Low-Latency Neural Speech Phase Prediction based on Parallel Estimation Architecture and Anti-Wrapping Losses for Speech Generation Tasks

Yang Ai, *Member, IEEE*, Zhen-Hua Ling, *Senior Member, IEEE*

***Abstract*—This paper presents a novel neural speech phase prediction model which predicts wrapped phase spectra directly from amplitude spectra. The proposed model is a cascade of a residual convolutional network and a parallel estimation architecture. The parallel estimation architecture is a core module for direct wrapped phase prediction. This architecture consists 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. We mathematically demonstrate that the anti-wrapping function should possess three properties, namely parity, periodicity and monotonicity. We also achieve low-latency streamable phase prediction by combining causal convolutions and knowledge distillation training strategies. For both analysis-synthesis and specific speech generation tasks, experimental results show that our proposed neural speech phase prediction model outperforms the iterative phase estimation algorithms and neural network-based phase prediction methods in terms of phase prediction precision, efficiency and robustness. Compared with HiFi-GAN-based waveform reconstruction method, our proposed model also shows outstanding efficiency advantages while ensuring the quality of synthesized speech. To the best of our knowledge, we are the first to directly predict speech phase spectra from amplitude spectra only via neural networks.**

***Index Terms*—speech phase prediction, parallel estimation architecture, anti-wrapping loss, low-latency, speech generation**

I. 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) [2]–[4], bandwidth extension (BWE) [5]–[7] and speech synthesis (SS) [8]–[13], 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. In addition to phase prediction precision, the efficiency, latency and robustness are also important metrics for evaluating phase prediction methods. The efficiency represents the generation speed and is a criterion for determining real-time performance. The latency refers to the duration of future input that are necessary for predicting current output. High efficiency and low latency are strict requirements for many practical application scenarios such as telecommunication. The robustness reflects the general applicability of phase prediction methods when faced with inputs of varying amplitude spectra.

In the early days, researchers mainly focused on iterative phase estimation algorithms, of which the Griffin-Lim algorithm (GLA) [14] is one of the most well-known algorithms. The GLA is based on alternating projection and iteratively estimates the phase spectra from amplitude spectra via the short-time Fourier transform (STFT) and inverse STFT (ISTFT). Due to its ease of implementation, the GLA has been widely used in speech generation tasks [11]–[13]. However, the GLA always causes unnatural artifacts in the reconstructed speech, meaning that there is still a large gap between the estimated phase and natural phase. Hence, several improved algorithms, such as the fast Griffin-Lim algorithm (FGLA) [15] and alternating direction method of multipliers (ADMM) [16], have also been also proposed to boost the performance of the GLA. Recently, Kobayashi *et al.* [17] applied three alternating reflection-based iterative algorithms developed in the optics community, i.e., the averaged alternating reflections (AAR) [18], [19], relaxed AAR (RAAR) [20], and hybrid input-output (HIO) [21]–[23] algorithms, to acoustic applications and clearly outperformed GLA families. However, these iterative algorithms always limit the reconstructed speech quality due to the influence of the initial phase and exhibit poor robustness when estimating phase spectra from degraded amplitude spectra. Besides, these iterative algorithms often require the amplitude spectra of an entire utterance as input, resulting in high latency.

With the development of deep learning, several neural network-based phase prediction methods have been gradually proposed. We roughly divide them into three categories of methods. The first is the GLA simulation method [24]–

This work is the extended version of our conference paper [1] published at 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023).

This work was funded by the National Nature Science Foundation of China under Grant 62301521 and U23B2053, the Anhui Provincial Natural Science Foundation under Grant 2308085QF200, and the Fundamental Research Funds for the Central Universities under Grant WK2100000033.

Y. Ai and Z.-H. Ling are with the National Engineering Research Center of Speech and Language Information Processing, University of Science and Technology of China, Hefei, 230027, China (e-mail: yangai@ustc.edu.cn, zhling@ustc.edu.cn).

Corresponding author: Zhen-Hua Ling.

[26]. For example, Masuyama *et al.* [25], [26] proposed deep Griffin-Lim iteration (DeGLI), which employs trainable neural networks to simulate the GLA process and achieve iterative phase reconstruction. However, the prediction target of such methods is the complex spectra rather than the phase spectra. The second is the two-stage method [27]–[29]. For example, Masuyama *et al.* [27] first predicted phase derivatives (i.e., the group delay and instantaneous frequency) by two parallel deep neural networks (DNNs), and then the phase was recursively estimated by a recurrent phase unwrapping (RPU) algorithm [30] from the predicted phase derivatives. Prior-distribution-aware method is the last category [31], [32]. Takamichi *et al.* [31], [32] assume that the phase follows a specific prior distribution (i.e., the von Mises distribution and sine-skewed generalized cardioid distribution) and employ a DNN to predict the distribution parameters of the phase. However, the phase predicted by the DNN still needs to be refined using the GLA. Obviously, all the abovementioned phase prediction methods need to combine neural networks with some convolutional iterative algorithms, which inevitably leads to cumbersome operations, increased complexity, low efficiency and high latency.

In addition to the phase prediction methods mentioned above, several speech waveform reconstruction methods, such as vocoders [33], also include implicit and indirect phase prediction within waveform synthesis. Recently, HiFi-GAN [34] vocoder has demonstrated exceptional performance on reconstructed waveform and has been widely applied in speech synthesis. The HiFi-GAN cascades multiple upsampling layers and residual convolution networks to gradually upsample the input mel spectrograms to the sampling rate of the final waveform while performing non-causal residual convolutional operations. Hence, the phase prediction is implicitly incorporated within waveform prediction. The HiFi-GAN utilizes adversarial losses [35] defined on the waveform, ensuring the generation of high-fidelity waveforms. However, HiFi-GAN still has limitations in terms of generation efficiency, training efficiency and latency, due to the direct waveform prediction, adversarial training and non-causal convolutions, respectively. To our knowledge, predicting speech wrapped phase spectra directly from amplitude spectra using only 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 direct 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. Due to the periodic nature and wrapping property of the phase, some conventional loss functions, such as L1 loss and mean square error (MSE), are disabled for phase prediction and cause error expansion issue. 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. These losses are defined between the phase spectra predicted by the model and the natural ones. The anti-wrapping function calculates the true error between the predicted value and the natural value, and we demonstrate that the function requires three properties, i.e., parity, periodicity, and monotonicity. We have also employed knowledge distillation to train an all-causal convolution-based neural phase prediction model. This approach has enabled us to achieve precise streamable phase prediction, thereby facilitating its effective utilization in low-latency scenarios. Experimental results show that our proposed model outperforms the GLA [14], RAAR [17] and von Mises distribution DNN-based phase prediction method [31], [32], in terms of both phase prediction precision and efficiency. Compared with the HiFi-GAN vocoder-based waveform reconstruction method [34], our proposed model demonstrates a significant efficiency advantage while maintaining the same quality of synthesized speech. Our proposed model exhibits near-natural reconstructed speech quality according to the mean opinion score (MOS) results and reaches 19.6x real-time generation on a CPU with low latency. When the proposed method is initially applied to specific speech generation tasks (i.e., using degraded amplitude spectra as input), it shows better stability and robustness than iterative algorithms. Ablation studies also certify that the parallel estimation architecture and anti-wrapping losses are extremely important for successful phase prediction.

The main contribution of this work is the realization of direct prediction of the wrapped phase spectra only by neural networks with high prediction precision, high generation efficiency, low latency and high robustness. Our proposed model is easy to implement, simple to operate and adaptable to integrate into specific speech generation tasks, such as SE, BWE and SS, due to its trainable property.

This paper is organized as follows. In Section II, we briefly review the representative iterative speech phase estimation algorithms and neural network-based speech phase prediction methods. In Section III, we provide details of the model structure, training criteria and improvements made for low-latency streamable inference of the proposed neural speech phase prediction model. In Section IV, we present our experimental results. Finally, we give conclusions in Section V.

II. Related Works

In this section, we briefly introduce two iterative speech phase estimation algorithms (i.e., the GLA [14] and RAAR [17]) and a neural network-based speech phase prediction method (i.e., the von Mises distribution DNN-based method [31], [32]). They are compared with our proposed neural phase prediction model in Section IV.

A. Griffin-Lim Algorithm (GLA)

The GLA [14] is an alternating projection algorithm and iteratively estimates the phase spectra from amplitude spectra via the STFT and 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 $\hat{\boldsymbol{P}}^{[0]} \in \mathbb{R}^{F\times N}$ to zero matrix, i.e., the initial complex spectrum $\hat{\boldsymbol{S}}^{[0]} = \boldsymbol{A}\odot e^{j\hat{\boldsymbol{P}}^{[0]}} = \boldsymbol{A}$, where $\odot$ represents the element-wise multiplication. Finally iterate the following formula from $i = 1$ to I:

$$\hat{\boldsymbol{S}}^{[i]} = P_C(P_A(\hat{\boldsymbol{S}}^{[i-1]})), \tag{1}$$

where I is the total number of iterations. P_C and P_A are two core projection operators defined as follows:

$$P_C(\boldsymbol{X}) = STFT(ISTFT(\boldsymbol{X})), \tag{2}$$

$$P_A(\boldsymbol{X}) = \boldsymbol{A} \odot \boldsymbol{X} \oslash |\boldsymbol{X}|, \tag{3}$$

where $\boldsymbol{X} \in \mathbb{C}^{F\times N}$. $\oslash$ and $|\cdot|$ represent the element-wise division and amplitude calculation, respectively. The final estimated phase spectrum $\hat{\boldsymbol{P}}^{[I]} \in \mathbb{R}^{F\times N}$ is contained in the complex spectrum $\hat{\boldsymbol{S}}^{[I]} \in \mathbb{C}^{F\times N}$. The final speech waveform is reconstructed from $\hat{\boldsymbol{S}}^{[I]}$ by ISTFT. The GLA can be easily implemented and is popular in speech generation tasks. Since the GLA 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. Besides, the GLA also tends to limit the phase estimation efficiency due to its iterative estimation mode and extend the latency due to its whole-utterance estimation mode.

B. Relaxed Averaged Alternating Reflection (RAAR)

The RAAR was originally developed in the optics community, and was recently successfully applied in the field of speech phase estimation by Kobayashi *et al.* [17]. The RAAR is an alternating reflection algorithm and also iteratively estimates the phase spectra from the amplitude spectra. The core reflection operators R_C and R_A for the RAAR are designed based on the projection operators P_C and P_A as follows:

$$R_C(\boldsymbol{X}) = 2P_C(\boldsymbol{X}) - \boldsymbol{X}, \tag{4}$$

$$R_A(\boldsymbol{X}) = 2P_A(\boldsymbol{X}) - \boldsymbol{X}. \tag{5}$$

The RAAR adopts the same initialization manner as the GLA and then iteratively executes the following formula from $i = 1$ to I:

$$\hat{\boldsymbol{S}}^{[i]} = \frac{\beta}{2}\hat{\boldsymbol{S}}^{[i-1]} + R_C(R_A(\hat{\boldsymbol{S}}^{[i-1]})) + (1-\beta)P_A(\hat{\boldsymbol{S}}^{[i-1]}), \tag{6}$$

where $0 < \beta < 1$ is a relaxation parameter.

In the original paper [17], Kobayashi *et al.* have proven that the RAAR with $\beta = 0.9$ is an excellent speech phase estimation algorithm which outperforms the GLA families and other alternating reflection algorithms (i.e., the AAR and HIO). However, the iterative formula of the RAAR is more complicated than that of the GLA, which inevitably inhibits the generation efficiency and latency.

C. Von Mises Distribution DNN-based Method

The von Mises distribution DNN-based method [31], [32] realizes phase prediction by combining neural networks and GLA. It 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 a gated linear unit (GLU) [36] 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 GLA with 100 iterations.

In the original paper [31], [32], Takamichi *et al.* have proven that the von Mises distribution DNN-based method significantly outperforms the plain GLA. They also evaluate the effect of the GLA phase refinement, and the experimental results show that the refinement operation is necessary because the phase predicted by the DNN directly is unsatisfactory.

III. Proposed Method

In this section, we give details on the model structure and training criteria of our proposed neural speech phase prediction model and improvements for low-latency streamable phase prediction through knowledge distillation training strategy as illustrated in Figures 1 and 2.

A. 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 non-causal residual convolutional network (RCNet) and a parallel estimation architecture.

As shown in Figure 2(a), the non-causal RCNet utilizes multiple non-causal convolutional layers to effectively broaden the receptive field, thus ensuring precise restoration of the phase. The input log amplitude spectrum sequentially passes through a linear non-causal convolutional layer (kernel size $= k_0$ and channel size $= C$) and P parallel non-causal residual convolutional blocks (RCBlocks), all of which have the same input. Then, the outputs of these P RCBlocks are summed (i.e., skip connections), averaged, and finally activated by a leaky rectified linear unit (LReLU) [37]. Each RCBlock is formed by a cascade of Q non-causal sub-RCBlocks. In the q-th sub-RCBlock of the p-th RCBlock ($p = 1, \ldots, P$ and $q = 1, \ldots, Q$), the input is first activated by an LReLU, then passes through a linear non-causal dilated convolutional layer (kernel size $= k_p$, channel size $= C$ and dilation factor $= d_{p,q}$), then is activated by an LReLU again, passes through a linear non-causal convolutional layer (kernel size $= k_p$ and channel size $= C$), and finally superimposes with the input (i.e., residual connections) to obtain the output.

The parallel estimation architecture is a core module for the direct prediction of wrapped phases. It is inspired by the

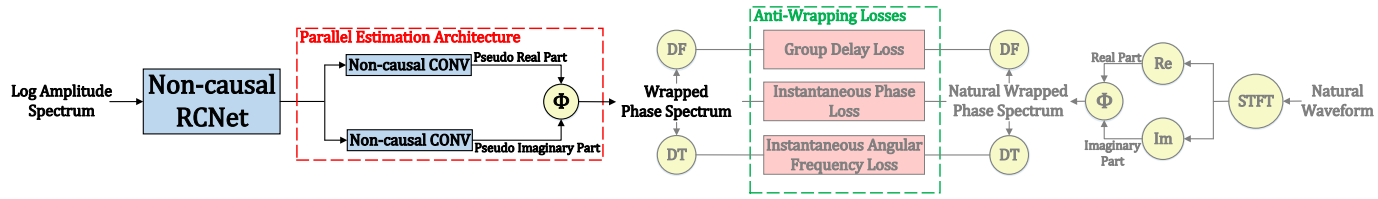

Fig. 1. Details of the proposed neural speech phase prediction model. Here, *RCNet*, *CONV*, *STFT*, *DF*, *DT*, *Re*, *Im* and Φ represent the residual convolutional network, linear convolutional layer, short-time Fourier transform, differential along frequency axis, differential along time axis, real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

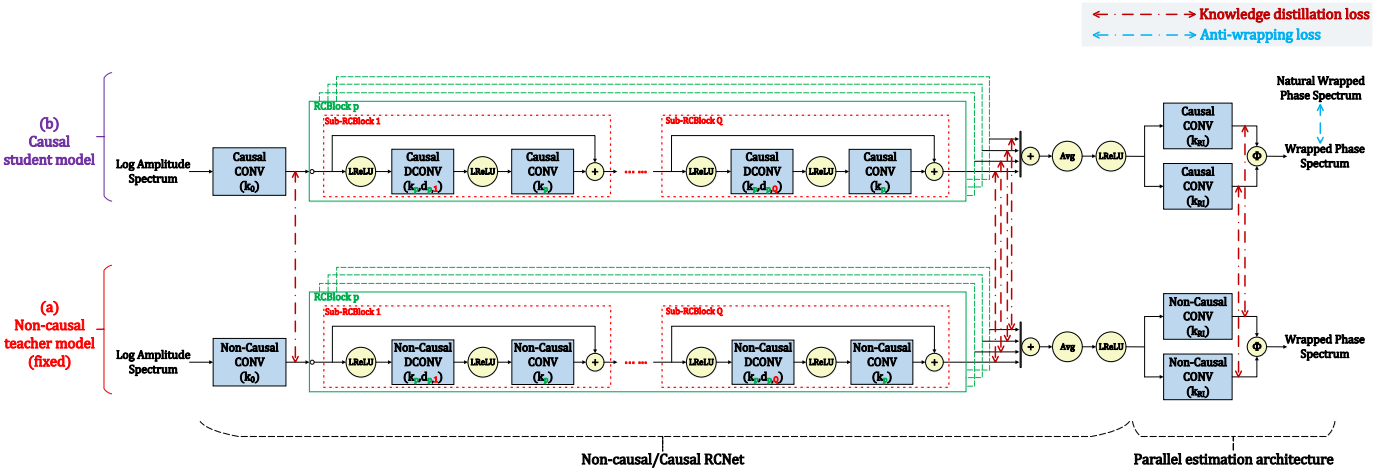

Fig. 2. Details of the residual convolutional network and the training procedure of low-latency streamable neural speech phase prediction model through knowledge distillation. Here, subfigure (a) represents a non-causal teacher model which is consistent with Figure 1. Subfigure (b) represents a causal student model. *RCNet*, *CONV*, *DCONV* and Φ represent the residual convolutional network, linear convolutional layer, linear dilated convolutional layer and phase calculation formula, respectively. k_* and $d_{*,*}$ denotes kernel size and dilation factor, respectively.

process of calculating the phase spectra from the real and imaginary parts of complex spectra and consists of two parallel linear non-causal convolutional layers (kernel size $= k_{RI}$ and channel size $= N$ for both layers) and a phase calculation formula $\mathbf{\Phi}$. We call the outputs of the two parallel layers as 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 $\mathbf{\Phi}$ as follows:

$$\hat{\boldsymbol{P}} = \mathbf{\Phi}(\hat{\boldsymbol{R}}, \hat{\boldsymbol{I}}). \tag{7}$$

Equation 7 is calculated element-wise. For $\forall R \in \mathbb{R}$ and $I \in \mathbb{R}$, we define

$$\mathbf{\Phi}(R, I) = \arctan\left(\frac{I}{R}\right) - \frac{\pi}{2} \cdot Sgn^*(I) \cdot [Sgn^*(R) - 1], \tag{8}$$

and $\mathbf{\Phi}(0, 0) = 0$. Sgn^* is a symbolic function defined as:

$$Sgn^*(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}. \tag{9}$$

Therefore, the range of values for the phase is $-\pi < \mathbf{\Phi}(R, I) \leq \pi$, meaning that the phase predicted by our model is wrapped and strictly restricted to the phase principal value interval. Obviously, the phase value does not depend on the absolute values of the pseudo real and imaginary parts but on their relative ratios and signs.

B. 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 3, 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

$$e = \min\{|\hat{P} - P|, 2\pi - |\hat{P} - P|\}. \tag{10}$$

For example, in Figure 3, 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 10 can be written in another form:

$$e = \left|\hat{P} - P - 2\pi \cdot round\left(\frac{\hat{P} - P}{2\pi}\right)\right|, \tag{11}$$

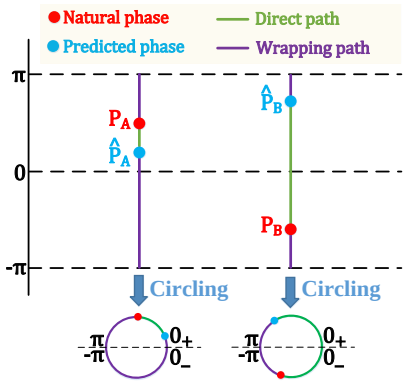

Fig. 3. An illustration explanation of the error expansion issue caused by phase wrapping.

where $round$ represents rounding. Obviously, Equation 11 is a function of error $\hat{P} - P$. We define a function $f_{line}(x)$ as follows:

$$f_{line}(x) = \left| x - 2\pi \cdot round\left(\frac{x}{2\pi}\right)\right|, x \in \mathbb{R}. \tag{12}$$

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

As shown in Figure 4(a), we draw the graph of the anti-wrapping function $f_{line}(x)$. Obviously, $f_{line}(x)$ is an even function with a period of 2π and exhibits monotonicity over half-periods. Actually, any function $f(x)$ that satisfies below parity, periodicity and monotonicity at the same time can be used as an anti-wrapping function to activate the direct error x and define loss between the predicted value and natural value.

- **Parity**: The anti-wrapping function $f(x)$ must be an even function because our goal is to promote the predicted value to approximate the natural value but ignore in which direction it is approximated.
- **Periodicity**: The anti-wrapping function $f(x)$ must be a periodic function with period 2π because this periodicity cleverly avoids the problem of error expansion caused by phase wrapping.
- **Monotonicity**: The anti-wrapping function $f(x)$ must be monotonically increasing in interval $[0, \pi]$ because the monotonicity ensures that the larger the true error $\left|x - 2\pi \cdot round\left(\frac{x}{2\pi}\right)\right|$, the larger the loss $f(x)$, which conforms to the definition rules of the loss function.

Figure 4(b)-(e) plot several typical convex anti-wrapping functions. Compared with the linear function $f_{line}(x)$, the rate of change of a convex function may be different at different error values x, thereby prompting the model to pay more attention to or ignore certain ranges of error values. In Section IV-F1, we will further explore the effect of different anti-wrapping functions on model performance through experiments.

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

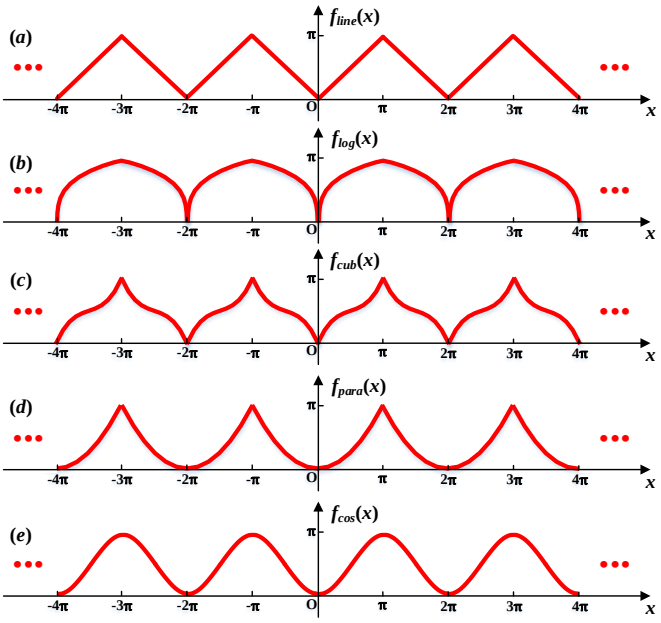

Fig. 4. Graphs of five typical anti-wrapping functions, including (a) linear function; (b) logarithmic function; (c) cubic function; (d) parabolic function and (e) cosine function.

as follows:

$$\mathcal{L}_{IP} = \mathbb{E}_{(\hat{\boldsymbol{P}}, \boldsymbol{P})} \overline{f\left(\hat{\boldsymbol{P}} - \boldsymbol{P}\right)}, \tag{13}$$

where $f(\boldsymbol{X})$ 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 through STFT, respectively. To ensure the continuity of the predicted wrapped phase spectrum along the frequency and time axes, we also define the group delay (GD) loss $\mathcal{L}_{GD}$ and instantaneous angular frequency (IAF) loss $\mathcal{L}_{IAF}$, which are both activated by the anti-wrapping function f to avoid the error expansion issue as follows:

$$\mathcal{L}_{GD} = \mathbb{E}_{(\Delta_{DF}\hat{\boldsymbol{P}}, \Delta_{DF}\boldsymbol{P})} \overline{f\left(\Delta_{DF}\hat{\boldsymbol{P}} - \Delta_{DF}\boldsymbol{P}\right)}, \tag{14}$$

$$\mathcal{L}_{IAF} = \mathbb{E}_{(\Delta_{DT}\hat{\boldsymbol{P}}, \Delta_{DT}\boldsymbol{P})} \overline{f\left(\Delta_{DT}\hat{\boldsymbol{P}} - \Delta_{DT}\boldsymbol{P}\right)}, \tag{15}$$

where Δ_{DF} and Δ_{DT} represent the differential along the frequency axis and time axis, respectively. Specifically, in Equation 14, we have

$$\Delta_{DF}\hat{\boldsymbol{P}} = \hat{\boldsymbol{P}}\boldsymbol{W}, \tag{16}$$

$$\Delta_{DF}\boldsymbol{P} = \boldsymbol{P}\boldsymbol{W}, \tag{17}$$

and

$$\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n, \ldots, \boldsymbol{w}_N], \tag{18}$$

$$\boldsymbol{w}_n = \left[\underbrace{0, \ldots, 0}_{1\text{st}}, \underbrace{1}_{n\text{-th}}, \underbrace{-1, 0, \ldots, 0}_{N\text{-th}}\right]^\top. \tag{19}$$

In Equation 15, we have

$$\Delta_{DT}\hat{\boldsymbol{P}} = \boldsymbol{V}\hat{\boldsymbol{P}}, \tag{20}$$

$$\Delta_{DT}\boldsymbol{P} = \boldsymbol{V}\boldsymbol{P}, \tag{21}$$

and

$$\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_f, \ldots, \boldsymbol{v}_F]^\top, \tag{22}$$

$$\boldsymbol{v}_f = \left[\underset{\text{1st}}{0}, \ldots, 0, \underset{f\text{-th}}{1}, -1, 0, \ldots, \underset{F\text{-th}}{0}\right]^\top. \tag{23}$$

Finally, the training criteria of our proposed neural speech phase prediction model are to minimize the final loss

$$\mathcal{L} = \mathcal{L}_{IP} + \mathcal{L}_{GD} + \mathcal{L}_{IAF}. \tag{24}$$

At the generation stage, first, the well-trained neural speech phase prediction model uses the log amplitude spectrum $\log \boldsymbol{A}$ as input and predicts the wrapped phase spectrum $\hat{\boldsymbol{P}}$. Then, the amplitude spectrum $\boldsymbol{A}$ and predicted phase spectrum $\hat{\boldsymbol{P}}$ are combined to a complex spectrum, and finally, the complex spectrum is converted to a waveform $\hat{\boldsymbol{x}}$ through ISTFT, i.e.,

$$\hat{\boldsymbol{x}} = ISTFT\left(\boldsymbol{A} \odot e^{j\hat{\boldsymbol{P}}}\right). \tag{25}$$

C. Low-Latency Streamable Phase Prediction by Causal Convolution and Knowledge Distillation

Some application scenarios have strict requirement on the latency and streamable inference mode such as real-time voice communication. The latency indicates the minimum amount of time needed for the model to initiate its operations. The proposed neural speech phase prediction model incorporates non-causal convolutions to enhance its modeling capacity. However, this inevitably results in increased latency. For a non-causal convolution operation with a kernel size of k and dilation factor of d, the number of future input samples required is

$$\zeta(k, d) = \left\lfloor \frac{(k-1)d}{2} \right\rfloor, \tag{26}$$

where $\lfloor \cdot \rfloor$ denotes flooring. Therefore, the latency measured in milliseconds of the proposed model is

$$l_{NSPP} = \{\zeta(k_0, 1) + \max_{p=1,\ldots,P} \left[\sum_{q=1}^{Q} \zeta(k_p, d_{p,q}) + Q\zeta(k_p, 1)\right] + \zeta(k_{RI}, 1)\} \cdot w_s, \tag{27}$$

where w_s is the window shift in milliseconds of the amplitude and phase spectra. It can be seen that when the window shift is long and the kernel size and dilation factor of convolutional layers are large, it will result in significant latency, which is undesirable in low-latency scenarios.

Therefore, as shown in Figure 2(b), we design a causal neural speech phase prediction model which can support low-latency streamable inference. It replaces all non-causal convolutions in the non-causal model as shown in Figure 2(a) with causal convolutions. Notably, the inference process of the causal model requires at least one frame of log amplitude spectrum input to initiate, thus, it incurs an inevitable latency equal to the window size (i.e., low latency). However, the use of causal convolutions, which cannot leverage future information, will inevitably lead to a reduction in phase prediction precision, despite achieving low latency. To bridge the gap between causal and non-causal models, we propose a knowledge distillation training strategy in which a non-causal teacher model guides the training of a causal student model. Specifically, we first train a non-causal neural speech phase prediction model (i.e., the teacher model) using the anti-wrapping loss depicted in Equation 24. Then, the non-causal teacher model fixes its parameters and provide training objectives for the causal neural speech phase prediction model (i.e., the student model). We define the output of the input convolutional layer, the outputs of P RCBlocks, the pseudo real part and the pseudo imaginary part of the student model as $\hat{\boldsymbol{O}}^I \in \mathbb{R}^{F\times C}$, $\hat{\boldsymbol{O}}_p^{RCB} \in \mathbb{R}^{F\times C}(p = 1, \ldots, P)$, $\hat{\boldsymbol{O}}^{PRP} \in \mathbb{R}^{F\times N}$ and $\hat{\boldsymbol{O}}^{PIP} \in \mathbb{R}^{F\times N}$, respectively. The outputs of the teacher model at corresponding positions are respectively denoted as $\tilde{\boldsymbol{O}}^I$, $\tilde{\boldsymbol{O}}_p^{RCB}$, $\tilde{\boldsymbol{O}}^{PRP}$ and $\tilde{\boldsymbol{O}}^{PIP}$. The knowledge distillation loss is defined as follows:

$$\begin{aligned}\mathcal{L}_{KD} = {} & \mathbb{E}_{(\hat{\boldsymbol{O}}^I, \tilde{\boldsymbol{O}}^I)} \overline{\left(\hat{\boldsymbol{O}}^I - \tilde{\boldsymbol{O}}^I\right)^2} \\ & + \sum_{p=1}^{P} \mathbb{E}_{(\hat{\boldsymbol{O}}_p^{RCB}, \tilde{\boldsymbol{O}}_p^{RCB})} \overline{\left(\hat{\boldsymbol{O}}_p^{RCB} - \tilde{\boldsymbol{O}}_p^{RCB}\right)^2} \\ & + \mathbb{E}_{(\hat{\boldsymbol{O}}^{PRP}, \tilde{\boldsymbol{O}}^{PRP})} \overline{\left(\hat{\boldsymbol{O}}^{PRP} - \tilde{\boldsymbol{O}}^{PRP}\right)^2} \\ & + \mathbb{E}_{(\hat{\boldsymbol{O}}^{PIP}, \tilde{\boldsymbol{O}}^{PIP})} \overline{\left(\hat{\boldsymbol{O}}^{PIP} - \tilde{\boldsymbol{O}}^{PIP}\right)^2}.\end{aligned} \tag{28}$$

The training target of the student model is to minimize a combination of the anti-wrapping loss and knowledge distillation loss, i.e.,

$$\mathcal{L}_{Student} = \mathcal{L}_{IP} + \mathcal{L}_{GD} + \mathcal{L}_{IAF} + \alpha_{KD}\mathcal{L}_{KD}, \tag{29}$$

where α_{KD} is a hyperparameter. Through training, the causal student model aims to approach the phase prediction capability of the non-causal teacher model while maintaining its advantage of low latency and streamable inference.

IV. Experiments

A. Data and Feature Configuration

A subset of the VCTK corpus [38] was adopted in our experiments[1]. 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 (i.e., $w_s = 5$), and the FFT point number was 1024 (i.e., $N = 513$).

B. Speech Generation Tasks

In our experiments, we apply contrastive phase prediction methods to the analysis-synthesis task and two specific speech generation tasks, including the BWE task and SS task. Figure

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

5 draws a simple flowchart of three tasks. Specifically, the detailed description of these three tasks is as follows.

1) Analysis-Synthesis Task: As shown in Figure 5, the analysis-synthesis task just recovered the 513-dimensional phase spectrum from the 513-dimensional natural amplitude spectrum by phase prediction methods and reconstructed the waveform by ISTFT.

2) BWE Task: As shown in Figure 5, the BWE task first adopted an amplitude extension model to predict the 256-dimensional high-frequency amplitude spectrum from the 257-dimensional low-frequency amplitude spectrum. Then, the 513-dimensional full-band amplitude spectrum was built by concatenating the low- and high-frequency amplitude spectra. Finally, the 513-dimensional phase spectrum was recovered from the full-band amplitude spectrum by phase prediction methods and the waveform was reconstructed by ISTFT. Here, the amplitude extension model was borrowed from our previous work [39] and included 2 bidirectional gated recurrent unit (GRU)-based recurrent layers, each with 1024 nodes (512 forward ones and 512 backward ones), 2 convolutional layers, each with 2048 nodes (filter width=9), and a feed-forward linear output layer with 256 nodes. The generative adversarial network (GAN) with two discriminators which conducted convolution along the frequency and time axis [39] was applied to the amplitude extension model at the training stage.

3) SS Task: For the SS task, we designed a neural vocoder framework with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis (SPSS). As shown in Figure 5, the vocoder framework first used an amplitude prediction model to complete the mapping from the 80-dimensional mel spectrogram to the 513-dimensional amplitude spectrum. Then, the 513-dimensional phase spectrum was recovered from the amplitude spectrum by phase prediction methods and the waveform was reconstructed by ISTFT. Here, the amplitude prediction model adopted the same structure as that used in the BWE task, except that the number of nodes in the feed-forward linear output layer was 513.

C. Comparison among Phase Prediction Methods

We conducted objective and subjective experiments to compare the performance of our proposed neural speech phase prediction model and other phase prediction methods for the analysis-synthesis task, BWE task and SS task. The descriptions of the phase prediction methods for comparison are as follows:

- **NSPP**: The proposed neural speech phase prediction model with latency as shown in Figure 1. In the non-causal RCNet, the kernel size of the input linear convolutional layer was $k_0 = 7$. There were 3 parallel RCBlocks (i.e., $P = 3$) in the RCNet, and each RCBlock was formed by concatenating 3 sub-RCBlocks (i.e., $Q = 3$). The kernel sizes of RCBlocks were $k_1 = 3$, $k_2 = 7$ and $k_3 = 11$, and the dilation factors of sub-RCBlocks within each RCBlock were $d_{*,1} = 1$, $d_{*,2} = 3$ and $d_{*,3} = 5$. The channel size of all the convolutional operations in the RCNet was $C = 512$. In the parallel estimation architecture, the kernel size of two parallel linear convolutional layers was $k_{RI} = 7$. We used the linear anti-wrapping function $f_{line}(x)$ as shown in Figure 4(a) at the training stage. $f_{line}(x)$ was given in Equation 12. The model was trained using the AdamW optimizer [40] with $\beta_1 = 0.8$ and $\beta_2 = 0.99$ on a single Nvidia 3090Ti GPU until 3100 epochs. 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. Based on the current configuration, the **NSPP** exhibited a latency of 330 ms, as calculated using Equation 27.
- **GLn**: The GLA [14] mentioned in Section II-A with n iterations ($n = 22$ and $n = 100$ were used in the experiments). The GLA required the amplitude spectra of an entire utterance as input, thus the latency of the **GLn** equaled to utterance length T in milliseconds.
- **RAARn**: The RAAR [17] mentioned in Section II-B with n iterations ($n = 13$ and $n = 100$ were used in the experiments). Same as the **GLn**, the latency of the **RAARn** was also equal to T.
- **DNN+GL100**: The von Mises distribution DNN-based phase prediction method [31], [32] mentioned in Section II-C. The phase spectra were first predicted by the DNN and then refined by the GLA with 100 iterations. We reimplemented it ourselves. The training configuration of the DNN is the same as that of **NSPP**. As mentioned in Section II-C, the DNN adopted the amplitude spectra at current and ± 2 frames, resulting in the latency of $2w_s$=10 ms. The latency of the **DNN+GL100** corresponded to the maximum value between the latencies of the DNN and GLA, i.e., $max\{10, T\}$.

To objectively evaluate the phase prediction precision, we calculated the average IP, GD and IAF losses on the test set. To objectively evaluate the reconstructed speech quality, two objective metrics used in our previous work [41] were adopted here, 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 [42]. Listeners were asked to give a naturalness score between 1 and 5, and the score interval was 0.5.

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

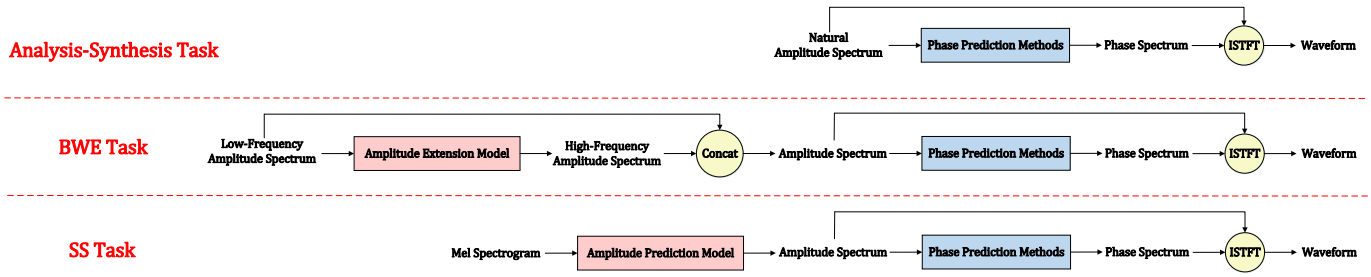

Fig. 5. A simple flowchart of the analysis-synthesis task, BWE task and SS task. Here, *Concat* and *ISTFT* represent concatenation and inverse short-time Fourier transform, respectively.

TABLE I
OBJECTIVE AND SUBJECTIVE EVALUATION RESULTS AMONG PHASE PREDICTION METHODS FOR THE ANALYSIS-SYNTHESIS TASK. HERE, "$a\times$" REPRESENTS $a\times$ REAL TIME.

	SNR(dB)↑	F0-RMSE(cent)↓	IP loss↓	GD loss↓	IAF loss↓	MOS↑	RTF↓
Natural Speech	–	–	–	–	–	3.93±0.063	–
NSPP	**8.26**	**10.0**	**1.479**	0.297	0.694	**3.86±0.065**	**0.051 (19.6×)**
GL22	2.70	66.4	1.570	0.302	0.768	2.07±0.073	0.053 (18.9×)
RAAR13	2.00	97.5	1.570	0.546	0.871	1.89±0.065	0.054 (18.5×)
GL100	3.35	32.5	1.569	0.218	0.505	3.46±0.074	0.23 (4.48×)
RAAR100	4.66	11.0	1.567	**0.179**	**0.271**	**3.89±0.065**	0.40 (2.48×)
DNN+GL100	5.03	13.2	1.537	0.209	0.484	3.70±0.068	0.29 (3.45×)

TABLE II
OBJECTIVE AND SUBJECTIVE EVALUATION RESULTS AMONG PHASE PREDICTION METHODS FOR THE BWE TASK.

	SNR(dB)↑	F0-RMSE(cent)↓	MOS↑
Natural Speech	–	–	4.15±0.050
NSPP	**8.18**	**10.8**	**4.09±0.052**
GL100	3.24	32.6	3.90±0.069
RAAR100	4.49	11.0	**4.10±0.053**
DNN+GL100	5.03	13.2	4.02±0.059

TABLE III
OBJECTIVE AND SUBJECTIVE EVALUATION RESULTS AMONG PHASE PREDICTION METHODS FOR THE SS TASK.

	SNR(dB)↑	F0-RMSE(cent)↓	MOS↑
Natural Speech	–	–	3.84±0.051
NSPP	**6.75**	**19.0**	**3.73±0.055**
GL100	3.14	39.4	3.50±0.068
RAAR100	3.92	22.7	3.64±0.061
DNN+GL100	4.02	22.5	3.66±0.062

For the analysis-synthesis task, BWE task and SS task, both the objective and subjective results are listed in Table I, Table II and Table III, respectively. Our proposed **NSPP** obtained the highest SNR and the lowest F0-RMSE among all methods for all three tasks. The IP loss, GD loss and IAF loss are only calculated for the analysis-synthesis task. Our proposed **NSPP** obtained the lowest IP loss but felled behind in two other metrics when compared to iterative algorithms. This indicates that our proposed model primarily achieved precise phase prediction by improving the IP loss compared with other methods. In our experiments, we discovered that reducing IP loss is challenging, which can be attributed to the sensitivity of instantaneous phase to waveform shifts [27]. Regarding the RTF results shown in Table I, our proposed **NSPP** was also an efficient model, reaching 19.6x real-time generation on a CPU. At the same generation speed, the GLA and RAAR could only iterate 22 rounds and 13 rounds (i.e., **GL22** and **RAAR13**), respectively, and their reconstructed speech quality was far inferior to that of **NSPP**. 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. Regarding the subjective results, the MOS score of the **NSPP** approached that of the natural speech for the analysis-synthesis task as shown in Table I, and the difference between the **NSPP** and **Natural Speech** was slightly insignificant ($p = 0.055$ of paired t-tests). The **GL100**, although fully iterated, still performed significantly worse than our proposed **NSPP** ($p < 0.01$) for all three tasks due to the audible unnatural artifact sounds. Compared with the **GL100**, the performance of the **DNN+GL100** was significantly improved ($p < 0.01$), which was consistent with the conclusion in the original paper [31], [32]. Nevertheless, our proposed **NSPP** still outperformed **DNN+GL100** in terms of both the reconstructed speech quality and generation speed for all three tasks. These results proved the precise phase prediction ability of our proposed model. Besides, compared with the **DNN+GL100**, the proposed **NSPP** was a fully neural network-based method without the extra phase refinement operation, which can be easily implemented. However, the subjective differences between the **NSPP** and **RAAR100** were not significant for both the analysis-synthesis task ($p = 0.38$) and the BWE task ($p = 0.98$). Interestingly, for the SS task, the MOS score of the **NSPP** was significantly higher than that of the **RAAR100** ($p < 0.01$). Obviously, the amplitude spectra used to recover the phase spectra in the analysis-synthesis task and BWE task were natural and semi-natural, respectively, but the amplitude spectra in the SS task were completely degraded. These results illustrated that our proposed **NSPP** had good

robustness, while the quality of the phase spectra recovered by the iterative algorithms (i.e., the GLA and RAAR) from the degraded amplitude spectra were obviously restricted. The proposed neural speech phase prediction model was more suitable for specific speech generation tasks.

D. Comparison with Waveform Reconstruction Method

Unlike iterative and neural network-based phase prediction methods, the waveform reconstruction methods were not originally designed for phase prediction. However, these waveform reconstruction methods implicitly incorporated phase prediction within waveform prediction. In this subsection, we compared our proposed **NSPP** with the HiFi-GAN vocoder (denoted by **HiFi-GAN**) using both objective and subjective evaluations. The description of the **HiFi-GAN** is as follows:

- **HiFi-GAN**: The v1 version of the HiFi-GAN vocoder [34]. We reimplemented it using the open source implementation[3]. We made small modifications to the open source code to fit our configurations. For a fair comparison with the **NSPP**, the input of the **HiFi-GAN** is 513-dimensional log amplitude spectra rather than 80-dimensional mel spectrograms. The upsampling ratios were set as $h_1 = 5$, $h_2 = 4$, $h_3 = 2$ and $h_4 = 2$. The latency calculation manner for the **HiFi-GAN** is similar with the **NSPP**. Although the **HiFi-GAN** incorporated more convolutional layers, the majority of its operations are conducted at a higher sampling rate (i.e., w_s is much smaller in Equation 27) relative to the original amplitude spectrum. Consequently, the latency of the **HiFi-GAN** amounted to a mere 101.4375 ms.

The objective evaluation results for the analysis-synthesis task are listed in Table IV. Our proposed **NSPP** slightly outperformed **HiFi-GAN** on the SNR and F0-RMSE metrics. However, considering the phase prediction precision, the phase continuity of the proposed **NSPP** was significantly superior to that of the **HiFi-GAN** according to the results of the GD loss and IAF loss, which confirmed the effectiveness of our proposed direct phase prediction manner. Regarding the generation efficiency, the RTF of **HiFi-GAN** on GPU was comparable to our proposed **NSPP** because GPUs allowed for parallel accelerated computations. However, on CPU, the **NSPP** exhibited significantly higher generation efficiency compared to the **HiFi-GAN**. Besides, due to the absence of GAN, the training time of **NSPP** is also much shorter than that of **HiFi-GAN** when using the same training mode and total number of training epochs (i.e., 3100 epochs). This validated the efficiency advantage of the **NSPP**.

Regarding the subjective evaluations, we conducted ABX preference tests on the Amazon Mechanical Turk platform to compare the subjective quality of the speeches generated by the **NSPP** and **HiFi-GAN**. 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 results are shown in Figure 6. There was no significant difference ($p > 0.01$) in subjective perception between the **NSPP** and **HiFi-GAN**, whether in analysis-synthesis, BWE or SS tasks. This finding suggests that the **NSPP** was on par with the **HiFi-GAN** in terms of reconstructed speech quality and robustness, while also offering a remarkable efficiency advantage.

[3]https://github.com/jik876/hifi-gan.

TABLE IV
OBJECTIVE EVALUATION RESULTS BETWEEN **NSPP** AND **HIFI-GAN** FOR THE ANALYSIS-SYNTHESIS TASK. HERE, "$a\times$" REPRESENTS $a\times$ REAL TIME.

	NSPP	**HiFi-GAN**
SNR(dB)↑	**8.26**	7.37
F0-RMSE(cent)↓	**10.0**	13.2
IP loss↓	**1.479**	1.483
GD loss↓	**0.297**	0.352
IAF loss↓	**0.694**	1.011
RTF (GPU)↓	**0.0065 (154×)**	0.0092 (109×)
RTF (CPU)↓	**0.051 (19.6×)**	0.60 (1.66×)
Training Time(h)↓	**27**	326

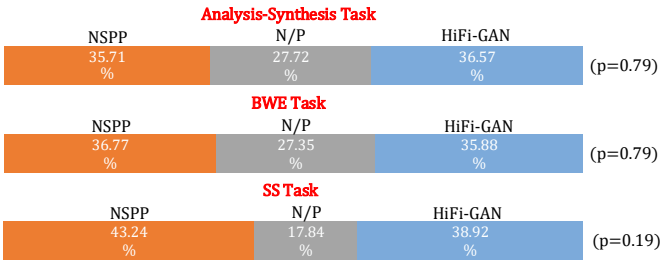

Fig. 6. Average preference scores (%) of ABX tests on speech quality between **NSPP** and **HiFi-GAN**, where N/P stands for "no preference" and p denotes the p-value of a t-test between two models.

It should be noted that the objective of this study is not to compare with other end-to-end speech generation methods. We are solely comparing the performance of different phase prediction methods when given different amplitude inputs. However, due to the trainable nature of the proposed neural speech phase prediction model, it can be easily integrated into end-to-end speech generation tasks to improve the phase quality, where the APNet vocoder [43] and MP-SENet [44] speech enhancement model serve as illustrative examples.

E. Evaluation on Low-Latency Streamable Phase Prediction

As discussed in Section IV-C and IV-D, our proposed **NSPP** exhibited a distinct advantage in latency compared to the **GLn**, **RAARn** and **DNN+GL100** when dealing with lengthy utterances. However, compared to the **HiFi-GAN**, the latency of our proposed **NSPP** was somewhat disappointing. Therefore, it is highly necessary to further reduce the latency of the proposed model, as discussed in Section III-C.

To validate the effectiveness of the low-latency streamable phase prediction method proposed in Section III-C, we compared the **NSPP** with the following two models:

TABLE V
OBJECTIVE EVALUATION RESULTS OF **NSPP**, **NSPP_CAUSAL** AND **NSPP_CAUSAL_KD** FOR THE ANALYSIS-SYNTHESIS TASK.

	IP loss↓	GD loss↓	IAF loss↓
NSPP	**1.479**	**0.297**	**0.694**
NSPP_causal	1.497	0.372	1.02
NSPP_causal_KD	1.499	0.309	0.766

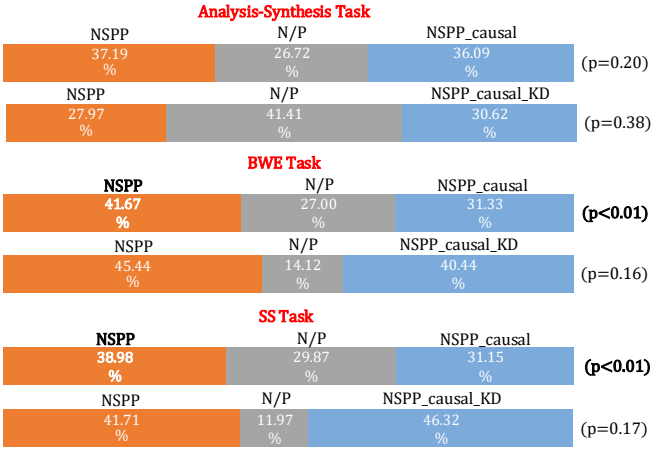

Fig. 7. Average preference scores (%) of ABX tests on speech quality for **NSPP_causal**, **NSPP_causal_KD** and **NSPP**, where N/P stands for "no preference" and p denotes the p-value of a t-test between two models.

- **NSPP_causal**: The causal neural speech phase prediction model trained only using the anti-wrapping losses (i.e., Equation 24).
- **NSPP_causal_KD**: The causal neural speech phase prediction model trained using the combination of anti-wrapping losses and knowledge distillation losses (i.e., Equation 29).

The aforementioned two models both have a 20 ms latency (i.e., the window size) and support streamable inference. We first compared the **NSPP** and **NSPP_causal**. The objective (i.e., three phase losses) and subjective (i.e., ABX tests) evaluation results are shown in Table V and Figure 7, respectively. Unsurprisingly, replacing non-causal convolutions with causal convolutions led to a significant decrease in the performance of the proposed model. Specifically, there is a noticeable increase in the GD and IAF losses of phase spectra predicted by **NSPP_causal**. In terms of perceptual quality, the **NSPP_causal** lagged significantly ($p < 0.01$) behind the **NSPP** in BWE and SS tasks, indicating a lack of robustness. To provide further evidence, we plotted the spectrograms of the reconstructed speech from the **NSPP** and **NSPP_causal** for the BWE task. As shown in Figure 8, the **NSPP_causal** experienced severe spectral interference, which may be the reason for the decline in auditory perception.

When the causal neural speech phase prediction model is integrated into the training process with knowledge distillation loss, remarkable improvements are observed. The GD and IAF losses of the **NSPP_causal_KD** approached the upper bound **NSPP** as listed in Table V. There are no significant subjective difference ($p > 0.05$) between the **NSPP** and **NSPP_causal_KD** for all tasks as shown in

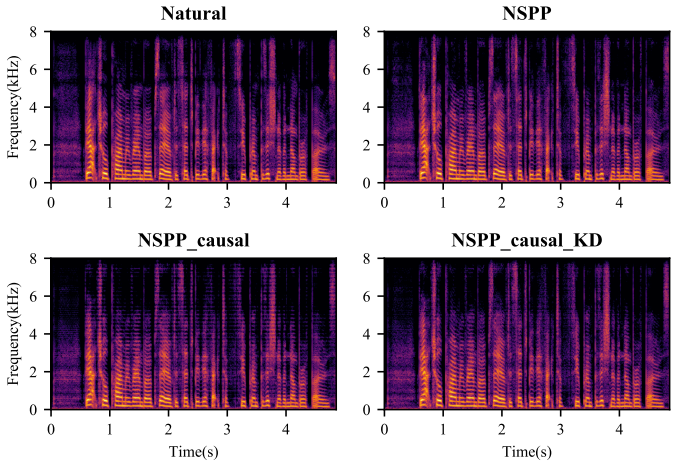

Fig. 8. A comparison among the spectrograms of the natural speech and speeches generated by **NSPP**, **NSPP_causal** and **NSPP_causal_KD** for the BWE task.

Figure 7. Compared with the **NSPP_causal**, the issue of spectral distortion also disappeared in the **NSPP_causal_KD** as shown in Figure 8. Through knowledge distillation, the student model successfully learned the knowledge from the teacher model. The above results strongly demonstrate that the combination of causal convolution and knowledge distillation reduced the latency of the proposed neural phase prediction model from a very high 330 ms to an extremely low 20 ms, while maintaining phase prediction precision, efficiency and robustness of the model.

F. Discussions

1) Effects of Different Anti-Wrapping Functions: As introduced in Section III-B, any function that conforms to the properties of parity, periodicity, and monotonicity can be used as an anti-wrapping function to activate the error between the predicted value and the natural value at the training stage. In this experiment, we studied the effect of different types of anti-wrapping functions on the performance of our proposed neural speech phase prediction model. The models used for comparison with the **NSPP** are shown as follows.

- **NSPP-log**: The proposed neural speech phase prediction model using the logarithmic anti-wrapping function $f_{log}(x)$ as shown in Figure 4(b) at the training stage. In the primary period, $f_{log}(x) = \frac{\pi}{ln(\pi+1)} ln(x+1), x \in (-\pi, \pi]$.
- **NSPP-cub**: The proposed neural speech phase prediction model using the cubic anti-wrapping function $f_{cub}(x)$ as shown in Figure 4(c) at the training stage. In the primary period, $f_{cub}(x) = \frac{4}{\pi^2}\left(x - \frac{\pi}{2}\right)^3 + \frac{\pi}{2}, x \in (-\pi, \pi]$.
- **NSPP-para**: The proposed neural speech phase prediction model using the parabolic anti-wrapping function $f_{para}(x)$ as shown in Figure 4(d) at the training stage. In the primary period, $f_{para}(x) = \frac{1}{\pi}x^2, x \in (-\pi, \pi]$.
- **NSPP-cos**: The proposed neural speech phase prediction model using the cosine anti-wrapping function $f_{cos}(x)$ as shown in Figure 4(e) at the training stage. In the primary period, $f_{cos}(x) = -\frac{\pi}{2}cos(x) + \frac{\pi}{2}, x \in (-\pi, \pi]$.

TABLE VI
SUBJECTIVE EVALUATION RESULTS AMONG FIVE NEURAL SPEECH PHASE PREDICTION MODELS FOR THE COMPARISON OF ANTI-WRAPPING FUNCTIONS ON THE ANALYSIS-SYNTHESIS TASK.

	MOS↑
Natural Speech	3.96±0.048
NSPP	**3.91±0.049**
NSPP-log	**3.90±0.048**
NSPP-cub	**3.93±0.048**
NSPP-para	3.79±0.050
NSPP-cos	3.84±0.053

The above four models and the **NSPP** shared the same settings except for the anti-wrapping function used during training. We compared the performance of these five models using the subjective evaluation for the analysis-synthesis task. MOS tests were conducted to compare the naturalness of the speeches reconstructed by these models. The subjective results are listed in Table VI. Obviously, the **NSPP**, **NSPP-log** and **NSPP-cub** all achieved excellent performance because their MOS scores were close to the natural one. However, **NSPP-para** and **NSPP-cos** were inferior to the other models. As shown in Figure 4, the commonality of $f_{line}(x)$, $f_{log}(x)$ and $f_{cub}(x)$ is that when the true error $\left|x - 2\pi \cdot round\left(\frac{x}{2\pi}\right)\right|$ is smaller, the rate of change of these functions is faster or remains the same. The above conclusion is opposite for functions $f_{para}(x)$ and $f_{cos}(x)$. These results indicated that an anti-wrapping function that paid less attention to activations for small error segments leads to a decrease in the phase prediction performance of the model (i.e., the $f_{para}(x)$ and $f_{cos}(x)$). At the small error segment (e.g., $x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$), the rate of change of the function should be faster than the rate of change of the error. For example, by comparing **NSPP-log** and **NSPP-para**, when they reduced the same loss value at the training stage, **NSPP-log** shrank the true error faster than **NSPP-para**. Quickly reducing the true error is the goal of model training. However, although the rates of change of function $f_{line}(x)$, $f_{log}(x)$ and $f_{cub}(x)$ were significantly different at the large error range (e.g., $x \in \left(-\pi, -\frac{\pi}{2}\right] \cup \left[\frac{\pi}{2}, \pi\right]$), the results of **NSPP**, **NSPP-log** and **NSPP-cub** were not significantly different. It is reasonable because we find that the true error was mostly concentrated in the small value segment (i.e., most $x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$). Taking the **NSPP** as an example, the converged mean values of the true errors of the IP, GD and IAF on the test set were 1.48, 0.297 and 0.694 respectively.

2) Ablation Studies: We then conducted several ablation experiments to explore the roles of some key modules in our proposed **NSPP**. Here, experiments were performed only on the analysis-synthesis task. 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 [31], [32].
- **NSPP wo AWF**: Removing the anti-wrapping function f 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 IP loss $\mathcal{L}_{IP}$ from the **NSPP** at the training stage.
- **NSPP wo GD**: Removing the GD loss $\mathcal{L}_{GD}$ from the **NSPP** at the training stage.
- **NSPP wo IAF**: Removing the IAF loss $\mathcal{L}_{IAF}$ from the **NSPP** at the training stage.

TABLE VII
OBJECTIVE EVALUATION RESULTS AMONG **NSPP** AND ITS ABLATED VARIANTS FOR THE ANALYSIS-SYNTHESIS TASK.

	SNR(dB)↑	F0-RMSE(cent)↓
NSPP	8.26	**10.0**
NSPP wo PEA	4.65	36.9
NSPP wo AWF	8.51	12.0
NSPP wo IP	4.95	21.2
NSPP wo GD	**8.95**	10.1
NSPP wo IAF	8.69	12.1

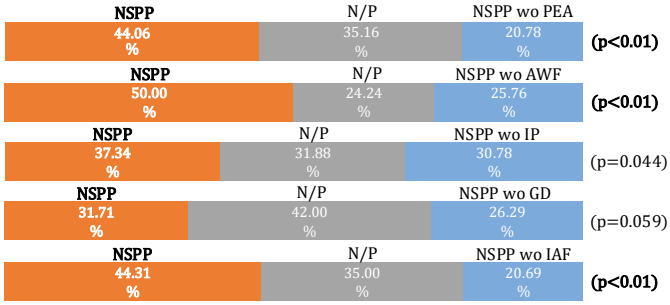

Fig. 9. Average preference scores (%) of ABX tests on speech quality between **NSPP** and its ablated variants for the analysis-synthesis task, where N/P stands for "no preference" and p denotes the p-value of a t-test between two models.

We also utilized SNR and F0-RMSE as objective metrics here for evaluating the reconstructed speech. Regarding the subjective evaluations, we conducted ABX preference tests on the Amazon Mechanical Turk platform to compare the differences between the **NSPP** and its ablated variants. The objective and subjective results are listed in Table VII and Figure 9, respectively. Additionally, we also provided the spectrograms of the speeches generated by the **NSPP** and its ablated variants in Figures 10 and 11 for visual analysis.

As expected, we can see that the **NSPP** outperformed the **NSPP wo PEA** significantly ($p < 0.01$) by analyzing both objective and subjective results listed in Table VII and Figure 9, respectively. 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. By comparing the spectrograms of the speeches generated by the **NSPP** and **NSPP wo PEA** in Figure 10, we can find that the spectrogram of the **NSPP wo PEA** was a little blurred. 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 a failure of anti-wrapping losses. These results indicated that the parallel estimation architecture was essential to wrapped phase prediction.

By comparing **NSPP** and **NSPP wo AWF** in Table VII, the SNR of the **NSPP wo AWF** was even higher and the F0-

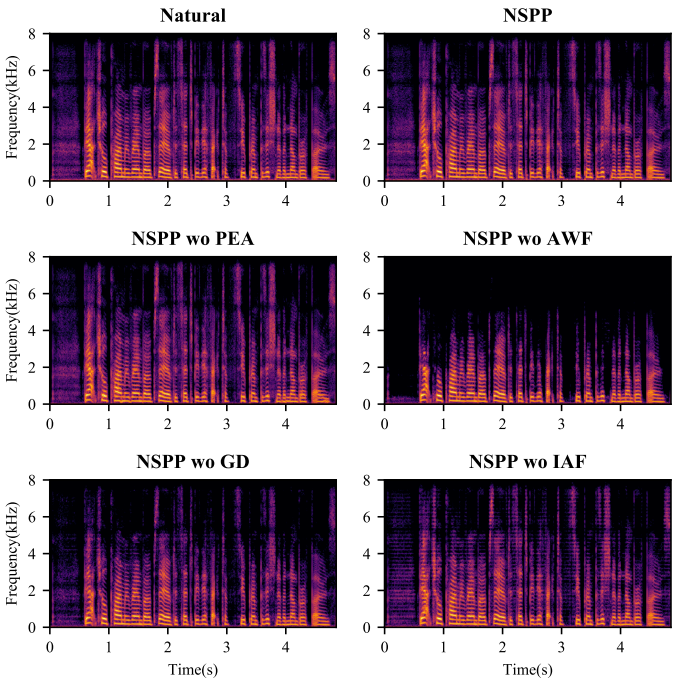

Fig. 10. A comparison among the spectrograms of the natural speech and speeches generated by **NSPP**, **NSPP wo PEA**, **NSPP wo AWF**, **NSPP wo GD** and **NSPP wo IAF** for the analysis-synthesis task.

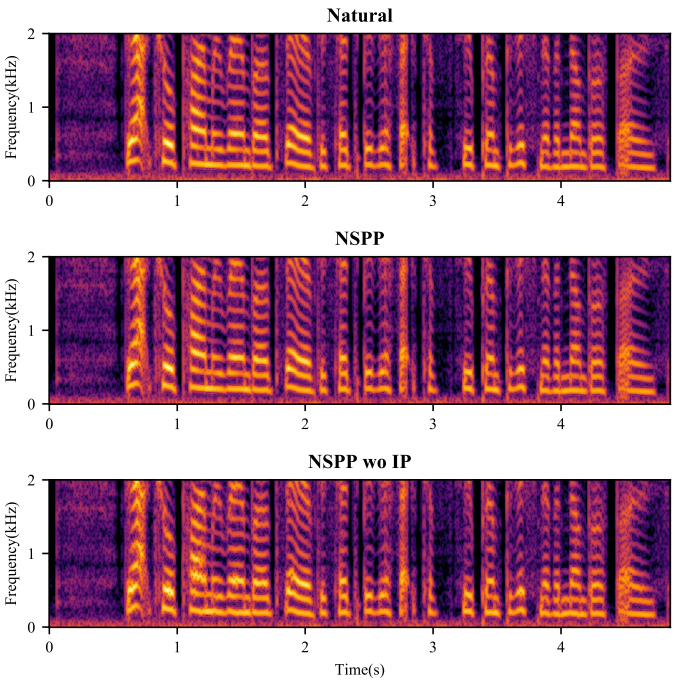

Fig. 11. A comparison among the low-frequency (0∼2000Hz) spectrograms of the natural speech and speeches generated by **NSPP** and **NSPP wo IP** for analysis-synthesis task.

RMSE of the **NSPP wo AWF** was comparable to that of the **NSPP**. However, the subjective results in Figure 9 indicated that the **NSPP** outperformed the **NSPP wo AWF** significantly ($p < 0.01$) in terms of speech quality, which proved that the anti-wrapping function was helpful for avoiding the error expansion issue. As shown in Figure 10, the high-frequency energy of the speech reconstructed by the **NSPP wo AWF** was completely suppressed, resulting in an extremely dull listening experience. This may be the reason for the poor ABX scores. Interestingly, there was 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**). The above experimental results also confirmed that the absence of high-frequency components had little effect on the SNR metric.

For the three losses, removing any loss led to poor performance of the model. However, each loss played a very different role. Removing $\mathcal{L}_{IP}$ (i.e., **NSPP wo IP**) led to a sharp drop in all objective metrics in Table VII. Regarding the ABX test results in Figure 9, the subjective difference between the **NSPP** and **NSPP wo IP** was slightly insignificant (p was slightly larger than 0.01). However, we found that the reconstructed speech quality of the **NSPP wo IP** indeed degraded. Figure 11 shows the low-frequency F0 and harmonic details of the spectrograms of the **NSPP** and **NSPP wo IP**. Obviously, the speech reconstructed by the **NSPP wo IP** exhibited few low-frequency spectrum corruption issues (see the range of 0.5∼2 seconds in Figure 11), resulting in F0 and harmonic structure distortion and blurry pronunciation. This is also the reason why the F0-RMSE of the **NSPP wo IP** was relatively poor. However, removing $\mathcal{L}_{GD}$ (i.e., **NSPP wo GD**) and $\mathcal{L}_{IAF}$ (i.e., **NSPP wo IAF**) did not cause significant deterioration on all objective metrics in Table VII. Although the subjective difference between the **NSPP** and **NSPP wo GD** was slightly insignificant (p was slightly larger than 0.01) in Figure 9, we can see from Figure 10 that the **NSPP wo GD** attenuated the overall spectral energy of the reconstructed speech, resulting in a mild dull listening experience. As shown in Figure 9, 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 (see Figure 10), causing annoying loud noise. Interestingly, the removal of $\mathcal{L}_{GD}$ and $\mathcal{L}_{IAF}$ did not destroy the F0 and harmonic structure, nor did it lead to mispronunciation (their F0-RMSEs were comparable to that of **NSPP**).

In conclusion, all ablated elements were indispensable for our proposed neural speech phase prediction model. The parallel estimation architecture and IP loss prevented the destruction of the F0 and spectral structure of speech. The anti-wrapping function and GD loss avoided the attenuation of high-frequency energy and dull hearing of speech. The IAF loss suppressed the appearance of loud noise caused by spectral horizontal lines.

V. 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 are 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. The anti-wrapping function should possess three properties, i.e., parity, periodicity and monotonicity. Low-latency streamable phase prediction is also achieved with the help of causal convolutions and knowledge distillation training strategies. Experimental results show that the proposed model outperforms the GLA, RAAR and von Mises distribution DNN-based phase prediction methods for both analysis-synthesis and specific speech generation tasks (i.e., the BWE and SS) in terms of phase prediction precision, efficiency and robustness. The proposed model is significantly faster in generation speed than HiFi-GAN-based waveform reconstruction method, while also having the same synthesized speech quality. Besides, the proposed model is easy to implement and also exhibits a fast training speed. Integrating the neural speech phase prediction model to more end-to-end speech generation tasks will be the focus of our future work.

REFERENCES

[1] Y. Ai and Z.-H. Ling, "Neural speech phase prediction based on parallel estimation architecture and anti-wrapping losses," in *Proc. ICASSP*, 2023, pp. 1–5.

[2] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. Interspeech*, 2013, pp. 436–440.

[3] Y. Xu, J. Du, L.-R. Dai, and C.-H. 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.

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

[5] Y. Wang, S. Zhao, W. Liu, M. Li, and J. Kuang, "Speech bandwidth expansion based on deep neural networks," in *Proc. Interspeech*, 2015, pp. 2593–2597.

[6] Y. Gu, Z.-H. Ling, and L.-R. Dai, "Speech bandwidth extension using bottleneck features and deep recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 297–301.

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

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

[9] 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," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[10] 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 *Proc. ICASSP*, 2018, pp. 4779–4783.

[11] S. Takaki, H. Kameoka, and J. 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.

[12] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," in *Proc. ICML*, 2019, pp. 4352–4362.

[13] Y. Saito, S. Takamichi, and H. Saruwatari, "Text-to-speech synthesis using STFT spectra based on low-/multi-resolution generative adversarial networks," in *Proc. ICASSP*, 2018, pp. 5299–5303.

[14] D. Griffin and J. 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.

[15] N. Perraudin, P. Balazs, and P. L. Søndergaard, "A fast Griffin-Lim algorithm," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[16] Y. Masuyama, K. Yatabe, and Y. Oikawa, "Griffin-Lim like phase recovery via alternating direction method of multipliers," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 184–188, 2018.

[17] T. Kobayashi, T. Tanaka, K. Yatabe, and Y. Oikawa, "Acoustic application of phase reconstruction algorithms in optics," in *Proc. ICASSP*, 2022, pp. 6212–6216.

[18] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Phase retrieval, error reduction algorithm, and fienup variants: a view from convex optimization," *JOSA A*, vol. 19, no. 7, pp. 1334–1345, 2002.

[19] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Finding best approximation pairs relative to two closed convex sets in hilbert spaces," *Journal of Approximation theory*, vol. 127, no. 2, pp. 178–192, 2004.

[20] D. R. Luke, "Relaxed averaged alternating reflections for diffraction imaging," *Inverse problems*, vol. 21, no. 1, p. 37, 2004.

[21] J. R. Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Optics letters*, vol. 3, no. 1, pp. 27–29, 1978.

[22] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Applied optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[23] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Hybrid projection–reflection method for phase retrieval," *JOSA A*, vol. 20, no. 6, pp. 1025–1034, 2003.

[24] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram," in *Proc. EUSIPCO*, 2018, pp. 2514–2518.

[25] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep Griffin-Lim iteration," in *Proc. ICASSP*, 2019, pp. 61–65.

[26] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep Griffin-Lim iteration: Trainable iterative phase reconstruction using neural network," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 1, pp. 37–50, 2020.

[27] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Phase reconstruction based on recurrent phase unwrapping with deep neural networks," in *Proc. ICASSP*, 2020, pp. 826–830.

[28] L. Thieling, D. Wilhelm, and P. Jax, "Recurrent phase reconstruction using estimated phase derivatives from deep neural networks," in *Proc. ICASSP*, 2021, pp. 7088–7092.

[29] N. B. Thien, Y. Wakabayashi, K. Iwai, and T. Nishiura, "Two-stage phase reconstruction using dnn and von mises distribution-based maximum likelihood," in *Proc. APSIPA*, 2021, pp. 995–999.

[30] D. C. Ghiglia and M. D. Pritt, "Two-dimensional phase unwrapping: theory, algorithms, and software," *A Wiley Interscience Publication*, 1998.

[31] S. Takamichi, Y. Saito, N. Takamune, D. Kitamura, and H. Saruwatari, "Phase reconstruction from amplitude spectrograms based on von-mises-distribution deep neural network," in *Proc. IWAENC*, 2018, pp. 286–290.

[32] S. Takamichi, Y. Saito, N. Takamune, D. Kitamura, and H. Saruwatari, "Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks," *Signal Processing*, vol. 169, p. 107368, 2020.

[33] H. Dudley, "The vocoder," *Bell Labs Record*, vol. 18, no. 4, pp. 122–126, 1939.

[34] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.

[36] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. ICML*, 2017, pp. 933–941.

[37] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.

[38] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2016.

[39] Y. Ai, Z.-H. Ling, W.-L. Wu, and A. Li, "Denoising-and-dereverberation hierarchical neural vocoder for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2036–2048, 2022.

[40] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2018.

[41] Y. Ai and Z.-H. 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.
[42] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *Proc. Interspeech*, 2011, pp. 3053–3056.
[43] Y. Ai and Z.-H. Ling, "APNet: An all-frame-level neural vocoder incorporating direct prediction of amplitude and phase spectra," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2145–2157, 2023.
[44] Y.-X. Lu, Y. Ai, and Z.-H. Ling, "MP-SENet: A speech enhancement model with parallel denoising of magnitude and phase spectra," in *Proc. Interspeech*, 2023, pp. 3834–3838.