# CIS-BWE: Chaos-Informed Speech Bandwidth Extension

Anonymous ACL submission

## Abstract

We design CIS-BWE, a novel adversarial Bandwidth Extension (BWE) framework that introduces two chaos-informed discriminators - Multi-Resolution Lyapunov Discriminator (MRLD) and Multi-Scale Detrended Fractal Analysis Discriminator (MSDFA) - for capturing the deterministic chaos from speech. MRLD exploits Lyapunov exponents to capture nonlinear chaotic fluctuations. MSDFA exploits detrended fluctuation analysis to quantify fractal-like, long-range temporal chaotic correlations. To the best of our knowledge, MRLD and MSDFA are included here for the first time with a complex-valued adversarial network to explore the chaotic study of speech reconstruction. We also introduce a novel complex-valued and dual-stream generator, which uses our newly proposed ConformerNeXt as a core block with Lattice interactions, acting as a gating mechanism by enabling controlled mixing of information across streams. We extensively optimize our design across five resolutions and use depth-wise separable convolutions to make our model lightweight yet powerful. Our CIS-BWE is tested with a de facto English and French dataset for clean and noisy speech for generalization. It achieves better performance across a total of eight subjective and objective evaluation metrics with a 40x reduction in discriminator size and overall 0.5x fewer parameters, establishing a new baseline in the BWE task.

## 1 Introduction and Related Work

Bandwidth Extension (BWE) does speech enhancement by reconstructing missing high frequencies from low-frequency data (Li and Lee, 2015). BWE is critical for wide range of Natural Language Processing (NLP) applications, such as Text-To-Speech (TTS) (Feng et al., 2019) and Automatic Speech Recognition (ASR) (Haws and Cui, 2019). BWE has been shown to preserve critical spectral cues and thus improve recognition accuracy on low-bandwidth inputs for TTS and ASR.

The advancement of deep learning transforms the BWE landscape. The early neural methods focused mainly on magnitude spectrogram enhancement (Li and Lee, 2015; Sui et al., 2024; Abreu and Biscainho, 2024; Hu et al., 2022; Lu et al., 2025), neglecting the phase due to its notorious noisy patterns and relying on vocoders for audio reconstruction (Ho et al., 2025; Liu et al., 2022a; Tamiti and Barua, 2025). However, due to the work of (Gerkmann et al., 2012; Lu et al., 2025; Tamiti et al., 2025; Tamiti and Barua, 2025), it is proven that enhancing phase together with magnitude (or real and imaginary) yields higher perceptual quality audio (Yin et al., 2020) at the cost of increased computational complexity. However, none of them considers the *chaotic modeling* of speech generation and hence, they miss the opportunity to improve their performance with less complexity. We refer to Appendix A.1 to understand the origin of chaotic presence in human speech.

Speech production is fundamentally non-linear with the presence of deterministic chaos due to the complex interaction between airflow and deformable vocal tract (Herzel et al., 1994). These chaotic dynamics are different from the ones present in phase and sensitively dependent on slow-variant features (Michael, 1999). Although the Multi Period Discriminator (MPD) and Multi Scale Discriminator (Kong et al., 2020), Multi Band Discriminator (Yang et al., 2021), and Multi-Resolution Spectrogram Discriminator (Jang et al., 2021) are proposed to determine the nonlinear cues among temporal structures, they lack a chaotic feature extraction framework. Therefore, they fail to capture the intricate chaotic features, leading to residual artifacts and temporal blurring (Kim et al., 2021). Moreover, they are usually parameter-heavy, resulting in extra computational overhead.

In this paper, we propose a novel class of chaos-

informed discriminators for capturing the deterministic chaos, which State-of-the-Art (SOTA) work overlooks. We design two chaos-inspired discriminators - Multi-Resolution Lyapunov Discriminator (MRLD) and Multi-Scale Detrended Fluctuation Analysis Discriminator (MSDFA). MRLD uses Lyapunov exponents (Oseledec, 1968) to capture rapid and nonlinear chaotic fluctuations. MSDFA uses detrended fluctuation analysis (Peng et al., 1994) to capture fractal-like temporal correlations. This is the first time that MRLD and MSDFA have been proposed to be included with complex-valued Generative Adversarial Networks (GANs) to explore the chaotic study of audio reconstruction, to the best of our knowledge. Our extensive design optimization across five resolutions and the use of depth-wise separable convolution make MRLD and MSDFA lightweight yet powerful, which requires 0.5x fewer parameters, a 40x reduction in discriminator size compared to SOTA (Lu et al., 2024a), with better performance across a wide range of subjective and objective metrics.

We also introduce a novel complex-valued and dual-stream generator, which uses ConformerNeXt as a core block with Lattice interactions, acting as a gating mechanism by enabling controlled mixing of information across streams. Proposed ConformerNeXt is a combination of Transformer-based Conformer (Gulati et al., 2020) for capturing global context and Convolution Neural Network (CNN)-based ConvNeXt (Liu et al., 2022b) for capturing local context efficiently present in speech. This optimized generator architecture simultaneously enhances magnitude and phase stream within a compact yet efficient architecture. We name our proposed model **CIS-BWE** (**C**haos-**I**nformed **S**peech BWE). Our key technical contributions are:

**(1)** For the first time, we introduce chaos-informed and parameter-efficient non-linear discriminators to capture deterministic chaos.

**(2)** We introduce dual-stream ConformerNeXt with Lattice interactions for controlled feature mixing for high-fidelity speech reconstruction.

**(3)** We extensively evaluate performance using six objective metrics: LSD, PESQ, STOI, SI-SDR, SI-SNR, and NISQA-MOS; two subjective metrics: MOS and Pairwise preference test; three real-time performance metrics: MAC, FLOP, and RTF (refer to Sections 3.1, 3.11, and 3.12 for full form).

## 2 Overall Architecture

The proposed architecture is illustrated in Fig. 1.

### 2.1 Generator Architecture

Due to the necessity of phase along with amplitude (Yin et al., 2020; Lu et al., 2024b), the generator of our CIS-BWE parallely receives both the amplitude and phase cues from the two computed synchronized feature maps. The magnitude spectrogram is obtained by taking the logarithm of the absolute value of the Short-Time Fourier Transform (STFT), and the phase spectrogram is obtained by taking the angle of the STFT. We define the narrow-band magnitude and phase spectrograms by $\mathbf{M}_{\text{nb}}$ and $\mathbf{\Phi}_{\text{nb}}$, respectively, in Eqn. 1.

$$\mathbf{M}_{\text{nb}} \in \mathbb{R}^{B\times F\times T}, \quad \mathbf{\Phi}_{\text{nb}} \in \mathbb{R}^{B\times F\times T} \tag{1}$$

where $B$ is the batch size, $F$ is the number of frequency bins ($F = \frac{n_\text{FFT}}{2} + 1$), and $T$ is the number of time frames. The generator's forward pass consists of the following three main stages:

**a) Dual Stream Processing:** The generator initially processes the narrow-band magnitude and phase in two separate streams and merges them into a common latent space at the end, harmonizing heterogeneous inputs into a unified projection.

**b) Lattice Block Interaction:** The Lattice block (Luo et al., 2020) ensures continuous controlled mixing of magnitude and phase streams that enables explicit exchange of information, reweighing each stream's contribution, and faster convergence. Although the model could get faster inference for the absence of this mixing, however, the absence of the mixing results in error accumulation in the streams,"muffled" artifacts in the reconstructed wideband audio signal, and unstable training. Therefore, we apply two successive one-dimensional Lattice blocks (**Lattice1D**), each of which interleaves the magnitude and phase streams via criss-cross connections by learnable scalars, denoted by $\alpha_1, \alpha_2, \beta_1$, and $\beta_2$ in Fig. 1. These scalars perform as a gating function by controlling the strength of cross-stream injection. These scalars are trained end-to-end and dynamically learn "where" and "how-much" cross-stream interactions are required.

Within each Lattice block, we insert a new module named **ConformerNeXt**. We design ConformerNeXt by replacing the standard Conformer's (Gulati et al., 2020) convolutional sub-module with a ConvNeXt (Liu et al., 2022b) block. This replacement significantly enhances the model's capacity by combining ConvNeXt's powerful hierarchical spatial feature extraction with Conformer's global self-attention. Our proposed ConformerNeXt takes

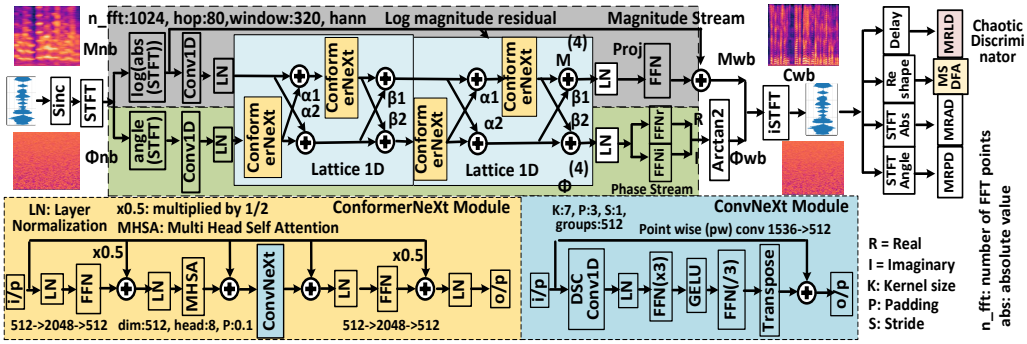


Figure 1: Our proposed CIS-BWE, containing Lattice net, ConformerNeXt, and chaos-informed discriminators.

two inputs instead of one, as opposed to (Luo et al., 2020). Moreover, the pre- and post-processing convolutions used in the original Lattice Nets in (Luo et al., 2020) are omitted. This variation provides the best choice for the inter-stream connectivity.

As there are two Lattice blocks and each contains one ConformerNeXt per stream, the full generator employs a total of four ConformerNeXt with cross-stream residual connections. After trying various combinations of connections and the number of ConformerNeXt, we find that this design provides the best trade-off between performance and a reduced number of parameters. The granular-level implementation details, like the number of filter channels, attention headcounts for ConformerNeXt are shown in Fig. 1 (see Appendix A.11).

**c) Residual Prediction and Synthesis:** After the final Lattice stage, separate output heads estimate the wideband residuals $\mathbf{M}^{(4)}$ and $\mathbf{\Phi}^{(4)}$, which are then passed through Layer Normalization (LN) and Linear Projection (Proj) blocks implemented by the Feed Forward Neural Network (FFN) to estimate the wide band residuals. The magnitude branch outputs a log-magnitude residual that is added to the narrow-band input to isolate and recover only missing high-frequency information, rather than remodeling the entire spectrum.

Moreover, previous work (Yin et al., 2020) shows that due to the noisy nature of the phase, it is very difficult to estimate the phase directly. To overcome these difficulties in direct phase estimation, the phase stream's output is fed into two FFNs for predicting "pseudo-real (R)" and "pseudo-imaginary (I)" residuals, shown in Eqn. 2. Finally, the wide-band phase is recovered by the "arctan2" function by stacking the magnitude and phase branch, and using Inverse STFT, the wide-band audio is reconstructed (see Eqn. 3).

$$R = \mathrm{FFN}_r\big(\mathrm{LN}(\mathbf{\Phi}^{(4)})\big); I = \mathrm{FFN}_i\big(\mathrm{LN}(\mathbf{\Phi}^{(4)})\big) \quad \Phi_{\mathrm{wb}} = \mathrm{atan2}(I, R) \tag{2}$$

$$\mathbf{C}_{\mathrm{wb}} = e^{\mathbf{M}_{\mathrm{wb}}}\big(\cos\Phi_{\mathrm{wb}} + i\sin\Phi_{\mathrm{wb}}\big) \tag{3}$$

## 2.2 Chaos-Informed Nonlinear Discriminator

Speech production is fundamentally a *non-linear dynamical process characterized by deterministic chaos* (Little et al., 2007). Discriminators used in traditional GANs (Tian et al., 2020), (Donahue et al., 2018), (Kumar et al., 2019) typically minimize the distance between reconstructed and original speech based on raw waveforms or spectrogram slices, but fail to detect those nonlinear chaotic cues. Therefore, generators produce over-smoothed and dull spectra (Cao et al., 2024). Here, we design two chaos-inspired nonlinear discriminators - MRLD and MSDFA. This is the first time that these two discriminators have been proposed to be included with complex-valued GANs to explore the chaotic study of audio reconstruction. They analyze long-range and hidden formant trajectories and micro-transients across equally spaced windows, output chaos-aware feature maps, and penalize any mismatch in sub-harmonic richness. Our approach results in a 40x reduction in discriminator size, 0.5x fewer parameters, and more realistic acoustics (see Sections 3.6, 3.4, & 3.8) with less over-smoothed spectra compared to SOTA models.

**a) Multi-Resolution Lyapunov Discriminator (MRLD):** We introduce Lyapunov Exponents (LE) (Oseledec, 1968; Wolf et al., 1985) to capture the rapid, nonlinear fluctuations and sensitivity to initial conditions in speech that spectrogram-based losses overlook. The LE is a measure of nonlinear dynamics used to quantify the rate of separation of infinitesimally close trajectories. Therefore, MRLD penalizes the mismatches in the Lyapunov spectra of real and generated signals and drives the generator to reproduce authentic deterministic chaotic behavior, yielding more lifelike speech.

A pseudo-code 1 is provided to explain how MRLD is implemented. MRLD divides each waveform into five non-overlapping windows $w \in \{64, 128, 256, 512, 1024\}$, computes local LE via delay-embedding and nearest-neighbor divergence,

and maps each segment to a single divergence rate (lines 1-7). Then, MRLD feeds these five exponent maps into five separate Single Resolution Discriminator (SRD). The detailed structure of the SRD is shown in Fig. 2. SRD uses a five-layer depthwise-separable (DSC) 2D Convolution with kernel size 5 (stride 2 for the first four layers, final kernel 3, 235k parameters) to discriminate real versus generated dynamics (refer to Appendix A.10 for details).

**Algorithm 1:** Pseudo-code of MRLD

**Require:** Raw waveform $x$, window sizes $\mathcal{W} = \{64, 128, 256, 512, 1024\}$
**Ensure:** Predicted label $y \in \{\text{real}, \text{generated}\}$
1: $\mathcal{F} \leftarrow []$ {Initialize feature list}
2: **for all** $w \in \mathcal{W}$ **do**
3: **for all** segment $x_i^w$ in $x$ with window size $w$ **do**
4: Delay-embed $x_i^w$ into vectors $\{y_j\}$ using dimension $d$, delay $\tau$
5: Compute $\lambda_i^w = \text{Avg}_j \left[ \frac{1}{\Delta} \log \left( \frac{\|y_{j+\Delta} - y_{j'+\Delta}\| + \epsilon}{\|y_j - y'_j\| + \epsilon} \right) \right]$
6: Append $\lambda_i^w$ to $\mathcal{F}$
7: **end for**
8: **end for**
9: Normalize and reshape $\mathcal{F}$ for SRD input
10: $y \leftarrow \text{SRD}_{\text{MRLD}}(\mathcal{F})$
11: **return** $y$

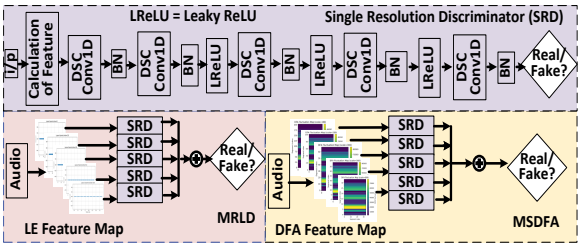


Figure 2: Implementation details of the MRLD, MSDFA, and SRD. We refer to Appendix A.10 for details.

**b) Multi-Scale Detrended Fluctuation Analysis Discriminator (MSDFA):** We introduce Detrended Fluctuation Analysis (DFA) (Peng et al., 1994) to quantify fractal-like, long-range temporal correlations that conventional spectrogram losses overlook. Therefore, by computing how root-mean-square fluctuations $F(n)$ grow with window size, MSDFA supervises the generator to ensure natural-sounding dynamics across syllabic, phonemic, and sub-phonemic scales. If omitted, the adversarial framework leads to muffled prosody even when amplitude and phase discriminators are present.

A pseudo-code 2 is provided to explain how MSDFA is implemented. We integrate classical DFA at five $n \in \{100, 200, 300, 500, 600\}$ samples, tile each $F(n)$ into a fixed $S \times S$ map, and feed the resulting tensor into five separate SRD modules. The SRD (see Fig. 2 and Appendix A.10) has a five-layer DSC 2D CNN (BN + LeakyReLU), whose adversarial loss back-propagates through the tiling, making MSDFA a light (≈247K parameters) yet powerful plug-in discriminator for BWE problems.

**Algorithm 2:** Pseudo-code of MSDFA

**Require:** Input waveform $x(t)$, scales $\mathcal{N} = \{100, 200, 300, 500, 600\}$
**Ensure:** Real/fake score $y \in \mathbb{R}$
1: $\mathcal{F} \leftarrow []$
2: **for all** $n \in \mathcal{N}$ **do**
3: Compute DFA fluctuation $F(n)$ on $x(t)$
4: Tile $F(n)$ into fixed-size map $M^{(n)} \in \mathbb{R}^{S \times S}$ and Append $M^{(n)}$ to $\mathcal{F}$
5: **end for**
6: $T \leftarrow \text{stack}(\mathcal{F}) \in \mathbb{R}^{S \times S \times 5}$
7: $y \leftarrow \text{SRD}_{\text{MSDFA}}(T)$
8: **return** $y$

**c) Multi-Resolution Amplitude & Phase Discriminators (MRAD & MRPD):** In addition to MRLD and MSDFA, we also use MRAD and MRPD in our adversarial framework. MRAD ensures that amplitude transients are captured in different granularities. MRPD stabilizes group delay and explores harmonic-phase relationships. We refer to (Lu et al., 2024b) for the implementation details of MRAD and MRPD. Similar to (Lu et al., 2024b), we use three resolutions, such as frequency bins = [512,128,512], hop sizes = [1024,256,1024], and window lengths = [2048,512, 2048]. Each resolution is fed to a 5-layer 2D CNN (varied kernels/strides, weight-norm) (see Appendix A.10).

### 2.3 Loss Functions

**Generator Losses:** We propose a total of six different generator loss functions shown in Table 1. Magnitude loss encourages accurate spectral amplitude reconstruction. Phase loss ensures faithful temporal alignment and phase continuity. Complex STFT loss jointly enforces faithful amplitude and phase reconstruction. Self consistency loss enforces synthesis consistency. Feature matching loss is critical as it penalizes subtle nuances enforced by non-linear, amplitude, and phase discriminators' feedback. Adversarial loss encourages realism in the waveform generated across multiple perceptual dimensions. The total Generator Loss is shown as:

$$\mathcal{L}_G = \mathcal{L}_{\text{mag}} + \mathcal{L}_{\text{pha}} + \mathcal{L}_{\text{com}} + \mathcal{L}_{\text{stft}} + \mathcal{L}_{\text{fm}} + \mathcal{L}_{\text{adv}}. \quad (4)$$

**Discriminator Losses:** Each discriminator $D_d$ is trained using a hinge loss objective, which specializes discriminators to become powerful critics of unnatural patterns by matching with the perceptual distribution of real speech (see Table 1). The total discriminator loss is shown in Eqn. 5.

$$\mathcal{L}_D = \sum_r \mathcal{L}_D^{\text{MRLD}} + \sum_s \mathcal{L}_D^{\text{MSDFA}} + \sum_r \mathcal{L}_D^{\text{MRAD}} + \sum_r \mathcal{L}_D^{\text{MRPD}} \quad (5)$$

| Loss function | Equation (MSE = Mean Square Error) | Terms |
|---|---|---|
| Magnitude Loss | $\mathcal{L}_{\text{mag}} = \lambda_{\text{mag}} \cdot \text{MSE}(\hat{M}, M)$ | $M$, $\hat{M}$: ground-truth and generated STFT magnitudes. $\lambda_{\text{mag}}$ = 45 |
| Phase Loss | $\mathcal{L}_{\text{pha}} = \lambda_{\text{pha}} \cdot (\mathcal{L}_{\text{IP}} + \mathcal{L}_{\text{GD}} + \mathcal{L}_{\text{IAF}})$ | IP = Instantaneous Phase difference, GD = Group Delay difference, IAF = Instantaneous Amplitude-Frequency difference. $\lambda_{\text{pha}}$ = 100. |
| Complex STFT Loss | $\mathcal{L}_{\text{com}} = \lambda_{\text{com}} \cdot \text{MSE}(\hat{C}, C)$ | $C$ and $\hat{C}$ are the complex-valued STFTs of the target and predicted signals. $\lambda_{\text{com}}$ = 90. |
| Self-Consistency Loss | $\mathcal{L}_{\text{stft}} = \lambda_{\text{stft}} \cdot \text{MSE}(\hat{C}, \tilde{C})$ | $\tilde{C}$ = STFT of the waveform reconstructed from predicted magnitude and phase. $\lambda_{\text{stft}}$ = 90. |
| Feature Matching Loss | $\mathcal{L}_{\text{fm}} = \sum_{d \in \mathcal{D}} \lambda_d \cdot \text{MSE}(f_d^{\text{real}}, f_d^{\text{fake}})$ | Feature maps from discriminator $d$, $d \in \{\text{MRLD}, \text{MSDFA}, \text{MRAD}, \text{MRPD}\}$ |
| Generator Hinge Loss | $\mathcal{L}_{\text{adv}} = \sum_{d \in \mathcal{D}} \lambda_d \cdot \mathbb{E}_{\hat{x} \sim p_G}[max(0, 1 - D_d(\hat{x}))]$ | $\hat{x} \sim P_G$: generated samples; $D_d(\hat{x})$: discriminator $d$ score; $\lambda_d$: weight on $d$'s adversarial term |
| Discriminator Hinge loss | $\mathcal{L}_D^{(d)} = \mathbb{E}_{x \sim p_{\text{data}}}[\max(0, 1 - D_d(x))] + \mathbb{E}_{\hat{x} \sim p_G}[\max(0, 1 + D_d(\hat{x}))]$ | $x \sim P_{\text{data}}$: real samples; $\hat{x} \sim P_G$: generated samples; $\max(0, 1 - D_d(x))$: real-hinge term ($D_d(x) \geq 1$); $\max(0, 1 + D_d(\hat{x}))$: fake-hinge term ($D_d(\hat{x}) \leq -1$) |

Table 1: Generator and discriminator loss functions. Code will be released after the acceptance of the paper.

where $\mathcal{L}_D^{\text{MRLD}}$, $\mathcal{L}_D^{\text{MSDFA}}$, $\mathcal{L}_D^{\text{MRAD}}$, and $\mathcal{L}_D^{\text{MRPD}}$ are MRLD, MSDFA, MRAD, and MRPD losses, respectively, for each resolution/scales.

# 3 Comprehensive Analysis

## 3.1 Evaluation Metrics

We assess intelligibility and perceptual quality of the reconstructed speech using six metrics: Log-Spectral Distance (LSD) (Erell and Weintraub, 1990) to quantify fine-grained spectral deviations; Short-Time Objective Intelligibility (STOI) (Taal et al., 2011) to evaluate speech intelligibility; Perceptual Evaluation of Speech Quality (PESQ) (Rix et al., 2001) to predict overall quality in line with human judgments; Scale-Invariant SDR (SI-SDR) (Le Roux et al., 2019) as a general distortion metric invariant to amplitude scaling; Scale-Invariant SNR (SI-SNR) (Luo and Mesgarani, 2018) to specifically gauge noise-related distortion; and Non-Intrusive Speech Quality Assessment (NISQA-MOS) (Mittag et al., 2021) for reference-free estimation of perceptual speech quality.

## 3.2 Hyperparameter and Configuration

The training of CIS-BWE involves carefully chosen hyperparameters. Learning rate is initialized with $2 \times 10^{-4}$ with exponential decay after each epoch with a decay factor of 0.999. AdamW optimizer with $\beta_1$ = 0.8, and $\beta_2$ = 0.99, and a weight decay of 0.01 are used for stable convergence. We use a batch size of 16 to balance between computational efficiency and memory utilization. All models are trained for a total of 50 epochs and per epoch takes around 25 minutes. **We list all the hyperparameters in Appendix A.12.** We use four NVIDIA RTX 4090 GPUs and Intel(R) Xeon(R) Silver 4310 CPUs (2.10 GHz) for computation.

## 3.3 Dataset and Preprocessing

We use the English VCTK (v0.92) (Yamagishi et al., 2019) (110 speakers) and MLS French corpus (Pratap et al., 2020) (114 speakers), to test in multi-lingual and cross-speaker setup. There are in total ∼1045 hours of speech in the two datasets with a sampling rate of 16 and 48 kHz. We load 16/48 kHz files, convert to mono channel, remove silence parts, downsample to simulate band-limited audio, sinc interpolate, and align length-wise. All test data have unseen speakers that are not included in the train data (i.e., cross speaker). We refer to Appendix A.2, A.7, A.8 for detailed explanation.

## 3.4 Discriminator Ablation Study

Table 2 represents the ablation results reported based on five objective evaluation metrics.

**Row①:** When we only use MRPD (phase) and MSDFA (fractal dynamics), the model receives feedback only on fine-grained periodicity and long-range temporal self-similarity, resulting in the poorest performance (NISQA-MOS = 2.32). However, a higher SI-SNR indicates that the generator can generate phase-consistent results without generating natural-sounding envelopes.

**Row② and ③:** MRLD (deterministic chaos) and MSDFA show an improved performance (NISQA-MOS = 3.58) as MRLD encourages realistic chaos, but without amplitude cues, the performance is lower. Using MRAD and MRPD in row ③, we obtain NISQA-MOS = 4.07 because we have both magnitude and phase cues. MRAD enforces the correct amplitude distributions, and MRPD aligns the notorious phase relationships. These combinations also score significantly better in LSD (1.11) and PESQ (1.61). These results show us the efficacy of the MRAD and MRPD and make them "indispensable" in our design choice.

**Row④ and ⑤:** We use MRLD instead of MRPD along with MRAD in row ④. This performs on par with MRAD and MRPD, as both phase and Lyapunov exponents capture two different types of deterministic chaos. Once again, when we use both MRLD and MRPD along with MRAD, we get an increase in NISQA-MOS to 4.14 from 4.08, which confirms the necessity of MRLD in

capturing deterministic chaos.

| SL | MPD | MRAD | MRPD | MRLD | MSDFA | LSD | STOI | PESQ | SNR | N-MOS |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ✗ | ✗ | ✓ | ✗ | ✓ | 1.22 | 0.89 | 2.05 | 9.47 | 2.32 |
| 2 | ✗ | ✗ | ✗ | ✓ | ✓ | 1.20 | 0.85 | 1.55 | 7.47 | 3.58 |
| 3 | ✗ | ✓ | ✓ | ✗ | ✗ | 1.11 | 0.86 | 1.61 | 7.87 | 4.07 |
| 4 | ✗ | ✓ | ✗ | ✓ | ✗ | 1.09 | 0.86 | 1.65 | 8.42 | 4.08 |
| 5 | ✗ | ✓ | ✓ | ✓ | ✗ | 1.06 | 0.85 | 1.58 | 7.78 | 4.14 |
| 6 | ✗ | ✓ | ✓ | ✓ | ✓ | 1.10 | 0.87 | 1.66 | 8.11 | 4.29 |
| Evaluating MPD with our proposed discriminators | | | | | | | | | | |
| 7 | ✓ | ✗ | ✗ | ✓ | ✗ | 1.23 | 0.85 | 1.53 | 6.95 | 3.58 |
| 8 | ✓ | ✗ | ✗ | ✓ | ✓ | 1.04 | 0.85 | 1.55 | 7.15 | 3.65 |
| 9 | ✓ | ✗ | ✗ | ✗ | ✓ | 1.24 | 0.85 | 1.52 | 7.08 | 3.79 |
| 10 | ✓ | ✓ | ✓ | ✗ | ✗ | 1.11 | 0.85 | 1.56 | 6.68 | 4.18 |
| Comparison of parameters among MPD and our proposed discriminators | | | | | | | | | | |
| 11 | 22M | 600.2k | 600.2k | 235.5k | 247.7k | | | | | |

Table 2: Ablation study on discriminators with their sizes for 2→16 kHz range. Here, N-MOS = NISQA-MOS and SNR = SI-SNR. Our MRLD + MSDFA in row ⑥ has in total 40x smaller parameters (22M vs 483.2k) compared to MPD in row ⑩ with better performance.

**Row⑥:** We give fractal analysis feedback to the generator by MSDFA along with MRLD, MRPD, and MRAD. These combined features provide strong cues to the generator, which results in significant boosts to NISQA-MOS from 4.14 to 4.29. Therefore, this set of combinations is used in our proposed CIS-BWE architecture.

### 3.5 Comparison with MPD

As SOTA models (Lu et al., 2024b,a) use MPD, we compare the performance of our proposed MRLD + MSDFA with MPD in row ⑦ to ⑩ of Table 2.

**Row⑩:** The combination of MPD + MRPD + MRAD gives NISQA-MOS of 4.18, which is lower than our proposed combinations of MRLD + MSDFA + MRAD + MRPD in Row⑥. This statement also holds for other metrics as well. As MRAD + MRPD is common in both cases, this shows that the MRLD + MSDFA performs better than MPD alone. Moreover, we are getting better performance with only 483.2k parameters in total for MRLD + MSDFA, compared to 22M parameters of MPD. This is a significant finding as our MRLD + MSDFA gives better performance compared to MPD with 40x smaller parameters (22M vs 483.2k). This will provide a stepping stone for smaller models in edge devices without sacrificing performance (refer to Appendix A.10 for details on discriminator).

### 3.6 Generator Architecture Ablation Study

To determine the optimal generator configuration, we performed three systematic ablations on core block selection, inter-stream connectivity, network depth, and Multilayer Perceptron (MLP) expansion ratio, summarized in Table 3 for 2→16 kHz range.

**Core Block Selection:** We consider two blocks: ConvNeXt and ConformerNeXt. The reason for choosing ConvNeXt is that we want to demonstrate the better performance of our ConformerNeXt over the SOTA ConvNext (Lu et al., 2024b,a). We separately use a total of 16 ConvNext (denoted by $\text{ConvNeXt}_{16}$ in row ①) and ConformerNeXt (denoted by $\text{ConformerNeXt}_{16}$ in row ②). The row ② achieves the highest NISQA-MOS of 4.44, a +0.13 gain over ConvNeXt in row ①, along with improvements in LSD (1.10 vs 1.12), STOI (0.87 vs 0.86), and PESQ (1.70 vs 1.63). In this way, we find the supremacy of the ConformerNeXt over ConvNeXt as a core block.

**Inter-Stream Connectivity:** We compare linear and Lattice Net for coupling magnitude and phase streams. Lattice Net in row ④ consistently outperforms the linear stream in row ③ in terms of all the six metrics. Therefore, we use Lattice Net as a cross-stream interaction for its superior controlled mixing of amplitude and phase stream via learnable scalars' gating mechanism (see Section 2.1).

**Depth and Head Count:** After getting the best core block and cross-connection scheme, we optimize the number of ConformerNeXt and head count in the generator. Reducing from 16 to 4 ConformerNeXt blocks (row ② vs row ④) yields a compact model with a 7x reduction in size with compromising a small performance but still better/similar to SOTA models (Lu et al., 2024b,a) in Table 4. We further evaluated multi-head self-attention by reducing from 8 to 4 heads (row ⑤), leading to a minor drop in NISQA-MOS (4.25 vs 4.29) compared to row ⑫. We also test the hidden dimension of the linear network with one-fourth (row ⑥), which degrades NISQA-MOS to 3.60.

### 3.7 Making the Discriminator Efficient

Here, we explain how we make our discriminators small yet efficient in terms of all six metrics. The performance of our generator is highly correlated with the number of scales or resolutions used in the discriminators. In rows ① to ④ of Table 3, we use three windows or scales to calculate features from three different resolutions. However, when we increase the number of scales to 5, the feature maps capture more fine-grained as well as coarse-grained temporal patterns to provide better performance, which is shown in rows ⑤ to ⑦ of Table 3. Due to the larger scale of 5, though it might take slightly more train time due to calculate more features, our discriminators can guide the generator for more faithful reconstruction without any increase in the number of parameters.

| SL | Architecture | Size | Discriminators | H | LSD ↓ | STOI ↑ | PESQ ↑ | SI-SDR ↑ | SI-SNR ↑ | NISQA-MOS ↑ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ConvNeXt$_{16}$ | 106.34M | [MRLD + MSDFA + MRA/PD]x3 | - | 1.12 | 0.86 | 1.63 | 7.89 | 7.88 | 4.31 |
| 2 | ConformerNeXt$_{16}$ | 223.2M | [MRLD + MSDFA + MRA/PD]x3 | 8 | 1.10 | 0.87 | 1.70 | 7.93 | 7.90 | 4.44 |
| 3 | ConformerNeXt$_4$, Linear | 64.23M | [MRLD + MSDFA + MRA/PD]x3 | 8 | 1.12 | 0.86 | 1.57 | 7.66 | 7.62 | 3.72 |
| 4 | ConformerNeXt$_4$ | 32.7M | [MRLD + MSDFA + MRA/PD]x3 | 8 | 1.09 | 0.87 | 1.71 | 8.56 | 8.54 | 4.03 |
| 5 | ConformerNeXt$_4$ | 33.5M | [DSC(MRLD+MSDFA) + MRA/PD]×5 | 4 | 1.10 | 0.87 | 1.67 | 8.24 | 8.20 | 4.25 |
| 6 | ConformerNeXt$_4$, MLP /4 | 16.67M | [DSC(MRLD+MSDFA+MRA/PD)]×5 | 4 | 1.12 | 0.87 | 1.68 | 8.01 | 7.98 | 3.60 |
| 7 | ConformerNeXt$_4$ (Proposed) | 33.5M | [DSC(MRLD + MSDFA)]x5 + [MRA/PD]x3 | 8 | 1.10 | 0.87 | 1.66 | 8.14 | 8.11 | 4.29 |

Table 3: Generator architecture ablation study for 2→16 kHz range. SI-SNR and SI-SDR use dB unit. H = number of attention heads in the multi-head self attention of the ConformerNeXt block.

Moreover, we use DSC in our discriminators to shrink their sizes. In rows ① to ④ of Table 3, we use normal convolution, but in rows ⑤ to ⑦, we try different combinations of DSC with our discriminators. For example, in rows ⑤ and ⑦: DSC in MRLD and MSDFA + normal convolution in MRAD and MRPD; in row ⑥: DSC in all MRLD, MSDFA, MRAD and MRPD.

We implement DSC by factorizing standard convolutions into $K \times 1$ depthwise steps (per channel), followed by $1 \times 1$ point-wise convolutions for cross-channel mixing. This reduces the computational complexity from $\mathcal{O}(K \times C_{\text{in}} \times C_{\text{out}})$ to $\mathcal{O}(K \times C_{\text{in}} + C_{\text{in}} \times C_{\text{out}})$. Here, K is kernel size, $C_{\text{in}}$ is input and $C_{\text{out}}$ is the output channel dimension.

**Final Design (row ⑦):** Based on all these ablations, our final generator employs a total of 4 ConformerNeXt blocks (each with 8 heads and linear projection hidden dimension ×4), interconnected via Lattice Net, resolution of 5 together with DSC in MRLD and MSDFA, resolution of 3 together with normal convolution in MRAD and MRPD. This configuration achieves the best trade-off between perceptual quality (NISQA-MOS = 4.29), computational efficiency, and parameter compactness (≈ 33.5 M parameters). We refer to Appendix A.11 and A.10 for final parameter count.

### 3.8 Comparative Analysis with Baselines

Table 4 compares our CIS-BWE against three baselines - EBEN (Hauret et al., 2023), AERO (Mandel et al., 2023), and AP-BWE (Lu et al., 2024a) - over three extension ranges for English-VCTK and MLS French datasets. EBEN is a Pseudo Quadrature Mirror Filter-based model, AERO is a complex-valued model, and AP-BWE is a dual-stream for amplitude and phase prediction model.

Compared to unprocessed speech in VCTK dataset, CIS-BWE significantly does a 3.3x reduction in LSD, a 1.72x increase in STOI, a 2.17x increase in PESQ, and a 1.51x increase in NISQA-MOS for 4→16 kHz. Table 4 also indicates that AP-BWE is the best-performing model in baselines for NISQA-MOS. Our proposed CIS-BWE exceeds AP-BWE in NISQA-MOS, LSD, PESQ, and STOI for all three frequency ranges. However, CIS-BWE gives a similar performance for SI-SDR and SI-SNR compared to AP-BWE. Please note that LSD is a measure for over-smoothing, and NISQA-MOS, PESQ, and STOI are measures of perceptual audio quality. Our model outperforms the best-performing baseline, AP-BWE, for the perceptual and over-smoothing metrics with almost 2.18x fewer parameters (72M vs 33M). This also holds for the MLS French dataset, indicating CIS-BWE outperforms AP-BWE and all other baselines in multilingual settings as well.

To qualitatively illustrate the improved reconstruction performance by our CIS-BWE, we include Fig. 4, which shows that chaotic cues, such as vocal fry, sharp onset, and creaky voice, are reconstructed more precisely by our CIS-BWE (almost similar to the ground truth) compared to the best-performing baseline AP-BWE. It is an indication that our approach of chaotic modeling using chaos-informed discriminators is outperforming other nonlinear discriminators, such as MPD, with fewer parameters (see Section 3.5).

We also refer to Table 12 for more details on performance across different frequency ranges.

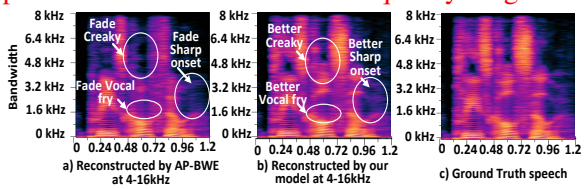


Figure 3: 4-16 kHz extended speech by CIS-BWE.

### 3.9 Study for Noisy Conditions

We use eight different noise sources from the AURORA dataset (Pearce and Hirsch, 2000) and use five different SNRs: -10, -5, 0, 5, and 10 dB for each of the noise sources to separately create the noisy dataset from the clean VCTK and MLS French. The eight noise sources are chosen from stationary and non-stationary types, such as airports, babble, cars, exhibitions, train stations, city streets, speech-shaped noise, and white Gaussian noise. We test our model separately for English and French noisy datasets. The average results for all noise sources and SNRs are shown in Table

| Method | Size | Data | NISQA-MOS | | | STOI | | | PESQ | | | SI-SDR | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| Unprocessed | - | VCTK | 2.79 | 3.67 | 4.43 | 0.55 | 0.61 | 0.61 | 1.15 | 1.51 | 1.41 | -11.03 | -8.07 | -6.07 | -10.53 | -7.62 | -5.63 | 3.27 | 2.27 | 2.85 |
| | | MLS | 2.29 | 3.12 | - | 0.51 | 0.57 | - | 1.05 | 1.34 | - | -14.15 | -18.89 | - | -13.67 | -17.71 | - | 3.48 | 2.48 | - |
| EBEN, 2023 | 29.7M | VCTK | 2.59 | 2.69 | 2.53 | 0.89 | 0.98 | 0.98 | 2.64 | 3.69 | 3.71 | 11.94 | 19.94 | 20.82 | 11.94 | 19.94 | 20.83 | 1.03 | 0.78 | 0.92 |
| | | MLS | 2.14 | 2.28 | - | 0.87 | 0.96 | - | 2.47 | 3.55 | - | 11.81 | 18.24 | - | 11.74 | 18.38 | - | 1.17 | 0.88 | - |
| AERO, 2023 | 36.4M | VCTK | 2.79 | 2.75 | 2.88 | 0.83 | 0.94 | 0.99 | 2.62 | 3.65 | 3.69 | 13.60 | 20.70 | 21.56 | 13.60 | 20.70 | 21.56 | 1.09 | 0.97 | 0.75 |
| | | MLS | 2.38 | 2.37 | - | 0.81 | 0.92 | - | 2.49 | 3.45 | - | 13.47 | 19.41 | - | 13.37 | 19.19 | - | 1.21 | 1.04 | - |
| AP-BWE, 2024 | 72M | VCTK | 3.86 | 3.97 | 4.49 | 0.94 | 0.99 | 0.99 | 2.55 | 3.69 | 3.72 | 13.42 | 18.26 | 20.86 | 13.35 | 18.07 | 20.74 | 0.96 | 0.74 | 0.75 |
| | | MLS | 3.18 | 3.42 | - | 0.92 | 0.99 | - | 2.51 | 3.62 | - | 13.57 | 17.39 | - | 13.17 | 16.33 | - | 1.11 | 0.84 | - |
| CIS-BWE (our) | 33.5M | VCTK | **4.24** | **4.26** | **4.53** | **0.95** | **0.99** | **0.99** | **2.64** | **3.72** | **3.75** | **13.24** | **18.13** | **19.53** | **13.15** | **17.98** | **19.44** | **0.95** | **0.72** | **0.71** |
| | | MLS | **3.47** | **3.91** | - | **0.92** | **0.99** | - | **2.58** | **3.71** | - | **13.29** | **17.68** | - | **12.86** | **16.81** | - | **1.10** | **0.82** | - |

Table 4: Comparison for clean English-VCTK and MLS French dataset (MLS does not have 48 kHz speech).

5, indicating that CIS-BWE outperforms the best-performing baseline AP-BWE.

| Method | Data | NISQA-MOS | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| EBEN | MLS | 0.84 | 0.92 | - | 4.12 | 5.20 | - | 1.43 | 1.13 | - |
| | VCTK | 1.01 | 1.08 | 1.15 | 4.23 | 5.31 | 6.01 | 1.41 | 1.11 | 1.00 |
| AERO | MLS | 1.05 | 1.09 | - | 4.21 | 5.24 | - | 1.45 | 1.16 | - |
| | VCTK | 1.52 | 1.12 | 1.19 | 4.39 | 5.38 | 6.15 | 1.43 | 1.13 | 1.01 |
| AP-BWE | MLS | 2.01 | 2.19 | - | 4.10 | 5.14 | - | 1.42 | 1.17 | - |
| | VCTK | 2.74 | 3.15 | 3.71 | 4.36 | 5.31 | 6.11 | 1.37 | 1.07 | 0.89 |
| CIS-BWE | MLS | **3.14** | **3.24** | **-** | **4.51** | **5.71** | **-** | **1.40** | **1.15** | **-** |
| | VCTK | **3.72** | **3.79** | **3.89** | **4.52** | **5.75** | **6.22** | **1.35** | **1.06** | **0.88** |

Table 5: Analysis for noisy VCTK and MLS dataset.

### 3.10 SNR-Wise Noise Analysis

Table 6 indicates that our proposed CIS-BWE outperforms the best-performing baseline AP-BWE for all noisy SNRs, especially for NISQA-MOS, indicating that CIS-BWE generates better quality speech in noisy scenarios. It is intuitive that the noisiest SNR of -10 dB gives the worst result, and the least noisy SNR of 10 dB gives the best performance. We refer to Appendix A.3 and A.4 for more details on noise type- and SNR-wise analysis.

| SNR | Model | NISQA-MOS | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| -10 | AP-BWE | 2.02 | 2.55 | 2.98 | -0.60 | -0.04 | 1.17 | 1.49 | 1.18 | 0.91 |
| | CIS-BWE | **2.73** | **3.05** | **3.37** | **0.22** | **1.09** | **1.21** | **1.45** | **1.18** | **0.90** |
| -5 | AP-BWE | 2.52 | 2.86 | 3.54 | 2.86 | 3.48 | 4.44 | 1.45 | 1.11 | 0.88 |
| | CIS-BWE | **3.56** | **3.51** | **3.73** | **3.35** | **4.23** | **4.01** | **1.54** | **1.12** | **0.90** |
| 0 | AP-BWE | 2.83 | 3.21 | 3.80 | 5.06 | 5.93 | 6.80 | 1.39 | 1.06 | 0.86 |
| | CIS-BWE | **3.90** | **3.91** | **3.96** | **5.17** | **6.31** | **6.80** | **1.34** | **1.05** | **0.85** |
| 5 | AP-BWE | 3.05 | 3.42 | 4.07 | 6.53 | 7.74 | 8.64 | 1.34 | 1.02 | 0.84 |
| | CIS-BWE | **4.15** | **4.08** | **4.12** | **6.38** | **7.87** | **9.18** | **1.32** | **1.01** | **0.82** |
| 10 | AP-BWE | 3.22 | 3.70 | 4.22 | 7.62 | 9.02 | 9.77 | 1.29 | 0.97 | 0.82 |
| | CIS-BWE | **4.27** | **4.29** | **4.31** | **7.17** | **8.92** | **10.90** | **1.28** | **0.96** | **0.81** |

Table 6: SNR-wise performance for noisy VCTK.

### 3.11 Computational Complexity

Table 7 shows the computational complexity using Multiply Accumulate Operations (MACs), Floating Point Operations per second (FLOPs), Real-Time Factor (RTF), and inference time. Due to optimization of the generator and discriminators (see Sections 3.6, 3.7, A.10, A.11), CIS-BWE uses 0.5x fewer parameters, MACs, and FLOPs compared to AP-BWE. This means CIS-BWE needs ∼0.5x less memory usage than AP-BWE while maintaining superior perceptual quality as shown by NISQA-MOS scores in Table 4. CIS-BWE also has a lower /similar RTF and inference time (GPU), indicating its effectiveness in real-time streaming services (see Appendix A.5 for detailed analysis).

| Model | Freq. | Par.(M) | MAC(M) | FLOP(M) | RTF(GPU) | Inf.(ms) |
|---|---|---|---|---|---|---|
| AP-BWE | 4-16 kHz | 72.07 | 14236.65 | 28473.31 | 0.0023x | 14.93 |
| AP-BWE | 16-48 kHz | 72.07 | 14236.65 | 28473.31 | 0.0025x | 16.60 |
| CIS-BWE | 4-16 kHz | **33.74** | **6790.86** | **13581.73** | **0.0025x** | **13.60** |
| CIS-BWE | 16-48 kHz | **33.74** | **6790.86** | **13581.73** | **0.0028x** | **16.64** |

Table 7: Computational complexity of CIS-BWE. The hardware configuration is provided in Section 3.2.

### 3.12 Subjective Analysis

For a subjective comparison of CIS-BWE against SOTA AP-BWE and unprocessed audio, we select a panel of 10 persons. We use 5-point (1=bad to 5=excellent) Mean Opinion Score (MOS) ratings and Pairwise preference tests. In Fig. 4, we present the MOS results separately for male and female speakers with the overall mean. AP-BWE performs better for only male speakers, while CIS-BWE outperforms for female speakers and overall. In the Pairwise preference test, CIS-BWE outperforms SOTA AP-BWE by a margin of 11% (see Appendix A.13 for detailed explanation). These results provide strong evidence that our CIS-BWE consistently generates higher perceptual quality audio, which is favored by a wide range of listeners.

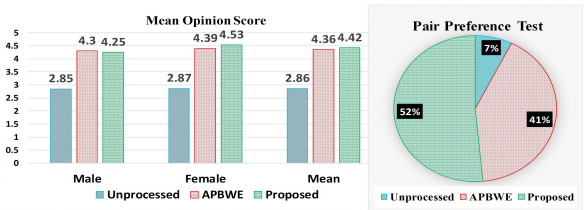


Figure 4: Results of MOS and Pairwise preference test.

## 4 Conclusion

We propose CIS-BWE, an adversarial model for speech BWE. To the best of our knowledge, for the first time, we incorporate chaotic dynamics of speech for improved perceptual quality in a dual-stream GAN-based framework. The efficacy of CIS-BWE is shown across a wide range of performance metrics. We believe that our chaos-informed discriminators will be adopted in the future in a wide range of NLP applications in the domain of generative speech for TTS and ASR tasks.

## 5 Acknowledgment

We are thankful to the VCTK and MLS data collection team for creating and releasing the VCTK and MLS corpus and the AP-BWE authors (Lu et al., 2024a) for providing their GitHub codebase, which we have substantially modified and extended for this work. The codebase is under MIT license and open source. We also express gratitude to the participants of the subjective evaluation tests for their contributions to the listening tests. We acknowledge that we have used Elicit for finding relevant papers, and used ChatGPT for debugging codes, and finding grammatical errors.

## 6 Limitations

Our current study provides better performance than SOTA baselines, demonstrated by leveraging multiple datasets, assessing robustness under noisy environments, and validating performance in multilingual and cross-speaker contexts. However, the training time/epoch for CIS-BWE is slightly higher than the SOTA AP-BWE (17 minutes/epoch vs 25 minutes/epoch) because of the involved large number of chaotic features, controlled cross-stream mixing, and nonlinear calculations.

## 7 Potential Risks / Ethical Considerations

While the intention of designing the CIS-BWE model is for frequency restoration research purposes, it can be misused for potential secret eavesdropping, impersonation, or deepfake audio generation. This model has the potential for severe privacy and security risks. To avoid these, we have to be very careful to ensure transparency, protect consent, and always follow guidelines.

## References

Wallace Abreu and Luiz Wagner Pereira Biscainho. 2024. Aeromamba: An efficient architecture for audio super-resolution using generative adversarial networks and state space models. *arXiv preprint arXiv:2411.07364*.

Bajibabu Bollepalli, Lauri Juvela, and Paavo Alku. 2019. Generative adversarial network-based glottal waveform model for statistical parametric speech synthesis. *arXiv preprint arXiv:1903.05955*.

Yubing Cao, Yongming Li, Liejun Wang, and Yinfeng Yu. 2024. Vnet: A gan-based multi-tier discriminator network for speech synthesis vocoders. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4384–4389. IEEE.

Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*.

A. Erell and M. Weintraub. 1990. Estimation using log-spectral-distance criterion for noise-robust speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 853–856 vol.2.

Berthy Feng, Zeyu Jin, Jiaqi Su, and Adam Finkelstein. 2019. Learning bandwidth expansion using perceptually-motivated loss. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 606–610. IEEE.

W Tecumseh Fitch. 2025. Applying nonlinear dynamics to the voice: a historical perspective. *Philosophical Transactions B*, 380(1923):20240024.

Timo Gerkmann, Martin Krawczyk, and Robert Rehr. 2012. Phase estimation in speech enhancement—unimportant, important, or impossible? In *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, pages 1–5. IEEE.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020*, pages 5036–5040.

Julien Hauret, Thomas Joubaud, Véronique Zimpfer, and Éric Bavu. 2023. Eben: Extreme bandwidth extension network applied to speech signals captured with noise-resilient body-conduction microphones. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

David Haws and Xiaodong Cui. 2019. Cyclegan bandwidth extension acoustic modeling for automatic speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6780–6784. IEEE.

Hanspeter Herzel, David A. Berry, Ingo R. Titze, and M. Saleh. 1994. Analysis of vocal disorders with methods from nonlinear dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 4(3):341–348.

Chun-Wei Ho, Pin-Jui Ku, Hao Yen, Sabato Marco Siniscalchi, Yu Tsao, and Chin-Hui Lee. 2025. An investigation on combining geometry and consistency constraints into phase estimation for speech enhancement. *arXiv preprint arXiv:2507.02192*.

Pengfei Hu, Hui Zhuang, Panneer Selvam Santhalingam, Riccardo Spolaor, Parth Pathak, Guoming Zhang, and Xiuzhen Cheng. 2022. Accear: Accelerometer acoustic eavesdropping with unconstrained vocabulary. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1757–1773. IEEE.

Won Jang, Dan Lim, Jaesam Yoon, Bongwan Kim, and Juntae Kim. 2021. Univnet: A neural vocoder with multi-resolution spectrogram discriminators for high-fidelity waveform generation. *arXiv preprint arXiv:2106.07889*.

Jack J Jiang, Yu Zhang, and Jennifer Stern. 2001. Modeling of chaotic vibrations in symmetric vocal folds. *The Journal of the Acoustical Society of America*, 110(4):2120–2128.

Ji-Hoon Kim, Sang-Hoon Lee, Ji-Hyun Lee, and Seong-Whan Lee. 2021. Fre-gan: Adversarial frequency-consistent audio synthesis. *arXiv preprint arXiv:2106.02297*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Kundan Kumar, Rithesh Kumar, Thibault De Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre De Brebisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. 2019. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630. IEEE.

Kehuang Li and Chin-Hui Lee. 2015. A deep neural network approach to speech bandwidth expansion. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4395–4399. IEEE.

Max Little, Patrick Mcsharry, Stephen Roberts, Declan Costello, and Irene Moroz. 2007. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Nature Precedings*, pages 1–1.

Haohe Liu, Woosung Choi, Xubo Liu, Qiuqiang Kong, Qiao Tian, and DeLiang Wang. 2022a. Neural vocoder is all you need for speech super-resolution. In *Interspeech*.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022b. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986.

Ye-Xin Lu, Yang Ai, Hui-Peng Du, and Zhen-Hua Ling. 2024a. Towards high-quality and efficient speech bandwidth extension with parallel amplitude and phase prediction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Ye-Xin Lu, Yang Ai, and Zhen-Hua Ling. 2025. Explicit estimation of magnitude and phase spectra in parallel for high-quality speech enhancement. *Neural Networks*, page 107562.

Ye-Xin Lu, Yang Ai, Zheng-Yan Sheng, and Zhen-Hua Ling. 2024b. Multi-stage speech bandwidth extension with flexible sampling rate control. *arXiv preprint arXiv:2406.02250*.

Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. 2020. LatticeNet: Towards Lightweight Image Super-Resolution with Lattice Block. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020 (Lecture Notes in Computer Science, vol 12367)*, pages 272–289. Springer, Cham.

Yi Luo and Nima Mesgarani. 2018. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE.

Julia K MacCallum, Li Cai, Liang Zhou, Yu Zhang, and Jack J Jiang. 2009. Acoustic analysis of aperiodic voice: perturbation and nonlinear dynamic properties in esophageal phonation. *Journal of Voice*, 23(3):283–290.

Moshe Mandel, Or Tal, and Yossi Adi. 2023. Aero: Audio super resolution in the spectral domain. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

F Martinez, Antonio Guillamón, Javier Conte Alcaraz, and MC Alcaraz. 2002. Detection of chaotic behaviour in speech signals using the largest lyapunov exponent. In *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No. 02TH8628)*, volume 1, pages 317–320. IEEE.

Nicholas A May and Ronald C Scherer. 2023. The effects of vocal tract constrictions on aerodynamic measures in a synthetic vocal fold model. *The Journal of the Acoustical Society of America*, 154(5):3310–3320.

BANBOOK Michael. 1999. Speech characterization and synthesis by nonlinear methods. *IEEE Trans. Speech, Audio Processing*.

Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. 2021. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *arXiv preprint arXiv:2104.09494*.

V.I. Oseledec. 1968. A multiplicative ergodic theorem: Lyapunov characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc.*, 19:197–231.

David Pearce and Hans-Günter Hirsch. 2000. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, pages 29–32. ISCA Archive.

C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger. 1994. Mosaic organization of dna nucleotides. *Physical Review E*, 49(2):1685–1689.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

Drew Rendall. 2025. Nonlinear phenomena in animal vocalizations: do they reflect alternative functional modes of voice control,'leaked'cues to quality or condition, or both? *Philosophical Transactions B*, 380(1923):20240010.

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. 2001. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE.

Leyuan Sheng and Evgeniy N Pavlovskiy. 2019. Reducing over-smoothness in speech synthesis using generative adversarial networks. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 0972–0974. IEEE.

Yueyuan Sui, Minghui Zhao, Junxi Xia, Xiaofan Jiang, and Stephen Xia. 2024. Tramba: A hybrid transformer and mamba architecture for practical audio and bone conduction speech super resolution and enhancement on mobile and wearable platforms. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4):1–29.

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2011. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on audio, speech, and language processing*, 19(7):2125–2136.

Tarikul Islam Tamiti and Anomadarshi Barua. 2025. A practical approach to power saving in hearables using sub-nyquist sampling with bandwidth extension. *arXiv preprint arXiv:2506.22321*.

Tarikul Islam Tamiti, Biraj Joshi, Rida Hasan, Rashedul Hasan, Taieba Athay, Nursad Mamun, and Anomadarshi Barua. 2025. A high-fidelity speech super resolution network using a complex global attention module with spectro-temporal loss. *arXiv preprint arXiv:2507.00229*.

Chao Tao and Jack J Jiang. 2008. Chaotic component obscured by strong periodicity in voice production system. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 77(6):061922.

Qiao Tian, Yi Chen, Zewang Zhang, Heng Lu, Linghui Chen, Lei Xie, and Shan Liu. 2020. Tfgan: Time and frequency domain based generative adversarial network for high-fidelity speech synthesis. *arXiv preprint arXiv:2011.12206*.

Ingo R Titze. 2008. Nonlinear source–filter coupling in phonation: Theory. *The Journal of the Acoustical Society of America*, 123(5):2733–2749.

Alan Wolf, J. B. Swift, Harry L. Swinney, and John A. Vastano. 1985. Determining lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, 16(3):285–317.

Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, and 1 others. 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, pages 271–350.

Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. 2021. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 492–498. IEEE.

Dacheng Yin, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. 2020. Phasen: A phase-and-harmonics-aware speech enhancement network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9458–9465.

Zhaoyan Zhang. 2023. The influence of source-filter interaction on the voice source in a three-dimensional computational model of voice production. *The Journal of the Acoustical Society of America*, 154(4):2462–2475.

# A Appendix

## A.1 Chaotic Properties of Speech Generation

Speech production is fundamentally a *non-linear dynamical process characterized by deterministic chaos* (Jiang et al., 2001), (Fitch, 2025). Its generation is driven by aerodynamic forces, with visco-elastic vocal cords forming a self-sustained oscillatory system whose glottal pulses produce harmonic frequencies, pressure waves occurring at integer multiples of the fundamental frequency (f0) (Titze, 2008). According to the source-filter theory, these harmonics are then shaped by the vocal tract's filtering action—resonances in the throat, mouth, and nasal cavities dynamically amplify or attenuate certain harmonics, forming moving spectral peaks known as *formants* (Zhang, 2023).

Because the vocal cords receive pressure feedback from the vocal tract, this coupled system can undergo period-doubling, create sub-harmonic frequencies, and intermittently exhibit *chaotic behavior* indicated by positive Lyapunov exponents (Martinez et al., 2002). Consequently, speech naturally alternates between stable, quasi-periodic sounds (typical vowels) and chaotic segments, such

as creaky voice and stressed speech, resulting in subtle jittery fluctuations, turbulence, and irregular timing that purely linear models cannot capture. Generally, vowels exhibit quasi-periodicity interspersed with intermittent chaotic episodes (Tao and Jiang, 2008).

During sound excitation, unstable airflow, vortex shedding, and uneven vocal cord movements introduce additional turbulence and timing irregularities. Within the vocal tract, constricted passages like those forming fricatives produce local turbulence, while the reactive characteristics of supraglottal and subglottal airways feed pressure variations back to the vocal cords, creating a complex non-linear interaction that can either stabilize or destabilize cord oscillations (May and Scherer, 2023). Moderate coupling enriches harmonic content and clarifies formant structures, whereas strong coupling can induce chaotic behaviors, resulting in rough or harsh vocal qualities as observed in creaky voices, infant cries, or animal distress calls (Rendall, 2025).

A purely linear model overlooks critical aspects such as sub-harmonics, bifurcations, and aperiodic bursts, making synthetic speech sound unnaturally smooth (Sheng and Pavlovskiy, 2019). Moreover, diagnostic methods that rely on detecting chaotic indicators for early identification of vocal disorders would lose effectiveness. Contemporary speech synthesis and enhancement systems predominantly use linear models or perturbation parameters, failing to capture these complex, subtle dynamics of speech (MacCallum et al., 2009). Hence, to accurately represent these non-linear behaviors, models must incorporate non-linear glottal-flow representations or leverage adversarial networks with discriminators designed to recognize non-linear characteristics, ensuring both the deterministic aspects (such as harmonic and formant structures) and chaotic elements (including noise bursts and timing irregularities) are faithfully reproduced (Bollepalli et al., 2019).

### A.2 Dataset Pre-Processing Pipeline

We use VCTK dataset, which is an established benchmark extensively use in Speech processing tasks. We extensively check and ensure that VCTK dataset does not contain any Personal Identifying Information (PIN), abusive contents, or any harmful that might be harmful for any individual, group, or others. We at first index all audio files by reading each line from `training.txt` and `test.txt` files, by extracting base filenames (without extensions) and by splitting on the "`|`" character. Out of 110 English native speakers with different accents reading Herald Newspaper articles, we have used 102 speakers for training the CIS-BWe and 8 speakers for testing the efficacy of the CIS-BWE. In total we have used 88,329 audio recordings for training and testing the CIS-BWE. For reducing disk I/O overhead, each audio cache in memory for up to `n_cache_reuse` accesses (default set to 1). Then we trim the silent portion of the audio files by deleting portion of audios taking the start and end of silences from the `vctk-silences.0.92.txt`. After that, audios are loaded by `torchaudio.load`, and stereo (dual) channel signals are converted into mono (single) by averaging across channels. These mono waveforms are then resampled to the high-resolution (HR) target of 16/48 KHz by sinc interpolation if requires. Furthermore, a low-resolution (LR) version is created by first downsampling the audio to 2/4/8/16/24 KHz and then upsampled back to 16/48 KHz by sinc interpolation which is given as input to the CIS-BWE model. For using the audios in training mode (`split=True`), we randomly crop an 8,000-sample segment, which is approximately 167 ms from HR and LR signals both. If the files are shorter than this length, then zero-padding is applied to ensure similar segment lengths.

For MLS French dataset, we use similar preprocessing as we do for VCTK. We use the default validation and test split as given in the corpus. For training set, at first we use the default 9 hours provided by the dataset. However, due to smaller subset of the dataset, CIS-BWE performs poorly due to sub-optimal training. Therefore, we choose the full dataset of $\sim$1000 hours to train the models.

### A.3 Noisy Dataset Preparation

We have collected noises from AURORA dataset. We choose eight different types of noise and 5 different SNR levels to mix with the clean speech to simulate real world environment. We continuously concatenate the same noise file back-to-back in a loop until each file is 5 minutes long without any cross-fades. For each clean speech file, we randomly choose one noise type and one SNR out of 40 different possible combinations. Then we randomly crop a segment exactly the same length as the clean audio. Then we scale the noise to fulfill the chosen SNR and add it with the clean audio. A CSV log file is used to track the noise type, the SNR, the start-end index for the crop for repro-

| SNR | Samples | | | NISQA-MOS | | | STOI | | | PESQ | | | SI-SDR | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| -10 | 558 | 558 | 593 | 2.73 | 3.05 | 3.375 | 0.64 | 0.74 | 0.73 | 1.11 | 1.23 | 1.16 | 0.29 | 1.17 | 0.67 | 0.22 | 1.09 | 0.61 | 1.558 | 1.18 | 0.93 |
| -5 | 585 | 585 | 608 | 3.56 | 3.51 | 3.73 | 0.73 | 0.82 | 0.8 | 1.19 | 1.45 | 1.25 | 3.47 | 4.35 | 4.13 | 3.35 | 4.23 | 4.01 | 1.54 | 1.12 | 0.9 |
| 0 | 606 | 606 | 601 | 3.9 | 3.91 | 3.96 | 0.79 | 0.87 | 0.85 | 1.28 | 1.71 | 1.4 | 5.12 | 6.38 | 6.9 | 5.17 | 6.31 | 6.8 | 1.49 | 1.07 | 0.88 |
| 5 | 580 | 580 | 576 | 4.15 | 4.08 | 4.12 | 0.83 | 0.91 | 0.89 | 1.38 | 2.04 | 1.6 | 6.44 | 7.93 | 9.28 | 6.38 | 7.87 | 9.18 | 1.43 | 1.03 | 0.86 |
| 10 | 608 | 608 | 559 | 4.27 | 4.29 | 4.31 | 0.86 | 0.94 | 0.93 | 1.50 | 2.35 | 1.84 | 7.18 | 8.93 | 10.95 | 7.17 | 8.92 | 10.9 | 1.38 | 0.99 | 0.84 |

Table 8: Comparative SNR-wise analysis over three extension ranges of our proposed CIS-BWE on VCTK noisy dataset.

| Noise Types | Samples | | | NISQA-MOS | | | STOI | | | PESQ | | | SI-SDR | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| AWGN | 348 | 348 | 378 | 3.73 | 3.34 | 3.23 | 0.77 | 0.82 | 0.82 | 1.29 | 1.49 | 1.40 | 5.13 | 6.17 | 7.25 | 5.04 | 6.09 | 7.19 | 1.49 | 1.35 | 0.90 |
| Airport | 372 | 372 | 411 | 3.77 | 3.92 | 4.16 | 0.77 | 0.87 | 0.85 | 1.31 | 1.84 | 1.49 | 4.60 | 5.87 | 6.28 | 4.56 | 5.84 | 6.19 | 1.48 | 1.01 | 0.85 |
| Babble | 384 | 384 | 348 | 3.67 | 3.9 | 4.07 | 0.76 | 0.87 | 0.85 | 1.28 | 1.79 | 1.46 | 4.18 | 5.62 | 6.02 | 4.15 | 5.59 | 5.93 | 1.47 | 1.01 | 0.86 |
| Car | 354 | 354 | 370 | 3.75 | 3.86 | 4.04 | 0.78 | 0.87 | 0.85 | 1.29 | 1.81 | 1.46 | 4.87 | 5.99 | 6.33 | 4.79 | 5.93 | 6.27 | 1.50 | 1.03 | 0.88 |
| Exhibition | 333 | 333 | 366 | 3.94 | 3.97 | 4.26 | 0.80 | 0.88 | 0.85 | 1.33 | 1.93 | 1.47 | 5.39 | 6.61 | 7.19 | 5.34 | 6.57 | 7.09 | 1.46 | 1.01 | 0.87 |
| SSN | 386 | 386 | 359 | 3.49 | 3.59 | 3.26 | 0.77 | 0.84 | 0.82 | 1.32 | 1.68 | 1.37 | 3.46 | 4.66 | 4.78 | 3.37 | 4.55 | 4.63 | 1.46 | 1.15 | 0.99 |
| Station | 380 | 380 | 349 | 3.59 | 3.78 | 4.12 | 0.76 | 0.86 | 0.85 | 1.27 | 1.78 | 1.48 | 4.31 | 5.64 | 6.29 | 4.20 | 5.54 | 6.18 | 1.49 | 1.01 | 0.85 |
| Street | 380 | 380 | 356 | 3.81 | 3.88 | 4.01 | 0.78 | 0.86 | 0.84 | 1.30 | 1.81 | 1.42 | 4.93 | 6.15 | 6.27 | 4.88 | 6.09 | 6.20 | 1.48 | 1.03 | 0.86 |

Table 9: Comparative noise type–wise analysis over three extension ranges of our proposed CIS-BWE on the VCTK noisy dataset.

| SNR | Samples | | | NISQA-MOS | | | STOI | | | PESQ | | | SI-SDR | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| -10 | 558 | 558 | 593 | 2.02 | 2.55 | 2.98 | 0.60 | 0.72 | 0.73 | 1.08 | 1.19 | 1.15 | -0.52 | 0.05 | 1.24 | -0.60 | -0.04 | 1.17 | 1.49 | 1.18 | 0.91 |
| -5 | 585 | 585 | 608 | 2.52 | 2.86 | 3.54 | 0.69 | 0.80 | 0.80 | 1.15 | 1.38 | 1.26 | 2.96 | 3.62 | 4.54 | 2.86 | 3.48 | 4.44 | 1.45 | 1.11 | 0.88 |
| 0 | 606 | 606 | 601 | 2.83 | 3.21 | 3.80 | 0.76 | 0.86 | 0.85 | 1.24 | 1.63 | 1.41 | 5.13 | 6.03 | 6.87 | 5.06 | 5.93 | 6.80 | 1.39 | 1.06 | 0.86 |
| 5 | 580 | 580 | 576 | 3.05 | 3.42 | 4.07 | 0.81 | 0.90 | 0.89 | 1.35 | 1.93 | 1.61 | 6.57 | 7.81 | 8.68 | 6.53 | 7.74 | 8.64 | 1.34 | 1.02 | 0.84 |
| 10 | 608 | 608 | 559 | 3.22 | 3.70 | 4.22 | 0.84 | 0.93 | 0.92 | 1.49 | 2.26 | 1.80 | 7.63 | 9.05 | 9.77 | 7.62 | 9.02 | 9.77 | 1.29 | 0.97 | 0.82 |

Table 10: Comparative SNR-wise analysis over three extension ranges of our proposed AP-BWE on VCTK noisy dataset.

ducibility.

### A.4 SNR- and Noise type-Wise Noise Analysis in Details

**Table 8 (CIS-BWE):** Performance increases significantly as SNR increases from $-10$ to 10 dB, with consistent gains in NISQA-MOS, STOI, PESQ, SI-SDR, and SI-SNR, and a corresponding drop in LSD. The widest band (16–48 kHz) yields the *strongest* SI-SDR/SI-SNR improvements and the *lowest* LSD at high SNRs—evidence which shows that richer high-frequency cues are being faithfully reconstructed in the presence of noises due to the non-linear feedback from chaos discriminators. Overall, CIS-BWE remains robust even at $-10$ dB and performance scales linearly to 10 dB.

**Table 10 (AP-BWE):** AP-BWE follows the same linear SNR trend, but its absolute quality and intelligibility lags CIS-BWE across bands, with the gap most visible under low SNRs and wider extensions.

CIS-BWE also consistently performs robustly in Tables 9 and 11, where results are showed according to the noise types. CIS-BWE performs better in real-world noises such as Exhibition, Airport, and Street noises. AP-BWE also follows a similar pattern; however, CIS-BWE outperforms AP-BWE.

### A.5 Details on Computational Complexity

Table 7 has the detailed analysis on computational complexity.

**Model Size (Parameters):** AP-BWE has 72.07M parameters, more than double that of CIS-BWE (33.74M). This means AP-BWE consumes $\sim$2$\times$ more memory for model weights and intermediate activations. CIS-BWE is more memory-efficient, making it easier to deploy on limited-resource devices.

**Computational Cost (MACs & FLOPs):** AP-BWE requires 14.2B MACs and 28.5B FLOPs per inference, whereas CIS-BWE only requires 6.79B MACs and 13.6B FLOPs. Again, CIS-BWE achieves over 50% reduction in operations, which directly reduces inference cost and memory bandwidth.

**Real-Time Factor (RTF, GPU):** AP-BWE runs at 0.0023–0.0025$\times$ RTF, while CIS-BWE achieves 0.0025–0.0028$\times$ RTF. Lower RTF means faster-than-real-time inference. Although both models are highly efficient on the GPU, CIS-BWE consistently shows a similar RTF compared to AP-BWE.

**Inference time (GPU):** AP-BWE has an inference time of 14.93 ms and 16.6 ms, while CIS-BWE achieves 13.6 ms and 16.64 ms for 4-16 kHz and 16-48 kHz frequency range, respectively. Due to optimization of the generator and discrimina-

| Noise Types | Samples | | | NISQA-MOS | | | STOI | | | PESQ | | | SI-SDR | | | SI-SNR | | | LSD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 | 4–16 | 8–16 | 16–48 |
| AWGN | 348 | 348 | 378 | 2.68 | 2.73 | 3.76 | 0.75 | 0.81 | 0.80 | 1.25 | 1.52 | 1.35 | 4.99 | 6.33 | 6.54 | 4.90 | 6.22 | 6.52 | 1.40 | 2.73 | 0.88 |
| Airport | 372 | 372 | 411 | 2.73 | 3.27 | 3.89 | 0.74 | 0.86 | 0.85 | 1.28 | 1.75 | 1.49 | 4.44 | 5.34 | 6.28 | 4.41 | 5.28 | 6.22 | 1.39 | 3.27 | 0.83 |
| Babble | 384 | 384 | 348 | 2.67 | 3.26 | 3.82 | 0.74 | 0.85 | 0.84 | 1.28 | 1.70 | 1.45 | 4.28 | 5.01 | 5.93 | 4.23 | 4.94 | 5.88 | 1.38 | 3.26 | 0.83 |
| Car | 354 | 354 | 370 | 2.87 | 3.23 | 3.66 | 0.75 | 0.85 | 0.84 | 1.26 | 1.71 | 1.45 | 4.55 | 5.42 | 6.26 | 4.49 | 5.34 | 6.21 | 1.38 | 3.23 | 0.85 |
| Exhibition | 333 | 333 | 366 | 2.91 | 3.24 | 3.94 | 0.77 | 0.86 | 0.85 | 1.30 | 1.79 | 1.47 | 5.41 | 6.15 | 6.91 | 5.37 | 6.07 | 6.87 | 1.34 | 3.24 | 0.84 |
| SSN | 386 | 386 | 359 | 2.60 | 3.14 | 3.20 | 0.73 | 0.83 | 0.82 | 1.23 | 1.64 | 1.39 | 3.15 | 4.31 | 4.81 | 3.05 | 4.18 | 4.65 | 1.47 | 3.14 | 0.97 |
| Station | 380 | 380 | 349 | 2.64 | 3.19 | 3.78 | 0.74 | 0.85 | 0.85 | 1.25 | 1.68 | 1.49 | 4.08 | 5.09 | 6.39 | 4.01 | 4.99 | 6.33 | 1.39 | 3.19 | 0.83 |
| Street | 380 | 380 | 356 | 2.81 | 3.18 | 3.63 | 0.75 | 0.84 | 0.83 | 1.26 | 1.72 | 1.41 | 4.67 | 5.66 | 6.11 | 4.63 | 5.59 | 6.09 | 1.37 | 3.17 | 0.84 |

Table 11: Comparative noise type–wise analysis over three extension ranges of our proposed AP-BWE on the VCTK noisy dataset.

tors (see Sections 3.6, 3.7, A.10, A.11), CIS-BWE achieves performance gain in all eight subjective and objective metrics with a lower/similar inference time.

| Freq. range | LSD | STOI | PESQ | SI-SDR | SI-SNR | NISQA-MOS |
|---|---|---|---|---|---|---|
| 2-16 kHz | 1.1068 | 0.8739 | 1.66 | 8.1487 | 8.118 | 4.2979 |
| 2-48 kHz | 1.21 | 0.8526 | 1.1836 | 6.963 | 6.9662 | 3.9987 |
| 4-48 kHz | 1.099 | 0.933 | 1.4921 | 12.045 | 11.99 | 4.2294 |
| 8-48 kHz | 1.092 | 0.933 | 1.506 | 12.39 | 12.33 | 3.975 |
| 12-48 kHz | 0.873 | 0.9976 | 3.1253 | 17.085 | 16.95 | 4.4854 |
| 24-48 kHz | 0.6531 | 0.9989 | 4.1822 | 23.42 | 23.38 | 4.5254 |

Table 12: Performance over different frequency ranges.

### A.6 Study for Different Frequency Ranges

Table 12 investigates performance across different frequency ranges, indicating a clear pattern: *as the gap between the narrow and target band reduces, performance improves*. Therefore, the broadest reconstruction (2-48 kHz) is the hardest, giving the poorest scores, and the narrowest reconstruction (24-48 kHz) is the easiest, giving the best scores.

### A.7 Dataset Class and DataLoader

Above preprocessing steps are defined in a custom PyTorch `Dataset` class. After initialization, we shuffle audio file lists with a fixed random seed (`random.seed(1234)`) for ensuring reproducibility. The `__getitem__` function handles loading of data (or reuse of cache), apply resampling, apply segmentation, and convert to mono channel, and return a tuple of 1-D tensors, which contains 8,000 samples. Total length of the dataset (`__len__`) is equal to the number of files in each split. During training process, we initialize a `DataLoader` class which uses four worker processes (`num_workers=4`) and uses `DistributedSampler` to ensure distribution of distinct partitions of the dataset.

### A.8 Time–Frequency Feature Extraction and Reconstruction

We design a feature extraction function, `amp_pha_stft`, which calculates the short-time Fourier transform (STFT) on audio segments using [win_size, hop_size, fft] = [320, 80, 1024] and a Hann window parameter. Using generated complex spectrogram $X \in \mathbb{C}^{F\times T}$, we derive two features for giving input to dual stream CIS-BWE. The is log-amplitude calculate using

$$M_nb = \log\big(|X| + 10^{-4}\big)$$

and instantaneous phase

$$\Phi_nb = \arg(X).$$

The output of the CIS-BWE are converted back into audio waveform by `amp_pha_istft`, which exponentiates the predicted log-amplitude, reconstructs the complex spectrogram, and applies an inverse STFT using the same windowing parameters to obtain the final HR audio.

### A.9 Inference Workflow

We load the trained CIS-BWE checkpoint to generate wide-band audio using narrow-band `.wav` inputs. The script initially loads the trained `CIS_BWE Model` checkpoint onto the specified device (GPU or CPU). It then recursively searches the input directory for narrowband `.wav` files. For each discovered file, the script:

1. Applies similar pre-processing and resampling steps for HR and LR creation.
2. Extracts log-amplitude and phase spectrograms as features via `amp_pha_stft`.
3. Feeds the two spectrogram features into each stream of the generator network.
4. Maps the narrowband audios to wideband by generating missing high frequency components
5. Applies `amp_pha_istft` to invert the output representations to waveforms.
6. Saves the output audio as 16 bit PCM `.wav` files at 16/48,KHz.
7. Logs the losses and total processing time for extensive analysis later.

### A.10 Parameter Breakdown for Discriminators

A layerwise parameter breakdown for each discriminator and grand total for all four discriminators in CIS-BWE are shown in Table 13.

### A.11 Parameter Breakdown for Generators

A layer-wise parameter breakdown for the CIS-BWE generator, including LatticeBlock1D parameters alongside pre-processing, ConformerNeXt blocks, and post-processing are shown in Table 14.

### A.12 Hyperparameters and Configuration

**Software Version:** We use an Anaconda virtual environment with Python 3.9.21, PyTorch 2.0.0+cu118, Torchaudio 0.15.0+cu118, Torchvision 0.15.0+cu118, and CUDA Toolkit 11.8.0.

For distributed training and potential scalability, we use the NCCL for multi-GPU training and TCP to initialize communication between processes.

The training and model hyperparameters for the CIS-BWE setup, with use cases and rationale are provided in Table 15.

### A.13 Subjective evaluation details

To evaluate the performance of the proposed CIS-BWE, a formal pair-preference listening test was conducted. A total of ten Bangladeshi under graduate student who self-reported normal-hearing (NH) participants—comprising four males and six females with an average age of 24—participated in the study. The participants voluntarily join to rate the audios without any compensation. At first they are trained on how to assign scores based on perceived perceptual quality of audios. They are also briefed about the purpose of the experiments, potential risks, and about the outcome of this paper. All participants were non-native English speakers and used soundproof headsets to ensure consistent and distraction-free listening conditions. Each participant evaluated 30 sets of speech samples. Every set included three randomly presented versions of the same utterance: (i) the unprocessed (noisy) signal, (ii) the baseline-enhanced signal using APBWE processing, and (iii) the speech processed by the proposed network. For reference, a clean version of the speech signal was also available, though it was not part of the evaluation. Participants rated the perceptual quality of each sample on a 5-point Mean Opinion Score (MOS) scale, where 1 indicates the lowest and 5 the highest quality. Additionally, they were asked to select the most preferred version from the three presented options. The test used 5–7-second speech clips selected from the VCTK dataset, which were processed under three different frequency band conditions: 2–16 kHz, 12–48 kHz, and 24–48 kHz. Individual pair-preference results were analyzed separately for both male and female participants across all band configurations. In the figure 5, we have presented the MATLAB interface that we use to conduct the subjective evaluation.

The findings in figure 4 clearly show that the speech enhanced by the proposed network was consistently and significantly preferred over both the unprocessed and baseline-processed versions. In particular, the proposed CIS-BWE achieved a 54.5% improvement in user preference compared to unprocessed speech and a 2% improvement over the APBWE baseline. These results highlight the network's robust ability to enhance perceptual speech quality under noisy and reverberant conditions.

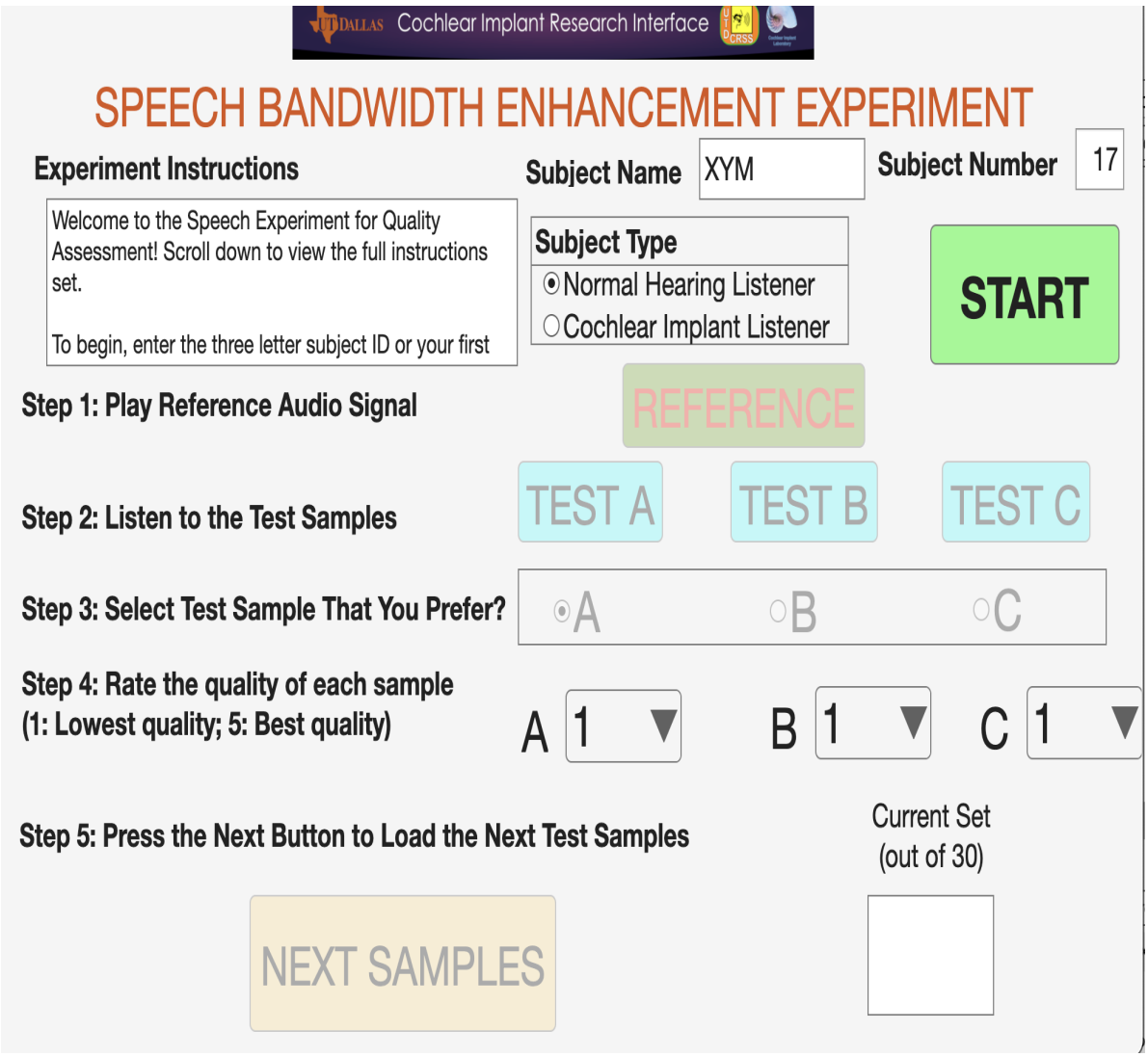


Figure 5: The MATLAB interface used in Subjective Tests.

| Discriminator | Stage | Layer Type | In→Out | Kernel | Stride | Padding | Params |
|---|---|---|---|---|---|---|---|
| MRLD (per scale) | Block 1 | Depthwise Conv1d | 1→1 | 5 | 2 | 2 | 6 |
| | | Pointwise Conv1d | 1→32 | 1 | 1 | 0 | 64 |
| | | BatchNorm1d + LReLU(0.1) | 32→32 | – | – | – | 64 |
| | Block 2 | Depthwise Conv1d | 32→32 | 5 | 2 | 2 | 192 |
| | | Pointwise Conv1d | 32→64 | 1 | 1 | 0 | 2 112 |
| | | BatchNorm1d + LReLU(0.1) | 64→64 | – | – | – | 128 |
| | Block 3 | Depthwise Conv1d | 64→64 | 5 | 2 | 2 | 384 |
| | | Pointwise Conv1d | 64→128 | 1 | 1 | 0 | 8 320 |
| | | BatchNorm1d + LReLU(0.1) | 128→128 | – | – | – | 256 |
| | Block 4 | Depthwise Conv1d | 128→128 | 5 | 2 | 2 | 768 |
| | | Pointwise Conv1d | 128→256 | 1 | 1 | 0 | 33 024 |
| | | BatchNorm1d + LReLU(0.1) | 256→256 | – | – | – | 512 |
| | Final | Depthwise Conv1d | 256→256 | 3 | 1 | 1 | 1 024 |
| | | Pointwise Conv1d | 256→1 | 1 | 1 | 0 | 257 |
| | | BatchNorm1d | 1→1 | – | – | – | 2 |
| | | | | | | **MRLD total (per scale)** | 47 113 |
| MSDFA (per scale) | Block 1 | Depthwise Conv2d | 1→1 | 3×3 | 1 | 1 | 10 |
| | | Pointwise Conv2d | 1→32 | 1×1 | 1 | 0 | 64 |
| | | BatchNorm2d + LReLU(0.2) | 32→32 | – | – | – | 64 |
| | Block 2 | Depthwise Conv2d | 32→32 | 3×3 | 2 | 1 | 320 |
| | | Pointwise Conv2d | 32→64 | 1×1 | 1 | 0 | 2 112 |
| | | BatchNorm2d + LReLU(0.2) | 64→64 | – | – | – | 128 |
| | Block 3 | Depthwise Conv2d | 64→64 | 3×3 | 2 | 1 | 640 |
| | | Pointwise Conv2d | 64→128 | 1×1 | 1 | 0 | 8 320 |
| | | BatchNorm2d + LReLU(0.2) | 128→128 | – | – | – | 256 |
| | Block 4 | Depthwise Conv2d | 128→128 | 3×3 | 2 | 1 | 1 280 |
| | | Pointwise Conv2d | 128→256 | 1×1 | 1 | 0 | 33 024 |
| | | BatchNorm2d + LReLU(0.2) | 256→256 | – | – | – | 512 |
| | Block 5 | Depthwise Conv2d | 256→256 | 3×3 | 1 | 1 | 2 560 |
| | | Pointwise Conv2d | 256→1 | 1×1 | 1 | 0 | 257 |
| | | BatchNorm2d | 1→1 | – | – | – | 2 |
| | | | | | | **MSDFA total (per scale)** | 49 549 |
| MRAD (per res) | Conv 1 | Conv2d, WeightNorm | 1→64 | 7×5 | 2×2 | 3×2 | 2 304 |
| | Conv 2 | Conv2d, WeightNorm | 64→64 | 5×3 | 2×1 | 2×1 | 61 504 |
| | Conv 3 | Conv2d, WeightNorm | 64→64 | 5×3 | 2×2 | 2×1 | 61 504 |
| | Conv 4 | Conv2d WeightNorm | 64→64 | 3×3 | 2×1 | 1×1 | 36 928 |
| | Conv 5 | Conv2d, WeightNorm | 64→64 | 3×3 | 2×2 | 1×1 | 36 928 |
| | Conv_post | Conv2d, WeightNorm | 64→1 | 3×3 | 1×1 | 1×1 | 577 |
| | | | | | | **MRAD total (per res)** | 199 745 |
| MRPD (per res) | Conv 1 | Conv2d WeightNorm | 1→64 | 7×5 | 2×2 | 3×2 | 2 304 |
| | Conv 2 | Conv2d WeightNorm | 64→64 | 5×3 | 2×1 | 2×1 | 61 504 |
| | Conv 3 | Conv2d WeightNorm | 64→64 | 5×3 | 2×2 | 2×1 | 61 504 |
| | Conv 4 | Conv2d WeightNorm | 64→64 | 3×3 | 2×1 | 1×1 | 36 928 |
| | Conv 5 | Conv2d WeightNorm | 64→64 | 3×3 | 2×2 | 1×1 | 36 928 |
| | Conv_post | Conv2d WeightNorm | 64→1 | 3×3 | 1×1 | 1×1 | 577 |
| | | | | | | **MRPD total (per res)** | 199 745 |
| | | | | | | **Grand total (all discriminators)** | 1 681 780 |

Table 13: Layer-wise parameter breakdown, per-discriminator totals, and grand total for all four discriminators in CIS-BWE.

| Stage / Component | Layer Type | In→Out | Kernel | Stride | Padding | Heads | Params |
|---|---|---|---|---|---|---|---|
| | | **Pre-processing** | | | | | |
| Pre-mag convolution | Conv1d | 513→512 | 7×7 | 1×1 | 7×1 | – | 1 839 104 |
| Pre-pha convolution | Conv1d | 513→512 | 7×7 | 1×1 | 7×1 | – | 1 839 104 |
| Pre-mag LayerNorm | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| Pre-pha LayerNorm | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| | | **ConformerNeXtBlock (per block breakdown)** | | | | | |
| | | **FFN & Norm** | | | | | |
| $\text{Norm}_1$ | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| $\text{FFN}_1$ - $\text{Linear}_1$ + GELU + Droput(0.1) | Linear | 512→2 048 | – | – | – | – | 1 050 624 |
| $\text{FFN}_1$ - $\text{Linear}_2$ + Dropout(0.1) | Linear | 2 048→512 | – | – | – | – | 1 049 088 |
| $\text{Norm}_2$ | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| | | **Self-Attention** | | | | | |
| Self-Attention | MultiHeadSelfAttention (embed=512) | 512→512 | – | – | – | 8 | 1 050 624 |
| | | **ConvNeXt Components** | | | | | |
| Depthwise Conv1d | Depthwise Conv1d | 512→512 | 7 | 1 | 3 | – | 4 096 |
| ConvNeXt—Norm | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| ConvNeXt—PWConv1 + GELU | Linear | 512→1 536 | 1 | 1 | 0 | – | 787 968 |
| ConvNeXt—PWConv2 | Linear | 1 536→512 | 1 | 1 | 0 | – | 786 944 |
| ConvNeXt—Gamma | Learned scale | 512→512 | – | – | – | – | 512 |
| | | | | | | **Total per ConformerNeXtBlock** | 6 834 688 |
| | | | | | | **ConformerNeXtBlock total (4 blocks)** | 27 338 752 |
| | | **LatticeBlock1D** | | | | | |
| LatticeBlock1D (per block) | Two-branch fusion using one ConformerNeXt Block + 4 scalars | 512→512 | – | – | – | – | $4 \times 4$ |
| | | | | | | **Total LatticeBlock1D (4 blocks)** | 27 338 768 |
| | | **Post-processing** | | | | | |
| Post-mag LayerNorm | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| Post-mag FFN | Linear | 512→513 | – | – | – | – | 263 169 |
| Post-pha LayerNorm | LayerNorm | 512→512 | – | – | – | – | 1 024 |
| FFN_r post-pha (real) | Linear | 512→513 | – | – | – | – | 263 169 |
| FFN_i post-pha (imag) | Linear | 512→513 | – | – | – | – | 263 169 |
| | | | | | | **Total generator parameters** | 31 808 531 |

Table 14: Layer-wise parameter breakdown for the CIS-BWE generator, including LatticeBlock1D parameters alongside Pre-processing, ConformerNeXt blocks, and Post-processing.

| Hyperparameter | Value | Use Case & Rationale |
|---|---|---|
| Number of GPUs | 1 | Ensure faster training and inference by effectively leveging parallel processing of CUDA cores |
| Max epochs | 50 | Provide enough weight updates for convergence yet avoid overfitting. |
| Batch size | 16 | Balance between gradient stability with computational resource constraints. |
| Initial learning rate | $2 \times 10^{-4}$ | Find balance between convergence with training stability. |
| Adam $\beta_1$ | 0.8 | Optimizer momentum parameter set to adapt quickly to adversarial non-stationarity. |
| Adam $\beta_2$ | 0.99 | Optimizer second moment estimate parameter for stable variance control. |
| Learning–rate decay | 0.999 | Decrease LR to subtly fine-tune weights toward convergence. |
| Random seed | 1234 | Ensure to generate same results across different run |
| ConvNeXt channels | 512 | Provide enough capacity to capture features. |
| ConformerNeXt blocks | 4 | Provide enough parameter without sacrificing performance. |
| Segment size (samples) | 8000 | Capture sufficient audio context for effective BWE. |
| FFT size (n_fft) | 1024 | Balance between frequency resolution against computational load. |
| Hop length | 80 | Overlap chosen to smooth ripple effects without increasing computational load. |
| Window length | 320 | Balance time–frequency resolution in STFT. |
| High-rate sampling rate (Hz) | 16 K / 48 K | Define wide-band frequency ranges. |
| Low-rate sampling rate (Hz) | 2 K / 4 K / 8 K / 16 K / 24 K | Different differncy range to evaluate the robustness of the model. |
| Subsampling ratio | 2 / 4 / 8 / 12 / 24 | Downsampling factors corresponding to low-rate configurations. |
| Number of data-loading workers | 4 | Parallel I/O to maximize throughput |
| Distributed backend | `nccl` | Efficient GPU-to-GPU communication |
| Distributed init URL | `tcp://localhost:54321` | Local rendezvous for single-node distributed setup. |
| Distributed world size | 1 | Single-process distributed for clean scaling. |
| MRLD window sizes | 64, 128, 256, 512, 1024 | Multi-scale Lyapunov analysis to capture deterministic chaotic features at different resolutions. |
| MSDFA scales | 100, 200, 300, 500, 600 | Range of DFA scales for fractal dimension analysis in discriminator. |
| MRAD resolutions (n_fft, hop, win) | (512,128,512), (1024,256,1024), (2048,512,2048) | Multi-resolution STFT settings to capture amplitude dynamics. |
| MRPD resolutions (n_fft, hop, win) | (512,128,512), (1024,256,1024), (2048,512,2048) | Multi-resolution STFT settings to capture phase dynamics. |

Table 15: Training and model hyperparameters for the CIS-BWE setup, with use cases and rationale