NEURAL SPEECH PHASE PREDICTION BASED ON PARALLEL ESTIMATION ARCHITECTURE AND ANTI-WRAPPING LOSSES

Yang Ai, Zhen-Hua Ling

National Engineering Research Center of Speech and Language Information Processing
University of Science and Technology of China, Hefei, P.R.China
yangai@ustc.edu.cn, zhling@ustc.edu.cn

ABSTRACT

This paper presents a novel speech phase prediction model which predicts wrapped phase spectra directly from amplitude spectra by neural networks. The proposed model is a cascade of a residual convolutional network and a parallel estimation architecture. The parallel estimation architecture is composed of two parallel linear convolutional layers and a phase calculation formula, imitating the process of calculating the phase spectra from the real and imaginary parts of complex spectra and strictly restricting the predicted phase values to the principal value interval. To avoid the error expansion issue caused by phase wrapping, we design anti-wrapping training losses defined between the predicted wrapped phase spectra and natural ones by activating the instantaneous phase error, group delay error and instantaneous angular frequency error using an anti-wrapping function. Experimental results show that our proposed neural speech phase prediction model outperforms the iterative Griffin-Lim algorithm and other neural network-based method, in terms of both reconstructed speech quality and generation speed.

***Index Terms*—** speech phase prediction, parallel estimation architecture, anti-wrapping loss, neural network, phase wrapping

1. INTRODUCTION

Speech phase prediction, also known as speech phase reconstruction, recovers speech phase spectra from amplitude spectra and plays an important role in speech generation tasks. Currently, several speech generation tasks, such as speech enhancement (SE) [1, 2, 3], bandwidth extension (BWE) [4, 5, 6] and speech synthesis (SS) [7, 8, 9, 10], mainly focus on the prediction of amplitude spectra or amplitude-derived features (e.g., mel spectrograms and mel cepstra). Therefore, speech phase prediction is crucial for waveform reconstruction in these tasks. However, limited by the issue of phase wrapping and the difficulty of phase modeling, the precise prediction of the speech phase remains a challenge until now.

The Griffin-Lim algorithm [11] is a well-known iterative phase estimation method which is widely used in several speech generation tasks. However, the Griffin-Lim algorithm always causes unnatural artifacts in the reconstructed speech. With the development of deep learning, Takamichi *et al.* [12, 13] proposed a von-Mises-distribution deep neural network (DNN) for phase prediction. However, the phase predicted by the DNN still needs to be refined using the Griffin-Lim algorithm. Masuyama *et al.* [14] proposed a DNN-based two-stage method which first predicted phase derivatives by DNNs, and then the phase was recursively estimated by a recurrent phase unwrapping algorithm. To our knowledge, predicting speech wrapped phase spectra directly from amplitude spectra using neural networks has not yet been thoroughly investigated.

Due to the phase wrapping property, how to design 1) suitable architectures or activation functions to restrict the range of predicted phases for direct wrapped phase prediction and 2) loss functions suitable for phase characteristics, are the two major challenges for phase prediction based on neural networks. To overcome these challenges, we propose a neural speech phase prediction model based on a parallel estimation architecture and anti-wrapping losses. The proposed model passes the input log amplitude spectra through a residual convolutional network and a parallel estimation architecture to predict the wrapped phase spectra directly. To restrict the output phase values to the principal value interval and predict the wrapped phases directly, the parallel estimation architecture imitates the process of calculating the phase spectra from the real and imaginary parts of complex spectra, and it is formed by two parallel convolutional layers and a phase calculation formula. To avoid the error expansion issue caused by phase wrapping, we propose the instantaneous phase loss, group delay loss and instantaneous angular frequency loss activated by an anti-wrapping function at the training stage. Experimental results show that our proposed model can achieve higher reconstructed speech quality than the iterative Griffin-Lim algorithm and the von-Mises-distribution DNN-based method. In addition, our proposed model also exhibits the fastest generation speed, reaching 19.6x real-time on a CPU.

This paper is organized as follows. In Section 2, we briefly review the representative iterative speech phase estimation algorithm and neural network-based speech phase prediction method, respectively. In Section 3, we provide details on our proposed neural speech phase prediction model. In Section 4, we present our experimental results. Finally, we give conclusions in Section 5.

2. RELATED WORK

2.1. Iterative phase estimation

This subsection briefly describes the well-known iterative Griffin-Lim algorithm [11]. It iteratively estimates the phase spectra from amplitude spectra via the short-time Fourier transform (STFT) and inverse STFT (ISTFT). Assume that the amplitude spectrum is $\boldsymbol{A} \in \mathbb{R}^{F \times N}$, where F and N are the total number of frames and frequency bins, respectively. Then initialize the phase spectrum $\hat{\boldsymbol{P}} \in \mathbb{R}^{F \times N}$ to zero matrix and iteratively execute the following formulas until convergence:

$$\boldsymbol{S} = STFT\left[ISTFT\left(\boldsymbol{A}e^{j\hat{\boldsymbol{P}}}\right)\right], \tag{1}$$

This work was partially funded by the National Nature Science Foundation of China under Grant 61871358 and the Fundamental Research Funds for the Central Universities.

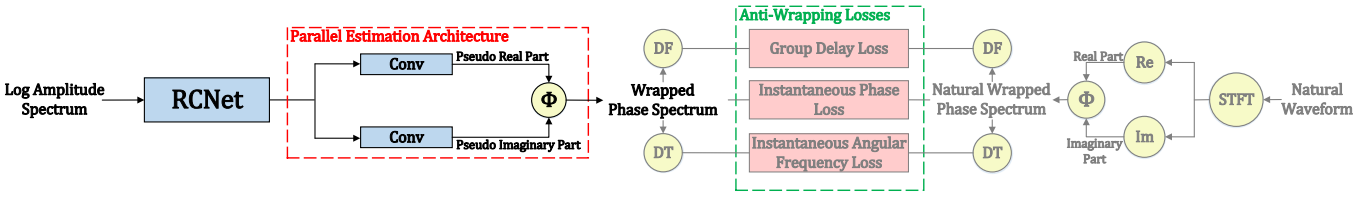

Fig. 1. Details of the proposed neural speech phase prediction model. Here, *RCNet*, *Conv*, *STFT*, *DF*, *DT*, *Re*, *Im* and Φ represent the residual convolutional network, linear convolutional layer, short-time Fourier transform, differential along frequency axis, differential along time axis, real part calculation, imaginary part calculation and phase calculation formula, respectively. Gray parts do not appear during generation.

$$e^{j\hat{\boldsymbol{P}}} = \boldsymbol{S} \oslash |\boldsymbol{S}|, \tag{2}$$

where $\oslash$ and $|\cdot|$ represent the element-wise division and amplitude calculation, respectively. The Griffin-Lim algorithm can be easily implemented and is popular in speech generation tasks. Since the iterative algorithm always gives a local optimal solution, the reconstructed speech quality is limited by the influence of the initial phase and there are obvious artifacts in the reconstructed speech.

2.2. Neural network-based phase prediction

This subsection briefly describes the von-Mises-distribution DNN-based method [12, 13]. This method assumes that the phase follows a von Mises distribution and then uses a DNN to predict the mean parameter of the phase distribution from the input log amplitude spectra at current and ± 2 frames. The mean parameter is regarded as the predicted phase. The DNN is composed of three 1024-unit feed-forward hidden layers activated by gated linear unit (GLU) [15] and a linear output layer. A multi-task learning strategy with phase loss and group delay loss is adopted to train the DNN. The phase loss and group delay loss are formed by activating the phase error and group delay error using a negative cosine function, respectively. Finally, the phase predicted by the DNN is set as the initial phase and refined by the Griffin-Lim algorithm with 100 iterations.

3. PROPOSED METHODS

3.1. Model structure

As shown in Figure 1, the proposed neural speech phase prediction model predicts the wrapped phase spectrum $\hat{\boldsymbol{P}} \in \mathbb{R}^{F\times N}$ directly from the input log amplitude spectrum $\log \boldsymbol{A} \in \mathbb{R}^{F\times N}$ by a cascade of a residual convolutional network (RCNet) and a parallel estimation architecture.

In the RCNet, the input sequentially passes through a linear convolutional layer (kernel size=7 and channel size=512) and three parallel residual convolutional blocks (RCBlocks). Then, the outputs of these three RCBlocks are summed (i.e., skip connections), averaged, and finally activated by a leaky rectified linear unit (LReLU) [16]. Each RCBlock is formed by a cascade of three sub-RCBlocks. In each sub-RCBlock, the input is first activated by an LReLU, then passes through a linear dilated convolutional layer, then is activated by an LReLU again, passes through a linear convolutional layer, and finally superimposes with the input (i.e., residual connections) to obtain the output. The kernel sizes of all the convolutional operations in the three RCBlocks are 3, 7, and 11, respectively, and the channel sizes are 512. The dilation factors of the dilated convolutional operations in the three sub-RCBlocks for each RCBlock are 1, 3, and 5, respectively.

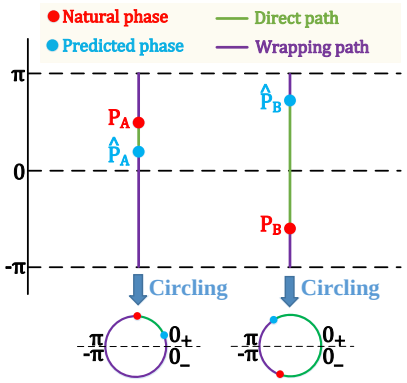

Fig. 2. An illustration explanation of the error expansion issue caused by phase wrapping.

The parallel estimation architecture is inspired by the process of calculating the phase spectra from the real and imaginary parts of complex spectra, and consists of two parallel linear convolutional layers (kernel size=7 and channel size=N) and a phase calculation formula $\boldsymbol{\Phi}$. The outputs of the two parallel layers are the pseudo real part $\hat{\boldsymbol{R}} \in \mathbb{R}^{F\times N}$ and pseudo imaginary part $\hat{\boldsymbol{I}} \in \mathbb{R}^{F\times N}$, respectively. Then the wrapped phase spectrum $\hat{\boldsymbol{P}}$ is calculated by $\boldsymbol{\Phi}$ as follows:

$$\hat{\boldsymbol{P}} = \boldsymbol{\Phi}(\hat{\boldsymbol{R}}, \hat{\boldsymbol{I}}). \tag{3}$$

Equation 3 is calculated element-wise. For $\forall R \in \mathbb{R}$ and $I \in \mathbb{R}$, we define

$$\boldsymbol{\Phi}(R, I) = \arctan\left(\frac{I}{R}\right) - \frac{\pi}{2} \cdot Sgn^*(I) \cdot [Sgn^*(R) - 1], \tag{4}$$

and $\boldsymbol{\Phi}(0, 0) = 0$. When $x \geq 0$, $Sgn^*(x)$ is equal to 1; otherwise, it is equal to -1. Formula $\boldsymbol{\Phi}$ strictly restricts the predicted phase to the principal value interval $(-\pi, \pi]$ for direct wrapped phase prediction.

3.2. Training criteria

Due to the wrapping property of the phase, the absolute error $e_a = |\hat{P} - P|$ between the predicted phase $\hat{P}$ and the natural phase P might not be their true error. As shown in Figure 2, assuming that the phase principal value interval is $(-\pi, \pi]$, there are two paths from the predicted phase point $\hat{P}_*$ to the natural one P_*, i.e., the direct path (corresponding to the absolute error) and the wrapping path (corresponding to the wrapping error). Visually, we can connect the vertical line segment between $-\pi$ and π end to end into a circle,

according to the wrapping property of the phase. Obviously, the wrapping path must pass through the boundary of the principal value interval, and the wrapping error is $e_w = 2\pi - |\hat{P} - P|$. Therefore, the true error between $\hat{P}$ and P is

$$e = \min\{|\hat{P} - P|, 2\pi - |\hat{P} - P|\}. \tag{5}$$

For example, in Figure 2, the true error between $\hat{P}_A$ and P_A is the absolute error, but the true error between $\hat{P}_B$ and P_B is the wrapping error. This means that the absolute error and the true error satisfy $|\hat{P} - P| \geq e$, resulting in *error expansion issue* when using the conventional L1 loss or mean square error (MSE) loss. Equation 5 can be written in another form:

$$e = \left| \hat{P} - P - 2\pi \cdot round\left(\frac{\hat{P} - P}{2\pi}\right) \right|, \tag{6}$$

where $round$ represents the rounding. Obviously, Equation 6 is a function of error $\hat{P} - P$. We define a function $f_{AW}(x)$ as follows:

$$f_{AW}(x) = \left| x - 2\pi \cdot round\left(\frac{x}{2\pi}\right) \right|, x \in \mathbb{R}. \tag{7}$$

f_{AW} is an anti-wrapping function which can avoid the error expansion issue caused by phase wrapping because $f_{AW}(\hat{P} - P) = e$.

Specifically, we define the instantaneous phase loss $\mathcal{L}_{IP}$ between the wrapped phase spectrum $\hat{\boldsymbol{P}}$ predicted by our model and the natural wrapped phase spectrum $\boldsymbol{P} = \boldsymbol{\Phi}(\boldsymbol{R}, \boldsymbol{I})$ as follows:

$$\mathcal{L}_{IP} = \mathbb{E}_{(\hat{\boldsymbol{P}}, \boldsymbol{P})} \overline{f_{AW}\left(\hat{\boldsymbol{P}} - \boldsymbol{P}\right)}, \tag{8}$$

where $f_{AW}(\boldsymbol{X})$ means element-wise anti-wrapping function calculation for matrix $\boldsymbol{X}$, and $\overline{\boldsymbol{Y}}$ means averaging all elements in the matrix $\boldsymbol{Y}$. $\boldsymbol{R}$ and $\boldsymbol{I}$ are the real and imaginary parts of the complex spectrum extracted from the natural waveform. To ensure the continuity of the predicted wrapped phase spectrum along the frequency and time axes, we also define the group delay loss $\mathcal{L}_{GD}$ and instantaneous angular frequency loss $\mathcal{L}_{IAF}$, which are both activated by the anti-wrapping function f_{AW} to avoid the error expansion issue as follows:

$$\mathcal{L}_{GD} = \mathbb{E}_{(\Delta_{DF}\hat{\boldsymbol{P}}, \Delta_{DF}\boldsymbol{P})} \overline{f_{AW}\left(\Delta_{DF}\hat{\boldsymbol{P}} - \Delta_{DF}\boldsymbol{P}\right)}, \tag{9}$$

$$\mathcal{L}_{IAF} = \mathbb{E}_{(\Delta_{DT}\hat{\boldsymbol{P}}, \Delta_{DT}\boldsymbol{P})} \overline{f_{AW}\left(\Delta_{DT}\hat{\boldsymbol{P}} - \Delta_{DT}\boldsymbol{P}\right)}, \tag{10}$$

where Δ_{DF} and Δ_{DT} represent the differential along the frequency axis and time axis, respectively. Finally, the training criteria of our proposed model is to minimize the final loss

$$\mathcal{L} = \mathcal{L}_{IP} + \mathcal{L}_{GD} + \mathcal{L}_{IAF}. \tag{11}$$

4. EXPERIMENTS

4.1. Data and feature configuration

A subset of the VCTK corpus [17] was adopted in our experiments. We selected 11,572 utterances from 28 speakers and randomly divided them into a training set (11,012 utterances) and a validation set (560 utterances). We then built the test set, which included 824 utterances from 2 unseen speakers (a male speaker and a female speaker). The original waveforms were downsampled to 16 kHz for the experiments. When extracting the amplitude spectra and phase spectra from natural waveforms, the window size was 20 ms, the window shift was 5 ms, and the FFT point number was 1024 (i.e., $N = 513$).

Table 1. Objective and subjective evaluation results among phase prediction methods. Here, "$a\times$" represents $a\times$ real time.

	SNR(dB)↑	F0-RMSE(cent)↓	RTF↓	MOS↑
GT	–	–	–	3.97±0.052
NSPP	**8.26**	**10.0**	**0.051 (19.6×)**	**3.95±0.055**
GL22	2.70	66.4	0.053 (18.9×)	2.92±0.10
GL100	3.35	32.5	0.23 (4.48×)	3.73±0.069
DNN+GL100	5.03	13.2	0.29 (3.45×)	3.86±0.057

4.2. Comparison among phase prediction methods

We first conducted objective and subjective experiments to compare the performance of our proposed neural speech phase prediction model and other phase prediction methods. Note that the object for comparison here is the speech waveforms reconstructed from the amplitude spectra and the predicted phase spectra through ISTFT. The descriptions of methods for comparison are as follows[1]:

- **NSPP**: The proposed neural speech phase prediction model. The model details are given in Section 3. The model was trained using the AdamW optimizer [18] with $\beta_1 = 0.8$ and $\beta_2 = 0.99$ on a single Nvidia 3090Ti GPU. The learning rate decay was scheduled by a 0.999 factor in every epoch with an initial learning rate of 0.0002. The batch size was 16, and the truncated waveform length was 8000 samples (i.e., 0.5 s) for each training step. The model was trained until 3100 epochs.
- **GLn**: The iterative Griffin-Lim phase estimation algorithm [11] mentioned in Section 2.1 with n iterations ($n = 22$ and $n = 100$ were used in the experiments).
- **DNN+GL100**: The von-Mises-distribution DNN-based phase prediction method [12, 13] mentioned in Section 2.2. The phase spectra were first predicted by the DNN and then refined by the Griffin-Lim algorithm with 100 iterations. We reimplemented it ourselves. The training configuration of the DNN is the same as that of **NSPP**.

Two objective metrics used in our previous work [19] were adopted here to compare the reconstructed speech quality, including the signal-to-noise ratio (SNR), which was an overall measurement of the distortions of both amplitude and phase spectra, and root MSE of F0 (denoted by F0-RMSE), which reflected the distortion of F0. To evaluate the generation efficiency, the real-time factor (RTF), which is defined as the ratio between the time consumed to generate all test sentences using a single Intel Xeon E5-2680 CPU core and the total duration of the test set, was also utilized as an objective metric. Regarding the subjective evaluation, mean opinion score (MOS) tests were conducted to compare the naturalness of the speeches reconstructed by these methods. In each MOS test, twenty test utterances reconstructed by these methods along with the natural utterances were evaluated by at least 30 native English listeners on the crowdsourcing platform of Amazon Mechanical Turk[2] with anti-cheating considerations [20]. Listeners were asked to give a naturalness score between 1 and 5, and the score interval was 0.5.

Both the objective and subjective results are listed in Table 1. Our proposed **NSPP** obtained the highest SNR and the lowest F0-RMSE among all methods. Regarding the subjective results of MOS scores, the **NSPP** also outperformed the other three methods

[1]Source codes are available at `https://github.com/yangai520/NSPP`. Examples of generated speech can be found at `https://yangai520.github.io/NSPP`.

[2]`https://www.mturk.com`.

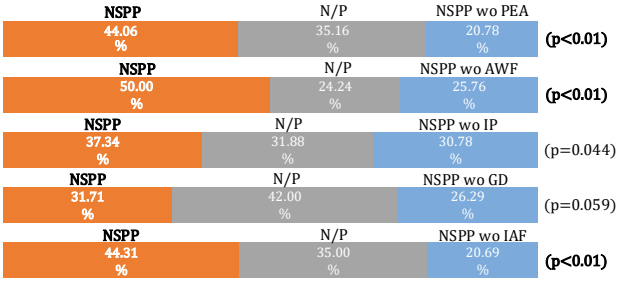

Fig. 3. Average preference scores (%) of ABX tests on speech quality between **NSPP** and its ablated variants, where N/P stands for "no preference" and p denotes the p-value of a t-test between two models.

significantly ($p < 0.01$ of paired t-tests). Besides, the MOS score of the **NSPP** also approached that of the ground truth natural speech (i.e., the **GT** in Table 1), and the difference between the **NSPP** and **GT** was insignificant ($p = 0.55$). These results proved the precise phase prediction ability of our proposed model. Regarding the RTF, our proposed **NSPP** was also an efficient model, reaching 19.6x real-time generation on a CPU. At the same generation speed, the Griffin-Lim algorithm could only iterate 22 rounds (i.e., the **GL22**), and the reconstructed speech quality was far inferior to **NSPP**. The **GL100**, although fully iterated, still performed worse than our proposed **NSPP** due to the audible unnatural artifact sounds. Compared with the **GL100**, the performance of the **DNN+GL100** was significantly improved, which was consistent with the conclusion in the original paper [12, 13]. However, our proposed **NSPP** outperformed the **DNN+GL100** in terms of both the reconstructed speech quality and generation speed. Besides, the **NSPP** was a fully neural network-based method without the extra phase refinement operation, which can be easily implemented. The **NSPP** was also proven to be universal, as it exhibited excellent performance on phase prediction for unseen speakers in the test set. It is also worth mentioning that the training speed of the **NSPP** was also fast, with a training time of 27 hours on this dataset using a single Nvidia 3090Ti GPU.

4.3. Ablation studies

We then conducted several ablation experiments to explore the roles of some key modules in our proposed **NSPP**. The ablated variants of the **NSPP** for comparison included the following:

- **NSPP wo PEA**: Removing the parallel estimation architecture from the **NSPP**. The output of the residual convolutional network passes through a linear layer without activation to predict the phase spectra, which is the same way as used in the von-Mises-distribution DNN-based method [12, 13].
- **NSPP wo AWF**: Removing the anti-wrapping function f_{AW} from the **NSPP** and adopting L1 losses for $\mathcal{L}_{IP}$, $\mathcal{L}_{GD}$ and $\mathcal{L}_{IAF}$ at the training stage.
- **NSPP wo IP**: Removing the instantaneous phase loss $\mathcal{L}_{IP}$ from the **NSPP** at the training stage.
- **NSPP wo GD**: Removing the group delay loss $\mathcal{L}_{GD}$ from the **NSPP** at the training stage.
- **NSPP wo IAF**: Removing the instantaneous angular frequency loss $\mathcal{L}_{IAF}$ from the **NSPP** at the training stage.

We conducted ABX preference tests on the Amazon Mechanical Turk platform to compare the differences between the **NSPP** and its ablated variants. In each ABX test, twenty utterances were randomly selected from the test set reconstructed by two comparative models and evaluated by at least 30 native English listeners. The listeners were asked to judge which utterance in each pair had better speech quality or whether there was no preference. In addition to calculating the average preference scores, the p-value of a t-test was used to measure the significance of the difference between two models.

The ABX test results are shown in Figure 3. As expected, we can see that the **NSPP** outperformed the **NSPP wo PEA** significantly ($p < 0.01$). Specifically, the speech reconstructed by the **NSPP wo PEA** exhibited annoying loud noise similar to electric current, which significantly affected the sense of hearing due to the imprecise phase prediction. One possible reason is that it was difficult for neural networks without the parallel estimation architecture to restrict the range of predicted phases, leading to failure of few anti-wrapping losses. These results indicated that the parallel estimation architecture was essential to wrapped phase prediction. The **NSPP** also outperformed the **NSPP wo AWF** significantly ($p < 0.01$), which proved that the anti-wrapping function was helpful for avoiding the error expansion issue. We also find that the high-frequency energy of the speech reconstructed by the **NSPP wo AWF** was completely suppressed, resulting in an extreme dull listening experience. Interestingly, there were no obvious mispronunciation or F0 distortion in the speech reconstructed by the **NSPP wo AWF** (F0-RMSE=12.0 cent, comparable to that of **NSPP**). For the three losses, removing $\mathcal{L}_{IAF}$ (i.e., **NSPP wo IAF**) led to a significant subjective performance degradation ($p < 0.01$), manifested in the presence of obvious spectral horizontal stripes in the reconstructed speech, causing annoying loud noise. When removing $\mathcal{L}_{IP}$ (i.e., **NSPP wo IP**) and $\mathcal{L}_{GD}$ (i.e., **NSPP wo GD**), the ABX test results show that the subjective differences were slightly insignificant (p was slightly larger than 0.01). However, we found that the reconstructed speech quality of the **NSPP wo IP** and **NSPP wo GD** indeed degraded, because the speech reconstructed by the **NSPP wo IP** exhibited few low-frequency spectrum corruption issues, resulting in F0 distortion and blurry pronunciation (F0-RMSE=21.2 cent, significantly higher than that of **NSPP**), and the **NSPP wo GD** attenuated the overall spectral energy of the reconstructed speech, resulting in a mild dull listening experience.

5. CONCLUSION

In this paper, we have proposed a novel neural speech phase prediction model, which utilizes a residual convolutional network along with a parallel estimation architecture to directly predict the wrapped phase spectra from input amplitude spectra. The parallel estimation architecture is a key module which consists of two parallel linear convolutional layers and a phase calculation formula, strictly restricting the output phase values to the principal value interval. The training criteria of the proposed model is to minimize a combination of the instantaneous phase loss, group delay loss and instantaneous angular frequency loss, which are all activated by an anti-wrapping function to avoid the error expansion issue caused by phase wrapping. Experimental results show that the proposed model outperforms the iterative Griffin-Lim algorithm and the von-Mises-distribution DNN-based method, regarding the reconstructed speech quality. Besides, the proposed model is easy to implement and exhibits a fast training speed and generation speed. Ablation studies demonstrate that the parallel estimation architecture, anti-wrapping function and three losses are all useful. Applying the neural speech phase prediction model to concrete speech generation tasks (e.g., SE, BWE and SS) will be the focus of our future work.

6. REFERENCES

[1] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013, pp. 436–440.

[2] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[3] Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee, "T-GSA: Transformer with gaussian-weighted self-attention for speech enhancement," in *Proc. ICASSP*, 2020, pp. 6649–6653.

[4] Yingxue Wang, Shenghui Zhao, Wenbo Liu, Ming Li, and Jingming Kuang, "Speech bandwidth expansion based on deep neural networks," in *Proc. Interspeech*, 2015, pp. 2593–2597.

[5] Yu Gu, Zhen-Hua Ling, and Li-Rong Dai, "Speech bandwidth extension using bottleneck features and deep recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 297–301.

[6] Kehuang Li, Zhen Huang, Yong Xu, and Chin-Hui Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Proc. Interspeech*, 2015, pp. 2578–2582.

[7] Heiga Zen, Keiichi Tokuda, and Alan W Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[8] Shinji Takaki, Hirokazu Kameoka, and Junichi Yamagishi, "Direct modeling of frequency spectra and waveform generation based on phase recovery for dnn-based speech synthesis.," in *Proc. Interspeech*, 2017, pp. 1128–1132.

[9] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[10] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[11] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[12] Shinnosuke Takamichi, Yuki Saito, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari, "Phase reconstruction from amplitude spectrograms based on von-mises-distribution deep neural network," in *Proc. IWAENC*, 2018, pp. 286–290.

[13] Shinnosuke Takamichi, Yuki Saito, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari, "Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks," *Signal Processing*, vol. 169, pp. 107368, 2020.

[14] Yoshiki Masuyama, Kohei Yatabe, Yuma Koizumi, Yasuhiro Oikawa, and Noboru Harada, "Phase reconstruction based on recurrent phase unwrapping with deep neural networks," in *Proc. ICASSP*, 2020, pp. 826–830.

[15] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, 2017, pp. 933–941.

[16] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, p. 3.

[17] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al., "Superseded-CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2016.

[18] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2018.

[19] Yang Ai and Zhen-Hua Ling, "A neural vocoder with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 839–851, 2020.

[20] Sabine Buchholz and Javier Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *Proc. Interspeech*, 2011, pp. 3053–3056.