INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

NU-Wave: A Diffusion Probabilistic Model for Neural Audio Upsampling

Junhyeok Lee[1]*, Seungu Han*[1,2]
[1]MINDs Lab Inc., Republic of Korea
[2]Seoul National University, Republic of Korea
{jun3518, hansw0326}@mindslab.ai

Abstract

In this work, we introduce *NU-Wave*, the first neural audio upsampling model to produce waveforms of sampling rate 48kHz from coarse 16kHz or 24kHz inputs, while prior works could generate only up to 16kHz. NU-Wave is the first diffusion probabilistic model for audio super-resolution which is engineered based on neural vocoders. NU-Wave generates high-quality audio that achieves high performance in terms of signal-to-noise ratio (SNR), log-spectral distance (LSD), and accuracy of the ABX test. In all cases, NU-Wave outperforms the baseline models despite the substantially smaller model capacity (3.0M parameters) than baselines (5.4-21%). The audio samples of our model are available at https://mindslab-ai.github.io/nuwave, and the code will be made available soon.

Index Terms: diffusion probabilistic model, audio super-resolution, bandwidth extension, speech synthesis

1. Introduction

Audio super-resolution, neural upsampling, or bandwidth extension is the task of generating high sampling rate audio signals with full frequency bandwidth from low sampling rate signals. There have been several works that applied *deep neural networks* to audio super-resolution [1, 2, 3, 4, 5, 6, 7]. Still, prior works used 16kHz as the target frequency, which is not considered a high-resolution. Since the highest audible frequency of human is 20kHz, high-resolution audio used in multimedia such as movies or musics uses 44.1kHz or 48kHz.

Similar to recent works in the image domain [8, 9, 10], audio super-resolution works also used deep generative models [4, 5], mostly focused on generative adversarial networks (GAN) [11]. On the other hand, prior works in neural vocoders, which is inherently similar to audio super-resolution due to its local conditional structure, adopted a variety of generative models such as autoregressive models [12, 13], flow-based models [14, 15], and variational autoencoders (VAE) [16]. Recently, diffusion probabilistic models [17, 18] were applied to neural vocoders [19, 20], with remarkably high perceptual quality. Details of diffusion probabilistic models and their applications are explained in Section 2.1 and 2.2.

In this paper, we introduce *NU-Wave*, a conditional diffusion probabilistic model for neural audio upsampling. Our contributions are as follows:

1. NU-Wave is the first deep generative neural audio upsampling model to synthesize waveforms of sampling rate 48kHz from coarse 16kHz or 24kHz inputs.
2. We adopted diffusion probabilistic models for the audio upsampling task for the first time by engineering neural vocoders based on diffusion probabilistic models.
3. NU-Wave outperforms previous models [2, 5] including the GAN-based model in both quantitative and qualitative metrics with substantially smaller model capacity (5.4-21% of baselines).

2. Related works

2.1. Diffusion probabilistic models

Diffusion probabilistic models (diffusion models for brevity) are trending generative models [17, 18], which apply iterative Monte-Carlo Markov chain sampling to generate complex data from simple noise distribution such as normal distribution. Markov chain of a diffusion model consists of two processes: the *forward process*, and the *reverse process*.

The *forward process*, also referred to as the *diffusion process*, gradually adds Gaussian noise to obtain whitened latent variables $y_1, y_2, \dots, y_T \in \mathbb{R}^L$ from the data $y_0 \in \mathbb{R}^L$, where L is time length of the data. Unlike other generative models, such as GAN [11] and VAE [21], the diffusion model's latent variables have the same dimensionality as the input data. Shol-Dickstein *et al.* [17] applied a fixed noise variance schedule $\beta_{1:T} := [\beta_1, \beta_2, \dots, \beta_T]$ to define the forward process $q(y_{1:T}|y_0) := \prod_{t=1}^{T} q(y_t|y_{t-1})$ as the multiplication of Gaussian distributions:

$$q(y_t|y_{t-1}) := \mathcal{N}\left(y_t; \sqrt{1-\beta_t}\, y_{t-1}, \beta_t I\right). \tag{1}$$

Since the sum of Gaussian distributions is also a Gaussian distribution, we can write y_t, the latent variable at a timestep t, as $q(y_t|y_0) := \mathcal{N}\left(y_t; \sqrt{\bar{\alpha}_t}\, y_0, (1-\bar{\alpha}_t)I\right)$, where $\alpha_0 := 1$, $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. $\beta_{1:T}$ are predefined hyperparameters to match the forward and the reverse distributions by minimizing $D_{KL}(q(y_T|y_0)||\mathcal{N}(0, I))$. Ho *et al.* [18] set these values as a linear schedule of $\beta_{1:T} = \text{Linear}(10^{-4}, 0.02, 1000)$.

The *reverse process* is defined as the reverse of the diffusion where the model, which is parametrized by θ, learns to estimate the added Gaussian noise. Starting from y_T, which is sampled from Gaussian distribution $p(y_T) = \mathcal{N}(y_T; 0, I)$, the reverse process $p_\theta(y_{0:T}) := p(y_T) \prod_{t=1}^{T} p_\theta(y_{t-1}|y_t)$ is defined as the multiplication of transition probabilities:

$$p_\theta(y_{t-1}|y_t) := \mathcal{N}\left(y_{t-1}; \mu_\theta(y_t, t), \sigma_t^2 I\right), \tag{2}$$

where $\mu_\theta(y_t, t)$ and σ_t^2 are the model estimated mean and variance of y_{t-1}. Ho *et al.* suggested to set variance as timestep dependant constant. Latent variable y_{t-1} is sampled from the distribution $p(y_{t-1}|y_t)$ where ϵ_θ is model estimated noise and $z \sim \mathcal{N}(0, I)$:

$$y_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(y_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon_\theta(y_t, t)\right) + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t} z. \tag{3}$$

Diffusion models approximate the probability density of data $p_\theta(y_0) := \int p_\theta(y_{0:T}) dy_{1:T}$. The training objective of the diffusion model is minimizing the variational bound on negative log-likelihood without any auxiliary losses. The variational bound of the diffusion model is represented as KL divergence of the forward process and the reverse process. Ho *et al.* [18]

http://dx.doi.org/10.21437/Interspeech.2021-36

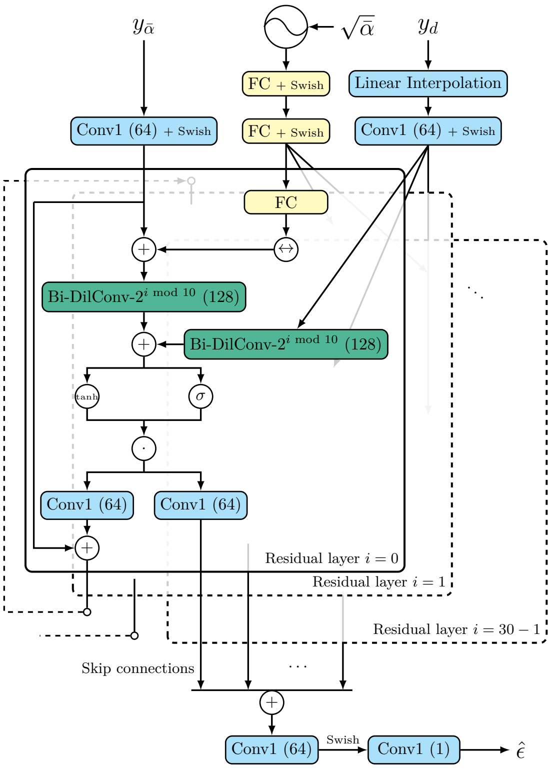

Figure 1: *The network architecture of NU-Wave. Noisy speech $y_{\bar{\alpha}}$, downsampled speech y_d, and noise level $\sqrt{\bar{\alpha}}$ are inputs of model. The model estimates noise $\hat{\epsilon}$ to reconstruct y_0 from $y_{\bar{\alpha}}$.*

reparametrized the variational bound to a simple loss which is connected with denoising score matching and Langevin dynamics [22, 23, 24]. For a diffused noise $\epsilon \sim \mathcal{N}(0, I)$, the loss resembles denoising score matching:

$$\mathbb{E}_{t,y_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\,y_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon, t\right)\right\|_2^2\right]. \tag{4}$$

2.2. Conditional diffusion models as a neural vocoder

Neural vocoders adopted diffusion models [19, 20] by modifying them to a conditional generative model for x, where x is a local condition like mel-spectrogram. The probability density of a conditional diffusion model can be written as $p_\theta(y_0|x) := \int p_\theta(y_{0:T}|x)dy_{1:T}$. Furthermore, Chen *et al.* [20] suggested training the model with the continuous noise level $\sqrt{\bar{\alpha}}$, which is uniformly sampled between adjacent discrete noise levels $\sqrt{\bar{\alpha}_t}$ and $\sqrt{\bar{\alpha}_{t-1}}$, instead of the timestep t. Continuous noise level training allows different noise schedules during training and sampling, while a discrete timestep trained model is needed the same schedule. Along with the continuous noise level condition, Chen *et al.* [20] replaced the L_2 norm of Eq.(4) to L_1 norm for empirical training stability.

3. Approach

3.1. Network architecture

In this section, we introduce NU-Wave, a conditional diffusion model for audio super-resolution. Our method directly generates waveform from waveform without feature extraction such as short-time Fourier transform spectrogram or phonetic posteriorgram. Figure 1 illustrates the architecture of the NU-Wave. Based on the recent success of diffusion-based neural vocoders *DiffWave* [19] and *WaveGrad* [20], we engineer their architectures to fit audio super-resolution task. To build a downsampled signal conditioned model, NU-Wave model $\epsilon_\theta : \mathbb{R}^L \times \mathbb{R}^{L/r} \times \mathbb{R} \to \mathbb{R}^L$ takes the diffused signal $y_{\bar{\alpha}} := \sqrt{\bar{\alpha}}\,y_0 + \sqrt{1-\bar{\alpha}}\,\epsilon$, the downsampled signal y_d and the noise level $\sqrt{\bar{\alpha}}$ as the inputs and its output $\hat{\epsilon} := \epsilon_\theta(y_{\bar{\alpha}}, y_d, \sqrt{\bar{\alpha}})$ estimates diffused noise ϵ. Similar to DiffWave$_{\text{BASE}}$ [19], our model has $N = 30$ residual layers with 64 channels. In each residual layer, bidirectional dilated convolution (*Bi-DilConv*) with kernel size 3 and dilation cycle $[1, 2, \ldots, 512] \times 3$ is used. Noise level embedding and local conditioner are added before and after the main Bi-DilConv to provide conditions of $\sqrt{\bar{\alpha}}$ and y_d. After similar parts, there are two main modifications from the original DiffWave architecture: noise level embedding and local conditioner.

Noise level embedding. DiffWave includes an integer timestep t in the inputs to indicate a discrete noise level [19]. On the other hand, inputs of WaveGrad contain a continuous noise level $\sqrt{\bar{\alpha}}$ to have a different number of iterations during inference [20]. We implement the noise level embedding using 128-dimensional vector similar to sinusoidal positional encoding introduced in Transformer [25]:

$$\begin{aligned} E(\sqrt{\bar{\alpha}}) = \Big[&\sin\left(10^{-[0:63]\times\gamma} \times C\sqrt{\bar{\alpha}}\right), \\ &\cos\left(10^{-[0:63]\times\gamma} \times C\sqrt{\bar{\alpha}}\right)\Big]. \end{aligned} \tag{5}$$

We set a linear scale with $C = 50000$ instead of 5000 introduced by Chen *et al.* [20], because the embedding with $C = 5000$ cannot effectively distinguish among the low noise levels. γ in noise level embedding is set to 1/16. The embedding is fed to the two shared fully connected (FC) layers and a layer-specific FC layer and then added as a bias term to the input of each residual layer.

Local conditioner. We modified the local conditioner to fit the audio upsampling task. After various experiments, we discovered that the receptive field for the conditional signal y_d needs to be larger than the receptive field for the noisy input $y_{\bar{\alpha}}$. For each residual layer, we apply another Bi-DilConv with kernel size 3 for the conditioner and the same dilation cycle as the main Bi-DilConv to make the receptive field of y_d nearly twice as large as that of $y_{\bar{\alpha}}$. We hypothesize that having a large receptive field for the local conditions provides more useful information for upsampling. While DiffWave upsampled the local condition with transposed convolution [19], we adopt linear interpolation to build a single neural network architecture that could be utilized for different upscaling ratios.

3.2. Training objective

We discovered that the L_1 norm scale of small timesteps and large timesteps differ by a factor of almost 10. Substituting the L_1 norm to log-norm offers stable training results by scaling the losses from different timesteps. Besides, audio super-resolution tasks need to substitute conditional factor from mel-spectrogram x to downsampled signal y_d as:

$$\mathbb{E}_{\bar{\alpha},y_0,y_d,\epsilon}\left[\log\left\|\epsilon - \epsilon_\theta\left(y_{\bar{\alpha}}, y_d, \sqrt{\bar{\alpha}}\right)\right\|_1\right]. \tag{6}$$

Algorithm 1 illustrates the training procedure for NU-Wave which is similar to the continuous noise level training suggested by Chen *et al.* [20]. The details of the filtering process are described in Section 4.5. During sampling, we could utilize an untrained noise schedule with a different number of iterations by using continuous noise level training. Thus, $T, \alpha_{1:T}, \beta_{1:T}$ in Algorithm 1 and Algorithm 2 need not be identical.

Algorithm 1 Training.

```
repeat
    $y_0 \sim q(y_0)$
    $y_d = \text{Subsampling}(\text{Filtering}(y_0), r)$
    $t \sim \text{Uniform}(\{1, 2, \dots, T\})$
    $\sqrt{\bar{\alpha}} \sim \text{Uniform}(\sqrt{\bar{\alpha}_t}, \sqrt{\bar{\alpha}_{t-1}})$
    $\epsilon \sim \mathcal{N}(0, I)$
    Take gradient descent step on
      $\nabla_\theta \log \left\| \epsilon - \epsilon_\theta \left( \sqrt{\bar{\alpha}}\, y_0 + \sqrt{1 - \bar{\alpha}}\, \epsilon, y_d, \sqrt{\bar{\alpha}} \right) \right\|_1$
until converged
```

Algorithm 2 Sampling.

```
$y_T \sim \mathcal{N}(0, I)$
for $t = T, T-1, \dots, 1$ do
    $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
    $\sigma_t = \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t}$
    $y_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( y_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(y_t, y_d, \sqrt{\bar{\alpha}_t}) \right) + \sigma_t z$
end for
return $\hat{y} = y_0$
```

3.3. Noise schedule

We use linear noise schedule $\text{Linear}(1 \times 10^{-6}, 0.006, 1000)$ during training following the prior works on diffusion models [18, 19, 20]. We adjust the minimum and the maximum values from the schedule of WaveGrad ($\text{Linear}(1 \times 10^{-4}, 0.005, 1000)$) [20] to train the model with a large range of noise levels. Based on empirical results, $\sqrt{\bar{\alpha}_T}$ should be smaller than 0.5 to generate clean final output y_0. During inference, the sampling time is proportional to a number of iterations T, thus we reduced it for fast sampling. We tested several manual schedules and found that 8 iterations with $\beta_{1:8} = [10^{-6}, 2 \times 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 9 \times 10^{-1}]$ are suitable for our task. Besides we did not report the test results with 1000 iterations as they are similar to that of 8 iterations.

4. Experiments

4.1. Dataset

We train the model on the VCTK dataset [26] which contains 44 hours of recorded speech from 108 speakers. Only the mic1 recordings of the VCTK dataset are used for experiments and recordings of *p280* and *p315* are excluded as conventional setup. We divide audio super-resolution task into two parts: *SingleSpeaker* and *MultiSpeaker*. For the *SingleSpeaker* task, we train the model on the first 223 recordings of the first speaker, who is labeled *p225*, and test it on the last 8 recordings. For the *MultiSpeaker* task, we train the model on the first 100 speakers and test it on the remaining 8 speakers.

4.2. Training

We train our model with the Adam [27] optimizer with the learning rate 3×10^{-5}. The *MultiSpeaker* model is trained with two NVIDIA A100 (40GB) GPUs, and the *SingleSpeaker* model is trained with two NVIDIA V100 (32GB) GPUs. We use the largest batch size that fits memory constraints, which is 36 and 24 for A100s and V100s. We train our model for two upscaling ratios, $r = 2, 3$. During training, we use the 0.682 seconds patches ($32768 - 32768 \bmod r$ samples) from the signals as the input, and during testing, we use the full signals.

4.3. Evaluation

To evaluate our results quantitatively, we measure the signal-to-noise ratio (SNR) and the log-spectral distance (LSD). For a reference signal y and an estimated signal $\hat{y}$, SNR is defined as $\text{SNR}(\hat{y}, y) = 10 \log_{10}(\|y\|_2^2 / \|\hat{y} - y\|_2^2)$. There are several works reported that SNR is not effective in the upsampling task because it could not measure high-frequency generation [2, 5]. On the other hand, LSD could measure high-frequency generation as spectral distance. Let us denote the log-spectral power magnitudes Y and $\hat{Y}$ of signals y and $\hat{y}$, which is defined as $Y(\tau, k) = \log_{10} |S(y)|^2$ where S is the short-time Fourier transform (STFT) with the Hanning window of size 2048 samples and hop length 512 samples, τ and k are the time frame and the frequency bin of STFT spectrogram. The LSD is calculated as following:

$$\text{LSD}(\hat{y}, y) = \frac{1}{\mathcal{T}} \sum_{\tau=1}^{\mathcal{T}} \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left(\hat{Y}(\tau, k) - Y(\tau, k) \right)^2}. \quad (7)$$

For qualitative measurement, we utilize the ABX test to determine whether the ground truth data and the generated data are distinguishable. In a test, human listeners are provided three audio samples called *A*, *B*, and *X*, where *A* is the reference signal, *B* is the generated signal, and *X* is the signal randomly selected between *A* or *B*. Listeners classify that *X* is *A* or *B*, and we measure their accuracy. ABX test is hosted on the Amazon Mechanical Turk system, each of *SingleSpeaker* and *MultiSpeaker* task is examined by near 500 and 1500 cases. We only tested it for the main upscaling ratio $r = 2$.

4.4. Baselines

We compare our method with linear interpolation, U-Net [28] like model suggested by Kuleshov *et al.* [2], and MU-GAN [5]. To compare with our waveform-to-waveform model, we choose these models because U-Net is a basic waveform-to-waveform model and MU-GAN is a waveform-to-waveform GAN-based model with minimal auxiliary features in audio super-resolution. We implemented and trained baseline models on same the 48kHz VCTK dataset. While U-Net model was implemented as details in Kuleshov *et al.* [2], we modified number of channels (down: $\min(2^{b+2}, 32)$, up: $\min(2^{12-b}, 128)$ for block index $b = 1, 2, ..., 8$) in MU-GAN for stable training. We early-stopped training baseline models based on their LSD metrics Each model's number of parameters is provided in Table 1, NU-Wave only requires 5.4-21% parameters than baselines. Hyperparameters of baseline models might not be fully optimized for the 48kHz dataset.

Table 1: *Comparison of model size. Our model has the smallest number of parameters as 5.4-21% of other baselines.*

	U-Net	MU-GAN	NU-Wave (Ours)
#parameter(M)↓	56	14	**3.0**

4.5. Implementation details

To downsample with in-phase, the filtering consists of STFT, fill zero to high-frequency elements, and inverse STFT (iSTFT). STFT and iSTFT use the Hanning window of size 1024 and hop size 256. The thresholds for high-frequency are the Nyquist frequency of the downsampled signals. We cut out the leading and trailing signals 15dB lower than the max amplitude.

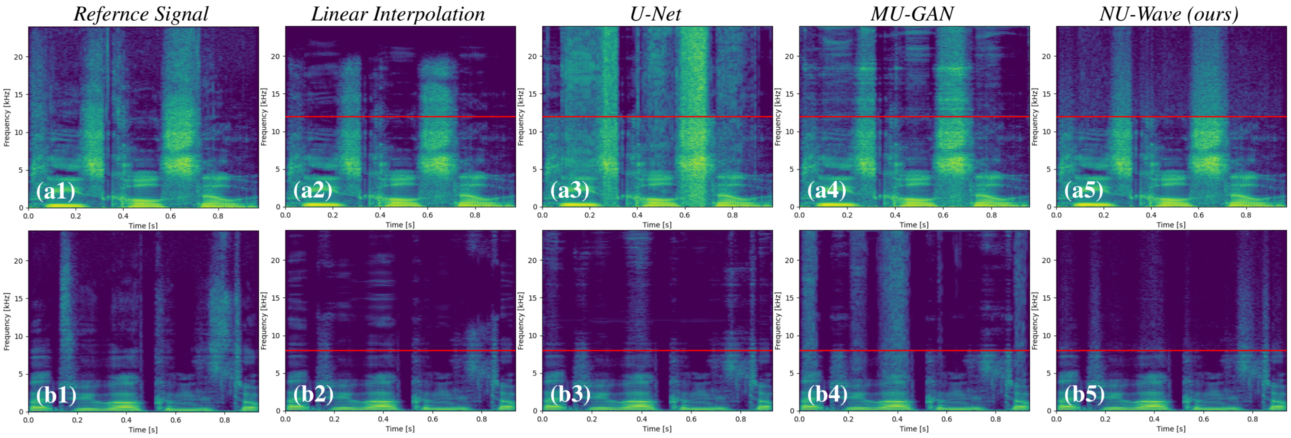

Figure 2: *Spectrograms of reference and upsampled speeches. Red lines indicate the Nyquist frequency of downsampled signals.* **(a1-a5)** *are samples of $r = 2$, MultiSpeaker (p360_001) and* **(b1-b5)** *are samples of $r = 3$, SingleSpeaker (p225_359). More samples are available at https://mindslab-ai.github.io/nuwave*

5. Results

As illustrated in Figure 2, our model can naturally reconstruct high-frequency elements of sibilants and fricatives, while baselines tend to generate that resemble low-frequency elements by flipping along with downsampled signal's Nyquist frequency.

Table 2 shows the quantitative evaluation of our model and the baselines. For all metrics, NU-Wave outperforms the baselines. Our model improves SNR value by 0.18-0.9 dB from the best performing baseline, MU-GAN. The LSD result achieves almost half ($r = 2$: 43.8-45.0%, $r = 3$: 55.6-57.3%) of linear interpolation's LSD, while the best baseline model (MU-GAN, $r = 2$, *SingleSpeaker*) only achieves 62.2%. We tested our model five times and observed that the standard deviations of the metrics are remarkably small, in the factor of 10^{-5}.

Table 2: *Results of evaluation metrics. Upscaling ratio (r) is indicated as* ×2, ×3. *Our model outperforms other baselines for both SNR and LSD.*

Model	SingleSpeaker		MultiSpeaker	
	SNR ↑	LSD ↓	SNR ↑	LSD ↓
Linear ×2	9.69	1.80	11.1	1.93
U-Net ×2	10.3	1.57	9.86	1.47
MU-GAN ×2	10.5	1.12	12.3	1.22
NU-Wave ×2	**11.1**	**0.810**	**13.2**	**0.845**
Linear ×3	8.04	1.72	8.71	1.74
U-Net ×3	8.81	1.64	10.7	1.41
MU-GAN ×3	9.44	1.37	11.7	1.53
NU-Wave ×3	**9.62**	**0.957**	**12.0**	**0.997**

Table 3: *The accuracy and the confidence interval of the ABX test. NU-Wave shows the lowest accuracy (51.2-52.1%) indicating that its outputs are indistinguishable from the reference.*

Model	SingleSpeaker	MultiSpeaker
Linear ×2	55.5 ± 0.04%	52.3 ± 0.03%
U-Net ×2	52.9 ± 0.04%	52.5 ± 0.03%
MU-GAN ×2	**52.1** ± 0.04%	**51.3** ± 0.03%
NU-Wave ×2	**52.1** ± 0.04%	**51.2** ± 0.03%

Table 3 shows the accuracy of the ABX test. Since our model achieves the lowest accuracy (52.1%, 51.2%) close to 50%, we can confidently claim that NU-Wave's outputs are almost indistinguishable from the reference signals. While it is notable to mention that the confidence interval of our model and MU-GAN is mostly overlapped, the difference in accuracy was not statistically significant.

6. Discussion

In this paper, we applied a conditional diffusion model for audio super-resolution. To the best of our knowledge, NU-Wave is the first audio super-resolution model that produces high-resolution 48kHz samples from 16kHz or 24kHz speech, and the first model that successfully utilized the diffusion model for the audio upsampling task.

NU-Wave outperforms other models in quantitative and qualitative metrics with fewer parameters. While other baseline models obtained the LSD similar to that of linear interpolation, our model achieved almost half the LSD of linear interpolation. This indicates that NU-Wave can generate more natural high-frequency elements than other models. The ABX test results show that our samples are almost indistinguishable from reference signals. However, the differences between the models were not statistically significant, since the downsampled signals already contain the information up to 12kHz which has the dominant energy within the audible frequency band. Since our model outperforms baselines for both *SingleSpeaker* and *MultiSpeaker*, we can claim that our model generates high-quality upsampled speech for both the seen and the unseen speakers.

While our model generates natural sibilants and fricatives, it cannot generate harmonics of vowels as well. In addition, our samples contain slight high-frequency noise. In further studies, we can utilize additional features, such as pitch, speaker embedding, and phonetic posteriorgram [4, 7]. We can also apply more sophisticated diffusion models, such as DDIM [29], CAS [30], or VP SDE [31], to reduce upsampling noise.

7. Acknowledgements

The authors would like to thank Sang Hoon Woo and teammates of MINDs Lab, Jinwoo Kim, Hyeonuk Nam from KAIST, and Seung-won Park from SNU for valuable discussions.

8. References

[1] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "Dnn-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *INTERSPEECH*, 2015, pp. 2578–2582.

[2] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," in *Workshop of International Conference on Learning Representations*, 2017.

[3] 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)*. IEEE, 2018, pp. 646–650.

[4] X. Li, V. Chebiyyam, K. Kirchhoff, and A. Amazon, "Speech audio super-resolution for speech recognition." in *INTERSPEECH*, 2019, pp. 3416–3420.

[5] S. Kim and V. Sathe, "Bandwidth extension on raw audio via generative adversarial networks," *arXiv preprint arXiv:1903.09027*, 2019.

[6] S. Birnbaum, V. Kuleshov, Z. Enam, P. W. W. Koh, and S. Ermon, "Temporal film: Capturing long-range sequence dependencies with feature-wise modulations." in *Advances in Neural Information Processing Systems*, 2019, pp. 10 287–10 298.

[7] N. Hou, C. Xu, V. T. Pham, J. T. Zhou, E. S. Chng, and H. Li, "Speaker and phoneme-aware speech bandwidth extension with residual dual-path network," in *INTERSPEECH*, 2020, pp. 4064–4068.

[8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[9] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[10] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "Pulse: Self-supervised photo upsampling via latent space exploration of generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2437–2445.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[12] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[13] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*, 2018, pp. 2410–2419.

[14] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[15] W. Ping, K. Peng, K. Zhao, and Z. Song, "Waveflow: A compact flow-based model for raw audio," in *International Conference on Machine Learning*, 2020, pp. 7706–7716.

[16] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *International Conference on Machine Learning*, 2020, pp. 7586–7598.

[17] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, 2015, pp. 2256–2265.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020, pp. 6840–6851.

[19] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, 2021.

[20] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *International Conference on Learning Representations*, 2021.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.

[22] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.

[23] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems*, 2019, pp. 11 918–11 930.

[24] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 12 438–12 448.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[26] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit(version 0.92)," 2016. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/3443

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[28] 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*. Springer, 2015, pp. 234–241.

[29] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[30] A. Jolicoeur-Martineau, R. Piché-Taillefer, R. T. d. Combes, and I. Mitliagkas, "Adversarial score matching and improved sampling for image generation," in *International Conference on Learning Representations*, 2021.

[31] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021.