©2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Online Phase Reconstruction via DNN-based Phase Differences Estimation

Yoshiki Masuyama, Graduate Student Member, IEEE, Kohei Yatabe, Member, IEEE, Kento Nagatomo, and Yasuhiro Oikawa, Member, IEEE

Abstract—This paper presents a two-stage online phase reconstruction framework using causal deep neural networks (DNNs). Phase reconstruction is a task of recovering phase of the short-time Fourier transform (STFT) coefficients only from the corresponding magnitude. However, phase is sensitive to waveform shifts and not easy to estimate from the magnitude even with a DNN. To overcome this problem, we propose to use DNNs for estimating differences of phase between adjacent time-frequency bins. We show that convolutional neural networks are suitable for phase difference estimation, according to the theoretical relation between partial derivatives of STFT phase and magnitude. The estimated phase differences are used for reconstructing phase by solving a weighted least squares problem in a frame-by-frame manner. In contrast to existing DNN-based phase reconstruction methods, the proposed framework is causal and does not require any iterative procedure. The experiments showed that the proposed method outperforms existing online methods and a DNN-based method for phase reconstruction.

Index Terms—Real-time spectrogram inversion, group delay, instantaneous frequency, time-frequency analysis, low-latency.

I. INTRODUCTION

PHASE reconstruction of short-time Fourier transform (STFT) coefficients is important for various audio technologies such as speech enhancement [1]–[6], audio source separation [7]–[11], and text-to-speech synthesis [12]–[15]. As the structure of audio signals is apparent in the magnitude of STFT coefficients, ordinary methods for these technologies have focused on manipulating the magnitude. After obtaining the magnitude, the corresponding phase is required to reconstruct a time-domain signal. Although the phase of an observed signal is available in speech enhancement and audio source separation, it often causes artifacts and residual interference [16], [17]. In text-to-speech synthesis, the magnitude is generated from linguistic features, and thus the phase is fully unavailable. Hence, phase reconstruction of STFT coefficients is helpful for many applications.

As summarized in Fig. 1, various phase reconstruction methods have been studied, such as consistency-based methods [18]–[20], phase gradient heap integration (PGHI) [21],

Y. Masuyama is with the Department of Computer Science, Graduate School of Systems Design, Tokyo Metropolitan University, Hino, Tokyo 191-0065, Japan (e-mail: masuyama-yoshiki@ed.tmu.ac.jp).

K. Yatabe is with the Department of Electrical Engineering and Computer Science, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan (e-mail: yatabe@go.tuat.ac.jp).

K. Nagatomo and Y. Oikawa are with the Department of Intermedia Art and Science, Waseda University, Tokyo 169-8555, Japan (e-mail: jimijeffericking@akane.waseda.jp; yoikawa@waseda.jp).

	Offline processing	Online processing
Consistency-based approach	Griffin–Lim algorithm [18] Fast Griffin–Lim algorithm [19]	Real-time iterative spectrogram inversion [37]
Phase gradient heap integration	Phase gradient heap integration (PGHI) [21]	Real-time PGHI [38]
Sinusoidal-model- based approach		Single pass spectrogram inversion [22] Phase unwrapping [23]
DNN-based approach	Directional-statistics DNN [26, 27] Deep Griffin–Lim iteration [28, 29]	Proposed two-stage framework

1

Fig. 1. Comparison of offline and online phase reconstruction methods. Methods in the bottom half exploit the prior knowledge of the target signal.

sinusoidal-model-based methods [22], [23], and deep neural network (DNN)–based methods [24]–[32]. These methods can be divided into two categories: phase reconstruction with and without prior knowledge of a target signal.

As phase reconstruction methods without prior knowledge, consistency-based methods have been widely used [18]–[20]. These methods are based on the relation among nearby STFT coefficients owing to the window and its overlapping nature. To recover this relation, most consistency-based methods are given as iterative optimization algorithms. Meanwhile, PGHI [21] does not require such iterative procedures or the prior knowledge. PGHI is based on a relation between the magnitude and phase of the STFT coefficients [33], [34]; partial derivatives of phase can be analytically calculated from magnitude under some assumptions. This relation enables PGHI to reconstruct phase by integrating phase derivatives. As a result, PGHI has achieved promising results despite lacking any prior knowledge about the target signal.

In contrast, sinusoidal-model-based and DNN-based methods leverage prior knowledge about the target signal. The assumption of sinusoidal-model-based methods is that the target signal consists of sinusoids [22], [23]. This assumption allows one to approximate the phase derivative with respect to time by the frequency of each sinusoid. Then, the phase can be reconstructed by integrating the approximated derivative over time. However, such theoretically derived methods are only applicable to specific signals, and DNN-based methods are promising because DNNs can automatically learn prior knowledge from a training dataset. Since some DNN-based methods have been successfully applied to phase reconstruction [29], we also propose to use DNNs to exploit prior knowledge learned from the training dataset.

Manuscript received May 8, 2022; revised Aug 31, 2022; revised Oct 12, 2022; accepted Oct 26, 2022. (Corresponding author: Yoshiki Masuyama.)



Fig. 2. Illustration of the two-stage online phase reconstruction. As depicted in the top figure, the first stage estimates phase differences from the logmagnitude, and the second stage reconstructs phase from them. Instead of TPD, the DNN estimates a modified version of TPD called baseband phase delay (BPD) as shown in the bottom figure. The estimated BPD is converted to TPD by adding $2\pi\alpha m/M$. The second stage reconstructs the phase frameby-frame by solving a weighted least squares problem of complex STFT coefficients.

While many phase reconstruction methods are offline algorithms, online phase reconstruction is highly desired in a wide range of applications, including incremental text-tospeech [35] and low-latency audio source separation [36]. Therefore, except for the DNN-based methods, the aforementioned methods have been extended to the online setting [22], [23], [37], [38]. A promising method is an extension of PGHI called real-time PGHI (RTPGHI) [38]. Although it outperforms the consistency-based method [37] and the sinusoidal-model-based method [22], there remains room for improvement. First, the STFT phase-magnitude relation is valid only in the continuous setting, and thus estimated phase differences contain some errors in the discrete setting. Second, to approximate the phase derivatives, the centered difference scheme used in PGHI is not allowable in the online setting without look-ahead frames. Instead, RTPGHI uses the backward difference that results in lower performance than the original PGHI. The strong modeling capability of DNNs can improve the estimation accuracy of phase differences.

In this paper, we propose a DNN-based online phase reconstruction framework. As illustrated in Fig. 2, the proposed framework consists of two stages: (i) estimating the phase differences from the magnitude and (ii) reconstructing the phase from the estimated phase differences. First, we estimate the phase differences with respect to time (TPD) and frequency (FPD) by using causal DNNs. This DNN-based estimation is expected to be robust to the mismatch of the STFT phasemagnitude relation by leveraging the prior knowledge acquired from a training dataset. Second, we recurrently reconstruct phase from the estimated differences. To handle the phase differences efficiently, we treat them as the ratios of complex STFT coefficients. Then, the phase is reconstructed by solving a weighted least-squares problem of STFT coefficients. Through several experiments, we confirmed the effectiveness of the proposed two-stage framework compared to existing online phase reconstruction methods.

Note that this paper is related to our conference paper [30], in which we developed the basic concept of the two-stage phase reconstruction framework in the offline setting. In this paper, we extend it to an online method with improvement on all components, i.e., both the first and the second stages. The contributions of this paper are summarized as follows:

- proposing an online phase reconstruction framework using causal DNNs, while our previous work [30] focused on the offline setting;
- applying convolutional neural networks (CNNs) to estimate phase differences, which is motivated by the STFT phase-magnitude relation;
- presenting a novel method for reconstructing phase from its differences by solving the weighted least squares problem of complex STFT coefficients;
- investigating and comparing the performance of various online phase reconstruction methods.

The rest of the paper is organized as follows. In Section II, offline and online phase reconstruction problems are formulated. Section III explains the STFT phase-magnitude relation, PGHI, and its online extension, RTPGHI. DNN-based phase reconstruction methods are also reviewed. The proposed two-stage framework for DNN-based online phase reconstruction is introduced in Section IV. In Section V, the proposed method is compared with various online phase reconstruction methods, and then the effectiveness of both stages is investigated. Finally, Section VI concludes this paper.

II. PROBLEM FORMULATION

STFT¹ of a discrete signal χ with respect to a real symmetric window g of length L is defined as

$$X[m,n] = \sum_{l=-\lfloor L/2 \rfloor}^{\lfloor L/2 \rfloor} \chi[l+\alpha n] g[l] e^{-2\pi i lm/M}, \qquad (1)$$

where X[m, n] is the (m, n)th entry of the STFT coefficients, i is the imaginary unit, α is a time shifting step, $n = 0, \ldots, N-1$ and $m = 0, \ldots, M-1$ are the time-frame and frequency indices, respectively. These symbols used in this section are summarized in Table I. Let us denote the magnitude and phase of the STFT coefficients **X** by **A** and **Φ**, respectively:

$$A[m,n] = |X[m,n]| \tag{2}$$

$$\Phi[m,n] = \operatorname{Arg}(X[m,n]), \tag{3}$$

where $\operatorname{Arg}(\cdot)$ returns the principal value of the complexargument of its input.

We consider the task of phase reconstruction that aims at estimating the target phase Φ while allowing the ambiguity of

¹In the literature of audio signal processing, the transform defined by (1) is commonly called STFT, while it is called the discrete Gabor transform in other communities [39], [40]. In this paper, we use the term STFT according to the former literature.

TABLE I LIST OF SYMBOLS USED IN SECTION II

	N. 11				
variables					
$\chi[l]$	The <i>l</i> th sample of a discrete time-domain signal				
g[l]	The <i>l</i> th sample of a window used in STFT				
X[m,n]	Complex STFT coefficient at the (m, n) th bin				
A[m,n]	STFT magnitude given by $ X[m, n] $				
$\Phi[m,n]$	STFT phase given by $Arg(X[m, n])$				
Notations with accents and subscripts					
$\widehat{(\cdot)}$	Estimate of its input				
()	Vector of its input at the <i>n</i> th time-frame,				
$(\cdot)_n$	e.g., $\mathbf{x}_n = [X[0, n], \dots, X[M-1, n]]^{T}$				
Maps					
$\mathscr{F}(\cdot)$	Mapping from the magnitude to the phase				
$\operatorname{Arg}(\cdot)$	Mapping from a complex scalar to its principal argument				

 $2\pi.$ That is, our objective is to construct the a mapping $\mathscr{F}(\cdot)$ such that:

$$\widehat{\mathbf{\Phi}} = \mathscr{F}(\mathbf{A}),\tag{4}$$

$$\boldsymbol{\Phi} \approx \boldsymbol{\Phi} + 2\pi \mathbf{N},\tag{5}$$

where $\mathbf{N} \in \mathbb{Z}^{M \times N}$ is an arbitrary integer-valued array.

Offline phase reconstruction uses the STFT magnitude at all time-frames as in (4), which can be written as follows:

$$\widehat{\mathbf{\Phi}} = \mathscr{F}(\mathbf{a}_0, \dots, \mathbf{a}_{N-1}), \tag{6}$$

where $\mathbf{a}_n = [A[0, n], \dots, A[M-1, n]]^\mathsf{T}$, and $(\cdot)^\mathsf{T}$ denotes the transpose. However, (6) is not applicable to the online setting. In real-time applications, we should estimate the phase at each time-frame only from the magnitudes up to the current time-frame and few look-ahead frames:

$$\boldsymbol{\phi}_n = \mathscr{F}(\dots, \mathbf{a}_n, \dots, \mathbf{a}_{n+N_{\mathrm{LA}}}),\tag{7}$$

where $\widehat{\phi}_n = [\widehat{\Phi}[0, n], \dots, \widehat{\Phi}[M - 1, n]]^\mathsf{T}$, and $N_{\mathrm{LA}} \in \mathbb{N}$ is the number of look-ahead frames. The number of time-frames that affect the current output depends on the map $\mathscr{F}(\cdot)$. When using a causal CNN [41], it depends on the receptive field of the CNN. Meanwhile, when using a recurrent neural network (RNN), the output at the current time-frame implicitly depends on the magnitudes at all the past time-frames. In this paper, a system is said to be causal if it does not require future information to compute its current output, i.e., $N_{\mathrm{LA}} = 0$.

III. RELATED WORKS

In this section, after explaining PGHI and RTPGHI, we review the DNN-based phase reconstruction methods. The symbols used in this section are listed in Table II.

A. Phase Gradient Heap Integration (PGHI)

PGHI is a non-iterative phase reconstruction method based on the STFT phase-magnitude relation derived from the definition of continuous STFT [21]. In the continuous setting, STFT of a function $y \in L^2(\mathbb{R})$ with respect to a window function $h \in L^2(\mathbb{R})$ is defined as

$$Y(f,t) = \int_{\mathbb{R}} y(\tau+t) h(\tau) e^{-2\pi i f \tau} d\tau$$
$$= \mathcal{A}(f,t) e^{i\varphi(f,t)}, \qquad (8)$$

 TABLE II

 List of Symbols Used in Section III

Variables				
\overline{y}	L^2 function as a signal			
h	L^2 window function			
Y	Continuous STFT of the function y			
\mathcal{A}	Magnitude of Y			
φ	Phase of Y			
$\widetilde{A}[m,n]$	Log-magnitude of the discrete STFT coefficient			
$V_{\rm c}[m,n]$	Phase derivative for time defined for the (m, n) th point			
$U_{c}[m, n]$	[m, n] Phase derivative for frequency defined for the (m, n) th point			
V[m,n]	V[m, n] Backward phase difference for time (TPD)			
U[m, n]	Backward phase difference for frequency (FPD)			
Maps				
$\mathcal{F}_{\theta}(\cdot)$	DNN for estimating phase from the given magnitude			
$\mathcal{L}(\cdot, \cdot)$	Periodic loss function			

where \mathcal{A} and φ represent the magnitude and phase, respectively. Let us define the Gaussian window as follows:

$$h(t) = \left(\frac{2}{\sigma^2}\right)^{1/4} e^{-\pi t^2/\sigma^2},$$
 (9)

where σ is a parameter of the Gaussian window. When using the Gaussian window for STFT, both magnitude and phase are partially differentiable with respect to both time and frequency. In addition, the following phase-magnitude relation of STFT can be derived [21]:

$$\frac{\partial}{\partial t}\varphi(f,t) = \frac{1}{\sigma^2}\frac{\partial}{\partial f}\log(\mathcal{A}(f,t)) + 2\pi f, \qquad (10)$$

$$\frac{\partial}{\partial f}\varphi(f,t) = -\sigma^2 \frac{\partial}{\partial t} \log(\mathcal{A}(f,t)).$$
(11)

This relation indicates that the phase derivatives can be analytically calculated from the corresponding log-magnitude. We can thus reconstruct the phase by integrating its gradient up to the global constant phase.

PGHI exploits the relations in (10) and (11) to compute the phase gradient in the discrete setting, where STFT is defined as (1). In PGHI, phase gradient is approximated by using the second order centered differences of log-magnitude:

$$\widehat{V}_{c}[m,n] = \frac{\alpha M}{2\beta} (\widetilde{A}[m+1,n] - \widetilde{A}[m-1,n]) + \frac{2\pi\alpha m}{M}, (12)$$

$$\widehat{U}_{\mathbf{c}}[m,n] = -\frac{\beta}{2\alpha M} (\widetilde{A}[m,n+1] - \widetilde{A}[m,n-1]), \qquad (13)$$

where $\widetilde{A}[m,n] = \log(A[m,n])$, and β is a constant depending on the window. Note that $\widehat{V}_{c}[m,n]$ and $\widehat{U}_{c}[m,n]$ approximate phase derivatives sampled on the time-frequency (T-F) grid.

The phase is reconstructed by numerically integrating the backward phase differences. Since there are multiple possible paths for the integration, PGHI adaptively chooses one of the following four integration paths:

$$\widehat{\Phi}[m,n] = \widehat{\Phi}[m,n-1] + \widehat{V}[m,n], \tag{14}$$

$$\widehat{\Phi}[m,n] = \widehat{\Phi}[m,n+1] - \widehat{V}[m,n+1], \qquad (15)$$

$$\widehat{\Phi}[m,n] = \widehat{\Phi}[m-1,n] + \widehat{U}[m,n], \qquad (16)$$

$$\widehat{\Phi}[m,n] = \widehat{\Phi}[m+1,n] - \widehat{U}[m+1,n],$$
 (17)



Fig. 3. Illustration of the phase derivatives and the backward phase differences. Green circles correspond to phases at the T-F grids. Red and blue arrows indicate phase differences with respect to time and frequency, respectively.

where $\widehat{V}[m,n]$ and $\widehat{U}[m,n]$ are approximate backward TPD and FPD, respectively. They are given by averaging the estimated phase gradient in (12) and (13):

$$\widehat{V}[m,n] = \frac{\widehat{V}_{c}[m,n] + \widehat{V}_{c}[m,n-1]}{2},$$
(18)

$$\widehat{U}[m,n] = \frac{\widehat{U}_{c}[m,n] + \widehat{U}_{c}[m-1,n]}{2}.$$
(19)

Note that the oracle backward TPD and FPD are given by

$$V[m,n] = \Phi[m,n] - \Phi[m,n-1],$$
 (20)

$$U[m,n] = \Phi[m,n] - \Phi[m-1,n],$$
 (21)

respectively. The relation between the phase derivatives and the backward phase differences are illustrated in Fig. 3. While the former is defined for every T-F points, the latter is defined as the relation between adjacent T-F bins.

In the numerical integration, PGHI omits phase at a T-F bin whose magnitude is small because phase differences should be unreliable at such T-F bins. Instead, random phase is assigned to such T-F bins for simplicity. We refer the reader to the paper [21] and codes² for more details of implementation.

B. Real-time PGHI (RTPGHI)

RTPGHI is an online extension of PGHI [38]. When allowing one look-ahead frame, i.e., $N_{\text{LA}} = 1$ in (7), the phase gradient can be approximated by the centered difference scheme as in (12) and (13). However, the centered difference in (13) is not applicable to the causal setting (i.e., $N_{\text{LA}} = 0$) because $\widetilde{A}[m, n + 1]$ is not accessible. Hence, RTPGHI approximates the phase derivative with respect to the frequency by using the second order backward time-difference of the log-magnitude:

$$\widehat{U}_{c}[m,n] = -\frac{\beta}{2\alpha M} (3\widetilde{A}[m,n]) - 4\widetilde{A}[m,n-1] + \widetilde{A}[m,n-2]).$$
(22)

²PGHI and RTPGHI are implemented in the phase retrieval toolbox (PHASERET): http://ltfat.github.io/phaseret/ [42].

Then, the phase is reconstructed via one of the integration paths except (15) because $\widehat{\Phi}[m, n+1]$ is not available.

Although RTPGHI achieved promising results, it has some limitations. First, the phase-magnitude relation in (10) and (11) is only valid for the continuous case, and hence the estimated TPD and FPD in (12) and (13) contain errors in the discrete setting. Moreover, this relation assumes the use of the Gaussian window which has infinite support in the time domain. Such a window is not allowed in real-time applications. Second, the experimental results in [38] showed that the second order backward difference approximation in (22) degrades the quality of the reconstructed signals from that of the centered difference in (13).

C. DNN-based Phase Reconstruction

DNN-based phase reconstruction has gained increasing attention because of strong modeling capability of DNNs [24]– [32]. A DNN-based method can handle various signals by learning prior knowledge from a training dataset. A straightforward approach is to model the map $\mathscr{F}(\cdot)$ in (4) by a DNN. When training such a DNN, we should consider the periodic nature of the target phase $\Phi[m, n]$ because phase is given as a complex-argument. Ordinary loss functions for a regression problem, including the mean squared error, are not suitable for training in such a situation.

To address this issue, several approaches have been presented. One approach uses a DNN to estimate complex STFT coefficients **X** instead of their phase Φ [24], [28], [29]. Another approach quantizes the target phases and estimates their indices [8], [9]. As a result of recasting the regression problem as a classification problem, the periodic nature of the phase is circumvented. Neither approach directly estimates the phase to avoid dealing with a circular variable.

A periodic loss function has been proposed to train a DNN that directly estimates the continuous circular phase [26], [27]. In this approach, a DNN $\mathcal{F}_{\theta}(\cdot)$ directly estimates the phase:

$$\widehat{\mathbf{\Phi}} = \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{A}), \tag{23}$$

where θ is a set of parameters of the DNN. To measure the error between the target phase $\Phi[m, n]$ and the estimated phase $\widehat{\Phi}[m, n]$, a periodic loss function satisfying

$$\mathcal{L}(\phi, \widehat{\phi}) = \mathcal{L}(\phi, \widehat{\phi} + 2\pi b) \tag{24}$$

is considered, where b is an arbitrary integer. For instance, the negative cosine loss function is given by

$$\mathcal{L}_{\cos}(\phi, \widehat{\phi}) = -\cos(\phi - \widehat{\phi}). \tag{25}$$

By using the periodic loss function, a DNN for estimating the continuous phase is trained as follows:

$$\min_{\boldsymbol{\theta}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathcal{L}_{\cos}(\Phi[m,n], \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{A})[m,n]), \quad (26)$$

where we omit the summation over the training dataset because the DNN treats each pair of A and Φ separately. The estimated phase has the ambiguity of 2π due to the use of the periodic loss function. This ambiguity is not a problem

TABLE III List of Symbols Used in Section IV

	Variables		
W[m,n]	Backward baseband phase difference (BPD)		
$\mathfrak{V}[m,n]$	Ratio between STFT coefficients at the adjacent time-frames		
$\mathfrak{U}[m,n]$	Ratio between STFT coefficients at the adjacent frequency-bins		
Ψ_n	Feature matrix for estimating the phase differences by DNNs		
$\mathbf{\Lambda}_n$	Diagonal matrix representing the reliability of estimated TPD		
Γ_n	Diagonal matrix representing the reliability of estimated FPD		
	Maps		
$\mathcal{W}(\cdot)$	Wrapping operator		
$\mathcal{G}_{\boldsymbol{\theta}_{time}}(\cdot)$	DNN for estimating BPD from the given magnitude		
$\mathcal{H}_{\boldsymbol{\theta}_{\mathrm{free}}}(\cdot)$	DNN for estimating FPD from the given magnitude		
$\mathcal{P}(\cdot)$	Autoregressive map for reconstructing phase from TPD and FPD		
$\texttt{Arg}(\cdot)$	Element-wise map from complex scalars to their principal arguments		

when calculating the complex STFT coefficients with the given magnitude as $A[m, n] \exp(i\Phi[m, n])$. The direct phase estimation with a DNN is still hard because a small perturbation of magnitude might imply a large phase difference.

IV. PROPOSED ONLINE PHASE RECONSTRUCTION

In this section, we propose a DNN-based online phase reconstruction framework that consists of two stages. Section IV-A shows the motivation of the two-stage framework. Then, its overview is introduced in Section IV-B. The detail of each stage is explained in Sections IV-C and IV-D, respectively. The weighting rule in the second stage is presented in Section IV-E. The symbols used in this section are summarized in Table III.

A. Motivation: Sensitivity of Phase to Waveform Shift

A DNN-based phase reconstruction method in [26] is formulated as $\widehat{\Phi} = \mathcal{F}_{\theta}(\mathbf{A})$. When training such a DNN in a supervised manner, not only the periodic nature of the phase but also the sensitivity to waveform shifts becomes a problem. Considering the Fourier transform, its phase is sensitive to waveform shifts, while its magnitude is shift-invariant. This is approximately true for STFT when waveform shifts are small. Furthermore, when the sign of a time domain signal is inverted, the corresponding STFT phase is shifted by π without changing magnitude. It is thus difficult to completely determine the phase from given magnitude³.

The effects of a waveform shift on STFT magnitude and phase are depicted in Fig. 4. We used an utterance in the LJ speech dataset⁴ and that shifted by 0.5 ms. Their TPD and FPD are depicted in the third and fourth rows where we wrap them by using the following wrapping operator:

$$W(\Phi) = \operatorname{Arg}(e^{i\Phi}).$$
 (27)

Phase of the shifted signal is noticeably different from the original one even though the magnitude remained almost the same. We thus expect that the phase itself is not easy to



⁴The LJ speech dataset is available in online: https://keithito.com/ LJ-Speech-Dataset/.



Fig. 4. Examples of the STFT magnitude, phase, and backward phase differences of an utterance and that shifted by 0.5 ms. The rightmost column shows the errors between the original and shifted ones.

estimate from the magnitude. In contrast, according to the third and fourth rows of Fig. 4, phase differences, TPD and FPD, were robust to the waveform shift.

The harmonic structure is apparent in the FPD but vague in the TPD. We thus modify TPD to BPD $[3]^5$:

$$W[m,n] = \mathcal{W}\left(V[m,n] - \frac{2\pi\alpha m}{M}\right).$$
 (28)

As depicted in the bottom row of Fig. 4, the harmonic structure is more clear in BPD than in TPD. We thus expect that BPD and FPD are easier to estimate by DNNs, where this expectation will be experimentally confirmed in Section V-H.

B. Overview: Two-stage Online Phase Reconstruction

The example in the previous subsection suggested that directly estimating phase is difficult because a DNN must connect small changes in magnitude to large differences in phase, as illustrated in Fig. 5-(i). Such an unstable map is not easy to model by a DNN. In contrast, we use DNNs to

⁵BPD (baseband phase delay) was introduced in a sinusoidal-modelbased phase reconstruction [3] and has also been used in DNN-based phase reconstruction [31]. This DNN-based phase reconstruction method uses the BPD to normalize the distribution of TPD. In Section IV-C, we will show the importance of the modification in (28) especially with CNNs.



(ii) DNN-based phase difference estimation (proposal)

Fig. 5. Comparison between (i) the existing DNN-based direct phase reconstruction and (ii) the first stage of the proposed framework. Although the estimation is performed separately for each time-frame, magnitude and phase differences at all time-frames are shown for visibility.

estimate BPD and FPD as depicted in Fig. 5-(ii). Since a small change in log-magnitude results in small changes in BPD and FPD, the maps from the log-magnitude to the phase differences should be easily modeled by DNNs. After estimating BPD and FPD, the phase is reconstructed in a frame-by-frame manner. This two-stage phase reconstruction is summarized in Fig. 2.

At the first stage, causal DNNs $\mathcal{G}_{\boldsymbol{\theta}_{\text{time}}}(\cdot)$ and $\mathcal{H}_{\boldsymbol{\theta}_{\text{freq}}}(\cdot)$ estimate BPD $\mathbf{w}_n \in \mathbb{R}^M$ and FPD $\mathbf{u}_n \in \mathbb{R}^{M-1}$ at the *n*th frame, respectively, as follows:

$$\widehat{\mathbf{w}}_n = \mathcal{G}_{\boldsymbol{\theta}_{\text{time}}}(\widetilde{\mathbf{a}}_{n-N_{\text{LB}}}, \dots, \widetilde{\mathbf{a}}_n), \tag{29}$$

$$\widehat{\mathbf{u}}_n = \mathcal{H}_{\boldsymbol{\theta}_{\text{freg}}}(\widetilde{\mathbf{a}}_{n-N_{\text{LB}}}, \dots, \widetilde{\mathbf{a}}_n), \tag{30}$$

where $N_{\text{LB}} \in \mathbb{N}$ is the number of look-back frames. We stress that both DNNs are causal and do not use magnitude at future time-frames, i.e., $N_{\text{LA}} = 0$.

At the second stage, the phase is reconstructed from the estimated phase differences. We design the following map $\mathcal{P}(\cdot)$ that computes the phase at the *n*th time-frame from that at the (n-1)th time-frame with the phase differences and magnitude:

$$\widehat{\boldsymbol{\phi}}_n = \mathcal{P}(\widehat{\boldsymbol{\phi}}_{n-1}, \widehat{\mathbf{v}}_n, \widehat{\mathbf{u}}_n, \mathbf{a}_n, \mathbf{a}_{n-1}), \quad (31)$$

where $\hat{\mathbf{v}}_n$ is the estimated TPD computed from the estimated BPD $\hat{\mathbf{w}}_n$. This map is constructed based on a weighted least squares problem of complex STFT coefficients. Its detail is postponed to Section IV-D. Since both stages do not require information on future time-frames, the proposed framework causally reconstructs the phase in a frame-by-frame manner.

C. First Stage: DNNs for Estimating Phase Differences

The proposed framework can use arbitrary causal DNNs as $\mathcal{G}_{\theta_{\rm time}}(\cdot)$ and $\mathcal{H}_{\theta_{\rm freq}}(\cdot)$. While fully connected neural networks (FCNs) have been used for DNN-based phase reconstruction [26], [27], [30]–[32], we present an efficient DNN architecture for the proposed framework.

According to the phase-magnitude relation, phase derivatives can be approximated by differences of log-magnitude in the surrounding T-F bins as in (12) and (13). In PGHI, the phase differences are approximated by averaging the phase derivatives at the T-F grids as in (18) and (19). These operations can be implemented by convolution in the T-F domain except for $2\pi \alpha m/M$ in (12). In the online setting, RTPGHI uses the second order backward difference in (22), which can also be implemented by convolution. While these mathematical formulations are concrete, we expect that estimation accuracy can be improved by exploiting prior knowledge of a target signal. For example, mixed derivative of phase is useful for analyzing harmonic signals [45], and instantaneous frequency of a sinusoidal component can be estimated from its spectral peak [22]. To acquire such complicated phase information from a dataset, DNNs should be effective.

We employ convolution layers that can efficiently aggregate information in the surrounding T-F bins. Our DNNs consist of the mean subtraction and 1-D frequency convolution layers (FreqConv) as in Fig. 6. We concatenate the log-magnitude up to the current time-frame and subtract its mean:

$$\Psi_n = \mathcal{N}(\widetilde{\mathbf{a}}_{n-N_{\rm LB}}, \dots, \widetilde{\mathbf{a}}_n), \tag{32}$$

where $\Psi_n \in \mathbb{R}^{M \times (N_{\text{LB}}+1)}$ is a frame-wise feature, and $\mathcal{N}(\cdot)$ subtracts the mean of its inputs. This map just changes the global magnitude within the inputted $N_{\text{LB}} + 1$ frames and retains the STFT phase-magnitude relation. The following first FreqConv layer treats temporal adjacencies of the inputted T-F bins as channels. As a result, the FreqConv layer can perform a causal convolution along the time-frame and mimic the operations in (13) and (22). In detail, the number of channels of the FreqConv layer corresponds to the kernel size of the causal convolutions along the time-frame. The frame-wise feature Ψ_n is passed to multiple FreqConv layers. We combine the FreqConv layers with the gating mechanism [46] as follows:

$$\begin{aligned} & \texttt{FreqGatedConv}(\Psi_n) \\ & = \texttt{Sigmoid}(\texttt{FreqConv}(\Psi_n)) \odot \texttt{FreqConv}(\Psi_n), (33) \end{aligned}$$

where the two FreqConv layers have different parameters. This mechanism can adaptively control the information passed to the next layer. Its effectiveness has been confirmed in DNN-based phase reconstruction [29]. In this first stage of the proposed framework, each feature Ψ_n is handled separately, and thus it is causal.

CNNs have difficulty of using the absolute T-F location due to their translation invariance. In the phase-magnitude relation in (12), absolute frequency information $2\pi\alpha m/M$ is required to compute the phase derivative with respect to time. It is thus difficult to estimate TPD by a CNN. In contrast, BPD removes



Fig. 6. Illustration of a DNN for estimation of BPD or FPD.

the absolute frequency information in (28) and is expected to be easily estimated by a CNN.

Supervised learning is straightforward for training DNNs that estimate BPD and FPD, because the pairs of magnitude and phase differences are easily calculated from time-domain signals. The target phase differences should be treated as circular variables because they inherit the periodic nature of the phase. Hence, we measure the errors of the estimated phase differences by the periodic loss functions as follows:

$$\mathcal{L}_{\rm BPD}(\mathbf{W}, \widehat{\mathbf{W}}) = \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} \mathcal{L}_{\rm cos}(W[m, n], \widehat{W}[m, n]), \quad (34)$$

$$\mathcal{L}_{\text{FPD}}(\mathbf{U}, \widehat{\mathbf{U}}) = \sum_{m=1}^{M-1} \sum_{n=0}^{N-1} \mathcal{L}_{\text{cos}}(U[m, n], \widehat{U}[m, n]).$$
(35)

The range of the estimated phase differences, $\widehat{W}[m,n]$ and $\widehat{U}[m,n]$, is not restricted to $[-\pi,\pi)$.

D. Second Stage: Online Phase Reconstruction From Phase Differences

As in (31), the second stage of the proposed framework $\mathcal{P}(\cdot)$ recurrently estimates the phase based on the estimated phase differences. The phase differences estimated by DNNs have ambiguity of 2π due to the use of the periodic loss function. The second stage of the proposed framework must take care of this ambiguity. Since it is not easy to directly handle such phase differences with the ambiguity, we propose to convert them to the ratios of complex STFT coefficients as follows:

$$\widehat{\mathfrak{V}}[m,n] = \frac{A[m,n]}{A[m,n-1]} e^{i\widehat{V}[m,n]},$$
(36)

$$\widehat{\mathfrak{U}}[m,n] = \frac{A[m,n]}{A[m-1,n]} \mathrm{e}^{\mathrm{i}\widehat{U}[m,n]},\tag{37}$$

where $\widehat{V}[m,n] = \widehat{W}[m,n] + 2\pi\alpha m/M$. Note that the oracle versions of these ratios are given by

$$\mathfrak{V}[m,n] = \frac{X[m,n]}{X[m,n-1]},$$
(38)

$$\mathfrak{U}[m,n] = \frac{X[m,n]}{X[m-1,n]},\tag{39}$$

which cannot be computed because the phase of X[m, n] is not available. This conversion is depicted in Fig. 7. The main advantage of this conversion is that the 2π ambiguity of the estimated phase differences is avoided.



Fig. 7. Illustration of the conversion from (i) the phase difference to (ii) the ratio of the complex STFT coefficients.

On the basis of the complex ratios, we formulate an optimization problem and estimate the phase by solving it. Let us consider the complex ratios at the *n*th time-frame $\hat{\mathbf{v}}_n$ and $\hat{\mathbf{u}}_n$ given by

$$\widehat{\mathbf{\mathfrak{v}}}_n = [\widehat{\mathfrak{V}}[0,n],\dots,\widehat{\mathfrak{V}}[M-1,n]]^{\mathsf{T}},\tag{40}$$

$$\widehat{\mathbf{u}}_n = [\widehat{\mathfrak{U}}[1, n], \dots, \widehat{\mathfrak{U}}[M-1, n]]^\mathsf{T}.$$
(41)

To enforce the complex ratios between successive time-frames close to $\hat{\mathbf{v}}_n$, we minimize the following function $\mathscr{T}(\cdot)$ with respect to an optimization variable $\mathbf{z}_n \in \mathbb{C}^M$:

$$\mathscr{T}(\mathbf{z}_n, \widehat{\mathbf{x}}_{n-1}, \widehat{\mathbf{b}}_n) = \|\mathbf{z}_n - \operatorname{diag}(\widehat{\mathbf{b}}_n)\widehat{\mathbf{x}}_{n-1}\|_{\mathbf{\Lambda}_n}^2,$$
 (42)

where $\hat{\mathbf{x}}_{n-1} = [\hat{X}[0, n-1], \dots, \hat{X}[M-1, n-1]]^{\mathsf{T}}$ is the estimated STFT coefficients at the previous time-frame, diag(·) returns the diagonal matrix whose diagonal elements are its input vector, and $\|\mathbf{z}\|_{\mathbf{\Lambda}_n}^2 = \mathbf{z}^{\mathsf{H}} \mathbf{\Lambda}_n \mathbf{z}$. Meanwhile, to enforce the complex ratios between adjacent frequencies close to $\hat{\mathbf{u}}_n$, the following function $\mathscr{S}(\cdot)$ is also minimized:

$$\mathscr{S}(\mathbf{z}_n, \widehat{\mathbf{u}}_n) = \|\mathbf{D}_n \mathbf{z}_n\|_{\mathbf{\Gamma}_n}^2, \tag{43}$$

where the matrix $\mathbf{D}_n \in \mathbb{R}^{M-1 imes M}$ is defined as

$$D_n[m-1, m-1] = -\widehat{\mathfrak{U}}[m, n],$$
(44)

$$D_n[m-1,m] = 1, (45)$$

and the other entries are zero. The weights Λ_n in (42) and Γ_n in (43) are diagonal matrices that reflect the reliability of the estimated TPD and FPD, respectively. The detail of the weights is explained in the next subsection. By using these two functions, the map $\mathcal{P}(\cdot)$ in (31) is realized as follows:

$$\widehat{\boldsymbol{\phi}}_n = \operatorname{Arg}(\underline{\mathbf{x}}_n), \tag{46}$$

$$\mathbf{x}_{n} = \operatorname*{argmin}_{\mathbf{z}_{n}} \mathscr{T}(\mathbf{z}_{n}, \widehat{\mathbf{x}}_{n-1}, \widehat{\mathbf{b}}_{n}) + \mathscr{S}(\mathbf{z}_{n}, \widehat{\mathbf{u}}_{n}), \qquad (47)$$

where (46) calculates the complex-argument element-wise, i.e., $\widehat{\Phi}[m,n] = \operatorname{Arg}(X[m,n])$. The solution of (47) $\underline{\mathbf{x}}_n = [X[0,n],\ldots,X[M-1,n]]^{\mathsf{T}}$ does not maintain the given magnitude A[m,n]. We thus modify it as $\widehat{X}[m,n] = A[m,n] \exp(i\widehat{\Phi}[m,n])$ and use the modified version in (42) for the next time-frame.

The optimization problem in (47) aims to estimate complex STFT coefficients that are consistent with the ratios calculated from the phase differences. It can be solved in a closed form:

$$\mathbf{x}_n = (\mathbf{\Lambda}_n + \mathbf{D}_n^\mathsf{T} \mathbf{\Gamma}_n \mathbf{D}_n)^{-1} \mathbf{\Lambda}_n \mathbf{y}_n, \tag{48}$$

where the *m*th entry of \mathbf{y}_n is given by $\hat{\mathbf{v}}[m, n] \widehat{X}[m, n-1]$. By using (48), the proposed method reconstructs the phase in a frame-by-frame manner without any iterative optimization.

E. Weighting Rule and Initialization

In the optimization-based phase reconstruction given in (46) and (47), the weights, Λ_n and Γ_n , are important to improve the quality of the reconstructed signal. The phase differences with large weights are maintained, and thus the weights must be designed based on the reliability of the estimated phase differences. We propose to design the *m*th diagonal entry of Λ_n and Γ_n by using the given magnitude:

$$\Lambda_n[m,m] = (A[m,n]A[m,n-1])^p,$$
(49)

$$\Gamma_n[m,m] = \gamma_0 (A[m,n]A[m-1,n])^p,$$
(50)

where p is a parameter for compressing or enhancing the magnitudes, and $\gamma_0 \ge 0$ is a parameter to balance the two weights. These weights are based on the assumption that the ratios of complex STFT coefficients are accurate when magnitude at the related T-F bins is large.

As a special case, the proposed method in (48) results in the integration of the estimated TPD over time when $\gamma_0 = 0$:

$$\overline{\Phi}[m,n] = \overline{\Phi}[m,n-1] + V[m,n].$$
(51)

This phase reconstruction was already used in a DNN-based method [25]. Its performance is limited because the relation between adjacent STFT coefficients in the frequency direction is neglected. The proposed method with $\gamma_0 > 0$ uses the relations in both time and frequency directions. Another related work [47] applies some weight designed from the given magnitude to training of DNNs that estimate phase. In contrast, we use the weights for reconstructing phase from the phase differences but not for training DNNs.

The recurrent phase reconstruction in (48) is not applicable to the initial time-frame because $\widehat{\mathbf{x}}_{n-1}$ is not given. We compute the phase at the initial time-frame $\widehat{\Phi}[m, 0]$ by accumulating the estimated FPD. In our preliminary experiments, however, it often resulted in a similar performance with other initialization methods, e.g., the zero and random phases. This should be because the estimated FPD is unreliable due to a small magnitude at the initial time-frame. The proposed method is robust against errors at the T-F bins with small magnitudes because the relations between the T-F bins with large magnitudes are emphasized by the weights in (49)–(50). In detail, if the T-F bins at the previous time-frame have small magnitude, the proposed method tries to maintain the estimated FPD at the current time-frame and neglect the phase at the previous time-frame.

V. EXPERIMENTS

In this section, we investigate the performance of the proposed DNN-based two-stage framework in online phase reconstruction. The experimental conditions are described in Section V-A. Section V-B compares the proposed framework with various online and offline phase reconstruction methods. The generalization capability and robustness of the proposed

8

TABLE IV EXPERIMENTAL CONDITIONS

Parameters of DNN Architecture				
# of FreqConv layers	1 + 1			
<pre># of FreqGatedConv layers</pre>	5			
# of channels	64			
Kernel size of FreqGatedConv layers	3			
Kernel size of FreqConv layers	1			
$N_{\rm LB}$	3			
# of parameters	206k			
Parameters for Training				
Optimizer	RAdam [50]			
Base learning rate	0.0004			
Batch size	32			
# of epochs	100			
# of warmup epochs	5			
Weight decay	10^{-6}			
Maximum norm of the gradients	10			

framework are shown in Sections V-C and V-D, respectively. The effectiveness of our CNN for the first stage is validated in Section V-E. We investigate the effect of the weight parameters, p and γ_0 , on the quality of the reconstructed signals in Section V-F. Section V-G demonstrates the effectiveness of the optimization-based phase reconstruction method in the second stage. Finally, the proposed two-stage framework is compared with direct phase reconstruction in Section V-H.

A. Experimental Conditions

1) Dataset and STFT Parameters: Evaluations were performed on the LJ speech dataset that consists of 13100 audio clips uttered by a female speaker. The audio clips were sampled at 22050 Hz and randomly splitted into three subsets: 12500 clips for training, 300 clips for validation, and 300 clips for testing as in [48]. During the training, the utterances were further divided into about 1-second-long segments (24064 samples). The validation set was used to optimize the hyperparameters for the second stage. STFT was computed with the Hann window, where the window size and shift size were 1024 and 256 samples, respectively. We used $ltfatpy^6$ to implement STFT and related transformations.

2) DNN Configuration and Training Setup: The DNN used in the following experiments is illustrated in Fig. 6. It consists of the mean subtraction layer, a FreqConv layer, and five FreqGatedConv layers followed by another FreqConv layer. We set the number of look-back frames $N_{\rm LB}$ to 3 based on the overlap of the window for STFT. Other configurations are summarized in Table IV.

To train the DNNs, we used the RAdam optimizer [50] for 100 epoch where the batch size was 32. We linearly warmed up the learning rate for 5 epochs to 0.0004 and adopt the halfperiod cosine scheduler [51]. We applied a weight decay of 10^{-6} and a gradient clipping of 10 for stable training, which were implemented in Pytorch [52].

⁶ltfatpy is available under: https://dev.pages.lis-lab.fr/ltfatpy/. It is a python version of LTFAT: http://ltfat.org/ [49].



Fig. 8. Boxplots of PESQ, ESTOI, and LSC for 300 reconstructed utterances. Blue and red boxes correspond to online and offline phase reconstruction methods, respectively. Higher PESQ and ESTOI indicate better sound quality. Lower LSC indicates better phase reconstruction.

3) Evaluation Metrics: The results of phase reconstruction were evaluated by three objective measures. The first one is the log-spectral convergence (LSC) [53] defined by

$$LSC(\widehat{\mathbf{X}}, \mathbf{A}) = 20 \log_{10} \frac{\|\mathbf{A} - |STFT(iSTFT(\mathbf{X}))|\|_{Fro}}{\|\mathbf{A}\|_{Fro}}, (52)$$

where $\widehat{X}[m,n] = A[m,n] e^{\widehat{\Phi}[m,n]}$, and $\|\cdot\|_{\text{Fro}}$ denotes the Frobenius norm. When the estimated phase is perfect, i.e., $\widehat{\Phi} = \Phi$, iSTFT and STFT do not alter the magnitude of \widehat{X} , and LSC becomes $-\infty$. The second and third ones are the wide-band extension of the perceptual evaluation of subjective quality (PESQ) [54] and the extended short-time objective intelligibility (ESTOI) [55]. These objective measures have been commonly used to evaluate naturalness and intelligibility of the results of phase-aware speech enhancement [56] and separation [57].

B. Comparison to Existing Online Phase Reconstruction

To validate the effectiveness of the proposed DNN-based phase reconstruction, we compared the proposed method with three online phase reconstruction methods: RTPGHI [38], a consistency-based phase reconstruction method called realtime iterative spectrogram inversion (RTISI) [37], and a sinusoidal-model-based phase reconstruction method called single pass spectrogram inversion (SPSI) [22]. For RTPGHI and RTISI, we set the number of look-ahead frames $N_{\rm LA}$ to 0, i.e., all methods were set causal. We also investigated the performance of two offline methods: PGHI and an offline version of the proposed method. In the offline proposed method, we additionally concatenated three look-ahead frames as the input of the DNNs. All existing methods are implemented in PHASERET [42]. In RTISI, the number of pertime-frame iterations was set to 5, which is the default value of PHASERET. We would like to stress that this is not a fair comparison because only the proposed method uses a DNN trained on utterances of the target speaker. This comparison, however, demonstrates the great potential of incorporating

TABLE V MEDIAN OF PESQ AND ESTOI OF 100 RECONSTRUCTED UTTERANCES FOR EACH SPEAKER.

		Female			Male	
	p225	p228	p229	p226	p227	p232
PESQ						
RTPGHI Proposed	3.52 4.31	3.35 4.22	3.27 4.25	3.40 4.27	3.19 4.29	3.38 4.31
	ESTOI					
RTPGHI Proposed	0.840 0.949	0.830 0.959	0.859 0.969	0.823 0.957	0.862 0.969	0.874 0.975

the prior knowledge into online phase reconstruction. The generalization capability of the proposed method is validated in the next subsection. Furthermore, the performance of the existing methods combined with the DNNs are investigated in Section V-G.

PESQ, ESTOI, and LSC of the reconstructed signals are summarized in Fig. 8. The two methods that do not utilize prior knowledge of target signals, RTPGHI and RTISI, resulted in similar performance. Although SPSI considers the sinusoidal model for target signals, it performed worse than the other methods in our experiment. This should be a consequence of the mismatch between the signal model and actual signals. The proposed method was able to outperform all of the existing methods. Note that the proposed method also outperformed the offline PGHI. This result confirms the advantage of leveraging prior knowledge of the target signals learned by DNNs. The offline proposed method substantially improved the performance from that of the online version. That is, the performance of the two-stage phase reconstruction can be improved by leveraging look-ahead frames.

C. Generalization for Unseen Speakers

To clarify the generalization capability of the proposed method, we evaluated it on unseen speakers. While the training and validation data were from the LJ speech dataset, the evaluation was performed on utterances of three females (p225, p228, p229) and three males (p226, p227, p232) from the VCTK corpus. For each speaker, 100 utterances were randomly selected and resampled at 22050 Hz as in [29].

The experimental results are summarized in Table V. Even on the unseen speakers, the proposed method outperformed RTPGHI which was the best reference method in the previous experiment. This result confirms the generalization capability of the proposed framework even when the training dataset consists of utterances of a single speaker. Training on utterances of multiple speakers might improve the generalization capability further.

D. Application to Mel-Spectrogram Inversion

In many applications, the given STFT magnitude contains some errors. To investigate the robustness against such errors, we validated RTPGHI and the proposed method on the magnitude recovered from that compressed to the mel scale. Mel-



Fig. 9. Average PESQ and ESTOI of utterances reconstructed from the degraded STFT magnitudes. The magnitudes in the linear scale with 513 bins were recovered from the mel-spectrograms in different number of bins.

spectrograms have been widely used as acoustic features in audio synthesis, and phase reconstruction has been applied to the magnitudes recovered from them [58], [59]. In this experiment, the power-compressed magnitude in the linear scale with 513 bins, $A[m, n]^{0.3}$, was converted to the mel scale with a smaller number of bins $M_{\text{mel}} \in \{80, 160, 240, 320, 400\}$. Then, the mel-scale magnitude is converted back to that in the linear scale with regular power by solving a nonnegative least squares problem, which is implemented in Librosa [60]. The power compression has been used to maintain the components with a small magnitude in least squares as in [61] and was effective for both RTPGHI and the proposed method. The smaller M_{mel} caused more error in the recovered linear-scale magnitude.

PESQ and ESTOI of the reconstructed signals are shown in Fig. 9. In addition to RTPGHI and the proposed method, we evaluated the recovered magnitude with the true phase. This is an upper bound of the performance of phase reconstruction. When $M_{\rm mel} > 80$, the proposed method substantially outperformed RTPGHI. Recently, DNNs have been used to recover the magnitude in the linear scale from that in the mel scale [58], [59]. These DNN-based methods should improve the quality of the recovered magnitude from the nonnegative least squares used in this experiment. We thus expect that the performance of the proposed method is improved by incorporating it with the DNN-based estimation of the magnitude in the linear scale.

E. Effectiveness of CNN to Estimate BPD and FPD

To validate the effectiveness of the proposed CNN for estimating BPD and FPD, we compared its estimation accuracy with that of an FCN. The CNN was the same as that used in the previous experiment (Fig. 6). The FCN comprised 3 gated linear units of 1024 units and a linear output layer as in [26], where its number of total parameters was 8929k which is about 43 times more than that of the CNN. The input of the FCN was the log-magnitudes up to the current time-frame as in (32), but we concatenated them along with the frequency direction. The training configuration was the same as in Section V-A2. The



Fig. 10. Histogram of the absolute wrapped error (AWE) of estimated BPD and FPD. The vertical axis is proportion to the total number of all T-F bins and audio clips. The number inside the above parentheses represents median.

estimated phase differences were evaluated by the following absolute wrapped error (AWE):

$$\mathcal{L}_{abs}(\phi, \widehat{\phi}) = \left| \mathcal{W}(\phi - \widehat{\phi}) \right|.$$
(53)

The histograms of AWE of the estimated BPD and FPD are illustrated in Fig. 10. These histograms are more biased towards the left when the estimates were more accurate. AWE of RTPGHI is summarized in the rightmost column, where TPD computed by (18) was converted to BPD. The accuracy of FPD was significantly worse than that of BPD because RTPGHI in the causal setting must use the second order backward difference in (22) instead of the centered difference for computing FPD. If the second order centered difference was used by allowing one look-ahead frame, the median of AWE was reduced to 0.389 from 0.701. Even though the DNNs did not use any look-ahead frames, they achieved notably better accuracy compared to RTPGHI. By efficiently aggregating information in the surrounding T-F bins, the CNN outperformed the FCN with 43 times fewer parameters. Consequently, as shown in Table VI, the objective measures of the reconstructed utterances were significantly improved by using the CNN for the estimation of phase differences.

To demonstrate the difficulty of estimating TPD using a CNN, we compared a CNN and an FCN by directly estimating TPD. Note that the proposed framework does not estimate TPD itself, and hence we trained another DNN for this experiment. The histograms of AWE for TPD estimation are



Fig. 11. Histogram of AWE of estimated TPD.



Fig. 12. Average PESQ and ESTOI on the validation set. The parameters varied on the logarithmic scale in both axes. We limit the color ranges to clarify the peaks, which results in saturation for $p > 10^{0.4}$.

depicted in Fig. 11. The FCN achieved performance similar to that for BPD in Fig. 10. In contrast, TPD estimation by CNN resulted in much more error compared to that of BPD. This result indicates that estimation of TPD is difficult for the CNN as discussed in Section IV-C. Hence, the conversion of the target from TPD to BPD is essential for the CNN.

F. Effect of the Weight Parameters p and γ_0

As discussed in Section IV-E, the weights are important in the second stage of the proposed framework. In our experiments, the optimal weight parameters were obtained by using the validation set, where the phase differences were estimated by the CNNs trained as Section V-A. The search range of pand γ_0 were set to [0.1, 10] and [1, 100], respectively.

Fig. 12 shows PESQ and ESTOI of the reconstructed signals on the validation set. The proposed framework performed well with wide range of parameters, i.e., it is not so sensitive to these parameters. ESTOI took the maximum value 0.996 at $p = 10^{-0.4}$ and $\gamma_0 = 10$, which also gives a high PESQ value. Hence, these parameters were used in the other experiments.



Fig. 13. Boxplots of PESQ, ESTOI, and LSC of signals reconstructed from the phase differences approximated by (18) and (19). The time integration of TPD [25] and the adaptive integration of TPD and FPD [38] are abbreviated as Time Int. and Adaptive Int., respectively.

G. Evaluation of Various Methods for Reconstructing Phase From Estimated Phase Differences

In this experiment, the second stage of the proposed framework was evaluated. We compared the second stage of the proposed framework with existing methods: the time integration of the estimated TPD used in [25] defined by (51), and the recurrent phase unwrapping (RPU) presented in our conference paper [30]. The difference between RPU and the proposal of this paper is the definition of least squares problems; RPU solves the least squares problem of phase without weighting, while the second stage of the proposed framework solves the least squares problem of complex STFT coefficients with weighting. We also evaluated the adaptive integration of TPD and FPD used in RTPGHI [38]. Note that, if the oracle phase differences are available, all methods can perfectly reconstruct the phase up to the global constant. To investigate their robustness to the error in the phase differences, this experiment used the analytic formulas in (18) and (19) or the CNNs to estimate the phase differences.

Fig. 13 summarizes the results using the phase differences computed by the analytic formulas in (18) and (19). The performance of the adaptive integration and the proposed method was significantly better than that of the time integration and RPU. The former group uses the magnitude to reconstruct the phase from the estimated phase differences, while the latter group does not. As a result, the former group is more robust to the estimation error at T-F bins with small magnitudes.

Fig. 14 shows the results using the phase differences estimated by the CNNs. According to Section V-E, the estimation accuracy of FPD was improved by using the CNN from that of (19). The performance was improved in all methods except for the time integration that does not use FPD. The proposed optimization-based method outperformed the other methods including the adaptive integration. This should be because the proposed method jointly optimizes all the phase at each timeframe based on the carefully designed weight.



Fig. 14. Boxplots of PESQ, ESTOI, and LSC of signals reconstructed from the phase differences estimated by the CNNs.



Fig. 15. Block diagram of the comparison between (i) conventional direct phase reconstruction and (ii) proposed two-stage phase reconstruction. The reconstructed phase $\widehat{\Phi}$ was evaluated through the BPD $\widehat{\underline{W}}$ and FPD $\widehat{\underline{U}}$ calculated from it. Although magnitude and phase for all time-frames are shown here, phase reconstruction was conducted in a frame-by-frame manner.

H. Comparison with DNN-based Direct Phase Reconstruction

In this experiment, the proposed two-stage framework is compared with the direct phase reconstruction using the FCN or CNN. For the direct phase reconstruction, the DNN was trained to minimize the negative cosine loss function of phase in (25). We further refined the estimated phase by an iterative offline phase reconstruction method called Griffin– Lim algorithm (GLA) as in the original paper [26]. The number of iterations of GLA was 100. The reconstructed phase was evaluated by AWE of the phase differences⁷. We also investigated the total performance of the proposed twostage framework in this way. To be specific, we evaluated not the phase differences estimated in the first stage but those computed from the output of the second stage as follows:

$$\underline{\widehat{W}}[m,n] = \widehat{\Phi}[m,n] - \widehat{\Phi}[m,n-1] - \frac{2\pi\alpha m}{M}, \qquad (54)$$

$$\underline{\widehat{U}}[m,n] = \widehat{\Phi}[m,n] - \widehat{\Phi}[m-1,n].$$
(55)

This evaluation is illustrated in Fig. 15.

The results of the DNN-based phase reconstruction methods using FCNs and CNNs are summarized in Figs. 16 and 17, respectively. The direct phase reconstruction resulted in the lowest performance regardless of the types of DNN⁸. This result indicates the difficulty of directly estimating phase as discussed in Section IV-A. In contrast, the two-stage framework with the FCNs successfully reconstructed phase up to global constant phase. Although the direct phase reconstruction performed better when GLA was applied, the proposed framework with the CNNs outperformed it. Note that this comparison is unfair because GLA is an iterative offline method that uses the information from all time-frames. Even though the proposed two-stage method is non-iterative and causal, it was able to outperform the combination of the DNN-based and iterative methods.

Although the proposed two-stage method performed better than the existing methods, there is room for improvement. According to Fig. 10, AWE of BPD and FPD estimated in the first stage using the FCNs were 0.417 and 0.480, respectively, and those using the CNNs were 0.091 and 0.117, respectively. The results in Figs. 16 and 17 indicates that the output of the second stage was worse than those obtained in the first stage in terms of phase differences. This implies that the second stage of the proposed framework still has room for improvement. Refinement of the second stage is left as a future work.

VI. CONCLUSION

In this paper, we have presented a two-stage online phase reconstruction framework. In the framework, BPD and FPD are estimated by the causal DNNs based on 1-D frequency convolution layers. Then, phase is reconstructed from the estimated phase differences by analytically solving the weighted least squares problem of complex STFT coefficients in a frame-byframe manner. We confirmed that the proposed method outperformed existing online phase reconstruction methods. We also demonstrated the effectiveness of the two-stage framework by comparison with the direct reconstruction method.

In future works, we will apply the two-stage online framework to low-latency speech enhancement and separation. In these applications, the noisy phase of an observed signal is useful to improve the estimation accuracy of the phase differences. Fine-tuning of the DNNs used in the first stage to maximize the quality of the signals reconstructed by the second stage is also a possible direction for improving the proposed framework. Together with improvement of the second stage as discussed in Section V-H, optimizing the whole process of the proposed framework is the next step for realizing a better phase reconstruction method.

⁸This is not due to the evaluation metric. We confirmed that the estimated phase had large error which is reflected in BPD and FPD of the figures.

⁷We did not evaluate the estimated phase directly because global phase shift is not reflected in the perceptual quality.



Fig. 16. Histogram of AWE of the differences of reconstructed phase using FCN. Median over all T-F bins and audio clips is given in the parentheses.



Fig. 17. Histogram of AWE of the differences of reconstructed phase using CNN. Median over all T-F bins and audio clips is given in the parentheses.

REFERENCES

 K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Commun.*, vol. 53, no. 4, pp. 465–494, Apr. 2011.

- [2] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 217–220, Mar. 2013.
- [3] M. Krawczyk and T. Gerkmann, "STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1931–1940, Dec. 2014.
- [4] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 55–66, Mar. 2015.
- [5] P. Mowlaee and J. Kulmer, "Harmonic phase estimation in singlechannel speech enhancement using phase decomposition and SNR information," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 9, pp. 1521–1532, Sep. 2015.
- [6] P. Mowlaee, R. Saeidi, and Y. Stylianou, "Advances in phase-aware signal processing in speech communication," *Speech Commun.*, vol. 81, pp. 1–29, Jul. 2016.
- [7] P. Magron, R. Badeau, and B. David, "Model-based STFT phase recovery for audio source separation," *IEEE/ACM Trans. Audio, Speech* and Lang. Proc., vol. 26, no. 6, pp. 1095–1105, Jun. 2018.
- [8] N. Takahashi, P. Agrawal, N. Goswami, and Y. Mitsufuji, "PhaseNet: Discretized phase modeling with deep neural networks for audio source separation," in *INTERSPEECH*, Sep. 2018, pp. 2713–2717.
- [9] J. Le Roux, G. Wichern, S. Watanabe, A. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 2, pp. 370– 382, May 2019.
- [10] Z.-Q. Wang, K. Tan, and D. Wang, "Deep learning based phase reconstruction for speaker separation: A trigonometric perspective," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2019, pp. 71–75.
- [11] Y. Masuyama, K. Yatabe, K. Nagatomo, and Y. Oikawa, "Joint amplitude and phase refinement for monaural source separation," *IEEE Signal Process. Lett.*, vol. 27, pp. 1939–1943, Oct. 2020.
- [12] 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 *INTERSPEECH*, Aug. 2017, pp. 1128– 1132.
- [13] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for STFT spectrograms," in *INTER-SPEECH*, Aug. 2017, pp. 3389–3393.
- [14] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, Aug. 2017, pp. 4006–4010.
- [15] Y. Saito, S. Takamichi, and H. Saruwatari, "Text-to-speech synthesis using STFT spectra based on low-/multi-resolution generative adversarial networks," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2018, pp. 5299–5303.
- [16] Y. Ephraim and D. Malah, "Speech enhancement using a minimummean square error short-time spectral amplitude estimator," *IEEE/ACM Trans. Acoust., Speech, Signal Proc.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.
- [17] P. Magron, R. Badeau, and B. David, "Phase recovery in NMF for audio source separation: An insightful benchmark," in *IEEE Int. Conf. Acoust.*, *Speech, Signal Process. (ICASSP)*, Apr. 2015, pp. 81–85.
- [18] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, Apr. 1984.
- [19] N. Perraudin, P. Balazs, and P. L. Søndergaard, "A fast Griffin-Lim algorithm," in *IEEE Workshop Appl. Signal Process. Audio Acoust.* (WASPAA), Oct. 2013, pp. 1–4.
- [20] Y. Masuyama, K. Yatabe, and Y. Oikawa, "Griffin–Lim like phase recovery via alternating direction method of multipliers," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 184–188, Jan. 2019.
- [21] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from STFT magnitude," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 5, pp. 1154–1164, May 2017.
- [22] G. T. Beauregard, M. Harish, and L. Wyse, "Single pass spectrogram inversion," in *IEEE Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2015, pp. 427–431.
- [23] P. Magron, R. Badeau, and B. David, "Phase reconstruction of spectrograms with linear unwrapping: Application to audio signal restoration," in *Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2015, pp. 1–5.

- [24] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrograms," in *Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2514–2518.
- [25] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Int. Conf. Learn. Represent. (ICLR)*, Apr. 2019.
- [26] 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 *Int. Workshop Acoust. Signal Enhance. (IWAENC)*, Sep. 2018, pp. 286–290.
- [27] —, "Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks," *Signal Process.*, vol. 169, p. 107368, Apr. 2020.
- [28] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep Griffin–Lim iteration," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2019, pp. 61–65.
- [29] —, "Deep Griffin–Lim iteration: Trainable iterative phase reconstruction using neural network," *IEEE J. Sel. Top. Signal Process.*, vol. 15, no. 1, pp. 37–50, Jan. 2021.
- [30] —, "Phase reconstruction based on recurrent phase unwrapping with deep neural networks," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2020, pp. 826–830.
- [31] L. Thieling, D. Wilhelm, and P. Jax, "Recurrent phase reconstruction using estimated phase derivatives from deep neural networks," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 7088–7092, Jun. 2021.
- [32] N. B. Thien, Y. Wakabayashi, K. Iwai, and T. Nishiura, "Two-stage phase reconstruction using DNN and von Mises distribution-based maximum likelihood," in Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC), Dec. 2021, pp. 995–999.
- [33] M. Portnoff, "Magnitude-phase relationships for short-time fourier transforms based on gaussian analysis windows," *IEEE Int. Conf. Acoust.*, *Speech, Signal Process. (ICASSP)*, pp. 186–189, Apr. 1979.
- [34] F. Auger, É. Chassande-Mottin, and P. Flandrin, "On phase-magnitude relationships in the short-time fourier transform," *IEEE Signal Process. Lett.*, vol. 19, no. 5, pp. 267–270, May 2012.
- [35] T. Yanagita, S. Sakti, and S. Nakamura, "Neural iTTS: Toward synthesizing speech in real-time with end-to-end neural text-to-speech framework," in *ISCA Speech Synth. Workshop (SSW)*, Sep. 2019, pp. 183–188.
- [36] P. Magron and T. Virtanen, "Online spectrogram inversion for lowlatency audio source separation," *IEEE Signal Process. Lett.*, vol. 27, pp. 306–310, Jan. 2020.
- [37] X. Zhu, G. T. Beauregard, and L. L. Wyse, "Real-time signal estimation from modified short-time fourier transform magnitude spectra," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 5, pp. 1645–1653, Jul. 2007.
- [38] Z. Pruša and P. L. Søndergaard, "Real-time spectrogram inversion using phase gradient heap integration," in *Int. Conf. Digit. Audio Effects* (*DAFx*), Sep. 2016, pp. 17–21.
- [39] H. G. Feichtinger and T. Strohmer, Gabor Analysis and Algorithms. Birkhäuser, Boston, MA, 1998.
- [40] K. Gröchenig, Foundations of Time-Frequency Analysis. Birkhäuser, Boston, MA, 2001.
- [41] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *ISCA Speech Synth. Workshop (SSW)*, Sep. 2016, p. 125.
- [42] Z. Pruša, "The phase retrieval toolbox," in AES Int. Conf. Semant. Audio, Jun. 2017.
- [43] E. J. Candès, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure and Appl. Math.*, vol. 66, no. 8, pp. 1241– 1274, Nov. 2013.
- [44] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," *Math. Programm.*, vol. 149, no. 1, pp. 47–81, Feb. 2015.
- [45] Y. Masuyama, K. Yatabe, and Y. Oikawa, "Model-based phase recovery of spectrograms via optimization on Riemannian manifolds," in *Int. Workshop Acoust. Signal Enhance. (IWAENC)*, Sep. 2018, pp. 126–130.
- [46] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," arXiv:1612.08083, 2016.
- [47] A. A. Nugraha, K. Sekiguchi, and K. Yoshii, "A deep generative model of speech complex spectrograms," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2019, pp. 905–909.

- [48] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2019, pp. 3617–3621.
- [49] Z. Průša, P. L. Søndergaard, P. Balazs, and N. Holighaus, "LTFAT: A Matlab/Octave toolbox for sound processing," in *Int. Symp. Computer Music Multidiscip. Res. (CMMR)*, Oct. 2013, pp. 299–314.
- [50] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Int. Conf. Learn. Represent. (ICLR)*, Apr. 2020.
- [51] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Int. Conf. Learn. Represent. (ICLR)*, Apr. 2017.
- [52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst.*, Dec. 2019.
- [53] N. Strumel and L. Daudet, "Signal reconstruction from STFT magnitude: a state of the art," in *Int. Conf. Digit. Audio Effects (DAFx)*, Sep. 2011, pp. 375–386.
- [54] P.862.2: Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs, ITU-T Std. P.862.2, 2007.
- [55] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 11, pp. 2009–2022, Aug. 2016.
- [56] N. Zheng and X. L. Zhang, "Phase-aware speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 63–76, Jan. 2019.
- [57] Z. Q. Wang, G. Wichern, and J. Le Roux, "On the compensation between magnitude and phase in speech separation," *IEEE Signal Process. Lett.*, vol. 28, pp. 2018–2022, Sep. 2021.
- [58] J. Lee, H. S. Choi, C. B. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end korean singing voice synthesis system," in *Interspeech*, Sep. 2019, pp. 2588–2592.
- [59] F. Yang, S. Yang, P. Zhu, P. Yan, and L. Xie, "Improving mandarin endto-end speech synthesis by self-attention and learnable Gaussian bias," in *IEEE Autom. Speech Recognit. Underst. Workshop (ASRU)*, Dec. 2019, pp. 208–213.
- [60] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *Python Science Conf.*, Jul. 2015, pp. 18–24.
- [61] S. Wisdom, J. R. Hershey, K. Wilson, J. Thorpe, M. Chinen, B. Patton, and R. A. Saurous, "Differentiable consistency constraints for improved deep speech enhancement," in *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2019, pp. 900–904.



Yoshiki Masuyama received his B.E. and M.E. degrees from Waseda University in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree with the Graduate School of Systems Design, Tokyo Metropolitan University.



Kohei Yatabe received the B.E., M.E., and Ph.D. degrees from Waseda University, in 2012, 2014, and 2017, respectively. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Tokyo University of Agriculture and Technology.



Kento Nagatomo received the B.E. and M.S. degrees from the Department of Intermedia Art and Science, Waseda University in 2019 and 2021.



Yasuhiro Oikawa received the B.E, M.E., and Ph.D. degrees in electrical engineering from Waseda University in 1995, 1997, and 2001, respectively. He is currently a Professor with the Department of Intermedia Art and Science, Waseda University. His research interests include communication acoustics and digital signal processing of acoustic signals. He is a member of ASJ, ASA, IEICE, IPSJ, VRSJ, and AIJ.