Iterative autoregression: a novel trick to improve your low-latency speech enhancement model

Pavel Andreev[1]*, Nicholas Babaev[1]*, Ivan Shchekotov[123], Azat Saginbaev[1], Aibek Alanov[24]

[1]Samsung AI Center, Moscow   [2]HSE University, Moscow
[3]Skolkovo Institute of Science and Technology, Moscow
[4]Artificial Intelligence Research Institute, Moscow   *equal contribution

p.andreev@samsung.com, n.babaev@partner.samsung.com

Abstract

Streaming models are an essential component of real-time speech enhancement tools. The streaming regime constrains speech enhancement models to use only a tiny context of future information. As a result, the low-latency streaming setup is generally considered a challenging task and has a significant negative impact on the model's quality. However, the sequential nature of streaming generation offers a natural possibility for autoregression, that is, utilizing previous predictions while making current ones. The conventional method for training autoregressive models is teacher forcing, but its primary drawback lies in the training-inference mismatch that can lead to a substantial degradation in quality. In this study, we propose a straightforward yet effective alternative technique for training autoregressive low-latency speech enhancement models. We demonstrate that the proposed approach leads to stable improvement across diverse architectures and training scenarios.

Index Terms: speech enhancement; low latency; autoregression.

1. Introduction

The problem of real-time streaming speech processing holds immense practical significance for modern digital hearing aids, acoustically transparent hearing devices, and telecommunication. While the upper limit of undetectable lag for live, real-time processing is a subject of investigation and debate, it is estimated to be around 5-30 milliseconds depending on the application [1, 2]. Since speech enhancement tools are usually employed in joint pipelines with other speech processing tools (such as echo cancellation) and within signal transmission channels, requirements for total delay are extremely strict and, for many applications, are hardly met by mainstream speech enhancement solutions that typically rely on algorithmic latency of more than 30-60 ms (by model design) [3, 4]. Thus, there is a significant demand for research devoted to low-latency (less than 10 ms) speech enhancement models.

The nature of streaming generation follows a sequential pattern that lends itself well to autoregression. The conventional approach for training autoregressive models is through "teacher forcing" [5], whereby the model is presented with past ground-truth samples to predict the subsequent ones during training. During the inference stage, the model utilizes its own samples for autoregressive conditioning (free-running mode) since ground-truth is not available. Teacher forcing is an efficient means of training and convergence as it can be parallelized effectively for convolution-based networks. However, its primary limitation is the mismatch between training and inference which can result in a significant degradation in quality during the test phase. We observe that autoregressive speech enhancement models rely heavily on ground-truth conditioning and are, therefore, particularly vulnerable to training-inference mismatch. The common approaches address this mismatch by utilizing free-running mode during training [6]. However, these methods are typically used for recurrent networks as their application to convolution-based networks substantially slows down training, which hampers practical application.

In this paper, we present a straightforward yet highly efficient algorithm for training autoregressive models that significantly reduces the training-inference mismatch. Our approach is based on the iterative substitution of ground-truth conditioning with the model's predictions in teacher-forcing mode. Specifically, we divide the entire training process into N stages, starting with the standard teacher forcing in the initial stage. In the second stage, the forward pass of the model comprises two steps: in the first step, model predicts conditioned on the ground-truth, in the second step, it predicts conditioned on the predictions from the first step. Similarly, at the n-th stage, the forward pass consists of n steps, and at each step, the model is conditioned on the predictions from the previous step. We refer to this algorithm as "iterative autoregression" (IA) and demonstrate that it helps to alleviate the training-inference mismatch caused by teacher forcing. Furthermore, we show that autoregressive conditioning offers significant advantages over non-autoregressive baselines across various training losses and neural architectures. Notably, our proposed IA algorithm is versatile and can potentially be applied to training autoregressive models beyond the speech enhancement domain.

2. Related Work

2.1. Low-latency speech enhancement

Low-latency speech enhancement has recently attracted significant attention from the research community. Many works [7, 8] explored time-domain causal neural architectures (e.g., ConvTasNet [9]) for this task, while others [10, 11, 12] utilize time-frequency domain architectures. We limit the scope of this paper to the improvement of time domain architectures, but the proposed method is not bound to a particular domain.

2.2. Autoregressive models

Autoregressive models in application to conditional waveform generation are frequently used for neural vocoding [13, 14, 15, 16]. WaveNet [13] is a pioneering work proposing a fully-convolutional autoregressive model that produces highly realistic speech samples conditioned on linguistic features. Similarly to this work, we use causal convolutions for autoregressive conditional generation, however, we use a very different type of conditional information (degraded waveform) and

generate waveform samples in chunks instead of one by one. CARGAN [16] proposes to combine autoregressive conditioning with the power of generative adversarial networks to mitigate artifacts during spectrogram inversion. We also combine autoregressive conditioning with adversarial training, however, we consider a different task and use much smaller chunk sizes (2-16 ms compared to 92 ms employed in [16]).

3. Proposed Method

3.1. Limitations of teacher forcing

The approach of teacher forcing [5] has been widely adopted to train autoregressive models [16, 13, 17]. In contrast to the free-running mode, teacher forcing does not require sequential non-parallelizable processing, which is particularly advantageous for convolutional autoregressive models that can be efficiently parallelized during training [13]. Without parallelization, training of such models could take an inordinate amount of time. For instance, for a 2-second audio fragment, autoregressive free-running inference on an NVIDIA v100 GPU takes 1000 times longer than the teacher-forcing inference (forward pass at the training stage) for the WaveNet model, even when using an efficient implementation with activation caching [18]. However, we have observed that the training-inference mismatch caused by teacher forcing can lead to a significant degradation in speech enhancement quality, as shown in Table 1. There are several techniques to alleviate this issue [6, 19, 20], but all of them rely on the ability to perform autoregressive inference in free-running mode during training, which can complicate the usage of such techniques in practice for convolution-based architectures.

3.2. Iterative autoregression

We have discovered an alternative approach for reducing the discrepancy between training and inference without necessitating the use of a time-consuming free-running mode during training. Our proposal involves the iterative replacement of autoregressive conditioning with the model's predictions in teacher-forcing mode. Initially, the model is trained using the standard teacher-forcing mode, with the autoregressive channel containing the ground-truth waveform. At the next stage, during the forward pass of the model, two steps are involved: (1) the model makes predictions in the teacher-forcing mode, (2) it predicts when the autoregressive channel contains its predictions from the first step. In each subsequent stage of training, the autoregressive input channel comprises the model's predictions as if they were obtained in the previous stage. Overall, in this training procedure, the model is conditioned on its own predictions in iterative manner. As the training proceeds, we gradually increase the order of predictions for the model to be conditioned on, i.e., increase the number of forwarding passes ("iterations") before computing loss and doing backward passes. Note that we propagate gradient only through the last forward pass. Considering the standard training pipeline which includes forward pass, calculating loss, backpropagation, and weights optimization, the suggested method affects only forward pass, specifically, the forward function of the model. Our method is summarized in Algorithm 1 and in Figure 1. We found that this iterative procedure greatly reduces the mismatch between training and free-running (inference) modes (see Figure 2). Since IA does not require free-running (chunk-by-chunk) predictions during training, each iteration of IA can be efficiently parallelized.

Algorithm 1 Iterative autoregressive training

Require: $f(x, c)$ - model that takes noisy audio x and autoregressive conditioning c; S - number of stages; $\mathrm{stage\_iters}$ - list of number of iterations in each stage; l - algorithmic latency of the model in timesteps; $\mathrm{shift}(x, n)$ - function, which removes the last n elements from the last dimension of the tensor x and pads n zeros in the beginning of x.

> **procedure** ITERATIVEFORWARDPASS($f, x, c, \mathrm{stage\_idx}$)
>   **for** $k = 1$ to stage_idx **do**
>   **with NO_GRAD do**
> $c \leftarrow \mathrm{shift}(c, l)$ ▷ Shifting
> $c \leftarrow f(x, c)$ ▷ Conditioning
>   **end for**
> $c \leftarrow \mathrm{shift}(c, l)$ **return** c
> **end procedure**
>
> **begin**
> **for** stage_idx = 0 to S **do**
>   **for** $i = 1$ to stage_iters[stage_idx] **do**
> $(x, y) = \mathrm{sample\_batch}()$
> $c \leftarrow y$
> $c \leftarrow$ ITERATIVEFORWARDPASS($f, x, c, \mathrm{stage\_idx}$)
> $\hat{y} \leftarrow f(x, c)$
> $\mathrm{loss} \leftarrow \mathrm{LossFunc}(y, \hat{y})$
> $\mathrm{loss.backward}()$
> $f.\mathrm{update}()$
>   **end for**
> **end for**

Formally speaking, the waveform generated in the free-running mode is a stationary point of the presented iterative process. Employing notation from Algorithm 1, it is easy to see that if c is a waveform generated in the free-running mode, then $c = \mathrm{IterativeForwardPass}(f, x, c, \mathrm{stage\_idx})$ for any stage_idx, since the prediction of each next chunk was conditioned on past predictions, i.e., the process resembles the free-running mode.

Another important insight is that $y^{fr}_{1:n} = \mathrm{IterativeForwardPass}(f, x, c, n)_{2:n+1}$, disregarding of c, where $y^{fr}_{1:n}$ are the first n chunks of free-running mode waveform and $\mathrm{IterativeForwardPass}(f, x, c, n)_{2:n}$ are 2 to $n + 1$ chunks of conditioining waveform produced by iterative forward pass. Indeed, the first chunk is predicted conditioned on padding for both iterative forward pass and free-running mode. Thus, for $n = 1$ the equation is true. Let the equation be true for $n - 1$. Since free-running mode and iterative forward pass were provided with the same conditioning $y^{fr}_{1:n-1}$, their predictions for n-th chunk are identical. Thus, by induction $y^{fr}_{1:n} = \mathrm{IterativeForwardPass}(f, x, c, n)_{2:n+1}$ for any n. This equality states that iterative autoregression forward pass is guaranteed to converge to the free-running mode waveform (shifted) in a finite number of steps being less or equal to the number of chunks in the waveform. In practice, we found that the process converges much faster than this number.

3.3. Model architecture

Typically, low-latency speech enhancement models are composed of causal neural layers (e.g, uni-directional LSTMs, causal convolutions, causal attention layers, etc.) operating either in time or frequency domains. We propose to directly con-
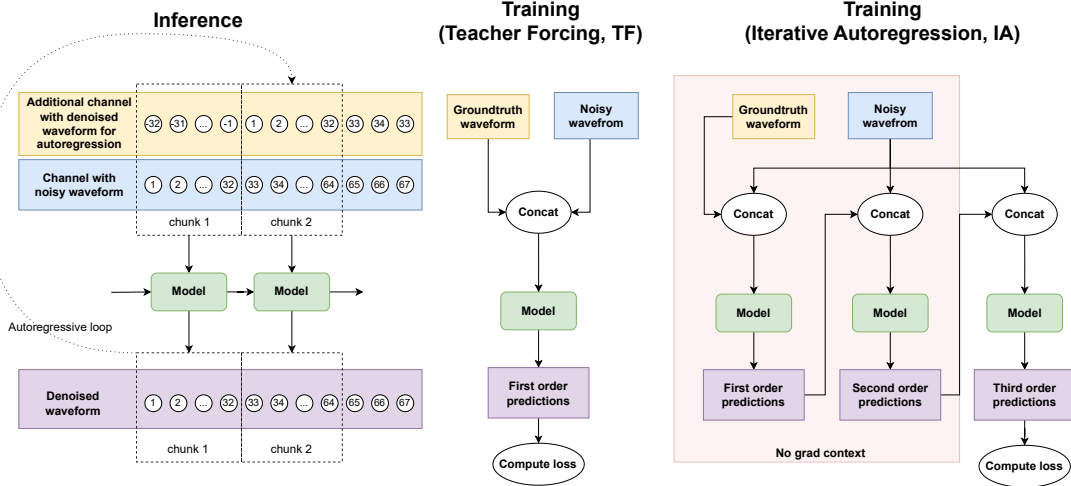
Figure 1: *Left: illustration of autoregressive conditioning for the model with algorithmic latency 32 timesteps (2 ms at 16 kHz sampling rate). Predicted timesteps from chunk 1 are re-used then making predictions for chunk 2 during inference (free-running mode). Middle: Illustration of teacher forcing training. Right: Stage 3 of the iterative autoregression training process. The model uses its own predictions to produce predictions of higher orders. We shift ground-truth waveform and predictions before forming a channel with autoregressive conditioning to avoid leakage of future information.*
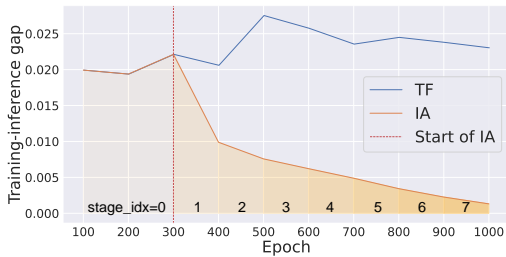

Figure 2: *Training-inference mismatch measured as the mean absolute error between model's outputs in inference (free running) and training modes, depending on training epoch. In contrast to teacher forcing (TF), iterative autoregression (IA) causes the output of the training mode to become closer to the output of the inference mode as the training proceeds. Thus, iterative autoregression allows for mitigating training-inference mismatch and improving quality.*
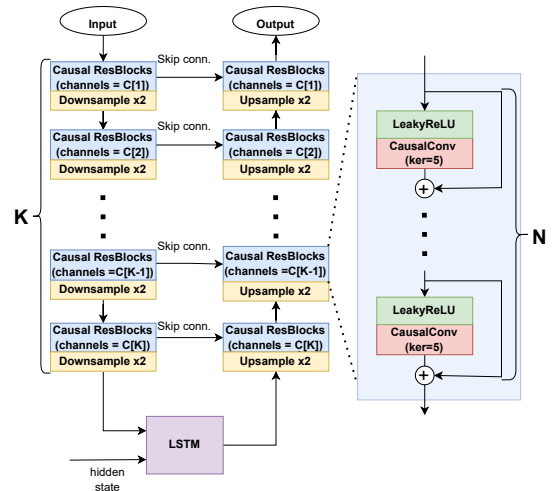

Figure 3: *WaveUNet+LSTM architecture. Parameter K regulates the overall depth of the UNet structure, parameter N determines the number of residual blocks within each layer, and array C determines the number of channels on each level of the UNet structure.*

catenate the autoregressive conditioning to the model's input (see Figure 1). For most of our experiments, we use a simple time domain architecture, which we call WaveUNet+LSTM. It is based on the famous convolutional encoder-decoder UNet architectures [21] and augmented with a one-directional LSTM layer at the bottleneck to enable a large receptive field to the past time steps similarly to [3]. We use a strided convolutional downsampling layer with kernel size 2 and stride 2 and nearest neighbor upsampling within the UNet structure. The algorithmic latency of this neural network is regulated by the number of downsampling/upsampling layers K and is equal to 2^K timesteps.

4. Experiments

Overall experimental setup   In all our experiments, we consider additive noise as the distortion to be removed from speech recordings. We conduct a number of experiments to test the pro-

ficiency of iterative autoregression in different training scenarios. For each experimental setting, we train the baseline model without autoregressive conditioning (w/o AR) and the model with autoregressive conditioning (w/ AR). The training conditions and models are identical except AR models are trained with iterative autoregression if not stated otherwise. The experiments can be divided into 5 settings, depending on the training scenario employed. In each training scenario, we change only one training condition (dataset/model architecture/loss/latency) while leaving other parameters as in the base configuration described below.

Metrics   We use state-of-the-art objective speech quality met-

ric UTMOS [22]. In our internal experiments, we found this metric has the best correlation among well-known objective metrics (e.g., SI-SDR, DNSMOS, PESQ) with human-assigned MOSes for speech enhancement task. We additionally report conventional metrics DNSMOS [23] and SI-SDR [24]. To validate the improvement provided by autoregression, we conduct subjective Comparison Category Rating (CCR) tests [25]. The referees listened to the output of a baseline (non-autoregressive) model and the corresponding output of an autoregression model and give a score from -3 (if the baseline's output is better) to 3 (if the autoregression's output is better). These tests result in Comparison Mean Opinion Scores (CMOS).

Training hyperparameters   We train all AR models for 1000 epochs and all non-AR models for 2000 epochs (so the training time is approximately the same for the corresponding runs), each epoch includes 1000 batch iterations. The best epoch is selected by maximizing the UTMOS on validation data. In all experiments, the batch size equals 16, the segment size is set to 2 s, Adam optimizer is used with the learning rate of 0.0002 and betas 0.8 and 0.9. We use 8 stages for iterative autoregressive training, the model is trained for 300 epochs during the initial (teacher forcing) stage and for 100 epochs during all other 7 stages. All experiments were conducted on NVIDIA v100 GPUs, the average experiment duration is around 10 days.

4.1. Experimental series description

Base configuration   For the main configuration, we fix the K, N, and C parameters of the model architecture to 7, 4, and [16, 24, 32, 48, 64, 96, 128], respectively. LSTM width equals 512. This configuration corresponds to 8 ms of algorithmic latency and 6 millions parameters. We use the time-domain $L1$ loss function to train the model. We employ VoiceBank-DEMAND dataset [26] which is a standard benchmark for speech-denoising systems. The train set consists of 28 speakers with 4 signal-to-noise ratios (SNR) (15, 10, 5, and 0 dB) and contains 11572 utterances. The test set (824 utterances) consists of 2 speakers unseen by the model during training with 4 SNR (17.5, 12.5, 7.5, and 2.5 dB). In this base configuration, we also test the proposed iterative autoregression training algorithm against teacher forcing (TF). It can be seen that autoregressive conditioning leads to much worse results than a non-autoregressive baseline when the model is trained with teacher forcing. The most characteristic artifacts that we have observed are the regions of silence that appear in the predicted waveform (we attach samples in supplementary material). The model seems to too heavily rely on ground-truth conditioning, to detect regions of speech and silence.

Different loss functions   In this series, we test the proficiency of autoregressive conditioning when training with different loss functions. Specifically, we replace $L1$ loss function with adversarial learning losses [27, 28] and SI-SNR loss [9]. One can see that the iterative autoregression improves the results regardless of the loss function employed for training. This result is particularly important in the case of adversarial losses, which do not require exact reconstruction of the ground-truth signal.

DNS dataset   In this experiment, we use the Deep Noise suppression (DNS) challenge dataset [29] instead of VoiceBank-DEMAND. We synthesized 100 hours of training data using provided codes and default configuration from Deep Noise Suppression challenge official repository. The models are tested on a hold-out data randomly selected and excluded from synthesized 100 hours of training data.

ConvTasNet   As an alternative architecture, we leveraged the ConvTasNet [9], a deep learning framework for end-to-end time-domain speech separation, which can be used as a speech enhancer as well. ConvTasNet consists of a linear encoder, decoder, and a temporal convolutional network (TCN) consisting of stacked 1-D dilated convolutional blocks. For experiments, the original architecture implementation was used with the parameters adjusted to match the algorithmic latency to 8 ms and the number of multiply-accumulate operations per second to 2 billion (2 GMAC, the same as in the base configuration).

Different latencies   To investigate the effect of algorithmic latency on autoregressive improvement, we additionally examine 3 models with latencies of 2 ms, 4 ms, and 16 ms. We tune the parameters of WaveUNet+LSTM architecture to adjust algorithmic latency while leaving the number of multiply-accumulate operations the same as in the base configurations.

Table 1: *Autoregressive training improves speech enhancement quality in different scenarios. All AR models are trained with iterative autoregression if not indicated otherwise (TF).*

Experiment	UTMOS	DNSMOS	SISDR	CMOS
Base configuration				
w/o AR	3.53	2.97	17.0	-
w/ AR (TF)	3.38	2.92	12.6	$-0.5^{\pm 0.08}$
w/ AR	**3.61**	**3.03**	**18.4**	$0.1^{\pm 0.05}$
Different losses				
w/o AR (adv.)	3.68	3.02	15.2	-
w/ AR (adv.)	**3.74**	**3.04**	**15.3**	$0.12^{\pm 0.04}$
w/o AR (si-snr)	3.51	2.95	17.0	-
w/ AR (si-snr)	**3.57**	**2.96**	**17.8**	$0.13^{\pm 0.05}$
DNS dataset				
w/o AR	2.42	2.98	14.5	-
w/ AR	**2.47**	**3.03**	**14.6**	$0.1^{\pm 0.05}$
ConvTasNet architecture				
w/o AR	3.08	2.86	15.3	-
w/ AR	**3.33**	**2.99**	**15.8**	$0.52^{\pm 0.06}$
Different latencies				
w/o AR (2 ms)	3.47	2.94	17.1	-
w/ AR (2 ms)	**3.55**	**2.98**	**18.3**	$0.04^{\pm 0.04}$
w/o AR (4 ms)	3.5	2.96	17.2	-
w/ AR (4 ms)	**3.59**	**3.02**	**18.6**	$0.16^{\pm 0.05}$
w/o AR (16 ms)	3.57	2.99	17.3	-
w/ AR (16 ms)	**3.64**	**3.02**	**18.6**	$0.09^{\pm 0.04}$

5. Conclusion

The proposed method of iterative autoregressive training allows for improving the quality of streaming speech enhancement models in all studied scenarios. Furthermore, it dramatically outperforms the conventional teacher forcing method, which fails to provide any improvements over the non-autoregressive baseline due to a high training-inference mismatch. We believe that the presented technique provides a practical alternative to teacher forcing and takes an important step toward improving streaming models by means of autoregression.

6. Acknowledgements

7. References

[1] S. Graetzer, J. Barker, T. J. Cox, M. Akeroyd, J. F. Culling, G. Naylor, E. Porter, R. Viveros Munoz *et al.*, "Clarity-2021 challenges: Machine learning challenges for advancing hearing aid processing," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2. International Speech Communication Association (ISCA), 2021, pp. 686–690.

[2] M. A. Stone and B. C. Moore, "Tolerable hearing aid delays. i. estimation of limits imposed by the auditory path alone using simulated hearing losses," *Ear and Hearing*, vol. 20, no. 3, pp. 182–192, 1999.

[3] A. Défossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," *Proc. Interspeech 2020*, pp. 3291–3295, 2020.

[4] X. Hao, X. Su, R. Horaud, and X. Li, "Fullsubnet: A full-band and sub-band fusion model for real-time single-channel speech enhancement," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6633–6637.

[5] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[6] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[7] Z. Tu, J. Zhang, N. Ma, and J. Barker, "A two-stage end-to-end system for speech-in-noise hearing aid processing," *Proc. Clarity*, pp. 3–5, 2021.

[8] K. Zmolikova and J. Cernock, "But system for the first clarity enhancement challenge," *Proceedings of Clarity*, pp. 1–3, 2021.

[9] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[10] S. Wang, G. Naithani, A. Politis, and T. Virtanen, "Deep neural network based low-latency speech separation with asymmetric analysis-synthesis window pair," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 301–305.

[11] Z.-Q. Wang, G. Wichern, S. Watanabe, and J. L. Roux, "Stft-domain neural speech enhancement with very low algorithmic latency," *arXiv preprint arXiv:2204.09911*, 2022.

[12] J. Liu and X. Zhang, "Inplace cepstral speech enhancement system for the icassp 2023 clarity challenge," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–2.

[13] A. van den 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," in *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[14] 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*. PMLR, 2018, pp. 2410–2419.

[15] R. Vipperla, S. Park, K. Choo, S. Ishtiaq, K. Min, S. Bhattacharya, A. Mehrotra, A. G. C. Ramos, and N. D. Lane, "Bunched lpcnet: Vocoder for low-cost neural text-to-speech system," *Proc. Interspeech 2020*, pp. 3565–3569, 2020.

[16] M. Morrison, R. Kumar, K. Kumar, P. Seetharaman, A. Courville, and Y. Bengio, "Chunked autoregressive gan for conditional waveform synthesis," *International Conference on Learning Representations*, 2021.

[17] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[18] T. L. Paine, P. Khorrami, S. Chang, Y. Zhang, P. Ramachandran, M. A. Hasegawa-Johnson, and T. S. Huang, "Fast wavenet generation algorithm," *arXiv preprint arXiv:1611.09482*, 2016.

[19] A. M. Lamb, A. G. ALIAS PARTH GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," *Advances in neural information processing systems*, vol. 29, 2016.

[20] Q. Dou, J. Efiong, and M. J. Gales, "Attention forcing for speech synthesis." in *Interspeech*, 2020, pp. 4014–4018.

[21] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *arXiv preprint arXiv:1806.03185*, 2018.

[22] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "Utmos: Utokyo-sarulab system for voicemos challenge 2022," *arXiv preprint arXiv:2204.02152*, 2022.

[23] C. K. Reddy, V. Gopal, and R. Cutler, "Dnsmos p. 835: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 886–890.

[24] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[25] R. C. Babak Naderi, Sebastian Moller, "Speech quality assessment in crowdsourcing: Comparison category rating method," 2021.

[26] C. Valentini-Botinhao *et al.*, "Noisy speech database for training speech enhancement algorithms and tts models," 2017.

[27] P. Andreev, A. Alanov, O. Ivanov, and D. Vetrov, "Hifi++: a unified framework for bandwidth extension and speech enhancement," *arXiv preprint arXiv:2203.13086*, 2022.

[28] I. Shchekotov, P. K. Andreev, O. Ivanov, A. Alanov, and D. Vetrov, "FFC-SE: Fast Fourier Convolution for Speech Enhancement," in *Proc. Interspeech 2022*, 2022, pp. 1188–1192.

[29] H. Dubey, V. Gopal, R. Cutler, A. Aazami, S. Matusevych, S. Braun, S. E. Eskimez, M. Thakker, T. Yoshioka, H. Gamper *et al.*, "Icassp 2022 deep noise suppression challenge," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9271–9275.