An Audio Frequency Unfolding Framework for Ultra-Low Sampling Rate Sensors

Zhihui Gao, Minxue Tang, Ang Li, Yiran Chen
Department of Electrical and Computer Engineering
{zhihui.gao, minxue.tang, ang.li630, yiran.chen}@duke.edu

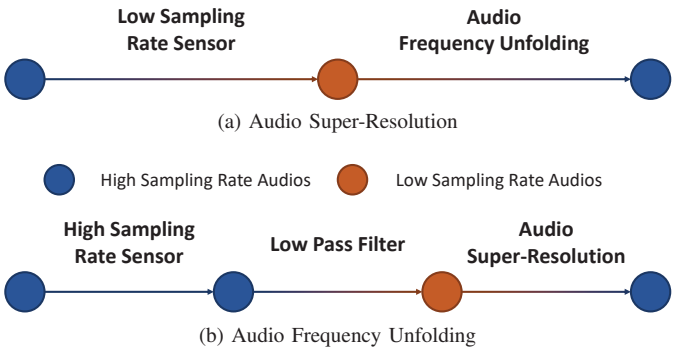
2022 23rd International Symposium on Quality Electronic Design (ISQED) | 978-1-6654-9466-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISQED54688.2022.9806149

***Abstract*—Recent audio super-resolution works have achieved significant success in promoting audio quality by improving a sensor's sampling rate, e.g., from 8 kHz to 48 kHz. However, these works fail to maintain the performance when the sampling rate at the sensor is ultra-low, where the audios suffer serious frequency aliasing. In this paper, we propose an audio frequency unfolding framework that efficiently reconstructs the aliasing audios to be perceptually recognizable. The intuition is that the audios generated by humans have a regular pattern on the spectrums; by learning such a regular pattern, our framework can reconstruct audio that sounds similar to real human voices. We evaluate our framework in a perceptual way: an automatic speech recognition (ASR) system is used to judge whether the words in the reconstructed audios can be correctly recognized. In the implementation based on AudioMNIST, when reconstructing the sampling rate from 2 kHz to 16 kHz, the recognition accuracy of the reconstructed audio reaches** 77.1%**.**

I. Introduction

The audio of human voices becomes an essential data source in many applications in reality, such as speech recognition [7], [2], user identification [5] and human localization [15], [19]. The common method to acquire these audios always requires a microphone with a high sampling rate. In general, a microphone with a sampling rate over 8 kHz can be considered speech-recognizable and with a sampling rate of 48 kHz is of good quality [10]. Such a high sampling rate of a microphone usually renders high power consumption, which limits the microphone's wider deployment on low-power devices. On the other hand, a microphone's being low-power means its low sampling rate, which suffers frequency aliasing according to the Nyquist sampling theorem. Besides the power consumption issue of the microphone, recent works [12], [18] focus on extracting audios from inertial measurement units (IMU). Compared to the microphones, the audios extracted from IMUs concentrate on the sound sources traveled from the solid mediums, less interfered by the noise source far away. However, the sampling rate of the IMU, much lower than that of a microphone [18], also suffers frequency aliasing. Hereby, given the benefits of the low-power microphones and the IMUs over the traditional microphones, is it possible to address the frequency aliasing problem, i.e., reconstructing their low sampling rates to a high sampling rate?

Recently, many efforts have been devoted to audio super-resolution [10], [11], [3], [21], which improves an audio's sampling rate to promote its quality. For example, the 8 kHz compressed audios can be promoted to 48 kHz high fidelity audios. However, the audio super-resolution does not match our problem. The input audios for super-resolution are collected by a high sampling rate sensor and pass a low pass filter, as shown in Fig. 1(a). This low pass filter ensures the audio's low-frequency band remains unchanged, where the frequency aliasing problem does not appear. In addition, the sampling rates of the input audios are not very low, i.e., they can still be recognized by human or automatic speech recognition (ASR) systems. As for audio frequency unfolding, the low sampling rate is due to the low sampling rate at the sensors, shown in Fig. 1(b). Since the ultra-low sampling rate of the sensor, serious frequency aliasing occurs. Besides, the ultra-low sampling rate significantly degrades the recognition accuracy.



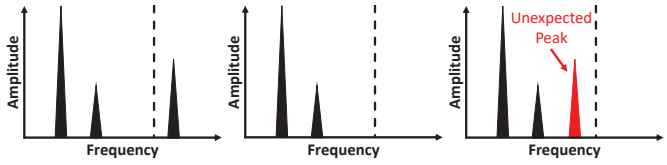
(a) Audio Super-Resolution

(b) Audio Frequency Unfolding

Fig. 1: The low sampling rate of the input audios for audio super-resolution is caused by the low pass filter (a) and that for audio frequency unfolding is caused by the low sampling rate sensor (b).

In this paper, we propose an audio frequency unfolding framework, which reconstructs the audio's low sampling rate to a recognizable high sampling rate. Our framework first transfers audio to the spectrogram and demonstrates the frequency unfolding on its spectrogram. Specifically, we build our frequency unfolding model on U-Net [14]. Based on the signal processing theorem, we develop a new pixel shuffle layer that reconstructs the spectrums more effectively.

To train the framework, we downsample original audio with a high sampling rate to produce the audio collected by a sensor with a low sampling rate. Then, our framework reconstructs this produced audio to new audio with a high sampling rate. This new audio is evaluated by comparing it to the original audio. Specifically, we adopt a reconstructed loss and a per-



(a) Original (b) Super-Resolution (c) Frequency Unfolding

Fig. 2: The original spectrums of human voice audios (a) and the input spectrums for super-resolution (b) and frequency unfolding (c).

ceptual loss. The former loss is the spectrogram discrepancy of the original audios and the reconstructed audios. In addition, a perceptual loss minimizes the acoustic features of the two audios, i.e., filter bank (FBank) in our implementation. This is because this feature fully determines the subjective feeling for human ears. Hence, the two audios with closed acoustic features perceptually sound similar.

Different from existing audio evaluation metrics, we evaluate our framework by an automatic speech recognition (ASR) system. Existing metrics, including signal-to-noise ratio (SNR) and log-spectral distance (LSD) [8], compare the details in the waveforms or spectrograms of audios. However, it is possible that two audios with totally different waveforms and spectrograms sound similar for human ears. These two audios are both acceptable in practice but are labeled as a bad performance by these two metrics. On the other hand, the ASR system, whose working principle is similar to human ears, judges audios by the words in the audios. In this way, as long as the two audios contain the same words, the reconstruction is regarded as successful.

II. Preliminary

Both audio super-resolution and audio frequency unfolding is designed to improve the sampling rate of audio. In this section, we first introduce the concept of audio super-resolution and several state-of-the-art related works. Then, we illustrate the audio frequency unfolding and how it is different from the audio super-resolution.

A. Audio super-resolution

Audio super-resolution refers to the process that promotes compressed audio with a low sampling rate to high fidelity audios with a high sampling rate. In reality, the original audios collected by microphones are too heavy to be directly used, so compression is exploited. Generally, the spectrum of human voice audio is composed of several peaks at different frequencies, as shown in Fig. 2(a). To compress these audios, a low-pass filter is adopted to filter out high-frequency peaks. In this way, the input spectrum in the low-frequency band remains unchanged after downsampling, shown in Fig. 2(b). As a result, although the overall quality of the audio is deprecated, the audio can still be mostly recognized by its spectrum in the low-frequency band.

In the field of signal processing, Whittaker-Shannon formular [20] and B-spline [17] are two traditional algorithms that improve the sampling rate of audios. According to the Nyquist sampling theorem, the recovered audios are still limited by the Nyquist sampling rate, which is half of the input audio's sampling rate. With the emergence of deep learning, one of the most widely adopted frameworks for audio super-resolution is based on U-Net [14]. This framework first extracts the features of audio in the time domain or the frequency domain by downsampling and then reconstructs the audio with a high sampling rate by upsampling [10], [11]. Besides U-Net, generative adversarial network (GAN) [6] is another framework that enables the super-resolution task. In GAN, a generator takes the low sampling rate audios as input and reconstructs the high sampling rate audios. Not being supervised, a discriminator is used to evaluate the quality of the reconstructed audios [3], [21].

B. Audio frequency unfolding

Audio frequency unfolding enables the audios collected by low sampling rate sensors to be recognizable, where there are two main challenges to be addressed.

As shown in Fig. 2(c), without a low-pass filter that removes the high-frequency peaks, directly using a low sampling rate sensor to collect audios folds the high-frequency peaks to the unexpected peaks in the low-frequency band. Such unexpected peaks seem no different from other low-frequency peaks and make the spectrum ambiguous. Therefore, the task of frequency unfolding is not only to reconstruct the missing peaks at high frequency but also to recognize the unexpected peaks and remove them.

Another challenge of the frequency unfolding is the ultra-low sampling rate of the input audio, e.g., 2 kHz, which is much lower than the Nyquist frequency of the human voice. As a result, the audio suffers a more serious frequency aliasing issue, which can neither be recognized by human ears nor the ASR systems.

Despite the two challenges, fortunately, there are regular patterns of human throats. In detail, for a given phoneme by the human voice, the peaks always appear at certain frequencies. As long as these regular patterns can be correctly recognized, the peaks in the spectrum can be correctly reconstructed. In this way, audio with a high sampling rate can be reconstructed and perceptually recognized.

III. Design

In this section, we first introduce the model architecture and how it fits the characteristic of audio frequency unfolding. Then, we illustrate the loss function for training the model, which avoids the over-fitting and benefits the training convergence.

A. Model Architecture

We mainly perform the frequency unfolding on the spectrogram, termed the spectrogram unfolding. Here the spectrogram is a series of spectrums by the short-time Fourier transform

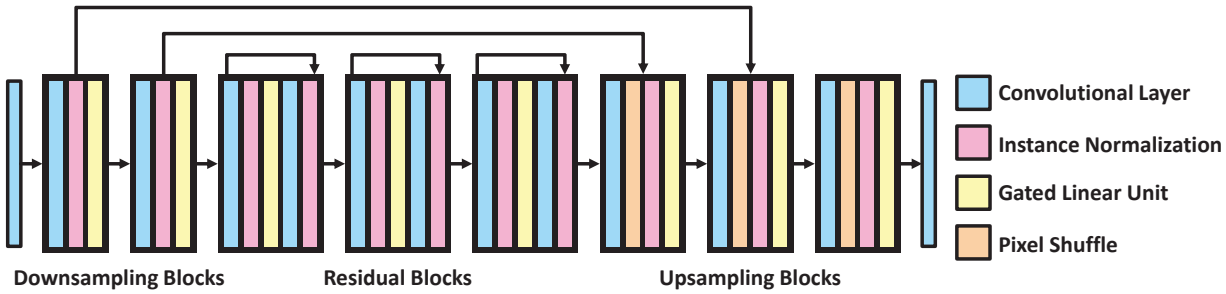
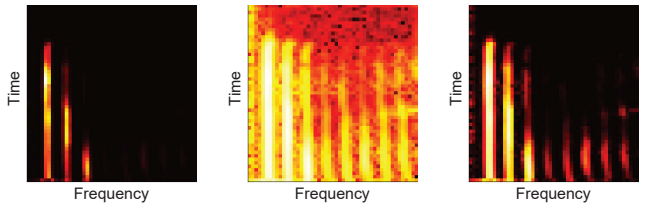


Fig. 3: The spectrogram unfolding model architecture, where downsampling blocks, residual blocks and upsampling blocks are used sequentially.

(STFT) over different time windows. In doing so, an audio-to-spectrogram and a spectrogram-to-audio processes are required before and after the spectrogram unfolding. Hence, there are three corresponding steps in our model: audio-to-spectrogram, spectrogram unfolding and spectrogram-to-audio.



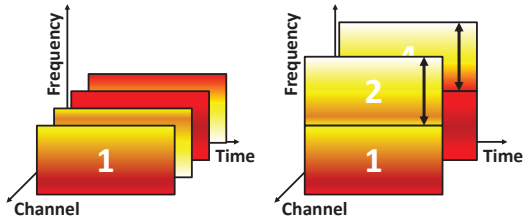
(a) Original (b) Direct Logarithm (c) Improved Logarithm

Fig. 4: The original spectrogram (a), the direct logarithm of the spectrogram (b) and the improved logarithm after adding a small constant (c).

Audio-to-spectrogram. This step transfers audio to a spectrogram by STFT. Specifically, in the audio analysis, the time window of STFT lasts for $25ms$ and the time interval between two time windows lasts for $10ms$ [7], [2].

To facilitate the spectrogram unfolding, we also need to calculate the logarithm after STFT, so that more details in the high-frequency band are presented in the spectrogram as shown in Fig. 4(a). However, directly taking the logarithm operation may amplify the noises. Specifically, the frequency bands with 0 power are sensitive to noises and after the logarithm operation, it shows small spots, as shown in Fig. 4(b). Such small spots can mislead the features to be extracted, and thus deprecate the training process afterward. To address this problem, we add a small constant to the spectrogram, which is larger than the noises but is still much smaller than the power in non-zero areas. As a result, this small constant can effectively remove the impact of the noises around the 0 values. The improved spectrogram with the small constant is shown in Fig. 4(c).

Spectrogram unfolding. We build the model architecture of spectrogram unfolding based on the U-Net backbone [14], [9], as shown in Fig. 3. There are three types of blocks in a sequence: the downsampling block, the residual block and the upsampling block. During the inference process, the input spectrogram's time dimension is always maintained; the frequency dimension is downscaled by 2 in the downsampling block and upscaled by 2 in the upsampling block. Since the output sampling rate is always greater than the input sampling rate, the number of the upsampling blocks is greater than that of the downsampling blocks. Note that the number of blocks in Fig. 3 is not the actual numbers we use. The block numbers vary for unfolding tasks with different input and output sampling rates. Within each block, we use the instance normalization (IN) [16] and gated linear unit (GLU) [4], both of which are commonly used in time series based model architectures.



(a) Before (b) After

Fig. 5: The pixel shuffle layer's working principle in our framework. The even number of the channels are flipped over the frequency dimension.

In an upsampling block, the upscale process is performed by a pixel shuffle layer, which reduces the channel dimension by half and doubles the frequency dimension. Different from the pixel shuffle layer in the traditional audio super-resolution, we develop a new pixel shuffle layer for audio frequency unfolding. As shown in Fig. 5, when unfolding, the even number of the old spectrograms are flipped around the frequency dimension. Such improvement is based on the famous signal processing theorem that the odd number of the spectrums are unchanged and the even number of the spectrums are flipped when the frequency aliasing occurs.

Spectrogram-to-audio. The last step is to transfer the reconstructed spectrogram to a high sampling rate audio. Since the phase is discarded in the spectrogram, we adopt an iterative

algorithm, fast Griffin-Lim algorithm (GLA) [13], to estimate the audio. In addition, the Hann window is exploited on each time window to improve the perceptual quality.

B. Loss Function

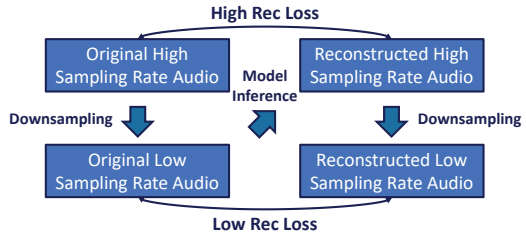


Fig. 6: The pipeline for the cyclic reconstructed loss calculation.

To train the model, we examine the discrepancy between the reconstructed audio and the original audio. In detail, there are two losses to measure such the discrepancy: a reconstructed loss and a perceptual loss. The reconstructed loss directly compares the two spectrograms; the perceptual loss is higher-level, which compares the acoustic features difference. We adopt the mean squared error (MSE) to measure the discrepancy.

The training of the reconstructed loss is easier than the perceptual loss. Thus, in the training progress, we weigh more on the reconstructed loss at first and reduce its weight gradually. On the other hand, the weight of the perceptual loss starts from a small value and increases over time.

Reconstructed loss. The pipeline of the reconstructed loss is cyclic, shown in Fig. 6. There are three steps in this pipeline: downsampling of the original audio, model inference and downsampling of the reconstructed audio. Given an original high sampling rate audio from the dataset, we generate its corresponding low sampling rate audio by downsampling in the time domain. The model reconstructs a high sampling rate audio using this downsampled audio. Finally, we demonstrate downsampling on the reconstructed high sampling rate audio to another low sampling rate audio. Since the last step of the model inference is the fast Griffin-Lim algorithm, which is iterative and hence non-differentiable, we bypass this step by directly performing downsampling on the spectrogram. So far, we have a pair of high sampling rate audios and a pair of low sampling rate audios.

There is a high reconstructed loss and a low reconstructed loss within the reconstructed loss, which are derived from the discrepancy of the pair of the high sampling rate audios and the pair of the low sampling rate audios, respectively. Generally, the weight of the high reconstructed loss is larger, which ensures the similarity between the reconstructed and the original audios. Also, the existence of the low reconstructed loss constraints the reconstructed audios to the input, avoiding the model outputs random audio irrelevant to the input.

Perceptual loss. The perceptual loss compares the discrepancy of acoustic features, filter banks (FBank), extracted from the original audios and the reconstructed audios. Note that without being cyclic, we only calculate the perceptual loss between the pair of the two high sampling rate audios. This is because the frequency dimension in the spectrograms of the low sampling rate audios is too coarse to extract precise FBank.

The introduction of the perceptual loss is non-trivial for two reasons. The first reason is based on an observation: human ears are only sensitive to acoustic features instead of detailed spectrograms. In other words, there exist two different spectrograms, with closed acoustic features, that sound similar for human ears. The usage of the reconstructed loss alone may constraint the reconstructed audios to the original audios' spectrograms, leading to model over-fitting. Replacing reconstructed loss by the high-level perceptual loss can effectively address this problem. In addition, this perceptual loss can also benefit the ASR systems, improving their recognition accuracy. Before inferring their model, most of the ASR systems extract acoustic features of audio, such as FBank, Mel-frequency cepstral coefficients (MFCC). Most of these features can be derived from the FBank. That is to say, the perceptual loss does not only minimize the FBank distance, but also ensures other acoustic features to be closed. With the closed input features to ASR systems, the recognition results tend to be the same. As a result, the reconstructed audios can be recognized by ASR systems at high accuracy.

IV. Evaluation

We implement our framework to reconstruct the low sampling rate audios from the dataset, AudioMNIST [1], where there are spoken 10 digits. The framework is evaluated by a typical ASR system, connectionist temporal classification (CTC) [7].

A. Experiment Setup

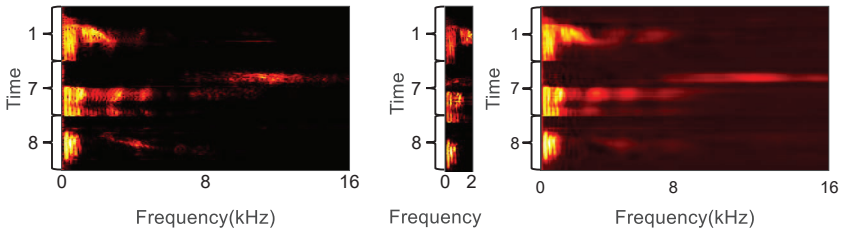
In this section, we explain the details of the dataset and the hardware where our model is trained and evaluated.

Dataset. We adopt the dataset, AudioMNIST [1] to train our model. Specifically, AudioMNIST contains 30000 audios of 0.5 second, which stands for spoken 10 digits (from 0 to 9). There are 60 speakers, including 48 males and 12 females, whose native languages include English, German, Chinese and Spanish. The original sampling rate of these audios is 48 kHz and we pre-downsample it to 16 kHz as the high sampling rate audios. We randomly split the 60 speakers into 50 with 25000 audios as the training set and 10 with 5000 audios as the evaluating set.

Hardware. We deploy our framework on the GPU, TITAN RTX with 24220 MBs. We train the model for 200 epochs, spending 140 minutes. The inference progress spends no more than 0.224 second to reconstruct audio with 0.5 second, revealing the feasibility of real-time inference.

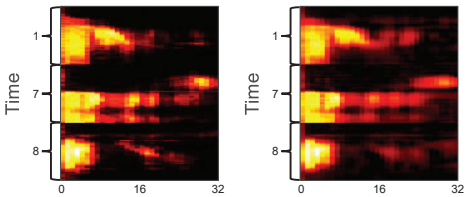
B. Evaluation Metric

Our framework is evaluated by the recognition accuracy of an ASR system. We do not adopt the metrics in existing works, such as signal-to-noise ratio (SNR) and log-spectral



(a) Original (b) Input (c) Reconstructed

Fig. 7: The original (a), the input (b) and the reconstructed (c) spectrograms for digits 1, 7 and 8 in a sequence.



(a) Original (b) Reconstructed

Fig. 8: The original (a) and the reconstructed (b) FBank over time for digits 1, 7 and 8 in a sequence.

distance (LSD) [8]. As we state in Sec. III-B, as long as the acoustic features are closed, the two audios with totally different waveforms or spectrograms may sound similar for human ears, both of which can be considered as effective outputs. In our evaluation, we adopt a typical ASR system, connectionist temporal classification (CTC) [7]. In our CTC implementation, we first extract the acoustic feature, FBank, from audios, followed by two convolutional layers, two long short-term memory (LSTM) layers and one fully connected layer. We train the ASR system for 500 epochs and it achieves the accuracy of over 99% on AudioMNIST's high sampling rate audios (16 kHz).

C. Overall Performance

We first show the evaluation results at a default setup, where we reconstruct the sampling rate from 2 kHz to 16 kHz. We set the dimension of FBank as 40 for the sampling rate of 16 kHz. We first show the details of reconstructing three audio samples, including the spectrograms and the acoustic features. Then, we show the recognition accuracy of this default setup.

Spectrogram and feature reconstruction. We pick three audio samples of the digits 1, 7 and 8 and show their original, input and reconstructed spectrograms in Fig. 7. The overall shapes of the spectrograms are similar at both the high-frequency band and the low-frequency band. However, the original spectrograms show more details while those details in the reconstructed spectrograms are blurred.

We also show the acoustic features, FBank, over time of the same audio samples in Fig. 8. Since the perceptual loss only takes FBanks into account and dominates the loss in the last several training epochs, the original features and the reconstructed features are very similar. Furthermore, different from the spectrograms, FBanks are more smooth, especially in the high-frequency band. Such smoothness makes the reconstructed spectrograms ignore the details in the original spectrograms and becomes blurred.

Recognition accuracy. The overall recognition accuracy of the default setup is 77.1% and the confusion matrix is shown in Fig. 9. According to the confusion matrix, we can find the accuracy varies over digits. Many digits, such as 1, 3 and 5, reach accuracy over 99% while some digits, such as 6 and 7,

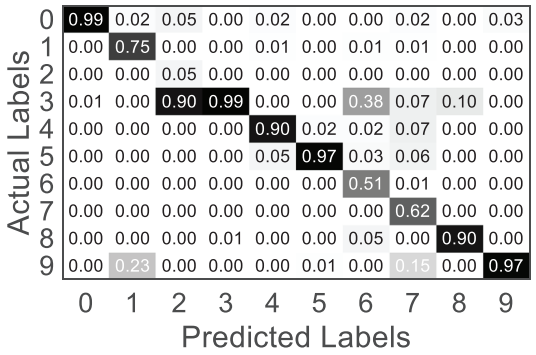
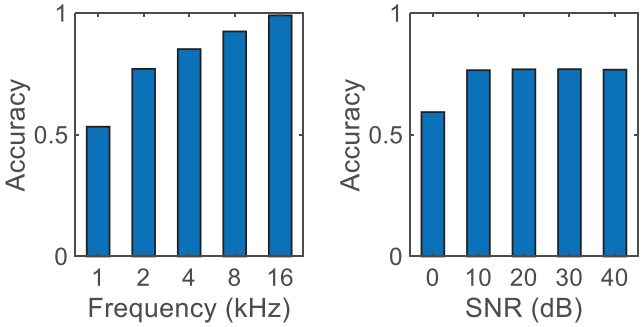


Fig. 9: The confusion matrix of the default setup, where we reconstruct the sampling rate from 2 kHz to 16 kHz.

have relatively low accuracy. Interestingly, almost all the digits 2 are mistaken as digit 3, which means the folded spectrogram of digit 2 is similar to that of digit 3. This illustrates the fact that when the frequency aliasing occurs, it is possible that two different sounds appear to be the same at a low sampling rate.

D. Impact of Factors



(a) Sampling Rate (b) SNR

Fig. 10: The recognition accuracy comparison over different factors: the input sampling rate (a) and the SNR of the input audios (b).

Input sampling rate. The sampling rate of the input audios is an important factor in our framework. This factor determines the lower bound of a sensor's sampling rate. In our evaluation, we fix the original sampling rate as 16 kHz and test our framework with the input sampling rate ranging from 1 kHz to 16 kHz. As shown in Fig. 10(a), the recognition accuracy maintains high when the input sampling rate is higher than 2 kHz. However, when the input sampling rate is 1 kHz, the spectrograms suffer serious frequency folding and the reconstructed audio can be partially recognized (53.3%). That is to say, the sampling rate of the sensor should be larger than 2 kHz. Moreover, when the input sampling rate reaches 16 kHz, which is exactly the sampling rate of the original audios, the recognition accuracy reaches 99.1%, closed to ASR's recognition accuracy of the original audios.

Signal-noise ratio. So far, the input audios are directly downsampled from the audios in the dataset, which are recorded in a quiet room with subtle background noises. However, it is not practical in reality: background noises are inevitably added to the input audios. Hence, we add background noises to the input audios and evaluate our framework's robustness against different signal-to-noise (SNR) levels. We try different SNR levels from 40dB to 0dB. As Fig. 10(b) shows, our framework maintains good recognition accuracy as long as the SNR is greater than 10 dB.

V. Conclusion

In this paper, we present a frequency unfolding framework that reconstructs the audio collected by the ultra-low sampling rate sensors. The proposed framework utilizes a U-Net-based model architecture and reconstructs the audios of better perceptual quality. Instead of the metric SNR or LSD, we adopt the recognition accuracy of the ASR system to evaluate the reconstructed audios by our framework. Exhaustive experiments show the reconstructed audios from 2 kHz to 16 kHz can be recognized by the ASR system at an accuracy of 77.1%. Our work enriches the usage of the sensors with low sampling rates, moving one step closer to the real-life adoption of ultra-low-power sensors.

Acknowledgement

This work was supported by NSF IIS-2140247, CCF-1822085 IUCRC for ASIC and membership from Ergomotion, Bayland Scientific.

References

[1] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek. Interpreting and explaining deep neural networks for classification of audio signals. *arXiv preprint arXiv:1807.03418*, 2018.

[2] C.-C. Chiu and C. Raffel. Monotonic chunkwise attention. *arXiv preprint arXiv:1712.05382*, 2017.

[3] Y. Cho, M. Chang, S. Lee, H. Lee, G. J. Kim, and J. Choo. Efficient adversarial audio synthesis via progressive upsampling. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3410–3414. IEEE, 2021.

[4] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.

[5] Y. Gao, Y. Jin, J. Chauhan, S. Choi, J. Li, and Z. Jin. Voice in ear: Spoofing-resistant and passphrase-independent body sound authentication. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–25, 2021.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[8] A. Gray and J. Markel. Distance measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(5):380–391, 1976.

[9] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6820–6824. IEEE, 2019.

[10] V. Kuleshov, S. Z. Enam, and S. Ermon. Audio super-resolution using neural nets. In *ICLR (Workshop Track)*, 2017.

[11] T. Y. Lim, R. A. Yeh, Y. Xu, M. N. Do, and M. Hasegawa-Johnson. Time-frequency networks for audio super-resolution. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 646–650. IEEE, 2018.

[12] Y. Michalevsky, D. Boneh, and G. Nakibly. Gyrophone: Recognizing speech from gyroscope signals. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 1053–1067, 2014.

[13] N. Perraudin, P. Balazs, and P. L. Søndergaard. A fast griffin-lim algorithm. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013.

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] S. Shen, D. Chen, Y.-L. Wei, Z. Yang, and R. R. Choudhury. Voice localization using nearby wall reflections. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[16] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[17] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. i. theory. *IEEE transactions on signal processing*, 41(2):821–833, 1993.

[18] T. Wang, S. Yao, S. Liu, J. Li, D. Liu, H. Shao, R. Wang, and T. Abdelzaher. Audio keyword reconstruction from on-device motion sensor signals via neural frequency unfolding. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–29, 2021.

[19] W. Wang, J. Li, Y. He, and Y. Liu. Symphony: localizing multiple acoustic sources with a single microphone array. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 82–94, 2020.

[20] E. Whitaker. On the functions which are represented by the expansion of interpolating theory. In *Proc. Roy. Soc. Edinburgh*, 1915.

[21] K. Zhang, Y. Ren, C. Xu, and Z. Zhao. Wsrglow: A glow-based waveform generative model for audio super-resolution. *arXiv preprint arXiv:2106.08507*, 2021.