INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Non-autoregressive End-to-End TTS with Coarse-to-Fine Decoding

Tao Wang[1,2], *Xuefei Liu*[1], *Jianhua Tao*[1,2,3], *Jiangyan Yi*[1], *Ruibo Fu*[1,2], *Zhengqi Wen*[1]

[1]National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing
[3]CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing

{tao.wang, xuefei.liu, jhtao, jiangyan.yi, ruibo.fu, zqwen}@nlpr.ia.ac.cn

Abstract

Most end-to-end neural text-to-speech (TTS) systems generate acoustic features autoregressively from left to right, which still suffer from two problems: 1) low efficiency during inference; 2) the limitation of "exposure bias". To overcome these shortcomings, this paper proposes a non-autoregressive speech synthesis model which is based on the transformer structure. During training, the ground truth of acoustic features is schedule masked. The decoder needs to predict the entire acoustic features by taking text and the masked ground truth. During inference, we just need a text as input, the network will predict the acoustic features in one step. Additionally, we decompose the decoding process into two stages so that the model can consider the information in the context. Given an input text embedding, we first generate coarse acoustic features, which focus on the meaning of sentences. Then, we fill in missing details of acoustic features by taking into account the text information and the coarse acoustic features. Experiments on a Chinese female corpus illustrate that our approach can achieve competitive results in speech naturalness relative to autoregressive model. Most importantly, our model speed up the acoustic features generation by $296\times$ compared with the autoregressive model based on transformer structure.

Index Terms: speech synthesis, non-autoregressive, schedule mask predicting, coarse-to-fine decoding

1. Introduction

Due to the powerful modeling capabilities of deep neural networks, end-to-end models [1–3] are proposed to simplify traditional TTS pipeline [4–7] with a single neural nerwork. They are mainly encoder-decoder [8] structure based on RNN [9] or transformer [3] structure. These models have significantly improved the quality of synthesized speech [1, 2, 10]. They have two common characteristics: firstly, the ground truth of acoustic features are fed to decoder to predict the next frame during training; secondly, they generate acoustic features autoregressively from left to right during inference. Since the above characteristics, those systems are still facing two challenges.

One challenge is the limitation of "exposure bias" [11]. The decoder is an autoregressive structure which will prevent the usage of future information during training and inference. Since the model has never been exposed to its own predictions, it will result in error accumulation at test time. Some methods have been proposed to address this problem [11–13]. For example, scheduled sampling [11], randomly selecting between previous ground truth element and generated element, has become the current dominant training procedure to fit RNNs based models. However, it can only alleviate this problem but cannot solve the problem fundamentally.

The other challenge is low efficiency during inference. Although CNN [14] network and transformer [3] structure can speed up the training over RNNs based model [1,2], the models still have to condition on the previous generated acoustic features to generate next frame. This is the essential reason for model autoregression. Due to the long sequence of acoustic features, autoregression models have to face the slow inference speed problem. This problem also limits the application of end-to-end speech synthesis models in a wider range of places, such as embedded, terminal, etc.

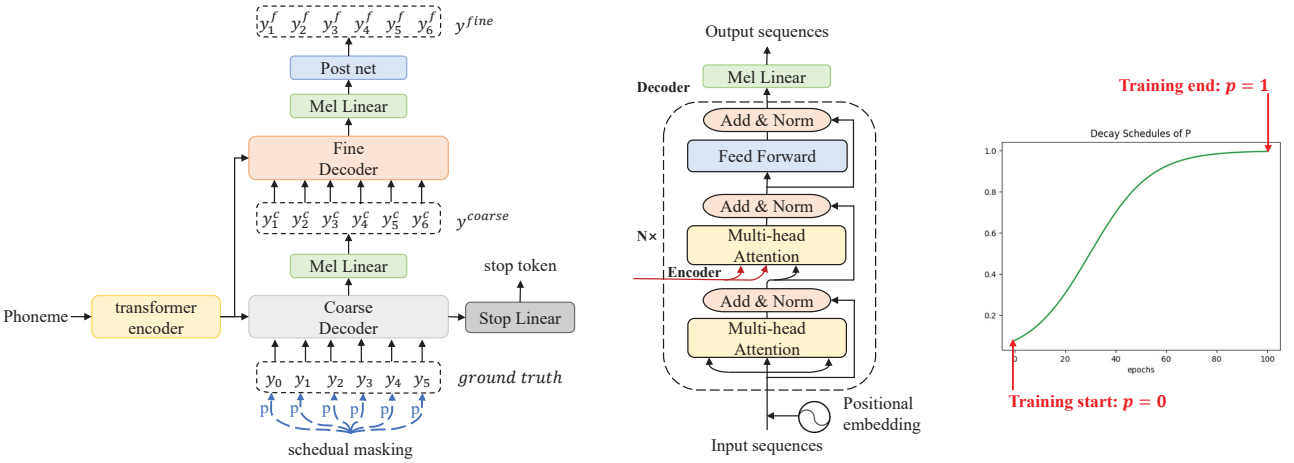
To overcome the above issues, non-autoregressive models have been proposed to some sequence generation tasks such as neural machine translation (NMT) [15–20], automatic speech recognition (ASR) [21] and TTS [22]. The main idea of non-autoregressive models is that systems predict sequences within constants number of interactions which does not depend on sequence's length. In fastspeech [22], the authors use a pre-trained phoneme duration predictor to predict hard alignment between a phoneme and its acoustic features. Recently mask-predict method [20] is proposed for encoder-decoder model. It is a conditional language model similar to BERT [23]. During training, some random words are replaced by a special mask token and the network is trained to predict original tokens. This way gets rid of the limitation that the model depends on the previous elements and enables the model can fuse the left and the right context, which can break the problem of "exposure bias" in autoregressive models. The method has been widely used in NMT tasks and the experiments get effective results.

In this paper, inspired by the idea of mask-predict [20], we propose a novel non-autoregressive end-to-end TTS model based on transformer structure. During training, acoustic features are randomly masked and the goal of the model is to predict the whole acoustic features and stop tokens. During inference, only a text sequence is needed and the model will output the whole features in one step. In addition, to further boosting the quality of speech, we decompose the decoding process into two stages. The first decoder predicts coarse acoustic features which focuses on the meaning of sentences. Subsequently, the second decoder fills in missing details by conditioning on the text information and the coarse acoustic features. Different from the first decoder, the second decoder foreknows what the basic structure of the speech looks like and the model can use it as global context to improve the prediction of the final details. Experimental results show that our framework achieves competitive performance compared to autoregressive models and the synthesis speed is much faster than the autoregressive model.

2. Background: autoregressive end-to-end TTS model

Firstly, let's understand how autoregressive TTS model works. Our goal is to generate acoustic features from text information. Given a text sequences $x = (x_1, x_2, \ldots, x_T)$ and its target acoustic features $y = (y_1, y_2, \ldots, y_{T'})$. We wish to estimate

http://dx.doi.org/10.21437/Interspeech.2020-1662



(a) The structure of non-autogressive TTS model (b) The structure of coarse decoder and fine decoder (c) Decay schedules of P in schedual

Figure 1: *(a) illustrates the whole structure of non-autoregressive model. During training, the input texts are inputted into the transformer encoder module. The encoder is composed of a stack of self-attention blocks. At decoding stage, the ground truth is first masked by a schedule masking mechanism, then the coarse acoustic features y^{coarse} are predicted based on text information and masked ground truth. The stop token is predicted by a linear projection after coarse decoder. Finally, the fine decoder combines the predicted acoustic features y_{coarse} and text information to predict the final acoustic features y^{fine}. (b) illustrates the details of the coarse decoder and fine decoder. (c) illustrates the probability P of each frame in ground truth being masked changes through the training epochs.*

$P(y|x;\theta)$, the TTS model predicting acoustic features y given input x, and θ is corresponding model parameters. The autoregressive model decomposes $P(y|x;\theta)$ into a multi-stage generation process, which can be expressed as:

$$P(y_0|x;\theta) = 1 \tag{1}$$
$$P(y_1|x;\theta) = P(y_1|y_0,x;\theta) * 1 \tag{2}$$
$$P(y_2,y_1|x;\theta) = P(y_2|y_1,y_0,x;\theta) * P(y_1|y_0,x;\theta) * 1 \tag{3}$$
$$\cdots$$
$$P(y|x;\theta) = \prod_{t=1}^{T'} P(y_t|y_{<t},x;\theta) \tag{4}$$

The formulas show autoregressive TTS model adopts previous targets $(y_1,\ldots,y_{t-1})$ as history to predict current target y_t in t_{th} step. y_0 is the start label of acoustic features.

3. Proposed non-autoregressive TTS model

The advantage of autoregression is that it can fuse context information to make the synthesized speech more natural, but the disadvantage is that it will lead to exposure bias and low efficiency during inference. To retain the strengths and eliminate the weaknesses, our proposed framework and the description of our model are shown in Fig. 1. In this section, we will present the ideas about schedule mask predicting and coarse-to-fine decoding in detail.

3.1. Schedule mask predicting

The reason for autoregression is that the model has to depend on previous elements to predict current acoustic feature. So the key insight to make it non-autoregressive is replacing the previous frames with some other else. One simple assumption is that each frame is independent. Under this assumption, we do not need to rely on previous acoustic features to predict the current frame. But this assumption is too strong since acoustic features are continuous and training will be extremely hard and not convergence. Inspired by the idea of mask-predict [20], our idea is replacing $y_{<t}$ with partial ground truth. A new token $<$ mask $>$ is introduced for training and inference. We randomly replace part of the ground truth with the token $<$ mask $>$ during training, which means the masked frames are independent to the predicted frames. The non-autoregressive TTS model can be expressed as:

$$P(y|x;\theta) = \prod_{t=1}^{T'} P(y_t|y_{t\neq<\text{mask}>},x;\theta) \tag{5}$$

$y_{t\neq<\text{mask}>}$ stands for the frames which not replaced by the $<$ mask $>$ token. Since we assume each frame is conditionally independent, those predictions can be done parallelly.

Different from the training stage, there is no ground truth during inference, we propose a schedule masking mechanism to make the model match the inference stage at the end of training. During training, this mechanism will randomly decide, whether we mask each frame with probability P_{mask}. When $P_{mask} = 0$, the model is trained exactly as autoregressive model; when $P_{mask} = 1$, the model is trained in the same setting as inference. With the training epoch growing, the probability of each frame being masked is increasing and the decoder will get less information of ground truth. At the end of training, the value of P_{mask} will be equal to 1, all ground truth will be masked and model can predict acoustic features based on input text only, which matches the inference stage's decoding setting. There are many functions that can satisfy this monotonically increasing from 0 to 1. In this paper, we use inverse sigmoid decay to compute the P_{mask} inspired by the schedule sampling method [11].

$$P_{mask} = 1 - \frac{k}{(k+\exp(i/k))} \tag{6}$$

where $k >= 1$ depends on the expected speed of convergence

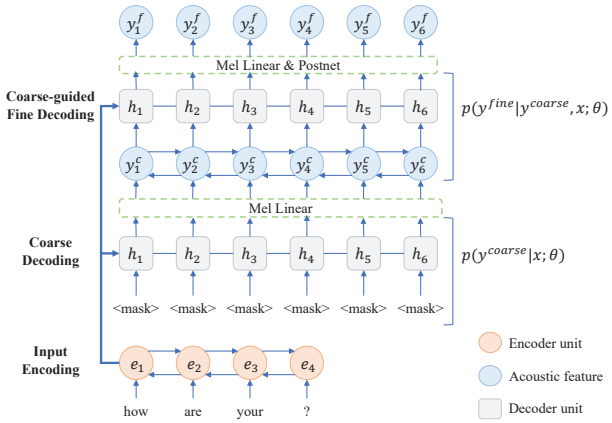


Figure 2: *At the inference stage, we first generate the coarse acoustic features y^{coarse} from text information only. Then, a fine decoder is used to fill in the missing details combining the coarse acoustic features and text information and predict finer acoustic feature y^{fine}.*

and i stands the num of training epoch. The function curve of P_{mask} is shown in the Fig. 1.

3.2. Coarse-to-fine decoding

An essential factor for autoregressive models can generate natural speech is that the prediction can combine the information of previous frames. While when we generate in a non-autoregressive form, since our assumption is that each frame is independent of each other, the model can not combine other frames' information. To solve this problem, we decompose decoding process into two stages, which is called coarse-to-fine decoding. Fig. 2 illustrates the coarse-to-fine decoding architecture, where consists of a coarse decoder and a fine decoder. The first stage of decoding is a coarse decoder which predicts coarse acoustic features from the global text information. Since there is no ground truth during inference, the coarse decoding can be expressed as:

$$P(y^{coarse}|x;\theta) = \prod_{t=1}^{T'} P\left(y_t|x;\theta\right) \tag{7}$$

It can be seen from the formula that the model predicts the acoustic features y^{coarse} based on text information merely at first stage. The predicted features y^{coarse} are relatively rough acoustic message because it lacks the context information of the acoustic features. Based on this, we perform the second decoding, which feed y^{coarse} and x to fine decoder to predict finer acoustic features, it can be expressed as:

$$P(y^{fine}|x;\theta) = \prod_{t=1}^{T'} P\left(y_t|y^{coarse}, x;\theta\right) \tag{8}$$

The fine decoder combines the global information of acoustic features and text information together, and generates more nature and expressive speech. There are two advantages to the coarse-to-fine decoding form. Firstly, the model can generate more natural speech while retaining parallel decoding, after generating the coarse acoustic features, the decoder knows what the basic meaning of the speech looks like, and the model can use it as global context to improve the prediction of the final details. Secondly, our approach fundamentally solves the issue of "exposure bias" and improves the stability of the system.

4. Experiments

In this section, we conduct experiments to evaluate our proposed method on a 20-hour, 16kHz, 16bit speech corpus, which is recorded by a professional chinese female speaker. We evaluate the performance of the model based on speech quality and inference speed. Furthermore, we explore the effects of coarse-to-fine decoding.

4.1. Setup

We train four systems to illustrate the effectiveness of our proposed model.

- **Baseline(tacotron2)** stands for autoregressive TTS model which the decoder is based on lstm. The structure details are same as tacotron2 in paper [2].
- **Baseline(transformer)** stands for autoregressive TTS model which is based on transformer structure. The structure details are same as the model in paper [10].
- **Transformer+mask** stands for non-autoregressive TTS model which is improved from **Baseline(transformer)** model by only using schedule masking.
- **Proposed method** stands for non-autoregressive TTS mode which is trained by adding coarse-to-fine decoding based on model **Transformer+mask**.

To get rid of the impact of model parameters on the model performance, all transformer-based models have the same encoder module. The transformer encoder is same as the encoder in [10]. The block numbers of encoders are all 3. Additionally, we ensure the block numbers of decoder in each transformer-based model are equal. Specifically, the block numbers of decoder in **Baseline(transformer)** and **Transformer+mask** are all 6. In **Proposed method**, the block number of coarse decoder is 3, and the block number of fine decoder is 3.

Acoustic features are extracted with 10 ms window shift. LPCNet [24] is utilized to extract 32-dimensional acoustic features, including 30-dimensional BFCCs [25], 1-dimensional pitch and 1-dimensional pitch correction parameter. Parameters of above models are all optimized using AdaDelta [26] with learning rate 0.001.

4.2. Subjective evaluation

We conduct Mean Opinion Score (MOS) listening test for audio quality on the test set. We keep the text content consistent among different models so as to exclude other interference factors, just examining audio quality. 20 listeners participated the evaluation. In each experimental group, 20 parallel sentences are selected randomly from testing sets of each system. Table 1 shows the MOS score of each system. Comparing with autoregressive model **Baseline(transformer)** and **Baseline(tacotron2)**, our proposed method has comparable results with them, this is because our model gets rid of the problem of "exposure bias" in the autoregressive model and we decompose decoding process into two stage to learn more natural speech. When we eliminate coarse-to-fine decoding model, which is the **Transformer+mask** model, we find that **Transformer+mask** is worse than **Proposed model** and other autoregressive models. This result suggests that decoding from coarse-level to fine-

Table 1: *The MOS score with 95% confidence intervals*

Model	MOS	TYPE
Baseline(tacotron2)	4.25 ± 0.07	autoregressive
Baseline(transformer)	4.18 ± 0.07	autoregressive
Transformer+mask	3.76 ± 0.08	non-autoregressive
Proposed model	4.20 ± 0.07	non-autoregressive

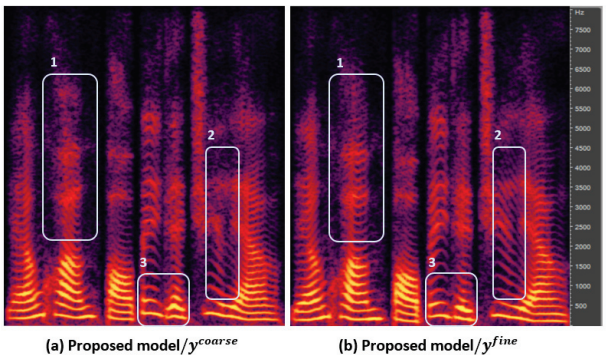


Figure 3: *The comparison of spectrums between coarse speech y^{coarse} and fine speech y^{fine}*

level is beneficial and removing the fine decoder harms performance since the decoder loss access to global contextual acoustic feature information.

4.3. Effectiveness of coarse-to-fine decoding

To further understand the role of coarse-to-fine decoding, firstly, we visualize some spectrograms of synthetic speech. Fig. 3 shows the generated spectrograms of coarse acoustic features y^{coarse} and fine acoustic features y^{fine}. In the high frequency part, we can find y^{fine} is more clear by observing part 1 and 2; in the low frequency, by observing part 3, we can find coarse speech's spectrum connection is not smooth which will cause the speech to sound unnatural. After fine decoder, the spectrum is more natural. The function of fine decoder is to improve the unnatural phenomenon of the low frequency part and to supplement the details of the high frequency part.

Secondly, Fig. 4 shows the acoustic features's loss function of the training process. We can find that y^{fine} in our **Proposed model** can reach the lowest level, which means it is the closest to the ground truth. Compared with the loss of **Transformer+mask**, although we guarantee the consistency of the model parameters, the **Proposed model** can be optimized to a better level. This shows that the decomposition of the decoding stage into two stages can help the model to combine the learned coarse acoustic information for further optimization and make the final acoustic features more precise.

4.4. Inference speed

We evaluate the inference speed of our proposed method with other autoregressive TTS model. The evaluation experiments are conduct on the serve with 52 Intel Xcon CPU, 512GB memory and 1 NVIDIA V100 gpu. It is worth mentioning that, in the design of the model, we have kept the model parameters as consistent as possible to eliminate their effects. We show the inference speed for acoustic features generation in Table 2. It can be seen that **Proposed model** speeds up acoustic features generation by $80\times$, compared with **Baseline(tacotron2)** model. **Proposed model** speeds up acoustic features generation by $296\times$, compared with **Baseline(transformer)** model. It shows that autoregressive generation greatly affects the speed of model and our proposed method can transform the model into a non-autoregressive form and increase the speed of synthesis effectively.

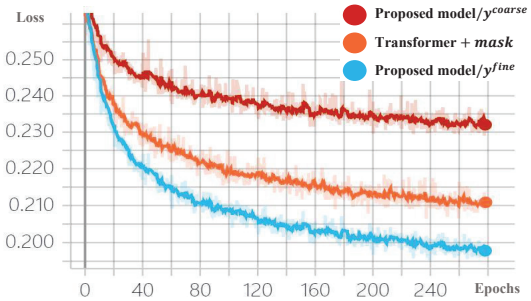


Figure 4: *The comparsion of loss function.*

Table 2: *The comparison of inference speed with 95% confidence intervals for proposed model and the baseline systems. The value of inference speed indicates how long it takes to synthesize 500 frames acoustic features.*

Model	Params	Inference(s)	Speedup
Baseline(tacotron2)	$1.87e7$	1.934 ± 0.210	/
Proposed model	$1.31e7$	0.024 ± 0.002	$80\times$
Baseline(transformer)	$1.29e7$	7.116 ± 0.189	/
Proposed model	$1.31e7$	0.024 ± 0.002	$296\times$

5. Conclusion

In this paper, we present a non-autoregressive end-to-end TTS model with coarse-to-fine decoding process, aiming to improve the synthesis speed by parallelly generating while ensuring the speech's quality. Firstly, relying on the schedule mask predicting, the autoregressive TTS model based on transformer can be changed into non-autoregressive form, which can generate speech in parallel to improve the synthesis speed. Secondly, based on coarse-to-fine decoding framework, our approach allows generating acoustic feature from coarse-level to fine-level, which is found to be very beneficial for speech's naturalness. Experiments demonstrate that the proposed model greatly speeds up synthesis, and the synthesized speech is comparable to the autoregressive model. We also verified the profits of coarse-to-fine decoding. Further, we will try to compress the non-autoregressive end-to-end TTS model so that the technique can be applied to a wider range of applications.

6. Acknowledgements

This work is supported by the National Key Research & Development Plan of China (No.2017YFC0820602), the National Natural Science Foundation of China (NSFC) (No.61831022, No.61901473, No.61771472, No.61773379) and Inria-CAS Joint Research Project (No.173211KYSB20170061 and No.173211KYSB20190049). This research is (partially) funded by Huawei Noah's Ark Lab. This work is also supported by the CCF-Tencent Open Research Fund.

7. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

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

[4] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[5] A. W. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis." in *in Eurospeech97*, 1997, pp. 601–604.

[6] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7962–7966.

[7] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *speech communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014.

[10] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Close to human quality TTS with transformer," *CoRR*, vol. abs/1809.08895, 2018. [Online]. Available: http://arxiv.org/abs/1809.08895

[11] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179.

[12] R. Liu, B. Sisman, J. Li, F. Bao, G. Gao, and H. Li, "Teacher-student training for robust tacotron-based tts," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6274–6278.

[13] H. Guo, F. K. Soong, L. He, and L. Xie, "A new gan-based end-to-end tts training algorithm," *arXiv preprint arXiv:1904.04775*, 2019.

[14] L. Yan, B. Yoshua, and H. Geoffrey, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher, "Non-autoregressive neural machine translation," *arXiv preprint arXiv:1711.02281*, 2017.

[16] J. Lee, E. Mansimov, and K. Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," *arXiv preprint arXiv:1802.06901*, 2018.

[17] J. Gu, Q. Liu, and K. Cho, "Insertion-based decoding with automatically inferred generation order," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 661–676, 2019.

[18] M. Stern, W. Chan, J. Kiros, and J. Uszkoreit, "Insertion transformer: Flexible sequence generation via insertion operations," *arXiv preprint arXiv:1902.03249*, 2019.

[19] S. Welleck, K. Brantley, H. Daumé III, and K. Cho, "Non-monotonic sequential text generation," *arXiv preprint arXiv:1902.02192*, 2019.

[20] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6114–6123.

[21] N. Chen, S. Watanabe, J. Villalba, and N. Dehak, "Non-autoregressive transformer automatic speech recognition," *arXiv preprint arXiv:1911.04908*, 2019.

[22] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[24] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[25] T. Gulzar, A. Singh, and S. Sharma, "Comparative analysis of lpcc, mfcc and bfcc for the recognition of hindi words using artificial neural networks," *International Journal of Computer Applications*, vol. 101, no. 12, pp. 22–27, 2014.

[26] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701