is non-trivial. A longer window captures more information but increases computational requirements, while a shorter window reduces computational costs but may fail to capture essential longer dependencies. We systematically examine the impact of context window length on prediction performance. We compare the performance of predicting



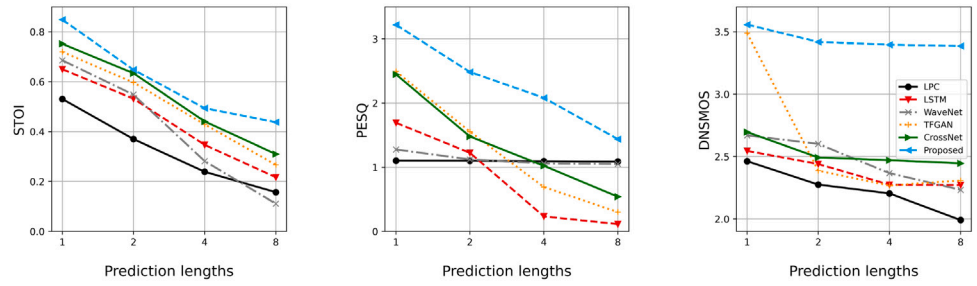
(a) STOI (b) PESQ (c) DNSMOS

Fig. 5. Effects of context window lengths (number of frames) on prediction performance. (a). STOI, (b). PESQ, and (c). DNSMOS.



(a) STOI (b) PESQ (c) DNSMOS

Fig. 6. Effects of prediction lengths (numbers of frames) using the multi-frame model. (a). STOI, (b). PESQ, and (c). DNSMOS.



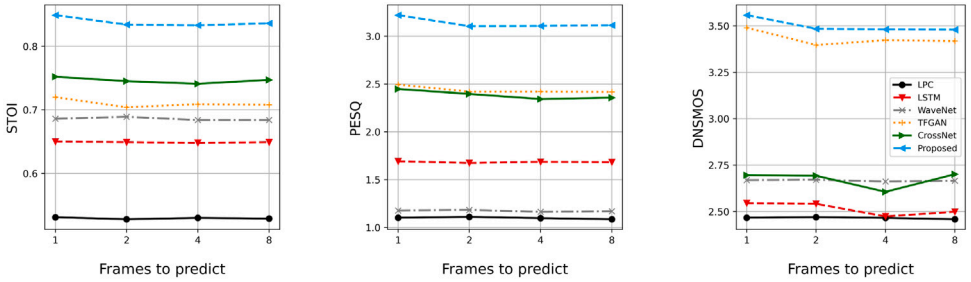
(a) STOI (b) PESQ (c) DNSMOS

Fig. 7. Effects of prediction lengths (numbers of frames) using the single-frame model. (a). STOI, (b). PESQ, and (c). DNSMOS.

a single frame using different context windows (10, 20, 40 and 80 frames), and display the results in Fig. 5. As expected, a longer context window yields better prediction performance. For TFGAN and TF-CrossNet, a context window of 10 frames is too small and notably reduces the prediction performance. The proposed method, on the other hand, achieves stable performance for different context windows. Even with only 10 frames (125 ms) as the context window, the proposed approach performs almost as well as with 80 frames in STOI and PESQ. In terms of DNSMOS, the prediction performance of the proposed model with a 10-frame context window drops a little compared to the long window of 80 frames. But with a context window of 20 frames, the proposed model predicts as well as with the long window.

5.2.3. Comparison of different prediction lengths

We now train the proposed and comparison models to predict 1, 2, 4, and 8 frames ahead and evaluate their speech prediction performance for different prediction lengths. Fig. 6 provides their prediction results. As expected, predicting further into the future is more difficult for all models, as evidenced by a clear performance drop across all metrics with increasing prediction length.



(a) STOI (b) PESQ (c) DNSMOS

Fig. 8. Single-frame prediction performance of multi-frame models. (a). STOI, (b). PESQ, and (c). DNSMOS.

Table 3
Ablation study of different techniques employed in the proposed model.

Model	STOI	PESQ	DNSMOS
Proposed (Emb+Token2Emb)	**0.849 ± 0.035**	**3.221 ± 0.198**	3.558 ± 0.207
(i) Token2Token	0.618 ± 0.028	2.424 ± 0.136	2.714 ± 0.148
(ii) Emb2Token	0.834 ± 0.031	3.137 ± 0.188	3.637 ± 0.225
(iii) Emb+Token2Token	0.835 ± 0.038	3.139 ± 0.204	**3.639 ± 0.218**
(iv) Emb2Emb	0.845 ± 0.033	3.220 ± 0.193	3.552 ± 0.196
(v) No $\mathcal{L}^{post}$	0.846 ± 0.040	3.215 ± 0.182	3.541 ± 0.202
(vi) Smaller network	0.821 ± 0.033	3.073 ± 0.184	3.343 ± 0.187

Additionally, since prediction is autoregressive, a model trained to predict only one frame can be used to predict multiple frames by performing recursive inference. Fig. 7 depicts the performance of recursive inference for predicting 1, 2, 4, and 8 speech frames, using the model trained for single-frame prediction. The differences between these two inference strategies are discussed in Section 3.3 earlier. Compared to multi-frame inference, single-frame recursive inference offers better flexibility as it can handle the prediction of an arbitrary number of frames. However, prediction errors may accumulate with each recursion. A comparison between Figs. 6 and 7 reveals that, although both inference strategies exhibit a clear performance drop when predicting more frames, recursive inference experiences more severe degradation. For instance, when predicting 8 frames, the proposed model and TFGAN yield notably low STOI and PESQ scores for recursive inferences. LPC's steady PESQ performance across different prediction lengths may be attributed to its consistent fundamental frequency estimation and tendency to repeat with longer predictions.

5.2.4. *Single-frame performance for multi-frame models*

As shown previously, multi-frame prediction demonstrates superior performance for longer speech prediction than single-frame recursive prediction. An intriguing question arises: Does multi-frame modeling sacrifice the quality of single-frame prediction? To investigate this, we conduct additional evaluations of the multi-frame models on the first predicted frame. As shown in Fig. 8, prediction performance does not significantly deteriorate, even when trained to predict 8 frames. This finding suggests that, while multi-frame training lacks flexibility for arbitrary prediction lengths, it offers better performance when the number of frames to predict is known in advance. In addition, multi-frame training allows for the prediction of fewer frames than the number of frames fixed during training with little performance degradation. This observation implies that multi-frame modeling can also be flexible; for example, a trained 8-frame model can be used to predict 1–8 frames, depending on the application.

5.3. *Ablation study*

Finally, we investigate the contributions of different components of the proposed model to prediction performance. The experiments are conducted on the WSJ0 dataset, evaluating the one-frame prediction capability of the following variants:

(i) Token2Token: Given acoustic tokens, predict the next-frame speech token.
(ii) Emb2Token: Given acoustic embeddings, predict the next-frame token.
(iii) Emb+Token2Token: Given acoustic embeddings and tokens, predict the next-frame token.
(iv) Emb2Emb: Given acoustic embeddings, predict the speech embeddings of the next frame, and recover speech tokens using the pre-stored codebook.
(v) No post-embedding loss: Remove the L_{post} term in the loss function (see Eq. (4)).
(vi) Smaller network: Use 4 transformer blocks instead of 12, and reduce the embedding dimension to 256.

Table 4
Comparison of different codecs.

	Prediction			Oracle		
	STOI	PESQ	DNSMOS	STOI	PESQ	DNSMOS
FACodec (Ju et al., 2024)	**0.849**	**3.221**	**3.558**	**0.889**	**3.420**	**3.659**
SoundStream (Zeghidour et al., 2021)	0.610	2.798	3.214	0.819	2.843	3.494
EnCodec (Défossez et al., 2023)	0.616	2.923	3.416	0.845	3.220	3.647
HiFiCodec (Yang et al., 2023a)	0.782	2.984	3.439	0.858	3.152	3.599

Table 5
Number of trainable parameters and macs for different models, where M indicates million.

	MACs (M)	# of parameters (M)
LPC (Kondo and Nakagawa, 2004)	–	–
LSTM (Lotfidereshgi and Gournay, 2018)	**0.187**	1.98
WaveNet (Oord et al., 2016)	10.62	3.75
TFGAN (Wang et al., 2021)	15.89	1.85
TF-CrossNet (Kalkhorani and Wang, 2024)	148.86	**1.71**
Proposed	41.07	41.09
Proposed smaller	4.80	4.73

The comparison results are given in Table 3. Compared to the proposed model (Emb+Token2Emb), other variants show various degrees of performance degradation. The Token2Token variant, although straightforward, does not yield good performance, as input tokens lose too much speech information, making speech prediction difficult (Hu et al., 2023; Wang et al., 2023). As shown in (ii), (iii) and (iv) rows, using acoustic embeddings facilitates training and produces better results. Adding both embedding and token features yields slightly better mean scores. Predicting acoustic embeddings instead of tokens improves STOI and PESQ but reduces DNSMOS scores. Moreover, matching prediction and input formats enables convenient autoregressive inference. Also, loss computation solely based on pre-quantizer embeddings produces slightly lower mean scores relative to the proposed training objective. Lastly, we employ a smaller code prediction model comprising only 4 transformer blocks. Despite the much reduced size and computation (see Table 5), the smaller model exhibits only a mild reduction of performance, and it still shows a competitive DNSMOS score and clearly better STOI and PESQ results compared to the baselines in Table 2.

For the proposed model, we also investigate using different pre-trained codecs for single-frame prediction and present their results in Table 4. The adopted FACodec is compared with SoundStream (Zeghidour et al., 2021), EnCodec (Défossez et al., 2023), and HiFiCodec (Yang et al., 2023a). Note that we adopt a 12.5 ms unit for FACodec, while the others use 20 ms per frame. SoundStream and EnCodec predict 8 code tokens per frame, whereas HiFiCodec[3] uses all 4 tokens, computing loss based on group RVQ computation. Oracle results are also listed in Table 4, where the ground-truth tokens are provided to each codec. FACodec clearly yields the best prediction performance, indicating that prediction performance is highly influenced by the chosen codec.

Finally, Table 5 compares computational costs of different methods, measured in terms of multiply-accumulate operations (MACs) per time frame and the number of trainable parameters. TF-CrossNet shows the highest computational complexity with 148.86M MACs, followed by our proposed method at 41.07M MACs. WaveNet and TFGAN have 10.62M and 15.89M MACs respectively, while LSTM has the lowest computational cost at 0.187M MACs. In terms of model size, our model is the largest. Despite the higher parameter count, our method has reasonable computational efficiency during inference. Moreover, the smaller version of our model described earlier is computational efficient, has a compact size, and produces better prediction performance than the baselines.

6. Conclusion

In conclusion, this paper proposes a speech prediction approach that combines a pretrained speech codec and a transformer decoder for autoregressive prediction. Unlike existing methods relying on auxiliary data or text, our approach operates solely on speech signals, using past acoustic information to predict future speech frames. Through comprehensive experiments on two tasks – PLC and frame-wise speech prediction – we demonstrate the superior predictive capability of the proposed model to classic and deep learning baselines. Our approach achieves the state-of-the-art PLC performance, as well as the best frame-wise prediction results. Furthermore, we systematically investigate factors that influence prediction performance, including context window length, prediction length, and training and inference strategies. Our findings provide valuable insights into the fundamental task of speech prediction, and can be applied to packet loss concealment and tasks that benefit from speech prediction such as low-latency speech enhancement and active noise control (Zhang and Wang, 2021).

[3] https://huggingface.co/Dongchao/AcademiCodec/blob/main/HiFi-Codec-16k-320d-large-universal.

CRediT authorship contribution statement

Heming Wang: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yufeng Yang:** Writing – review & editing, Visualization, Validation. **DeLiang Wang:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: DeLiang Wang reports financial support was provided by National Institute on Deafness and Other Communication Disorders. DeLiang Wang reports financial support and equipment, drugs, or supplies were provided by Ohio Supercomputer Center. DeLiang Wang reports financial support and equipment, drugs, or supplies were provided by Pittsburgh Supercomputing Center. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported in part by an NIDCD, United States (R01 DC012048) grant, the Ohio Supercomputer Center, and the Pittsburgh Supercomputing Center, United States (under NSF grant ACI-1928147).

Data availability

Data will be made available on request.

References

Adler, A., Emiya, V., Jafari, M.G., Elad, M., Gribonval, R., Plumbley, M.D., 2011. Audio inpainting. IEEE Trans. Audio Speech Lang. Process. 20, 922–932.
Agostinelli, A., Denk, T.I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., et al., 2023. MusicLM: Generating music from text. arXiv:2301.11325.
Andersen, S.V., Kleijn, W.B., Hagen, R., Linden, J., Murthi, M.N., Skoglund, J., 2002. iLBC-a linear predictive coder with robustness to packet losses. In: Proc. of IEEE Workshop. pp. 23–25.
Borsos, Z., Sharifi, M., Tagliasacchi, M., 2022. SpeechPainter: Text-conditioned speech inpainting. In: Proc. INTERSPEECH. pp. 431–435.
Chi, P.-H., Chung, P.-H., Wu, T.-H., Hsieh, C.-C., Chen, Y.-H., Li, S.-W., Lee, H.-Y., 2021. Audio Albert: A lite bert for self-supervised learning of audio representation. In: Proc. SLT. pp. 344–350.
Chung, Y.-A., Glass, J., 2020. Generative pre-training for speech with autoregressive predictive coding. In: Proc. IEEE ICASSP. pp. 3497–3501.
Chung, Y.-A., Hsu, W.-N., Tang, H., Glass, J., 2019. An unsupervised autoregressive model for speech representation learning. In: Proc. INTERSPEECH. pp. 146–150.
Défossez, A., Copet, J., Synnaeve, G., Adi, Y., 2023. High fidelity neural audio compression. Trans. Mach. Learn. Res. 2835–8856.
Diener, L., Sootla, S., Branets, S., Saabas, A., Aichner, R., Cutler, R., 2022. INTERSPEECH 2022 audio deep packet loss concealment challenge. In: Proc. INTERSPEECH. pp. 580–584.
Donahue, C., McAuley, J., Puckette, M., 2019. Adversarial audio synthesis. In: Proc. International Conference on Learning Representations. pp. 1–16.
Ebner, P.P., Eltelt, A., 2020. Audio inpainting with generative adversarial network. arXiv:2003.07704.
Garofolo, J., Graff, D., Paul, D., Pallett, D., 1993. CSR-I (WSJ0) complete LDC93S6A. In: Web Download. Linguistic Data Consortium, Philadelphia, p. 83.
Gold, B., Morgan, N., 2000. Speech and Audio Signal Processing. Wiley & Sons, New York.
Gunduzhan, E., Momtahan, K., 2001. Linear prediction based packet loss concealment algorithm for PCM coded speech. IEEE Trans. Speech Audio Process. 9, 778–785.
Haneche, H., Boudraa, B., Ouahabi, A., 2020. A new way to enhance speech signal based on compressed sensing. Measurement 151, 107117.
Hu, Y., Chen, C., Zhu, Q., Chng, E.S., 2023. Wav2code: Restore clean speech representations via codebook lookup for noise-robust asr. IEEE/ACM Trans. Audio Speech Lang. Process. 32, 1145–1156.
Janssen, A., Veldhuis, R., Vries, L., 1986. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. IEEE Trans. Acoust. Speech Signal Process. 34, 317–330.
Jiang, X., Peng, X., Zheng, C., Xue, H., Zhang, Y., Lu, Y., 2022. End-to-end neural speech coding for real-time communications. In: Proc. IEEE ICASSP. pp. 866–870.
Ju, Z., Wang, Y., Shen, K., Tan, X., Xin, D., Yang, D., Liu, Y., Leng, Y., Song, K., Tang, S., et al., 2024. NaturalSpeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. In: Proc. ICML. pp. 22605–22623.
Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., Oord, A., Dieleman, S., Kavukcuoglu, K., 2018. Efficient neural audio synthesis. In: Proc. ICML. pp. 2410–2419.
Kalkhorani, V.A., Wang, D.L., 2024. TF-CrossNet: Leveraging global, cross-band, narrow-band, and positional encoding for single- and multi-channel speaker separation. IEEE/ACM Trans. Audio Speech Lang. Process. 32, 4999–5009.
Kegler, M., Beckmann, P., Cernak, M., 2020. Deep speech inpainting of time-frequency masks. In: Proc. INTERSPEECH. pp. 3276–3280.
Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. Proc. ICLR.
Kondo, K., Nakagawa, K., 2004. A packet loss concealment method using recursive linear prediction. In: Proc. INTERSPEECH. pp. 2633–2636.
Kong, J., Kim, J., Bae, J., 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. Adv. Neural Inf. Process. Syst. 33, 17022–17033.
Lagrange, M., Marchand, S., Rault, J.-B., 2005. Long interpolation of audio signals using linear prediction in sinusoidal modeling. J. Audio Eng. Soc. 53, 891–905.
Li, N., Zheng, X., Zhang, C., Guo, L., Yu, B., 2022. End-to-end multi-loss training for low delay packet loss concealment. In: Proc. INTERSPEECH. pp. 585–589.
Lindblom, J., Hedelin, P., 2002. Packet loss concealment based on sinusoidal modeling. In: Speech Coding, IEEE Workshop Proceedings. pp. 65–67.

Liu, B., Song, Q., Yang, M., Yuan, W., Wang, T., 2022. PLCNet: Real-time packet loss concealment with semi-supervised generative adversarial network. In: Proc. INTERSPEECH. pp. 575–579.
Lotfidereshgi, R., Gournay, P., 2018. Speech prediction using an adaptive recurrent neural network with application to packet loss concealment. In: Proc. IEEE ICASSP. pp. 5394–5398.
Luo, Z., Shi, D., Gan, W.-S., Huang, Q., 2023. Delayless generative fixed-filter active noise control based on deep learning and Bayesian filter. IEEE/ACM Trans. Audio Speech Lang. Process..
Luo, Z., Shi, D., Ji, J., Shen, X., Gan, W.-S., 2024. Real-time implementation and explainable AI analysis of delayless CNN-based selective fixed-filter active noise control. Mech. Syst. Signal Process. 214, 111364.
Marafioti, A., Perraudin, N., Holighaus, N., Majdak, P., 2019. A context encoder for audio inpainting. IEEE/ACM Trans. Audio Speech Lang. Process. 27, 2362–2372.
Merazka, F., 2013. Packet loss concealment by interpolation for speech over IP network services. In: Proc. CIWSP. pp. 1–4.
Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al., 2017. Mixed precision training. In: Proc. ICLR.
Miller, G.A., Licklider, J.C., 1950. The intelligibility of interrupted speech. J. Acoust. Soc. Am. 22, 167–173.
Miotello, F., Pezzoli, M., Comanducci, L., Antonacci, F., Sarti, A., 2023. Deep prior-based audio inpainting using multi-resolution harmonic convolutional neural networks. IEEE/ACM Trans. Audio Speech Lang. Process..
Mokrỳ, O., Magron, P., Oberlin, T., Févotte, C., 2023. Algorithms for audio inpainting based on probabilistic nonnegative matrix factorization. Signal Process. 206, 108905.
Moliner, E., Lehtinen, J., Välimäki, V., 2023. Solving audio inverse problems with a diffusion model. In: Proc. IEEE ICASSP. pp. 1–5.
Montesinos, J.F., Michelsanti, D., Haro, G., Tan, Z.-H., Jensen, J., 2023. Speech inpainting: Context-based speech synthesis guided by video. In: Proc. INTERSPEECH. pp. 4459–4463.
Morrone, G., Michelsanti, D., Tan, Z.-H., Jensen, J., 2021. Audio-visual speech inpainting with deep learning. In: Proc. IEEE ICASSP. pp. 6653–6657.
Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. WaveNet: A generative model for raw audio. arXiv:1609.03499.
Pascual, S., Serrà, J., Pons, J., 2021. Adversarial auto-encoding for packet loss concealment. In: Proc. of WASPAA. pp. 71–75.
Perraudin, N., Holighaus, N., Majdak, P., Balazs, P., 2018. Inpainting of long audio segments with similarity graphs. IEEE/ACM Trans. Audio Speech Lang. Process. 26, 1083–1094.
Prablanc, P., Ozerov, A., Duong, N.Q., Pérez, P., 2016. Text-informed speech inpainting via voice conversion. In: Proc. EUSIPCO. pp. 878–882.
Prenger, R., Valle, R., Catanzaro, B., 2019. WaveGlow: A flow-based generative network for speech synthesis. In: Proc. IEEE ICASSP. pp. 3617–3621.
Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., Bengio, Y., 2020. Multi-task self-supervised learning for robust speech recognition. In: Proc. IEEE ICASSP. pp. 6989–6993.
Reddy, C.K., Gopal, V., Cutler, R., 2022. DNSMOS P. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In: Proc. IEEE ICASSP. pp. 886–890.
Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P., 2001. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In: Proc. IEEE ICASSP. pp. 749–752.
Rodbro, C.A., Murthi, M.N., Andersen, S.V., Jensen, S.H., 2006. Hidden Markov model-based packet loss concealment for voice over IP. IEEE Trans. Audio Speech Lang. Process. 14, 1609–1623.
Skoglund, J., Kozica, E., Linden, J., Hagen, R., Kleijn, W.B., 2008. Voice over IP: Speech transmission over packet networks. Springer Handb. Speech Process. 307–330.
Soni, M.H., Shah, N., Patil, H.A., 2018. Time-frequency masking-based speech enhancement using generative adversarial network. In: Proc. IEEE ICASSP. pp. 5039–5043.
Stimberg, F., Narest, A., Bazzica, A., Kolmodin, L., Gonzalez, P.B., Sharonova, O., Lundin, H., Walters, T.C., 2020. WaveNetEQ—Packet loss concealment with WaveRNN. In: Proc. of ACSSC. pp. 672–676.
Tan, X., Chen, J., Liu, H., Cong, J., Zhang, C., Liu, Y., Wang, X., Leng, Y., Yi, Y., He, L., et al., 2024. NaturalSpeech: End-to-End Text-to-Speech synthesis with Human-Level quality. IEEE Trans. Pattern Anal. Mach. Intell..
Tan, K., Wang, D.L., 2018. A convolutional recurrent neural network for real-time speech enhancement. In: Proc. of INTERSPEECH. pp. 3229–3233.
Valin, J.-M., Mustafa, A., Montgomery, C., Terriberry, T., Klingbeil, M., Smaragdis, P., Krishnaswamy, A., 2022. Real-time packet loss concealment with mixed generative and predictive model. In: Proc. INTERSPEECH. pp. 570–574.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.
Wang, J., Guan, Y., Zheng, C., Peng, R., Li, X., 2021. A temporal-spectral generative adversarial network based end-to-end packet loss concealment for wideband speech transmission. J. Acoust. Soc. Am. 150, 2577–2588.
Wang, X., Thakker, M., Chen, Z., Kanda, N., Eskimez, S.E., Chen, S., Tang, M., Liu, S., Li, J., Yoshioka, T., 2024. SpeechX: Neural codec language model as a versatile speech transformer. IEEE/ACM Trans. Audio Speech Lang. Process. 32, 3355–3364.
Wang, H., Yu, M., Zhang, H., Zhang, C., Xu, Z., Yang, M., Zhang, Y., Yu, D., 2023. Unifying robustness and fidelity: A comprehensive study of pretrained generative methods for speech enhancement in adverse conditions. arXiv:2309.09028.
Westhausen, N.L., Meyer, B.T., 2022. tPLCnet: Real-time deep packet loss concealment in the time domain using a short temporal context. In: Proc. INTERSPEECH. pp. 2903–2907.
Wu, H., Chen, X., Lin, Y.-C., Chang, K.-w., Chung, H.-L., Liu, A.H., Lee, H.-y., 2024. Towards audio language modeling - an overview. arXiv:2402.13236.
Xue, H., Peng, X., Lu, Y., 2024. Low-latency speech enhancement via speech token generation. In: Proc. IEEE ICASSP. pp. 661–665.
Yang, D.-H., Kim, D., Chang, J.-H., 2023b. Masked frequency modeling for improving packet loss concealment in speech transmission systems. In: Proc. WASPAA. pp. 1–5.
Yang, D., Liu, S., Huang, R., Tian, J., Weng, C., Zou, Y., 2023a. HiFi-Codec: Group-residual vector quantization for high fidelity audio codec. arXiv:2305.02765.
Yong, M., Davidson, G., Gersho, A., 1988. Encoding of LPC spectral parameters using switched-adaptive interframe vector prediction (speech coding). In: Proc. IEEE ICASSP. pp. 402–403.
Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., Tagliasacchi, M., 2021. SoundStream: An end-to-end neural audio codec. IEEE/ACM Trans. Audio Speech Lang. Process. 30, 495–507.
Zhang, H., Wang, D.L., 2021. Deep ANC: A deep learning approach to active noise control. Neural Netw. 141, 1–10.
Zhou, H., Liu, Z., Xu, X., Luo, P., Wang, X., 2019. Vision-infused deep audio inpainting. In: Proc. CVPR. pp. 283–292.