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

X-net: A Joint Scale Down and Scale Up Method for Voice Call

Liang Wen[1], *Lizhong Wang*[1], *Xue Wen*[1], *Yuxing Zheng*[1], *Youngo Park*[2], *Kwang Pyo Choi*[2]

[1] Samsung Research China-Beijing (SRC-B), Samsung Electronics, China
[2] Samsung Research, Samsung Electronics, South Korea

{liang001.wen,lz.wang,xue.wen,yxing.zheng,youngo.park,kp5.choi}@samsung.com

Abstract

This paper proposes X-net, a jointly learned scale-down and scale-up architecture for data pre- and post-processing in voice calls, as a means to bandwidth extension over band-limited channels. Scale-down and scale-up are deployed separately on transmitter and receiver to perform down- and upsampling. Separate supervisions are used on the submodules so that X-net can work properly even if one submodule is missing. A two-stage training method is used to learn X-net for improved perceptual quality. Results show that jointly learned X-net achieves promising improvement over blind audio super-resolution by both objective and subjective metrics, even in a lightweight implementation with only 1k parameters.

Index Terms: bandwidth extension, audio super-resolution, deep learning

1. Introduction

Audio bandwidth extension (BWE), or audio super-resolution, generates high-bandwidth audio from low-bandwidth audio. A typical use of BWE is for post-processing audio received over a band-limited channel to reconstruct missing high-frequency content. Most mainstream voice call codecs nowadays work under the WB (wideband speech) setting where speech is encoded up to 8kHz. Since this is well below the 20kHz human auditory limit, an BWE backend can be employed to help deliver higher clarity and overall perceived quality.

Blind audio bandwidth extension is a hassle-free setup to do BWE, where "blind" refers to BWE module seeing only the low-bandwidth input, but nothing of the upstream processing pipeline. This blind setup decouples BWE from upstream modules, therefore improves modularity and simplifies design choices. On the downside, it loses the chance to coordinate individual modules in a joint effort to boost performance. For example, the upsampling mechanism in blind BWE will generally not optimally match the downsampling mechanism across the channel.

This paper proposes a time-domain lightweight scale-down and scale-up design, codenamed X-net, for BWE over a low-bandwidth voice channel. It features jointly optimized neural networks for downsampling and upsampling, deployed as pre- and post-processing blocks to the codec. X-net assumes a U-net[1]-like architecture and forms an autoencoder [2] on top of it, therefore admits self supervision. A two-stage training plan is used to train X-net with time-domain and frequency-domain losses in tandem. Experimental results show X-net outperforms blind BWE under the same scale-up model by a large margin. With only 1k parameters in total, X-net achieves significant MOS (mean opinion score) improvement in EVS (Enhanced Voice Services) WB call environment [3].

2. Related work

Image super-resolution with deep learning

Super-resolution is a regression task to recover or generate high resolution image/video from low resolution image/video. This area has seen non-trivial growth in recent years [4], mostly thanks to advancements in deep learning [5]. BWE can be seen as the audio version of super-resolution. In this paper we adapt U-net [1], a successful deep architecture for super-resolution, to the job of BWE.

Frequency-domain BWE

Estimation of high-frequency features was traditionally based on the classical source-filter model of speech production, often implemented by codebooks [6][7], Gaussian mixture models (GMM)[8][9], or hidden Markov models (HMM)[10][11]. Li and Lee [12] appeared to be the first work that used deep neural networks (DNN) to address high-frequency feature prediction. Their network predicted high-bandwidth log power spectrum (LPS) from low-bandwidth LPS. Abel and Fingscheidt [13] proposed another frequency-domain method using a DNN to estimate a low-dimension cepstral representation of target high-bandwidth speech. EURECOM published a series of works [14][15][16] using autoencoders and adversarial learning for bandwidth extension. They worked with normalized high-frequency features like linear prediction cepstral coefficients (LPCCs).

Frequency-domain methods are typically associated with an inherent algorithmic delay due to block-wise processing. Care must be taken to manage such delay in real-time use cases like voice calls.

Time-domain BWE

Kuleshov et al. [17][18] proposed time-domain end-to-end audio super-resolution. It worked by predicting an additive correction term to be applied on an interpolated version of the low-bandwidth input. The underlying network was similar to U-net, with additional scaling layers for extra temporal context awareness. It was trained to minimize mean-square error (MSE) loss of the waveform. Their particular implementations are not suitable for BWE in voice calls due to non-streaming decoding and overall complexity.

Mixed time-frequency domain BWE

Time-domain MSE falls short in capturing perceptual quality of humans, which is better quantized in frequency domain. Lim et al. [19] proposed a time-frequency network (TFNet) that did BWE in both time- and frequency-domain branches then combined their predictions by spectral amplitude. Both branches of TFNet were supervised under the same time-domain MSE on combined output. On the other hand, Wang et al. [20][21] proposed to supervise time-domain BWE under both time- and frequency-domain objectives. While both

http://dx.doi.org/10.21437/Interspeech.2021-812

schemes were shown effective, they did require tuning extra hyper parameters to balance between the branches.

BWE in voice call

EVS WB, AMR WB and Opus WB are widely adopted wideband codecs as of today. The former two are common among telephone networks e.g. VoLTE (Voice over Long-Term Evolution), the latter more common in over-the-top (OTT) applications e.g. Google duo. During a voice call, the transmitter down-samples full-band voice from the microphone to 16kHz so that it matches the WB codecs.

Existing BWE methods for voice calls are implemented on the receiver side [19], and generally assume the "blind" setup of not knowing or controlling how transmitter works. In modern voice call systems, however, it is easy to exchange side information between parties so that non-blind BWE setup is indeed practical. In this paper we explore a setup where downsampling and BWE submodules are designed as a pair. This allows jointly optimizing both parts to improve received voice quality.

3. Proposed method

3.1. Framework

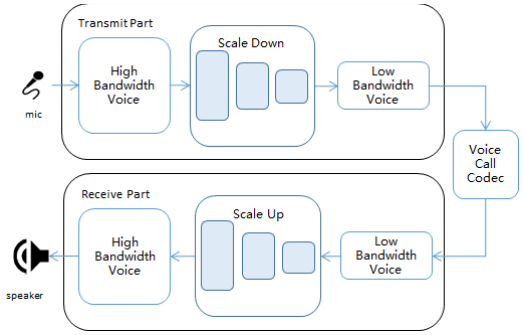

Figure 1. *Working environment of X-net*

Figure 1 shows the proposed X-net architecture in voice call context. It contains a scale-down module deployed with the transmitter, and a scale-up module deployed with the receiver. Given high-bandwidth voice signal y, the scale-down module down-samples it to low-bandwidth voice z that suits the codec:

$$z = \text{scale\_down}(y) \tag{1}$$

The scale-up module converts low-bandwidth codec output $\hat{z}$ back to high-bandwidth voice $\hat{y}$, i.e. it does BWE:

$$\hat{y} = \text{scale\_up}(\hat{z}) \tag{2}$$

The codec is a standard, fixed module that transmits low-bandwidth voice data. Scale-down, codec, and scale-up modules together form an autoencoder, with the codec at the bottleneck.

3.2. Network details

Both scale-down and scale-up operate in time domain. Scale-down contains stacked downsampling layers, each being 1D convolution without bias or nonlinearity. This makes scale-down a linear time-invariant downsampler but not necessarily an alias-free one. Scale-up contains stacked upsampling layers, each being 1D convolution with swish [23] nonlinearity. Inputs are duplicated in time before the scale-up convolutions. All scale-down and scale-up convolutions are causal, i.e. no look-ahead is used to compute each output point. Scale-down and scale-up together form a U-net-like construct similar to [17], but without skip connections in between. Consequently, intermediate down- and upsampling layers are not required to have matched feature map layouts.

Table 1 shows a typical lightweight CNN configuration of X-net designed for voice calls, where f, k and s denote the number of filters, kernel size and stride, respectively. We use the same number of filters and kernel size in corresponding layers of scale-down and scale-up.

Table 1. *X-net parameter setting*

	Scale Down	**Scale Up**
Layer 1	f=16, k=16, s = 1	f=16, k=16, s = 1
Layer 2	f=1, k=16, s = 2	f=1, k=16, s = 1

3.3. Training

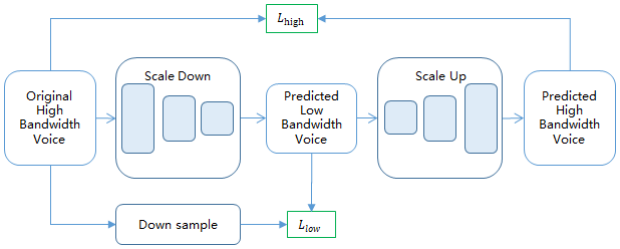

Figure 2. *Training X-net*

Figure 2 shows how we compose the objective for training X-net. Input to X-net is high-bandwidth voice y. During training X-net computes $\hat{z} = \text{scale\_down}(y)$ then $\hat{y} = \text{scale\_up}(\hat{z})$, regarding codec as straight-through. The training objective contains two loss terms on $\hat{z}$ and $\hat{y}$, respectively. For $\hat{z}$ we use a conventionally down-sampled version of y as the regression target:

$$L_{low} = Loss(\hat{z}, \text{downsample}(y)) \tag{3}$$

This loss term helps secure reasonable voice quality in the absence of a matched scale-down or scale-up module. For $\hat{y}$ we use the usual reconstruction loss:

$$L_{high} = Loss(\hat{y}, y) \tag{4}$$

The final objective combines these two loss terms:

$$L_{X-net} = \alpha \cdot L_{low} + \beta \cdot L_{high} \tag{5}$$

We use $\alpha = \beta = 0.5$ in our experiments.

3.3.1. Two-stage training

Rather than use only time-domain loss [17] or mix time- and frequency-domain losses [20], we apply a two-stage training plan with different objectives. Stage one uses time-domain MSE loss for minimized waveform distortion. Stage two switches to MSE on log spectral amplitude (LSA) which better captures perceptual similarity. More explicitly, we write

$$\begin{aligned} Loss_{time\_domain}(\hat{X}, X) &= \left\| \hat{X} - X \right\|_2 \\ Loss_{frequency\_domain}(\hat{X}, X) &= \left\| \text{LSA}(\hat{X}) - \text{LSA}(X) \right\|_2 \end{aligned} \tag{6}$$

where $\hat{X}$ stands for $\hat{z}$ or $\hat{y}$, and X is corresponding supervision target. The idea is that time-domain MSE in stage one is likely to land the networks into a well-conditioned neighbourhood of the phase landscape, so that stage two may proceed stably even without phase supervision. Switching instead of mixing losses also removes the need to tune extra hyperweights.

4. Experiments

4.1. PREVIEW: result summary

We briefly summarize our main findings before entering detailed explanations and results.

BWE with very small network is possible. Tested X-net with 1024 parameters is at least as good as a large but blind system. This strongly favours non-blind BWE over blind BWE.

Learned downsampling improves BWE. Both scale-up and baseline BWE can benefit from scale-down, even though scale-down is learned without baseline architecture in mind

Perceptual objective improves perceptual quality. LSD, POLQA and listening test favour X-net trained with LSA, while SNR metric favours baseline with time-domain MSE

4.2. Setup

4.2.1. Data

We train and evaluate our method on the VCTK dataset [24]. VCTK contains speech data from 109 native English speakers. Each speaker reads out approximately 400 different sentences. It contains 44 hours of speech data in total, which is divided into 78% training, 8% validation, and 14% testing, with no speaker overlap. Test examples include 225 wav files selected from speakers p336 to p361.

We use SWB (super wide band) voice at 32kHz sample rate as high-bandwidth voice, and WB voice at 16kHz sample rate as low-bandwidth voice. We use SSRC [25], a fast high-quality sample rate converter, to generate SWB and WB data from original VCTK data at 48kHz sample rate. Spectral analysis is performed using 60ms windows, which is 1920 sample points for SWB, 960 for WB.

4.2.2. Training details

Systems are developed on Keras [26] with TensorFlow [27] backend. Tested X-net implementation uses the lightweight configuration of Table 1. In two-stage training of X-net, stage one trained for 10 epochs and stage two trained for 50 epochs with early stop (patience=5). Adam [28] is used for optimization, with learning rate set at 0.0001.

4.2.3. Test conditions

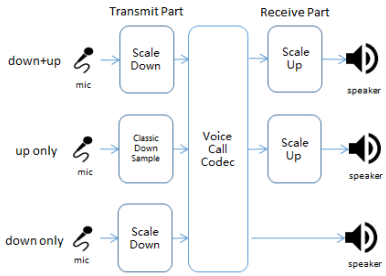

Figure 3. *Evaluating X-net*

Figure 3 shows three ways to deploy trained X-net in voice call systems, depending on the availability of X-net modules at the transmitter and the receiver. We explain them as follows.

- **'down+up'**: Paired scale-down and scale-up are deployed with transmitter and receiver, respectively. Receiver expects high-bandwidth voice.
- **'up only'**: Scale-up is deployed with receiver, matched to a classic downsampling method at the transmitter. Receiver expects high-bandwidth voice.
- **'down only'**: Scale-down is deployed with transmitter, without scale-up at receiver. Receiver expects low-bandwidth voice.

'down+up' and 'up only' conditions are tested to compare systems. 'down only' condition is tested to assess quality loss where matched scale-up is not present.

For baseline we use the blind time-domain BWE method of [17]. It features a U-net-like deep CNN with 4 downsampling then 4 upsampling layers, with residual connections between down- and upsampling layers at matched time scales. Baseline is evaluated under same test conditions as our system.

4.2.4. Metrics

We conduct objective and subjective evaluations. For objective evaluation we use signal-to-noise ratio (SNR), log-spectral distance (LSD) and POLQA (Perceptual Objective Listening Quality Assessment) [29]. SNR is computed between prediction x and reference y as

$$\mathrm{SNR}(x,y) = 10\log\frac{\|y\|_2^2}{\|x-y\|_2^2} \tag{7}$$

LSD captures the average difference between log amplitude spectrograms. It is defined by

$$\mathrm{LSD}(x,y) = \frac{2}{L\sqrt{K}}\sum_{l=1}^{L}\|\log X_l - \log Y_l\|_2 \tag{8}$$

where Y and X are STFT amplitudes of y and x, respectively, K is the STFT size, $l = 1, \dots, L$ is the frame index, and $\log(\cdot)$ is applied element-wise.

POLQA[29] is a standardized (ITU-T-P.863) objective test that simulates subjective mean opinion score (MOS) test for assessing voice quality over communication channels. POLQA MOS scores voice quality on a scale from 1 to 5 (higher score for better quality). In our experiments we calculate POLQA with the EVS WB codec at bit rates 13.2kbps and 24.4kbps.

For "purely" subjective evaluation we asked 30 evaluators, 15 male and 15 female, to rate final voice quality of X-net and that of a blind BWE using polynomial interpolation.

4.3. SNR and LSD results

Table 2 shows our objective SNR and LSD results. Baseline in the 'down+up' condition has been retrained with scale-down to remove input domain bias.

Table 2. *SNR and LSD results*

Test Condition	Model	SNR ↑	LSD ↓
'down+up'	X-net	16.15	**3.28**
	Baseline + scale-down	**18.5**	3.43
'up only'	X-net	10.71	3.98
	Baseline	13.57	5.38

On SNR the baseline performs better in both conditions. This is expected because 1) the baseline has higher model capacity, and 2) the baseline is trained solely on reconstruction MSE. On LSD we observe X-net performs better in both conditions, which can be attributed to the LSA objective in training stage 2. This is promising because LSD is closer to perceptual quality than time-domain SNR.

The effect of joint training, particularly that of a downsampler learned under BWE objective, is obvious from comparing 'down+up' against 'up only' results. This is also true with the baseline when it is co-trained with scale-down. It shows that scale-down has learned useful features which can be exploited by other BWE architectures which it had been trained with.

Figure 4 compares qualitative results under 'down+up' and 'up only' conditions. We select two sentences from speaker p345 of VCTK and run the test pipeline under both conditions (Figure 3). Spectrograms of their BWE results are shown alongside original SWB voice. While 'up only' setting can already recover the bandwidth to good extent, using a paired downsampler (i.e. scale-down) brings more energy and details back in the high-frequency range.

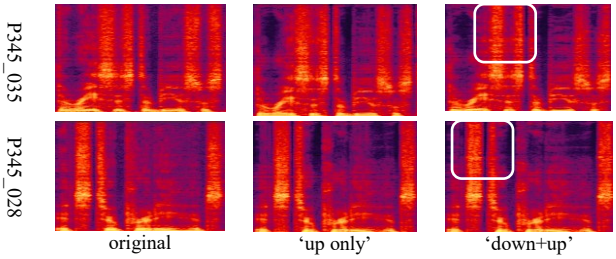

Figure 4. *Example spectrograms from X-net*

4.4. POLQA results

Table 3 shows POLQA MOS results of X-net and baseline in 'down+up' and 'up only' conditions. Results are computed with EVS WB codec at both 13.2 and 24.4bps rates. EVS WB/SWB voice without BWE is included for reference.

Table 3. *POLQA MOS results*

Test Condition	Model	EVS WB 13.2kbps	EVS WB 24.4kbps
'down+up'	X-net	**4.24**	**4.60**
	Baseline+scale-down	4.16	4.52
'up only'	X-net	4.22	4.54
	Baseline	4.15	4.43
Standard WB call		4.07	4.37
Standard SWB call		4.25	4.61

Bit rate makes the biggest difference in these results (~0.3), followed by the choice between X-net and baseline (~0.07). X-net is trained with an objective related to perceptual quality, therefore consistent with the idea behind POLQA. X-nets recovers >90% of the POLQA gap between WB and SWB at same bitrates. The effect of using a trained downsampler is less marked, especially with the lower bit rate, but still consistent. In all cases using BWE fares better than not using BWE.

Table 4. *POLQA results of scale-down*

Codec & Bitrate	Chebyshev-I	Scale Down
EVS WB 13.2kbps	4.07	4.08
EVS WB 24.4kbps	4.37	4.37

Table 4 compares scale-down with classic downsampling in the 'down-only' test condition. For the latter we use an 8th-order Chebyshev type I low-pass filter which is also used by [17]. No BWE is applied as the purpose is to assess distortion caused by scale-down without matched scale-up. The POLQA MOS results register no noticeable difference between the two downsampling methods. In other words, receiver without scale-up capability can expect same voice quality no matter if the transmitter uses scale-down or the good old Chebyshev filter for downsampling.

4.5. Listening test results

MOS results collected from 15 male and 15 female evaluators are plotted in Figure 5. The cyan bars are X-net results and the green bars are polynomial-interpolation blind BWE results. These results show clearly distinguishable listening experiences between the two BWE methods. Some evaluators reported in an informal post-interview that they believed X-net provided brighter and clearer voices.

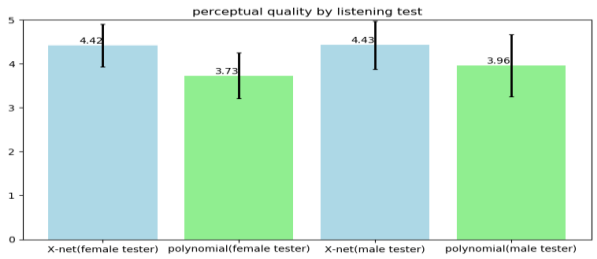

Figure 5. *Listening test results*

4.6. Runtime characteristics

Table 5 compares model size, computation load and algorithmic look-ahead between X-net and baseline [17]. Compared to the latter using a full-fledged U-net on the receiver, X-net has a lightweight design suitable for low-cost, low-delay voice calls, with 1024 parameters and 32M multiply-add operations per second, both split between the transmitter and receiver. X-net's fully causal design means it adds no inherent delay on top of the codec, compared to [17] that uses 356-sample look-ahead. X-net incurs less than 0.5ms latency on a Galaxy Note10 mobile phone with TensorFlow Lite.

Table 5. *Size, computation and look-ahead*

	Number of parameters	MMAC/sec	Look-ahead /sample points
X-net	1024	32.768	0
Baseline[17]	56,411,923	59003.904	356

5. Conclusion

In this paper we have presented X-net, a method for bandwidth extension over a band-limited voice codec. It jointly learns matched downsampling (scale-down) and upsampling (scale-up) modules as pre- and post-processing blocks to the codec, without changing the codec itself or the underlying channel. We have shown that a lightweight zero-lookahead implementation of X-net already outperforms a strong blind BWE baseline in simulated EVS WB environment. We have tentatively associated X-net's effectiveness with perception-related objective and learned downsampling, based on limited experiments. For the future work, we plan to further explore what scale-down has learned. Whether it learned to represent low-bandwidth speech more efficiently, or it cleverly hidden some high-frequency content inside low-bandwidth output. We will continue training work with codec to check whether performance is better than ignoring codec. X-net may also help design fully decoupled (therefore "blind") submodules or even new codecs. Understanding the mechanisms behind X-net, and the promises it hold, lay a rich landscape for further exploration.

6. References

[1] Olaf Ronneberger, Philipp Fischer, Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *arXiv* 1505.04597

[2] Geoffrey E Hinton and Richard S Zemel, "Autoencoders, minimum description length, and helmholtz free energy," *Advances in neural information processing systems*, pp. 3–3, 1994.

[3] Codec for Enhanced Voice Services (EVS); General overview, *3GPP TS26.441*, 2020.07

[4] Saeed Anwar, Salman Khan, and Nick Barnes,"A Deep Journey into Super-resolution A Survey", *arXiv* 1904.07523

[5] Hendrik Purwins,Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-yiin Chang, Tara Sainath, "Deep Learning for Audio Signal Processing", *Journal Of Selected Topics Of Signal Processing*, Vol. 13, No. 2, May 2019, pp. 206–219.

[6] U. Kornagel, "Spectral widening of telephone speech using an extended classification approach," in *EUSIPCO*, 2002, pp. 1–4

[7] S. Vaseghi, E. Zavarehei, and Q. Yan, "Speech bandwidth extension: Extrapolations of spectral envelop and harmonicity quality of excitation," in *ICASSP*, 2006, pp. 844–847.

[8] K. Park and H. S. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *ICASSP*, 2000, pp. 1843–1846.

[9] H. Seo, H. Kang, and F. K. Soong, "A maximum a posterior based reconstruction approach to speech bandwidth expansion in noise," in ICASSP, 2014, pp. 6087–6091.

[10] P. Jax and P. Vary, "Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden markov model," in *ICASSP*, 2003, pp. 680–683.

[11] G. Song and P. Martynovich, "A study of HMM-based bandwidth extension of speech signals," *Signal Processing*, vol. 89, no. 10, pp. 2036–2044, 2009.

[12] Kehuang Li and Chin-Hui Lee, "A deep neural network approach to speech bandwidth expansion," in *ICASSP*, 2015, pp. 4395–4399

[13] J. Abel and T. Fingscheidt, "Artificial speech bandwidth extension using deep neural networks for wideband spectral envelope estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 71–83, 2017

[14] P. Bachhav, M. Todisco, and N. Evans, "Exploiting explicit memory inclusion for artificial bandwidth extension," in *ICASSP*, 2018, pp. 5459–5463

[15] P. Bachhav et al., "Latent representation learning for artificial bandwidth extension using a conditional variational autoencoder," in *ICASSP*, 2019, pp. 7010–7014

[16] P. Bachhav, M. Todisco and N. Evans， "Artificial Bandwidth Extension Using Conditional Variational Auto-Encoders And Adversarial Learning", in *ICASSP*, 2020, pp. 6919-6923

[17] Volodymyr Kuleshov, S Zayd Enam, and Stefano Ermon, "Audio super-resolution using neural networks," 2017.

[18] S. Birnbaum, V. Kuleshov, Z. Enam, P. Koh, and S. Ermon, "Temporal film: Capturing long-range sequence dependencies with feature-wise modulations," *NeurIPS*, 2019.

[19] Teck Yian Lim, Raymond A. Yeh, Yijia Xu, Minh N. Do, Mark Hasegawa-Johnson, "Time-Frequency Networks For Audio Super-Resolution," *ICASSP* 2018, pp.646-650.

[20] Heming Wang and Deliang Wang, "Time-Frequency Loss For CNN Based Speech Super-Resolution", *ICASSP* 2020, pp. 861-865

[21] A. Pandey D. Wang, "A new framework for CNN based speech enhancement in the time domain," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 27, no. 7, pp. 1179–1188, 2019

[22] Aron van den Oord, S. Dieleman, H. Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv*, 2016

[23] P. Ramachandran, B. Zoph, and Q. V. Le, "Swish: a self-gated activation function," *arXiv* 1710.05941, 2017.

[24] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2017

[25] Naoki Shibata, https://github.com/shibatch/SSRC

[26] Keras, http://keras.io

[27] Tensorflow, http://www.tensorflow.org

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv* 1412.6980, 2014.

[29] POLQA: http://www.polqa.info/