Model-based speech enhancement with spectral envelope correction using stacked autoencoders

Wenhao Lu[a], Zhenya Zang[b], Feng Qin[c], Xia Dong[a], Jie Han[d], Zuozhou Pan[e,*] and Yiping Ke[a]

[a]College of Computing and Data Science, Nanyang Technological University, 50 Nanyang Ave, 639798, Singapore
[b]Department of Biomedical Engineering, University of Strathclyde, 106 Rottenrow East, Glasgow, G4 0NW, Glasgow, United Kingdom
[c]School of Physics and Electronics, Shandong Normal University, 88 Wenhua E Rd, Lixia District, Jinan, 250014, Shandong, China
[d]Department of Electrical and Computer Engineering, University of Alberta, 116 St & 85 Ave, Edmonton, T6G 1H9, Alberta, Canada
[e]College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou, 310018, , PR China

ARTICLE INFO

Keywords:
Speech enhancement
Harmonic noise model
Spectral envelope
Clustering
Stacked autoencoder

ABSTRACT

Speech enhancement aims to improve the intelligibility and perceptual quality of noisy speech signals. Many deep learning-based denoising approaches have been developed in the past decade. Their training objectives are to minimize the overall error between predicted and target signals with various mathematical metrics. However, enhancing the perceptual quality is more dependent on preserving the inherent speech characteristics than on overall signal matching. Neglecting this aspect may limit the improvements. Hence, we propose a speech enhancement system that combines the Harmonic Noise Model (HNM) with Stacked Autoencoder (SAE)-based spectral envelope correction. The HNM framework reconstructs the harmonic structure, which is a key spectral feature that contributes to timbre and perceived loudness. Since the parameters used for HNM reconstruction are corrupted by background noise, we design spectral envelope correction modules for restoration. These modules adopt a cluster-specific training strategy. Input data with similar characteristics are first grouped to guide the neural network in learning specific feature representations. Then, within each cluster, the associated SAE builds a robust mapping between clean and noisy parameters by mitigating redundancy and random perturbations in the data. Experimental results verify the effectiveness of our scheme across various noise types and input signal-to-noise levels.

1. Introduction

Speech enhancement is a signal-processing task that improves the intelligibility and perceptual quality of speech signals. It reduces the background noise effect so that distorted speech can be clearer and more understandable. Speech enhancement is extremely important in environments with significant ambient noise, such as crowded places or industrial settings. Its typical applications include telecommunications (Hsu, Lee and Bai, 2022; Tan, Zhang and Wang, 2021), hearing aids (Green, Hilkhuysen, Huckvale, Rosen, Brookes, Moore, Naylor, Lightburn and Xue, 2022; Kirton-Wingate, Ahmed, Gogate, Tsao and Hussain, 2023), and speech recognition (Trinh and Braun, 2022; Chen and Zhang, 2024).

In the past decade, thanks to the remarkable ability of modeling highly nonlinear relationships between noisy and clean speech signals, end-to-end learning-based methods have become the mainstream for speech enhancement. One of the pioneering works (Xu, Du, Dai and Lee, 2014) designed a deep neural network-based regressor which maps noisy log-power spectra to clean ones with a large-scale multi-noise-type training dataset. It applied a dropout technique to lower the risk of overfitting and global-variance equalization as a post-processing step to reduce over-smoothing artifacts observed in the enhancement stage. Samui, Chakrabarti and Ghosh (2019) presented a speech-enhancement framework that stacked Fuzzy Restricted Boltzmann Machines (FRBMs). By modeling the weights and biases of the FRBM as fuzzy numbers, the framework makes the network obtain a more robust and expressive representation capability under noisy training conditions. Cui, Zhang, Chen, Gao, Deng and Feng (2023) introduced an effective training strategy that operates on noisy speech signals in the time domain. An estimator is applied to measure the speech purity of noisy utterances. During the training stage, they combine a supervised learning process

wenhao.lu@ntu.edu.sg (W. Lu); zhenya.zang@strath.ac.uk (Z. Zang); qfxjtu@stu.xjtu.edu.cn (F. Qin); xia.dong@ntu.edu.sg (X. Dong); jhan8@ualberta.ca (J. Han); panzz@cjlu.edu.cn (Z. Pan); ypke@ntu.edu.sg (Y. Ke)
ORCID(s): 0000-0002-4842-2400 (W. Lu); 0000-0003-4952-6727 (Z. Zang); 0000-0002-0920-0274 (F. Qin); 0000-0003-2412-3120 (X. Dong); 0000-0002-8849-4994 (J. Han); 0000-0001-8178-9784 (Z. Pan); 0000-0001-9473-3202 (Y. Ke)

between clean and recovered signals with an unsupervised learning process on noisy signals to strengthen the network's generalization capability. The measured speech purity controls the weight of each learning type. The work of (Chen, Hu, Zou, Sun and Chng, 2023) provided an alternative way to lift the adaptability of enhancement systems for different noise conditions. Given clean speeches, it uses a generative adversarial network to simulate realistic noisy speech. With the artificially generated noisy-clean pairs, a denoising model is trained to minimize a scale-invariant signal-to-distortion ratio loss. Wang and Wang (2023) presented a cross-domain speech enhancement framework that integrates a time-domain supervised enhancement module with a complex-domain diffusion-based generative module. The first module produces a coarsely enhanced signal. This signal, along with a diffused noisy speech, a mask, and a noise level embedding, is subsequently fed to the second module. Based on a diffusion probabilistic model, the second module refines the masked noisy regions in the complex spectrogram to further enhance the speech quality. Welker, Richter and Gerkmann (2022) developed a diffusion-based denoising network by formulating a task-aware stochastic differential equation (SDE) in which the forward process gradually transforms clean speech into noisy speech, and the reverse SDE is learned via score matching to reconstruct clean complex spectrograms. In (Richter, Welker, Lemercier, Lay and Gerkmann, 2023), they further strengthened the network architecture by utilizing a multi-resolution U-Net backbone and incorporating progressive growing in both the contracting and expansive paths. Please note that all the above approaches focus on minimizing the overall difference between the predicted and clean signals with various mapping criteria, such as L1, L2, and cross-entropy losses. However, a good perceptual quality of the restored speech is more dependent on keeping inherent characteristics of original speech signals, rather than the overall signal matching based on mathematical metrics. Neglecting this aspect may limit improvements in perceptual quality.

In fact, human vocal mechanisms are intricately linked to auditory perception. When people speak, their vocal cords vibrate at a specific pitch frequency. This vibration produces a series of harmonics at frequencies that are multiples of the pitch frequency. It is called the "harmonic structure" in the frequency domain. Since this structure contributes directly to the voice timbre and indirectly influences the perceived loudness, some previous studies have attempted to enhance noisy speech by recovering the harmonics. To restore the harmonic magnitude, the work in (Huang, Bao and Wang, 2017) first applied a codebook-driven Bayesian approach to estimate the linear prediction (LP) coefficients and excitation variances of clean speech and noise. Then, with the estimated parameters, a Wiener filter is used to construct a denoised spectrum. The harmonic magnitude was finally sampled from this spectrum at harmonic frequencies. The drawback of this method is that Wiener filtering is based on the assumption of a stationary stochastic process. It is not appropriate for suppressing non-stationary noise in many real scenarios. Huang, Bao, Wang and Xiang (2020) trained a deep feedforward network to map the noise-corrupted logarithmic power harmonic magnitudes to the clean ones. To address the frame-dependent variations in the number of harmonics caused by different pitch periods, they used linear interpolation to fill in missing harmonic magnitudes based on neighboring harmonic values. The linear interpolation approach assumes that the variation of harmonic magnitudes across the frequency spectrum is smooth and predictable. However, it does not hold in the presence of non-stationary noise, resulting in discrepancies between the estimated and target harmonic magnitudes. Unlike the above methods focusing on harmonic magnitude restoration, the work in (Ping and Yafeng, 2022) estimated a harmonic spectrum by weighting the noisy spectrum with a mask. This initial estimation was subsequently refined through frame-to-frame smoothing operations and a harmonic-related gain function. The resultant spectrum facilitates the recovery of a natural harmonic structure in noisy speech. However, under a low signal-to-noise ratio (SNR) environment, the spectral components of the speech signal are highly overlapped with those of the background noise. It makes the mask hard to distinguish between speech signal and noise. As a result, many residual noise exists in the estimated harmonic spectrum. Hence, it is important to find an approach that can precisely restore the harmonic structure from noisy speeches and concurrently suppress background noise components in this structure, without making any explicit assumptions about the noise characteristics.

This paper presents a speech enhancement system incorporating a Harmonic Noise Model (HNM) analysis–synthesis framework (Griffin and Lim, 1988) with a stacked autoencoder-based (SAE) spectral envelope correction. Its aim is to improve the perceptual quality of noisy speeches by recovering the harmonic structure in the frequency spectrum. The HNM framework plays an important role in reconstructing clean harmonics with perceptually relevant acoustic parameters. As these parameters extracted from noise-corrupted speech frames are highly distorted, the spectral envelope correction modules effectively recover them from their degraded counterpart. The main contributions can be summarized as follows:

- We adopt the HNM analysis–synthesis framework to reconstruct clean speech, rather than directly attempting to denoising the given noisy speech. Since this model only depends on estimated acoustic cues to regenerate

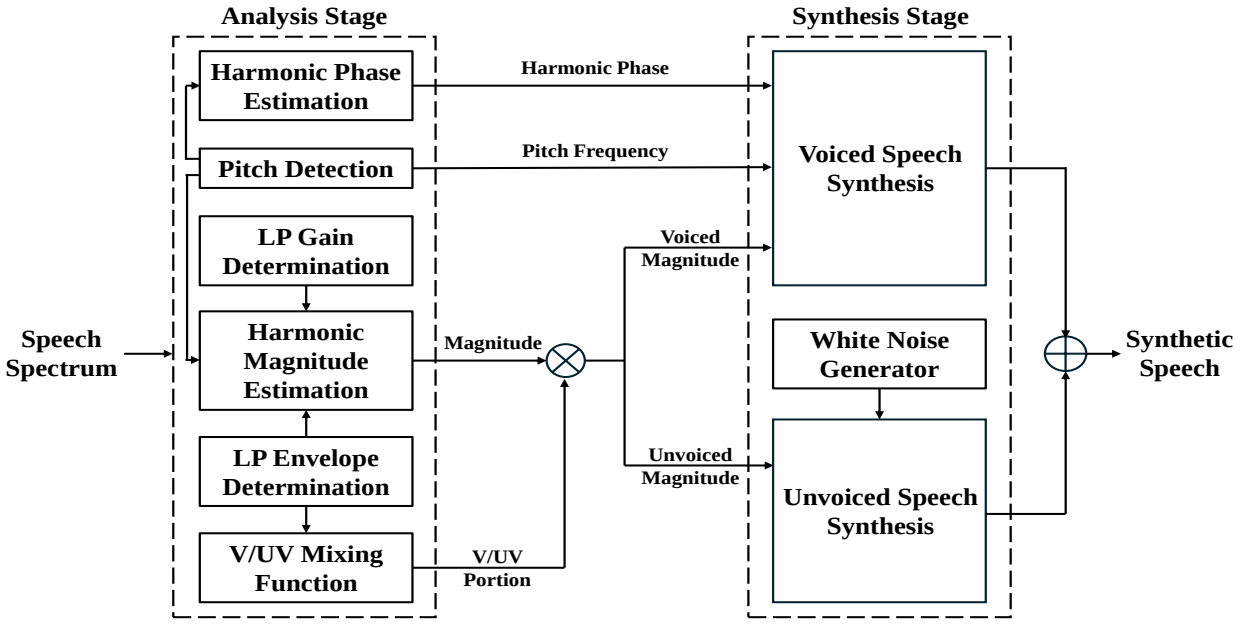

Figure 1: A general HNM analysis-synthesis framework.

the speech signal, any background noise components that overlap with the harmonic structure are inherently excluded during this process.

- We design a clustering coupled with a network-based correction process to recover spectral envelope. Unlike training a single network with input data, the clustering strategy groups inputs with similar characteristics, reducing the need for the network to simultaneously learn multiple input features and making each sub-network focus on the specific characteristics of its assigned cluster.

- Considering that each input pattern comprises slowly varying envelope information from consecutive frames, we choose the SAE to build a highly nonlinear relationship between the noise-corrupted and clean envelopes. Its encoder-decoder structure effectively mitigates redundancy in the input data, allowing the network to learn essential feature representation.

- To enhance the perceptual quality of the restored speech, we train networks to correct the spectral shape (which relates to timbre) and the spectral gain (which relates to loudness), respectively. In particular, given that the spectral gain is sensitive to short-time variations in the speech signal, we design an extra Hamming-weighted loss function to apply a higher weight to the central frame while gradually reducing it towards the edges when training SAEs for the gain correction.

The remainder of this paper is organized as follows. Section 2 introduces the background of HNM. In Section 3, we present the HNM-based speech enhancement with spectral envelope correction procedure. Experiments, as shown in Section 4, are used to evaluate the performance of the proposed enhancing scheme with various noise conditions. In Section 5, we discuss the time complexity of training the proposed system and measure its inference time. Concluding remarks are presented in Section 6.

2. Background

HNM is a signal representation framework designed to decompose speech into voiced (periodic) and unvoiced (aperiodic) components. It is extremely suited to analyze and synthesize speech signals that exhibit a clear periodic structure contaminated by stochastic disturbances. Due to the high quality of speech synthesized by HNM, we adopt the HNM analysis-synthesis as the framework of the proposed speech enhancement system. Fig. 1 illustrates the block diagram of HNM. In the analysis stage, the acoustic cues, i.e., the pitch frequency ω_0, the m-th harmonic magnitude

A_m, and the associated phase information φ_m, are extracted from the given speech. Note that the harmonic magnitude is obtained by sampling the spectral envelope at the pitch frequency and its multiples. Based on LP analysis, the spectral envelope in each speech frame can be described by two parameters: the normalized spectral envelope (LP envelope) and the spectral gain (LP gain). The magnitudes of the voiced and unvoiced components are derived from the estimated magnitude A_m weighed by a V/UV mixing function, respectively. In the synthesis stage, HNM treats the voiced component as the summation of harmonic-related sinusoidal functions in the form of $\sum \hat{A}_m \cos(m\omega_0 t + \varphi_m)$, where $\hat{A}_m$ is the m-th weighted harmonic magnitude. Unvoiced component is produced by artificial Gaussian white noise. The final output is the summation result of voiced and unvoiced components. In this section, we detail the key operations of HNM which greatly affect the perceptual quality of synthesized speech.

2.1. Pitch Detection

Pitch detection aims to estimate the fundamental frequency of human speech, which typically ranges from 50 Hz to 400 Hz. This frequency governs the periodicity of voiced sound and plays an important role in rebuilding voiced speech components. To identify the correct pitch, a set of candidate frequencies $\widetilde{\omega}_0$'s is evaluated within the aforementioned range. For each candidate, we compute a global-normalized matching error between the original speech magnitude spectrum $|S|$ and a pitch-dependent synthetic excitation magnitude spectrum $|E(\widetilde{\omega}_0)|$. The candidate that yields the smallest error is selected as the optimal pitch frequency. To quantify the matching error, the speech spectrum is divided into sub-bands corresponding to harmonic multiples of $\widetilde{\omega}_0$. Suppose there are totally $M(\widetilde{\omega}_0)$ sub-bands. From (Griffin and Lim, 1988), the matching error function is defined as

$$\mathrm{ERR}_1 = \frac{\sum_{m=1}^{M(\widetilde{\omega}_0)} \sum_{k=a_m}^{b_m} \left(|S(k)| - D_m\,|E(\widetilde{\omega}_0, k)|\right)^2}{(1 - \widetilde{P}_0 F_p) \sum_{m=1}^{M(\widetilde{\omega}_0)} \sum_{k=a_m}^{b_m} |S(k)|^2}, \tag{1}$$

where D_m is the estimated magnitude of the m-th sub-band ($m \in [1, M(\widetilde{\omega}_0)]$), given by

$$D_m = \frac{\sum_{k=a_m}^{b_m} |S(k)|\,|E(\widetilde{\omega}_0, k)|}{\sum_{k=a_m}^{b_m} |E(\widetilde{\omega}_0, k)|}. \tag{2}$$

In (1), $\widetilde{P}_0$ denotes the pitch period which is inversely related to the candidate pitch frequency $\widetilde{\omega}_0$, F_p represents the unit energy of the applied window in the speech frame, $S(k)$ denotes the k-th frequency bin of the original speech spectrum, a_m and b_m specify the lower and upper frequency boundaries of the m-th sub-band, and $E(\widetilde{\omega}_0, k)$ is the k-th frequency bin of the excitation signal spectrum with the pitch frequency $\widetilde{\omega}_0$. The functionality of $\frac{1}{1-\widetilde{P}_0 F_p}$ in (1) is as follows. When $\widetilde{P}_0$ is large (equivalently, $\widetilde{\omega}_0$ is small), the excitation spectrum $|E(\widetilde{\omega}_0)|$ becomes denser and more D_m are used to approximate the speech spectrum $|S|$. Consequently, the difference between $|S(k)|$ and $D_m\,|E(\widetilde{\omega}_0, k)|$ tends to be small. This bias makes the pitch detection prefer a large $\widetilde{P}_0$ (a small $\widetilde{\omega}_0$) candidate in nature, even if it is not the true one. To compensate for this bias, "$\frac{1}{1-\widetilde{P}_0 F_p}$" is incorporated into the matching error function. For example, given a large $\widetilde{P}_0$, $\frac{1}{1-\widetilde{P}_0 F_p}$ also becomes large. After being multiplied by the remaining terms in (1), it amplifies the overall matching error, thereby preventing the systematic favoring of a long-pitch-period candidate (a low-pitch-frequency candidate). Yu and Chan (1999) also reported another practical issue of (1). ERR_1 does not sufficiently penalize the mismatch occurring in low-energy harmonic bands, particularly when these bands lie between two adjacent high-energy regions. In such cases, gross pitch errors, e.g., pitch doubling or halving, potentially occur. To address this limitation, a similar but revised matching function is introduced (Yu and Chan, 1999), given by

$$\mathrm{ERR}_2 = \frac{1}{M(\widetilde{\omega}_0)(1 - \widetilde{P}_0 F_p)} \sum_{m=1}^{M(\widetilde{\omega}_0)} \left[\frac{\sum_{k=a_m}^{b_m} \left(|S(k)| - D_m |E(\widetilde{\omega}_0, k)|\right)^2}{\sum_{k=a_m}^{b_m} |S(k)|^2} \right]. \tag{3}$$

Unlike ERR_1 performing a global normalization over all sub-bands, ERR_2 adopts a per-band normalization strategy. It ensures that even low-energy bands contribute meaningfully to the pitch detection process. The final matching error in this study is the combination of ERR_1 and ERR_2 (Yu and Chan, 1999), i.e.,

$$\mathrm{ERR} = \mathrm{ERR}_1 + \mathrm{ERR}_2. \tag{4}$$

2.2. LP Envelope Estimation

In HNM-based speech reconstruction, once the optimal pitch frequency ω_0 is detected, the associated harmonic magnitudes A_m's are obtained by sampling the spectral envelope at integer multiples of ω_0. This process requires a proper representation of the spectral envelope. Here, we adopt an autoregressive (AR) model to represent it. It is because the envelope derived from the AR model is usually smooth, eliminating unnecessary spectral details that may make the synthesized speech sound unnatural or mechanical. Linear Predictive Coding (LPC) (Rao and Pearlman, 1996) is a widely used method for building this AR model. The frequency response of an LPC filter is given by

$$H\big(e^{j\varpi}\big) = \frac{G}{1 - \sum_{p=1}^{P} \gamma_p e^{-pj\varpi}}, \tag{5}$$

where ϖ is the normalized frequency variable between 0 and π, G is the spectral gain, P is the model order, and γ_p is the p-the LP coefficient ($p \in [1, P]$). For γ_p's, they are estimated through the Levinson–Durbin recursive algorithm (Durbin, 1960). In the rest of this paper, we refer to the normalized spectrum envelope "$\frac{1}{1-\sum_{p=1}^{P} \gamma_p e^{-pj\varpi}}$" as the LP envelope, and the spectral gain G as the LP gain. Note that the LP envelope indicates the spectral shape. Any change in the LP envelope can alter the perceived timbre, making a sound appear brighter, darker, richer, or more nasal.

In this study, the LPC coefficients are further transformed into line spectrum frequencies (LSFs). The reasons are twofold. First, LSFs have a straightforward acoustic interpretation. They are the frequencies that reflect the formant locations. Formants, as the perceptual acoustic cues, distinguish different vowel sounds based on their specific resonant frequencies. In contrast, LP coefficients lack any intuitive acoustic meaning. Second, as shown in Section 3.3, our proposed enhancement system includes a clustering operation. Since the LSF is bounded between 0 and π, it aids in reliable distance computations during the clustering process and improves algorithmic stability. By comparison, LPC coefficients are unbounded and therefore less suitable for such an operation.[1]

2.3. LP Gain Estimation

As stated in (5), the LPC transfer function decomposes the original spectral envelope into an LP envelope and an LP gain. For the LP gain G, it reflects the overall energy of a speech frame and therefore relates to the perceived loudness of the speech signal. This perceptual relevance makes the LP gain estimation important in speech reconstruction tasks. Let us denote the magnitude of the m-th sub-band for the speech spectrum and the LP envelope spectrum as $|S(m)|$ and $|\mathscr{S}(m)|$, respectively. The optimal LP gain is derived from minimizing the squared error between the target magnitude spectrum and the LP-envelope–scaled approximation, i.e.,

$$\sum_{m=1}^{M(\widetilde{\omega}_0)} \big(|S_m| - G|\mathscr{S}(m)|\big)^2. \tag{6}$$

By differentiating (6) with respect to G and setting the derivative to zero, the closed-form solution for G is given by

$$G = \frac{\sum_{m=1}^{M(\omega_0)} |S(m)|\,|\mathscr{S}(m)|}{\sum_{m=1}^{M(\omega_0)} |\mathscr{S}(m)|^2}. \tag{7}$$

2.4. V/UV Mixing

When people speak, the air expelled from the lungs traverses the glottal constriction, resulting in the vibration of the vocal cord. Concurrently, turbulent airflow is also produced due to the dynamic compression within the vocal tract (Flanagan and Cherry, 1969). This physical mechanism tells us that the voiced (from the vibration of the vocal cord) and unvoiced (from the turbulent airflow) components are mixed in the speech spectrum. In other words, each harmonic sub-band contains both voiced and unvoiced energies. Hence, we need to find out the mixing portions of these two components in the harmonic magnitude A_m. Considering that the unvoiced components have more mixing

[1] Some previous works also chose Mel-Frequency Cepstral Coefficients (MFCCs) as the speech feature. For example, the work of (Erro, Sainz, Navas and Hernáez, 2011a) converted the short-time spectrum into MFCCs through cepstral analysis with a Mel-scaled transformation. In the HNM synthesis stage, the harmonic magnitude is calculated from a function of pitch frequency and cepstral coefficients. Similar to the LP coefficients, MFCC is not explicitly bounded. It is not a good choice for clustering.

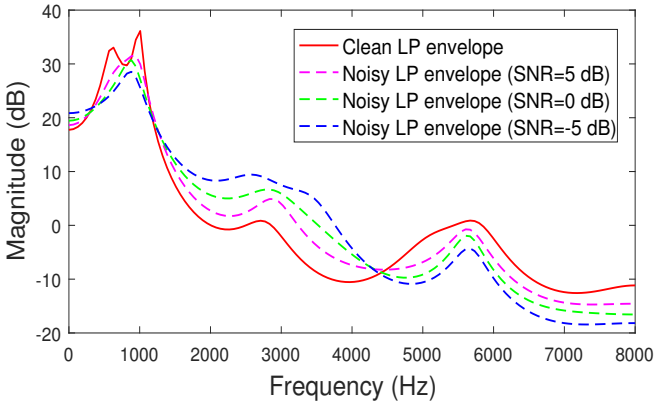

Figure 2: The LP envelope of a speech frame under clean and various pink noise conditions.

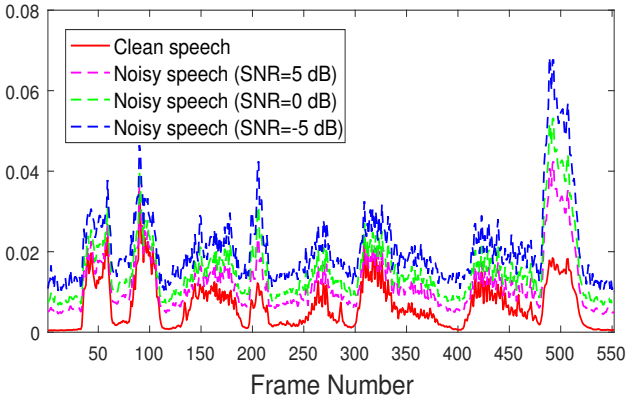

Figure 3: The LP gain contour of an utterance under clean and various pink noise conditions.

portion than the voiced ones when the LP envelope is flat, Yu and Chan (Yu and Chan, 1999) proposed a V/UV mixing function based on the spectral flatness measurement, given by

$$\text{mix}(\vartheta) = \begin{cases} 1 - \frac{\xi(\vartheta)}{2T_h}, & \xi(\vartheta) < T_h \\ \frac{T_h}{2\xi(\vartheta)}, & \xi(\vartheta) > T_h \end{cases}, \tag{8}$$

$$\xi(\vartheta) = \frac{1}{\pi - \vartheta} \int_{\vartheta}^{\pi} (\log |\mathscr{S}(\varpi)| - \varsigma(\vartheta))^2 \, d\varpi, \tag{9}$$

$$\varsigma(\vartheta) = \frac{1}{\pi - \vartheta} \int_{\vartheta}^{\pi} \log |\mathscr{S}(\varpi)| \, d\varpi, \tag{10}$$

where $|\mathscr{S}(\varpi)|$ is the magnitude of the LP envelope at the frequency ϖ, and T_h is a predefined threshold of spectral flatness degree. The magnitude of the voiced component is the product of A_m and $(1 - \text{mix}(\vartheta))$, while the magnitude of the unvoiced component is the product of A_m and $\text{mix}(\vartheta)$.

3. Proposed Speech Enhancement System

Section 2 tells us that as many of the acoustic cues in HNM are closely related to human auditory perception, the reconstructed utterance from a clean speech signal has a high perceptual quality. Suppose there is time-varying additive noise in the speech signal. A main concern is whether the utterance reconstructed from a noisy signal still remains a

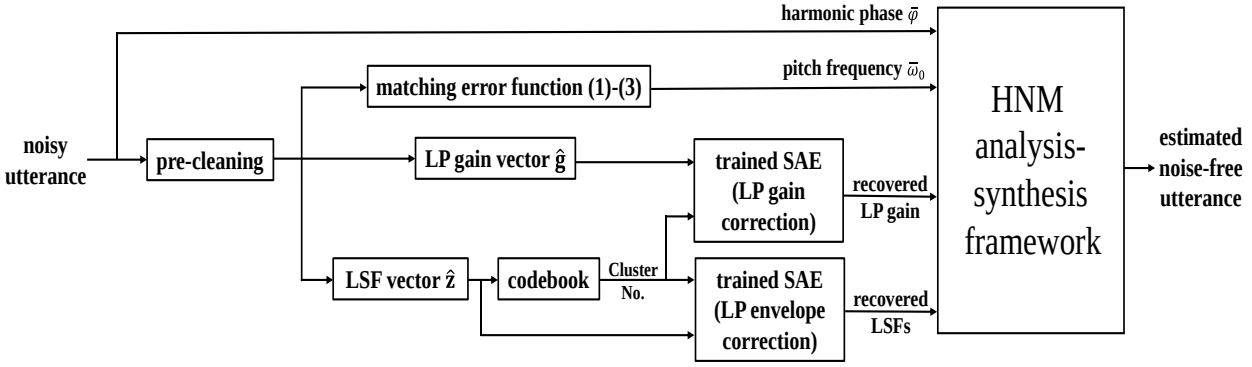

Figure 4: The architecture of the proposed speech enhancement system.

good quality. Fig. 2 depicts the changes in an LP envelope under various pink noise levels. It is evident that the spectral shape undergoes great distortion. For example, two original formants (between 500 and 1500 Hz) in the clean LP envelope degrade into a single one under noise conditions. The magnitude difference between the clean envelope (the red curve) and the noise-corrupted envelope (the blue curve) reaches around 10 dB at 2500 Hz. Fig. 3 shows the gain contour evolution of an utterance under various pink noise levels. Similarly, it can be observed that with the noise level increasing, the disparity in the gains of the noisy speech and its clean version progressively enlarges. Recall that the magnitudes of harmonic structures are sampled from the spectral envelope, and the spectral envelope is represented by LP envelope and gain as stated in (5). HNM-based reconstruction with these distorted acoustic cues might have an increased "harshness" or "sharpness" in the speech, leading to discomfort during listening. Therefore, it is necessary to perform corrections for LP envelope and gain.

In this section, we propose a speech enhancement system incorporating an HNM analysis–synthesis framework with an SAE-based spectral envelope correction. The general architecture is shown in Fig. 4. Given a noisy utterance, it is initially pre-cleaned by a log-spectral amplitude (LSA) estimator (Ephraim and Malah, 1985). For the pre-cleaned utterance, we first extract the associated LSF vectors $\hat{\mathbf{z}}$ (the parameters for LP envelope) and the LP gain vector $\hat{\mathbf{g}}$. Then, by comparing with each centroid in a codebook (obtained by offline clustering operation), $\hat{\mathbf{z}}$ is categorized into the cluster whose centroid is nearest to $\hat{\mathbf{z}}$. The resultant cluster index is concurrently assigned to the associated LP gain vector $\hat{\mathbf{g}}$. Both $\hat{\mathbf{z}}$ and $\hat{\mathbf{g}}$ are corrected by the trained SAEs (Bengio, Lamblin, Popovici and Larochelle, 2006; Hinton and Salakhutdinov, 2006) belonging to the selected clusters, respectively. The pitch detection is also conducted on the pre-cleaned signal with the matching error function stated in (1) to (4). Besides, the harmonic phase is obtained from the noisy utterance.[2] After that, the predicted pitch frequency $\bar{\omega}_0$, the corrected LSFs and LP gains, and the harmonic phase $\bar{\varphi}$ are fed to the HNM model. With the HNM analysis-synthesis framework in Fig. 1, we finally reconstruct the clean speech signal from the noisy one. In the remainder of this section, we will provide an elaborate description of the pre-cleaning, clustering, and LP parameter correction modules.

3.1. Pre-cleaning Operation

As shown in Fig. 4, when a noisy utterance is fed to the proposed enhancement system, it is first pre-cleaned by the LSA estimator. In this study, we consider that all the noisy speeches have low signal-to-noise ratio (SNR) levels, i.e., the input SNR is between -3 dB and 5 dB. From the basic signal processing theory, SNR is a measure that compares the power of a desired signal to the power of background noise. Speech signals with "$-3\text{dB} \leq \text{SNR} \leq 5\text{dB}$" typically imply

[2]Here, we do not apply phase recovery. It is based on a trade-off between perceptual improvement and inference time cost. From the perspective of perceptual improvement, while a recent phase-aware model (Chao, Yu, Fu, Lu and Tsao, 2022) (a fully phase-based enhancement method, denoted as PSE in Table 3) has demonstrated perceptual performance benefits, the improvement is limited. For example, as shown in Table 3, considering all the noise types and input SNR levels, the average PESQ improvement by PSE is only 0.15. In contrast, the average PESQ improvement by our system (fully magnitude-based enhancement method, denoted as HNM_SE) is 0.61. That means the phase information has "less importance" in speech enhancement (Wang and Lim, 1982). From the perspective of latency, as evaluated in Section 5.2, the proposed system already has a longer inference time than other magnitude-based enhancement methods due to the pre-cleaning operation and pitch detection in the HNM analysis-synthesis framework. Introducing an extra phase correction module would further increase the inference time, making the proposed system less practical. Hence, we exclude additional phase recovery in the proposed system.

Table 1
Average SNR results (dB) of denoised speech based on the LSA method.

Noise Type	Input SNR			
	-3dB	0dB	3dB	5dB
White	6.25	7.96	9.73	10.93
Pink	6.36	8.09	9.91	11.18
F16	6.03	7.82	9.68	10.96
Babble	2.97	5.30	7.60	9.13

that the noise component's power exceeds or is close to that of the signal itself. As a result, the useful speech signal information is masked by the external noise, making it difficult for further neural network training to extract effective features. The LSA estimator provides a coarse noise suppression result, which helps the subsequent LP parameter correction in Sections 3.4 and 3.5. To illustrate it, we randomly select a hundred clean utterances from the TIMIT database (Garofolo, Lamel, Fisher, Fiscus and Pallett, 1988). The associated sampling rate is 16 kHz. Each utterance is mixed with white noise, pink noise, F16 noise, and babble noise, respectively. The noise source is from the NOISEX-92 database (Varga and Steeneken, 1993). For each noise type, the input SNR is defined as -3 dB, 0 dB, 3 dB, and 5 dB, respectively. Given a noisy utterance, we measure SNR for its pre-cleaned version, i.e.,

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\sum_{t=1}^{T} x^2(t)}{\sum_{t=1}^{T}\left(x(t)-\mathscr{x}(t)\right)^2}\right), \tag{11}$$

where T is the number of samples in the signal, and $x(t)$ and $\mathscr{x}(t)$ are the noise-free and pre-cleaned speech signal in the time domain, respectively. "$x(t)-\mathscr{x}(t)$" is the residual noise after performing the pre-cleaning operation. Repeating the measurement in (11) for all the given utterances, we get the average SNR results. Table 1 shows the average SNR results of noisy utterances after they are processed by the LSA method. It can be seen that even for non-stationary noise (e.g., F16 and babble noise) with a low input SNR level (e.g., -3 dB), the pre-cleaned utterance still achieves an output SNR above 0 dB. Since an SNR above 0 dB means the signal components dominate over noise, the LSA method ensures that some essential speech features are preserved in the resultant utterances. Hence, using these pre-cleaned signals rather than noisy signals as input sources to train neural networks for LP parameter correction reduces the risk of misinterpreting noise artifacts as meaningful features and increases overall learning efficiency. In addition, the pre-cleaning operation makes the pitch detection more accurate. Fig. 5 depicts the magnitude spectrum of a speech signal under clean, noise, and pre-cleaned conditions. Clearly, the original pitch harmonics experience substantial distortion under the presence of F16 noise at the input SNR level of 0 dB, e.g., many of the harmonics in the low and middle frequency band are masked by noise components. Following the application of LSA, the harmonic structure is roughly retained. As mentioned before, the pitch detection depends on the spectrum matching error. Pre-cleaning operations strengthen the efficiency of pitch frequency extraction.

3.2. Data Preparation

With the pre-cleaned utterances, we prepare the input vectors for the subsequent clustering and SAE training operations. The target vectors for supervised training are derived from the associated clean utterances. Suppose the LP envelope in the r-th pre-cleaned speech frame needs to be recovered. As speech signal changes smoothly over a short time, inter-frame speech characteristics are combined as one block to provide context for accurate and natural feature representation in training networks. That is, we extract the pre-cleaned LP envelopes from the $(r-q)$-th frame to the $(r+q)$-th frame, parameterize them by LSFs, and bundle these LSFs as an input vector, i.e.,

$$\hat{\mathbf{z}} = \left[\hat{z}_1^{(r-q)} \cdots \hat{z}_P^{(r-q)}, \cdots, \hat{z}_1^{(r)} \cdots \hat{z}_P^{(r)}, \cdots, \hat{z}_1^{(r+q)} \cdots \hat{z}_P^{(r+q)}\right]^T, \tag{12}$$

where $z_P^{(r)}$ is the P-th LSF in the r-th pre-cleaned speech frame. Please be reminded that the LSFs are derived from the transformation of LP coefficients. The number of LSFs in each frame is the same as the order of the LPC filter, i.e., "P" in (5).

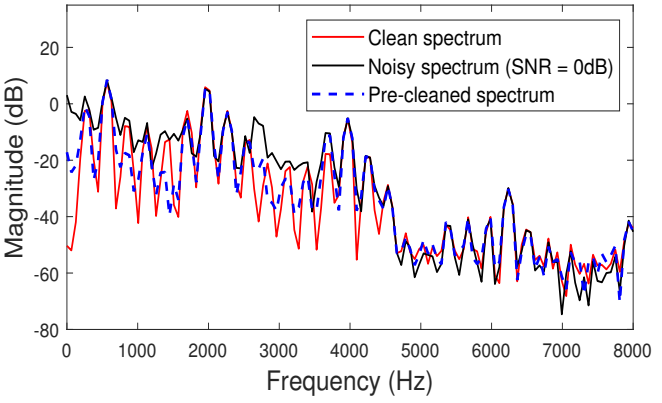

Figure 5: The magnitude spectra of clean, noisy (F16 noise, SNR=0 dB) and precleaned voiced speech signal.

3.3. Clustering (Codebook Generation)

The prepared input vectors are categorized into κ clusters. Note that various speech signals (e.g., different phonetic phonemes) have distinct spectral characteristics. By grouping vectors with similar LP envelope features together, clustering enables the subsequent network training to learn specific spectral shape features within the corresponding cluster.[3] A simple example illustrates its importance. Suppose there are two input vectors with the LSFs of vowel /u/ and vowel /i/, respectively. Generally, the first formant frequency (F1) of the vowel /u/ ranges from 300 to 500 Hz, and the second formant frequency (F2) lies between 800 and 1200 Hz. For the vowel /i/, its F1 is between 200 and 400 Hz, while its F2 is very high, reaching 2000 to 2500 Hz.[4] As the LSFs are closely related to the formant position, putting the input vectors of vowels /u/ and /i/ into one training set could severely weaken the network's learning capability. Thus, we cluster the prepared input vectors. Following the Linde-Buzo-Gray (LBG) algorithm (Linde, Buzo and Gray, 1980), the clustering operation randomly initializes a codebook which contains κ code vectors. Let us denote the code vector $\mathbf{c}_\ell$ as the centroid of the ℓ-th cluster ($\ell = 1, .., \kappa$). The notation of (12) is simplified as $\hat{\mathbf{z}} = \left[\hat{\boldsymbol{\varkappa}}_{r-q}^T, ..., \hat{\boldsymbol{\varkappa}}_r^T, ..., \hat{\boldsymbol{\varkappa}}_{r+q}^T\right]^T$, where $\hat{\boldsymbol{\varkappa}}_r = \left[\hat{z}_1^{(r)} ... \hat{z}_P^{(r)}\right]^T$. Also, let $\mathbf{c}_\ell = \left[\boldsymbol{\mathcal{C}}_{\ell,r-q}^T, ..., \boldsymbol{\mathcal{C}}_{\ell,r}^T, ..., \boldsymbol{\mathcal{C}}_{\ell,r+q}^T\right]^T$, where $\boldsymbol{\mathcal{C}}_{\ell,r}$ has the same dimension as $\hat{\boldsymbol{\varkappa}}_r$. Then, we compute the mean square error (MSE) distance between $\hat{\mathbf{z}}$ and $\mathbf{c}_\ell$, given by,

$$d(\hat{\mathbf{z}}, \mathbf{c}_\ell) = \frac{1}{2q+1} \sum_{\varrho=1}^{2q+1} \left(\hat{\boldsymbol{\varkappa}}_\varrho - \boldsymbol{\mathcal{C}}_{\ell,\varrho}\right)^T \left(\hat{\boldsymbol{\varkappa}}_\varrho - \boldsymbol{\mathcal{C}}_{\ell,\varrho}\right). \tag{13}$$

Based on the nearest distance criterion, $\hat{\mathbf{z}}$ is assigned to the cluster mapping with the target code vector. Once all the input vectors are allocated into clusters, the code vector $\mathbf{c}_\ell$ is updated to be the centroid of those $\hat{\mathbf{z}}$'s in the ℓ-th cluster. We repeat the above input vector assignment and the codebook update iteratively until the changes in the codebook are smaller than a predefined threshold or until the maximum number of iterations is reached. In Section 4.1, the selection of a proper κ value is discussed.

3.4. LP Envelope Correction

For each cluster, we train a neural network to correct the LP envelope. Two kinds of data are present. One comprises the input vectors with the LSFs of sequentially pre-cleaned frames. The collection of the N input vectors is denoted as $\hat{\mathbf{Z}} = \left[\hat{\mathbf{z}}_1, ..., \hat{\mathbf{z}}_N\right]$. Another comprises the target vectors with the corresponding clean LSFs. The collection of the corresponding target vectors is denoted as $\boldsymbol{\mathcal{Z}} = \left[\boldsymbol{\varkappa}_1, ..., \boldsymbol{\varkappa}_N\right]$, where $\boldsymbol{\varkappa}$ is a target vector expressed as $\boldsymbol{\varkappa} = \left[\varkappa_1^{(r-q)} ... \varkappa_P^{(r-q)}, ..., \varkappa_1^{(r)} ... \varkappa_P^{(r)}, ..., \varkappa_1^{(r+q)} ... \varkappa_P^{(r+q)}\right]^T$. Given the paired vectors $\hat{\mathbf{z}}$ and $\boldsymbol{\varkappa}$, an SAE is trained to correct the degraded LP envelopes within the associated cluster. It is a kind of neural network designed for learning hierarchical

[3] It is challenging for training a single network to learn all the features from the input dataset, especially when the dataset is large.
[4] For the spectral shape of vowel /u/ and vowel /i/, please refer to (Ferreira, 2007, Fig. 1).

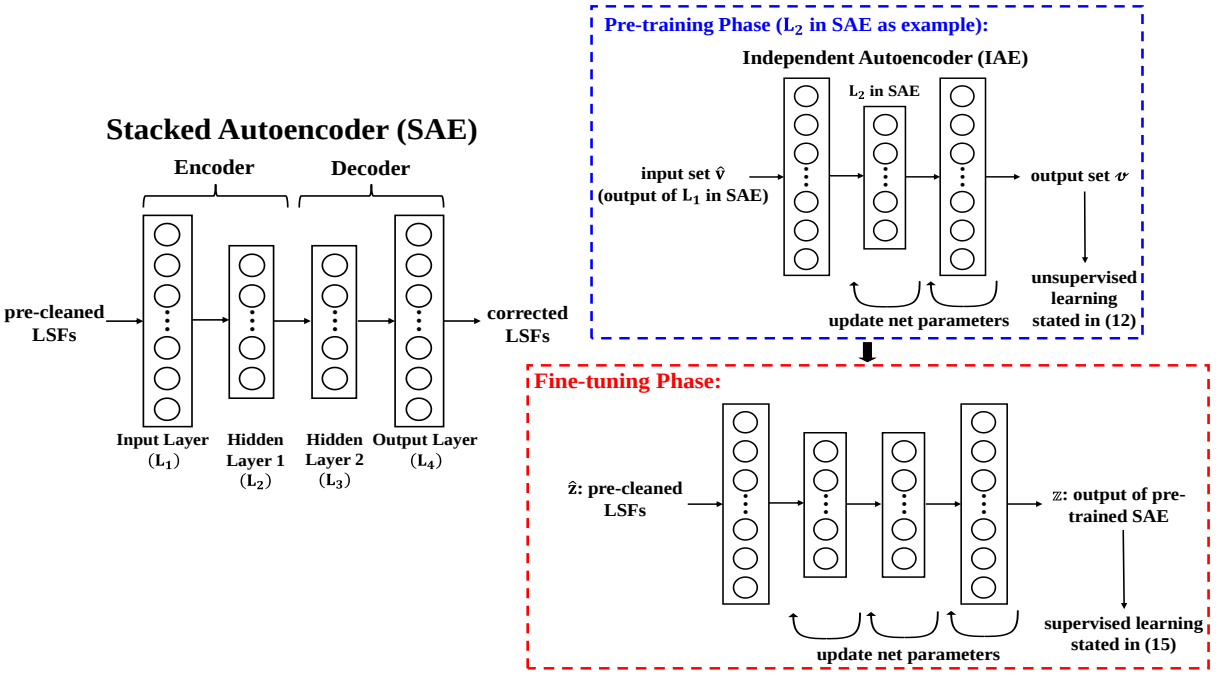

Figure 6: The structure of the applied SAE and its training procedure.

data representations. Fig. 6 illustrates the encoder-decoder architecture of the applied SAE. It consists of four layers. The first two layers perform as an encoder. When a vector is fed to the input layer, it undergoes a transformation which involves a linear combination of the input elements followed by a nonlinear activation function. The subsequent layer compresses the previous layer's output into a more condensed form, i.e., a lower-dimensional representation. This representation captures the essential features of the input vector. As the noise components are usually irrelevant to the speech signal, they are eliminated in the encoding process to some extent. The last two layers perform as a decoder to gradually reconstruct the original input. It mirrors the structure of the encoder but in reverse. When the encoder output is fed to the decoder as the input, each layer of the decoder attempts to reverse the transformation applied in its corresponding encoder layer and progressively expand the reduced representation back to the original input dimension. The decoder output generates the recovered LSFs. The activation function in the input layer and two hidden layers is the sigmoid function. The output layer adopts a linear activation function. Note that the SAE model is apt for processing the input data in our study. It is because the LSFs of continuous neighboring frames exhibit only small variations over short temporal spans, leading to a certain level of informational redundancy in each input vector. The SAE's encoder-decoder architecture is able to compress the input into a compact representation so that the redundancy is decreased.

Training a SAE consists of two stages, namely pre-training and fine-tuning. As shown in Fig. 6, during the pre-training stage, each layer (except the input layer) in the SAE is successively treated as the hidden layer of a three-layer independent autoencoder (IAE). Let us label the $\imath$-th layer in the SAE as $L_\imath$ ($\imath = 2, ..., n_\imath$), and the $\jmath$-th layer in the IAE as $L_\jmath$ ($\jmath = 1, ..., n_\jmath$). Suppose we take the layer $L_\imath$ as the hidden layer in IAE. Then, the input for the IAE is the output of the layer $L_{\imath-1}$ in the SAE. With an unsupervised strategy, the IAE is trained to develop a function capable of reconstructing the input as close as possible. Let the collection of input vectors for the IAE be $\hat{\mathbf{V}} = \left[\hat{\mathbf{v}}_1, \cdots, \hat{\mathbf{v}}_N\right]$, and the associated collection of net outputs (i.e., reconstructed inputs) be $\mathscr{V} = \left[\boldsymbol{v}_1, ..., \boldsymbol{v}_N\right]$, where both $\hat{\mathbf{v}}$ and $\boldsymbol{v}$ have the same dimensions. Also, let the output of the hidden layer in IAE be $\mathbf{h} = \left[h_1, ..., h_Y\right]$. The loss function of an IAE is defined as

$$\mathscr{L}_{\text{pre}} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{\mathscr{K}}\left(\hat{v}_i^{(j)} - v_i^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{\jmath=1}^{n_\jmath-1}\sum_{\mathscr{a}=1}^{\jmath_{\mathscr{a}}}\sum_{\mathscr{b}=1}^{\jmath_{\mathscr{b}}}\left(w_{\mathscr{b}\mathscr{a}}^{(\jmath)}\right)^2 + \eta\sum_{y=1}^{Y}\text{KL}(\rho||\rho_y), \tag{14}$$

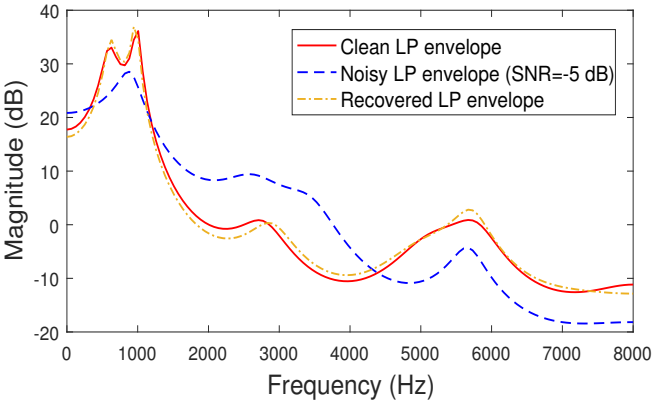

Figure 7: The LP envelope of a speech frame derived from clean, noisy (pink noise, SNR=-5dB) and SAE-corrected LSFs.

where $\mathscr{K}$ is the number of output neurons in the IAE, $\hat{v}_i^{(j)}$ and $v_i^{(j)}$ are the j-th element in the input vector $\hat{\mathbf{v}}$ and the output vector $\boldsymbol{v}$, respectively. The second term in (14) is a weight decay term to reduce the magnitude of the net weight, where λ is the weight decay parameter and $w_{\ell a}^{(\jmath)}$ is the connection weight between the a-th neuron ($a = 1, \cdots, \jmath_a$) in the layer $L_\jmath$ and the ℓ-th neuron ($\ell = 1, \cdots, \jmath_\ell$) in layer $L_{\jmath+1}$. The third term is a Kullback-Leibler (KL) divergence-based regularizer, given by

$$\eta \sum_{y=1}^{Y} \mathrm{KL}(\rho||\rho_y) = \eta \sum_{y=1}^{Y} \left(\rho \log \frac{\rho}{\rho_y} + (1-\rho) \log \frac{1-\rho}{1-\rho_y} \right), \tag{15}$$

$$\rho_y = \frac{1}{Y} \sum_{y=1}^{Y} h_y, \tag{16}$$

where η is the weight of the regularizer and ρ is a sparsity parameter. Suppose a hidden neuron is "active" if its output h_y is close to 1 and "inactive" if h_y is close to 0. ρ_y in (16) is the activation probability across all the Y neurons in the hidden layer, and ρ is the target sparsity level of the active hidden neurons. The KL regularizer aims at encouraging ρ_y to approach ρ. Thus, ρ must lie between 0 and 1. However, setting a large value of ρ (e.g., ρ close to 1) would make many hidden neurons frequently active, reducing the benefit of sparsity and preventing the autoencoder from learning compressed and essential representations. Conversely, setting an excessively small ρ value (e.g., ρ close to 0) imposes overly strict sparsity, which restricts the representational capacity of the autoencoder and leads to underfitting. Based on our training and test experiments with all cluster-specific stacked autoencoders used in this study, $\rho \in [0.01, 0.3]$ yielded satisfactory LP envelope correction performance. Note that given a value of ρ, the $\mathrm{KL}(\rho||\rho_y)$ in (15) is a piecewise monotonic function with respective to ρ_y. When $\rho_y \in (0, \rho]$, it is strictly monotonically decreasing, and when $\rho_y \in [\rho, 1)$, it is strictly monotonically increasing. For both cases, the codomain is $[0, +\infty)$. This property leads to a practical issue of using $\mathrm{KL}(\rho||\rho_y)$ as a penalty term. Specifically, $\mathrm{KL}(\rho||\rho_y)$ tends to infinity as ρ_y approaches 0 or 1, potentially destabilizing the training process. To address this issue, we introduce clipping constraints during the training procedure. If $\rho_y < 0.001$, we clip it to 0.001. If $\rho_y > 0.999$, we clip it to 0.999. With both the limitations on the ranges of ρ and ρ_y, $\mathrm{KL}(\rho||\rho_y)$ is prevented from approaching infinity. Consequently, the training stability is guaranteed. The net parameters are updated through the backpropagation algorithm. Once the training is completed, the connection weights between the input and hidden layers in the IAE are used as the initial weights between the layer $L_{\imath-1}$ and the layer $L_\imath$ in the SAE. The above procedures are subsequently repeated for each SAE layer until all the weights are initialized. The inference of a pre-trained SAE can be simply described as follows. An input vector $\hat{\mathbf{z}}$ is first compressed layer by layer to preserve its principal components. Then, the preserved components are applied for hierarchical input reconstruction.

After the SAE is pre-trained, we fine-tune it with noise-free bundled LSFs. The aim of fine-tuning is to refine the weights of the pre-trained SAE. During this stage, supervised learning is applied to equalize the output of the pre-trained SAE and the target vector based on the MSE metric. The input vector for the pre-trained SAE is the bundled pre-cleaned

LSFs $\hat{\mathbf{z}}$. The target vector is the corresponding noise-free LSFs $\boldsymbol{\varkappa}$. Given an input dataset $\hat{\mathbf{Z}}$, let the collection of the associated net outputs be $\mathbb{Z} = \left[\mathbb{z}_1, ..., \mathbb{z}_N\right]$. The loss function of fine-tuning is given by

$$\mathscr{L}_{\text{tune}} = \frac{1}{2N} \sum_{i=1}^{N} \sum_{\mathcal{J}=1}^{(2q+1)\times P} \left(\varkappa_i^{(\mathcal{J})} - \mathbb{z}_i^{(\mathcal{J})} \right)^2, \tag{17}$$

where "$(2q+1)\times P$" is the number of output neurons in the SAE, and $\varkappa_i^{(\mathcal{J})}$ and $\mathbb{z}_i^{(\mathcal{J})}$ are the $\mathcal{J}$-th element in the target vector $\boldsymbol{\varkappa}$ and the output vector $\mathbb{z}$, respectively. Again, the update of weights and biases is based on the backpropagation algorithm.

Fig. 7 shows an example of the LP envelope obtained from clean, noisy (pink noise, -5 dB), and our scheme's recovered LSFs. It can be seen that with the proposed LP envelope correction, the single formant (between 500 and 1500 Hz) in the noisy envelope is recovered to two formants again. Besides, the magnitude difference between the clean (the red curve) and recovered (the yellow curve) envelopes is very small. That means our scheme is useful for noise suppression.

3.5. LP Gain Correction

Similar to the LP envelope correction, building a cluster-specific LP gain correction also contains three steps. That is, given a pre-cleaned utterance, the LP gains extracted from continuous $(2q+1)$ speech frames are bundled as an input vector. This vector is then grouped into one of the clusters. Finally, a SAE corresponding to that cluster corrects the distorted LP gains. Instead of directly clustering the LP gains with the LBG algorithm, we use the LSF clustering results to guide the categorization of LP gains, i.e., the LP gain vector shares the same cluster index with the associated LSF vector. It is because compared with LP gain, LSFs are more robust speech features for clustering. LSFs (related to the speech resonant characteristics) have small changes over short time durations. On the contrary, LP gain (related to the energy within a speech frame) is sensitive to short-time variations in the speech signal, e.g., any change in the speaker's vocal intensity may lead to great fluctuation in the gain value. This indirect clustering method guarantees the consistency of feature grouping.

After the clustering operation, an SAE is trained to correct the degraded LP gains in the associated cluster. Each cluster has two kinds of data. One is the collection of the N input vectors, denoted as $\hat{\mathbf{G}} = \left[\hat{\mathbf{g}}_1, \cdots, \hat{\mathbf{g}}_N\right]$, where $\hat{\mathbf{g}}$ is a pre-cleaned LP gain vector given by $\hat{\mathbf{g}} = \left[\hat{g}^{(r-q)}, \cdots, \hat{g}^{(r+q)}\right]^T$. Another is the collection of target vectors, denoted as $\boldsymbol{\mathscr{G}} = \left[\boldsymbol{\mathscr{g}}_1, \cdots, \boldsymbol{\mathscr{g}}_N\right]$, where $\boldsymbol{\mathscr{g}}$ is a clean LP gain vector given by $\boldsymbol{\mathscr{g}} = \left[\mathscr{g}^{(r-q)}, \cdots, \mathscr{g}^{(r+q)}\right]^T$. The general structure of the applied SAE and the training procedures are the same as the description in Section 3.4, except for the fine-tuning phase. As discussed previously, the LP gain has more fluctuations than LSFs in a short time. Besides, in practice, only the corrected LP gain of the central frame is needed for further spectral envelope recovery. With these two considerations, we add Hamming weighting factors into the conventional MSE-based loss function in order to apply a higher weight to the central frame while gradually reducing it towards the edges. Let us denote the collection of the associated net outputs as $\mathbb{G} = \left[\mathbb{g}_1, \cdots, \mathbb{g}_N\right]$ for the given input dataset $\hat{\mathbf{G}}$. The Hamming-weighted loss function of fine-tuning is given by

$$\bar{\mathscr{L}}_{\text{tune}} = \frac{1}{2N} \sum_{i=1}^{N} \sum_{\zeta=1}^{2q+1} \text{ham}(\zeta) \left(\mathscr{g}_i^{(\zeta)} - \mathbb{g}_i^{(\zeta)} \right)^2, \tag{18}$$

where "2q+1" is the number of output neurons in the SAE, and $\mathscr{g}_i^{(\zeta)}$ and $\mathbb{g}_i^{(\zeta)}$ are the ζ-th element in the target vector $\boldsymbol{\mathscr{g}}_i$ and the output vector $\mathbb{g}_i$, respectively. In (18), ham(ζ) is the Hamming factor to weight the difference between the ζ-th target and output elements, given by

$$\text{ham}(\zeta) = 0.54 - 0.46\cos\left(\frac{2\pi\zeta}{2q}\right). \tag{19}$$

Fig. 8 shows the overall LP gain contours of an utterance under noise-free, pink noise, and recovery conditions. We can observe that external noise causes the gain contour of a clean utterance to drift greatly. After we apply the proposed gain correction, the recovered contour is much closer to the noise-free one. That means our scheme effectively reduces the gain fluctuation caused by noise.

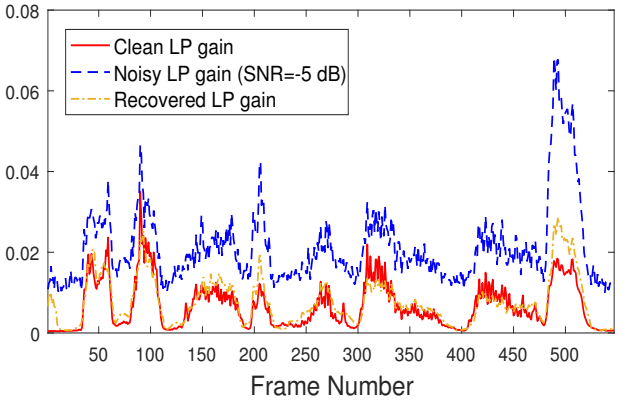

Figure 8: The clean, noisy (pink noise, SNR=-5dB) and SAE-corrected LP gain contours of an utterance.

4. Experimental Results

We have designed an HNM-based speech enhancement system with spectral envelope recovery via SAEs. This section would like to demonstrate its usefulness. SAEs are configured to handle the LP parameter correction tasks. A total of two hundred clean utterances from the TIMIT Corpus are used for the training set, while thirty other utterances are selected for each test. The associated noisy utterances, as the inputs for the proposed enhancing system, are generated by adding the clean versions with white noise, pink noise, F16 noise, and babble noise, respectively. These noises are from the NOISEX-92 database. They are used in both the training and test stages. Besides, three unseen noise types, the vacuum cleaner noise and the rain noise from the ESC-50 dataset (Piczak, 2015) and the home kitchen noise from the QUT-NOISE dataset (Dean, Sridharan, Vogt and Mason, 2010), are also chosen for the test to evaluate our scheme's generalization ability and robustness. The input SNR is set as -3 dB, 0 dB, 3 dB, and 5 dB, respectively. Such settings help us identify whether our scheme is robust at low input SNR levels. The noisy speeches are pre-cleaned by the LSA method. Both the pre-cleaned and noise-free utterances constitute a parallel training/test set. The sampling rate for speech signal is 16 kHz. A 256-sample Hamming window is applied to divide utterances into 16 ms speech frames with a 4 ms frameshift. In the LP envelope correction, every input vector contains the envelopes within 21 ($2q+1=21$) consecutive pre-cleaned speech frames. Each LP envelope is represented by 12-order ($P=12$) LSFs. For an SAE, the dimensions of each layer are 252 (12×21), 170, 170, and 252 (the same as the input layer due to the symmetric structure), respectively. Its training procedure has been shown in Section 3.4. Note that during training, all input vectors are grouped into several clusters. For each cluster, a separate SAE is independently trained. Due to space limitations, presenting the complete hyperparameter configurations for all these SAEs is impractical. Instead, we provide the common initial hyperparameter settings for the associated training processes. The number of epochs for pre-training and fine-tuning phases are 60 and 80, respectively. The settings of λ, η, and ρ in (14) are 10^{-6}, 10^{-3}, and 0.1, respectively. Similar to LP envelope correction, we also provide the common initial hyperparameter settings for the training of LP gain correction. The number of epochs for pre-training and fine-tuning phases are 60 and 100, respectively. λ, η, and ρ are initially set as 10^{-3}, 10^{-4}, and 0.1, respectively. In the rest of this section, all experiments are performed with the above experimental settings.

4.1. Cluster Number Selection

In the proposed speech enhancement system, each LSF vector is categorized into one of the κ clusters before it is corrected by a SAE. A practical issue is to ascertain the optimal number of clusters. This subsection determines the appropriate κ value. It is based on the elbow method (Umargono, Suseno and Gunawan, 2020). Suppose that κ is among $\{1, 2, 4, 8, 16, 32, 64\}$. For each candidate value, we use the LBG algorithm to cluster the LSF vectors in the training set. Once the clustering is completed, the Total Cluster Sum of Square (TCSS) is calculated, given by

$$\text{TCSS} = \sum_{i=1}^{N} d\left(\hat{\mathbf{z}}_i, \boldsymbol{c}_i^*\right), \tag{20}$$

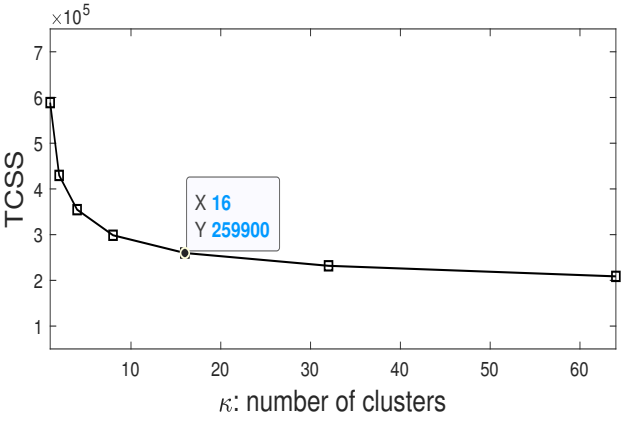

Figure 9: The TCSS values with different κ settings.

where $\boldsymbol{c}_i^*$ is the centroid of the cluster which the i-th input vector $\hat{\mathbf{z}}_i$ belongs to. We repeat the above procedure for all the κ candidates and record the corresponding TCSS results. The result is illustrated in Fig. 9. Note that the main concept of the elbow method is to find the point where the decrease in TCSS suddenly slows down. This point is also called the elbow point. In cluster analysis, the elbow point is considered as the optimal setting for the number of clusters. Prior to this point, adding more clusters greatly minimizes the TCSS, indicating tighter and more effective clustering. Beyond this point, further increasing the cluster number contributes less to the model improvement and may lead to overfitting problems. From Fig. 9, it is clear that "$\kappa = 16$" is the ideal number of clusters in our task.

4.2. Objective Performance Evaluation

In this subsection, we evaluate the performance of the proposed enhancement system based on various objective metrics. Six kinds of denoising methods are considered: a network with fully connected layers (Liu, Smaragdis and Kim, 2014) (denoted as FNN), a network that consists of convolutional layers (Park and Lee, 2016) (denoted as CNN), a generative adversarial network (Shin, Lee, Kim, Park and Han, 2023) (denoted as MetricGAN), a conditional diffusion model (Lu, Wang, Watanabe, Richard, Yu and Tsao, 2022) (denoted as CDiffuse), a phase recovery method (Chao et al., 2022) (denoted as PSE), and our proposed scheme as stated in Section 3 (denoted as HNM_SE).

There are four objective evaluation metrics, the cepstrum distance (CD) (Kitawaki, Nagabuchi and Itoh, 1988), the log spectral distortion (LSD) (Rabiner and Juang, 1993), the Perceptual Evaluation of Speech Quality (PESQ) (Recommendation, 2001), and the short-time objective intelligibility (Taal, Hendriks, Heusdens and Jensen, 2010) (STOI). CD is sensitive to the changes in the periodic components (i.e., the harmonics) of speech. LSD focuses on the discrepancy in the overall spectral structure. We use these two metrics to evaluate how well the recovered signal keeps the spectral features of the original clean one while removing noise distortions. For both LSD and CD, a small measurement value means a high similarity between the recovered and clean spectrum. PESQ measures the perceptual quality of the recovered speech. STOI evaluates speech intelligibility. For these two metrics, a higher value means a better recovery performance. We measure the difference between the recovered and clean utterances based on the above metrics. Besides, to explicitly evaluate the robustness of the proposed system, we further measure its PESQ performance degradation ratio (PDR) across different noise types when the input SNR is reduced from 5 dB to 0 dB, given by

$$\mathrm{PDR} = \frac{\mathrm{PESQ}_{5\mathrm{dB}} - \mathrm{PESQ}_{0\mathrm{dB}}}{\mathrm{PESQ}_{5\mathrm{dB}}}.$$

A smaller PDR indicates stronger robustness at lower input SNR levels. As a comparison, the average degradation ratio for the other five denoising methods is also computed. This value is obtained by first calculating the PDR for each method individually and then taking their arithmetic mean. It is denoted as AVE_PDR. The PESQ degradation ratio is selected as the robustness indicator because the proposed system is designed to enhance the perceptual quality. The average objective test results are summarized in Tables 2 and 3, respectively. The degradation ratio results are summarized in Table 4.

Table 2
Average CD and LSD results of denoising methods under various input SNR levels and noise types.

Noise Type	Method	CD				LSD			
		Input SNR				Input SNR			
		-3dB	0dB	3dB	5dB	-3dB	0dB	3dB	5dB
White	Noisy	9.763	9.528	9.155	8.823	2.722	2.611	2.470	2.361
	CNN	6.917	6.533	6.312	6.231	2.325	2.064	1.848	1.744
	FNN	7.097	6.933	6.791	6.746	2.591	2.371	2.197	2.111
	MetricGAN	9.622	9.304	8.844	8.435	2.449	2.201	2.033	1.953
	CDiffuse	9.469	9.219	8.865	8.587	2.319	2.140	1.989	1.898
	PSE	9.852	9.665	9.352	9.058	2.651	2.496	2.318	2.192
	HNM_SE	**5.088**	**4.731**	**4.416**	**4.252**	**1.642**	**1.410**	**1.234**	**1.173**
Pink	Noisy	9.367	8.946	8.387	7.956	2.420	2.310	2.175	2.071
	CNN	6.499	6.308	6.183	6.148	1.849	1.687	1.590	1.549
	FNN	6.941	6.794	6.689	6.659	2.504	2.285	2.108	2.015
	MetricGAN	9.270	8.835	8.288	7.905	2.033	1.926	1.822	1.757
	CDiffuse	8.893	8.475	7.903	7.467	2.083	1.935	1.801	1.723
	PSE	9.517	9.153	8.642	8.233	2.369	2.209	2.034	1.916
	HNM_SE	**4.929**	**4.539**	**4.229**	**4.079**	**1.547**	**1.356**	**1.204**	**1.147**
F16	Noisy	8.986	8.473	7.841	7.353	2.309	2.201	2.069	1.969
	CNN	6.921	6.533	6.495	6.566	2.098	1.900	1.742	1.664
	FNN	7.168	6.979	6.815	6.721	2.320	2.120	1.955	1.864
	MetricGAN	8.648	8.113	7.509	7.128	2.019	1.927	1.835	1.775
	CDiffuse	8.197	7.685	7.100	6.691	2.047	1.885	1.765	1.692
	PSE	9.194	8.728	8.138	7.685	2.259	2.111	1.947	1.835
	HNM_SE	**4.930**	**4.595**	**4.290**	**4.110**	**1.452**	**1.284**	**1.182**	**1.131**
Babble	Noisy	8.126	7.521	6.842	6.364	2.093	1.983	1.855	1.759
	CNN	6.774	6.444	6.295	6.323	1.924	1.815	1.726	1.688
	FNN	7.151	6.951	6.770	6.684	2.600	2.426	2.269	2.179
	MetricGAN	8.218	7.676	7.137	6.794	1.923	1.838	1.750	1.690
	CDiffuse	7.628	7.017	6.482	6.104	1.922	1.771	1.655	1.597
	PSE	8.401	7.830	7.181	6.720	2.038	1.902	1.753	1.651
	HNM_SE	**5.446**	**5.083**	**4.715**	**4.497**	**1.652**	**1.518**	**1.392**	**1.328**
Vacuum Cleaner	Noisy	7.858	7.749	7.572	7.407	3.079	2.958	2.792	2.660
	CNN	7.562	7.295	7.086	6.973	2.533	2.451	2.385	2.353
	FNN	7.306	7.219	7.110	7.031	3.071	2.836	2.617	2.481
	MetricGAN	8.159	7.878	7.511	7.219	**1.972**	**1.837**	**1.722**	**1.661**
	CDiffuse	8.088	7.872	7.489	7.147	2.482	2.257	1.991	1.845
	PSE	7.940	7.795	7.581	7.386	2.804	2.637	2.431	2.282
	HNM_SE	**7.180**	**6.780**	**6.339**	**6.053**	2.140	1.947	1.780	1.686
Rain	Noisy	8.122	7.988	7.772	7.572	3.432	3.307	3.134	2.996
	CNN	7.993	7.728	7.521	7.426	2.675	2.510	2.399	2.349
	FNN	8.292	8.178	8.002	7.854	3.404	3.154	2.905	2.742
	MetricGAN	8.102	7.778	7.345	6.993	**2.110**	**1.965**	**1.849**	1.784
	CDiffuse	8.073	7.855	7.460	7.053	3.227	2.856	2.479	2.148
	PSE	8.215	8.052	7.807	7.585	3.190	2.998	2.769	2.605
	HNM_SE	**7.691**	**7.234**	**6.756**	**6.443**	2.251	2.025	1.868	**1.677**
Home Kitchen	Noisy	7.490	7.065	6.556	6.177	1.883	1.775	1.642	1.543
	CNN	7.307	7.095	6.877	6.795	2.150	1.943	1.767	1.689
	FNN	7.579	7.442	7.318	7.262	2.388	2.188	2.033	1.950
	MetricGAN	7.119	6.740	6.405	6.195	1.649	1.612	1.591	1.576
	CDiffuse	7.462	6.995	6.499	6.162	1.681	1.584	1.506	1.458
	PSE	7.410	6.984	6.484	6.116	1.810	1.667	1.510	1.405
	HNM_SE	**6.356**	**5.911**	**5.517**	**5.284**	**1.531**	**1.414**	**1.313**	**1.255**

Table 2 shows both the LSD and CD results under various noise conditions. In the CD evaluation, HNM_SE performs better than other denoising methods in all of the input SNR levels and noise environments. For example, under the presence of F16 noise with the input SNR level of 5 dB, the CD value for our scheme is reduced by 37.4% compared to CNN, 38.85% to FNN, 42.34% to Metricgan, 38.57% to CDiffuse, and 46.51% to PSE. Note that end-to-end deep learning methods like MetricGAN and CDiffuse are trained in a purely data-driven manner. They may not adequately

Table 3
Average PESQ and STOI results of denoising methods under various input SNR levels and noise types.

Noise Type	Method	PESQ				STOI			
		Input SNR				Input SNR			
		-3dB	0dB	3dB	5dB	-3dB	0dB	3dB	5dB
White	Noisy	0.94	1.12	1.33	1.47	0.50	0.56	0.63	0.68
	CNN	1.55	1.75	1.88	1.93	0.58	0.65	0.71	0.74
	FNN	1.50	1.66	1.78	1.84	**0.63**	**0.68**	0.71	0.73
	MetricGAN	1.37	1.61	1.84	2.18	0.44	0.47	0.51	0.55
	CDiffuse	0.96	1.21	1.46	1.62	0.52	0.58	0.64	0.68
	PSE	1.00	1.19	1.41	1.55	0.50	0.57	0.63	0.68
	HNM_SE	**1.79**	**2.01**	**2.18**	**2.31**	0.60	0.66	**0.71**	**0.74**
Pink	Noisy	1.14	1.36	1.60	1.76	0.52	0.58	0.65	0.70
	CNN	1.81	1.96	2.07	2.12	**0.62**	0.67	0.71	0.73
	FNN	1.63	1.81	1.96	2.04	0.62	0.66	0.70	0.72
	MetricGAN	1.62	1.62	1.92	2.10	0.47	0.52	0.59	0.63
	CDiffuse	1.24	1.49	1.70	1.92	0.53	0.59	0.66	0.71
	PSE	1.27	1.49	1.72	1.88	0.52	0.59	0.66	0.71
	HNM_SE	**1.95**	**2.17**	**2.37**	**2.48**	0.61	**0.67**	**0.72**	**0.74**
F16	Noisy	1.09	1.31	1.56	1.73	0.53	0.60	0.67	0.72
	CNN	1.50	1.75	1.97	2.08	0.56	0.62	0.67	0.68
	FNN	1.76	1.97	2.15	2.25	0.61	0.68	0.73	0.75
	MetricGAN	1.41	1.62	1.93	2.09	0.46	0.50	0.55	0.58
	CDiffuse	1.25	1.52	1.78	1.94	0.51	0.55	0.61	0.63
	PSE	1.23	1.46	1.72	1.88	0.53	0.60	0.67	0.72
	HNM_SE	**1.85**	**2.12**	**2.34**	**2.45**	**0.63**	**0.69**	**0.74**	**0.77**
Babble	Noisy	1.27	1.49	1.73	1.89	0.51	0.59	0.66	0.70
	CNN	**1.60**	1.82	1.99	2.04	0.55	0.61	0.66	0.67
	FNN	1.54	1.75	1.93	2.01	0.56	**0.63**	0.68	0.71
	MetricGAN	1.43	1.61	1.87	1.98	0.47	0.52	0.57	0.60
	CDiffuse	1.39	1.68	1.97	2.14	0.49	0.55	0.61	0.64
	PSE	1.41	1.65	1.89	2.05	0.52	0.59	0.66	0.71
	HNM_SE	1.57	**1.84**	**2.07**	**2.19**	**0.56**	0.62	**0.69**	**0.72**
Vacuum Cleaner	Noisy	1.15	1.36	1.59	1.76	0.49	0.57	0.65	0.70
	CNN	1.52	1.67	1.80	1.87	0.53	0.60	0.66	0.69
	FNN	1.35	1.57	1.78	1.89	**0.55**	**0.62**	0.67	0.70
	MetricGAN	1.47	1.65	1.86	1.99	0.49	0.53	0.58	0.61
	CDiffuse	1.08	1.39	1.74	1.96	0.46	0.52	0.57	0.62
	PSE	1.31	1.54	1.77	1.92	0.49	0.57	0.65	0.70
	HNM_SE	**1.56**	**1.78**	**1.97**	**2.11**	0.52	0.60	**0.68**	**0.72**
Rain	Noisy	1.09	1.29	1.51	1.67	0.51	0.58	0.65	0.69
	CNN	1.46	1.66	1.80	1.87	**0.60**	**0.67**	**0.72**	**0.74**
	FNN	1.26	1.48	1.69	1.82	0.60	0.66	0.71	0.73
	MetricGAN	**1.62**	1.68	1.86	1.97	0.45	0.51	0.56	0.60
	CDiffuse	1.08	1.32	1.61	1.83	0.51	0.56	0.62	0.65
	PSE	1.21	1.42	1.64	1.79	0.51	0.58	0.65	0.70
	HNM_SE	1.41	**1.68**	**1.92**	**2.11**	0.53	0.60	0.67	0.71
Home Kitchen	Noisy	1.71	1.93	2.15	2.30	0.64	0.70	0.75	0.78
	CNN	1.81	1.99	2.09	2.11	0.63	0.68	0.72	0.73
	FNN	1.70	1.91	2.09	2.18	0.63	0.68	0.73	0.75
	MetricGAN	1.91	2.11	2.28	2.38	0.63	0.66	0.69	0.71
	CDiffuse	1.58	1.81	2.04	2.17	0.59	0.64	0.68	0.69
	PSE	1.86	2.07	2.28	2.42	0.64	0.70	**0.75**	**0.78**
	HNM_SE	**2.15**	**2.33**	**2.50**	**2.59**	**0.66**	**0.71**	0.74	0.76

preserve fine-grained harmonic details. On the other hand, HNM_SE benefits from the model-based analysis-synthesis framework, providing a more dedicated reconstruction of the spectral structure. In addition, the superior performance of HNM_SE on unseen noise demonstrates its generalization capability. Similarly, the LSD results also confirm the advantage of HNM_SE in preserving the spectral information. Again, HNM_SE achieves the lowest LSD under most of the noise conditions.

Table 4
PESQ Performance Degradation Ratio under different noise types.

Noise Type	HNM_SE	AVE_PDR
White	12.99%	18.76%
Pink	12.50%	16.94%
F16	13.47%	18.96%
Babble	15.98%	16.68%
Vacuum Cleaner	15.64%	18.72%
Rain	20.38%	18.63%
Home Kitchen	10.04%	12.09%

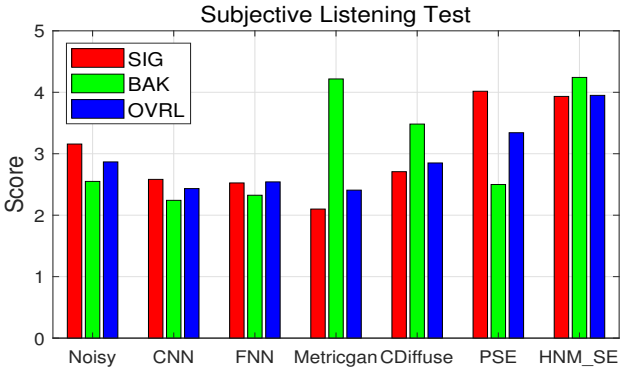

Figure 10: Average subjective test results in terms of SIG, BAK, and OVRL metrics.

Table 3 illustrates the evaluation results in terms of PESQ and STOI with various noise types and input SNR levels. It can be observed that the proposed HNM_SE has the best overall performance in PESQ scores. The average PESQ score of HNM_SE across all the noise conditions is 0.24 higher than that of the best baseline method (CNN). Since PESQ is a perceptually oriented metric, the improvement indicates that HNM_SE provides good gains in perceived speech quality. This enhancement can be attributed to HNM_SE's restoration of harmonic structure and spectral envelope, which are the acoustic cues closely aligned with human auditory perception. In contrast, data-driven baseline methods may fail to capture such useful spectral characteristics when operating solely on waveform (CDiffuse) or spectrogram-based representations (CNN, Metricgan). Apart from the PESQ performance, the proposed enhancement system shows a slight improvement in the intelligibility-oriented metric. Specifically, for HNM_SE, its average STOI score across all noise conditions is 0.04 higher than that of the noisy speech. As mentioned before, our scheme performs the denoising operation by correcting the LP spectral envelope (associated with timbre) and the LP gain (associated with loudness). Considering that these components contribute more directly to perceptual quality rather than to intelligibility, HNM_SE yields comparatively limited improvements in STOI.

Table 4 presents the PESQ performance degradation ratios obtained when the input SNR is decreased from 5 dB to 0 dB. For the four noise types that also appear in the training set (White, Pink, F16, and Babble Noise), HNM_SE consistently achieves lower degradation ratios than the average of the five baseline methods. It indicates that the proposed system maintains a more stable perceptual quality under challenging input SNR conditions. Besides, for the three unseen noise types (Vacuum Cleaner, Rain, and Home Kitchen Noise), our scheme still outperforms the baseline average in most cases. For example, under the home kitchen noise condition, the degradation ratio of HNM_SE is only 10.04%, while AVE_PDR is 12.09%. These observations collectively demonstrate the robustness of the proposed enhancement system.

4.3. Subjective Performance Evaluation

This subsection conducts a subjective listening test to evaluate enhanced speeches based on human listeners' perspectives. Such a test is necessary because certain perceptual artifacts or unnatural acoustic cues cannot be fully

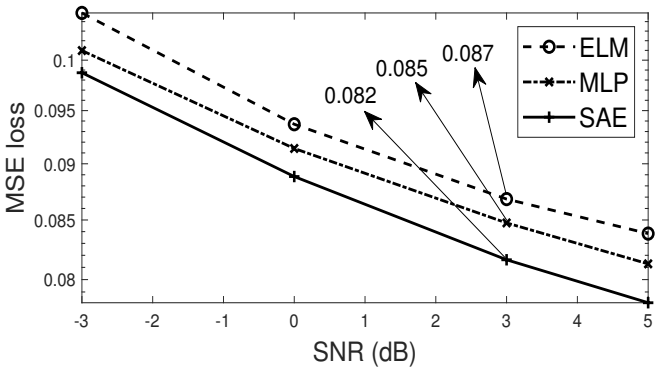

Figure 11: The MSE loss of the LP envelope correction from ELM, MLP, and SAE.

captured by objective metrics alone. It strictly follows the procedure mentioned in (Loizou, 2007). Six speech utterances corrupted by vacuum cleaner noise are randomly selected from the test set at an input SNR of 5 dB. Ten listeners are asked to successively listen to and rate the enhanced speech signals produced by the six denoising methods as well as the original noisy speeches. There are three subjective metrics: SIG, BAK, and OVRL. For SIG, it assesses the quality and naturalness of speech itself. For BAK, it measures the degree of background noise suppression regardless of the speech signal. OVRL provides an overall rating that considers speech clarity, noise reduction, and naturalness together. Each metric is scored from 1 to 5, where higher scores indicate better performance. The average results across all listeners and utterances are summarized in Fig. 10.

Fig. 10 illustrates the average SIG, BAK, and OVRL scores under the vacuum cleaner noise condition. We can see an apparent performance disparity across different enhancement methods. For example, both CNN and FNN get even lower scores than the noisy baseline. That means these data-driven models have limited generalization to unseen noise conditions. In contrast, MetricGAN and CDiffuse achieve notably high BAK scores, confirming their strong ability to suppress background noise. However, the associated SIG scores remain low. It suggests that over-noise suppression may have occurred, affecting the clarity and naturalness of the speech signal. PSE is designed to correct the phase of noisy speech, which contributes to lifting speech naturalness. Hence, its SIG score is competitive. Nevertheless, as phase correction alone has a restricted effect on noise removal, the resulting BAK score is almost the same as that of the noisy baseline. Unlike the above methods, the proposed HNM_SE method works well in both speech signal recovery and noise removal. Recall that our scheme effectively recovers both speech timbre (via LP envelope correction) and loudness (via LP gain estimation). In addition, since the HNM framework only depends on estimated acoustic cues to regenerate speech, any background noise components overlapped with the harmonic structure are inherently excluded during synthesis. The well performance in SIG and BAK metrics matches our design. Correspondingly, its OVRL score is the highest among all evaluated methods.

4.4. Effectiveness of SAE in LP Envelope Correction

In Section 3.4, we applied SAEs to map noisy LP envelopes with clean ones. The encoder-decoder architecture and "pre-training" plus "fine-tuning" make SAEs suitable for the correction task. This subsection verifies it. Three kinds of neural networks are considered. One is the SAE. The second is a three-layer extreme learning machine (ELM). For ELM, the number of neurons in the input and output layers is 252, the same as in SAE. The hidden layer contains 504 (252×2) neurons. The aim of choosing ELM for comparison is to find out whether the encoder-decoder SAE architecture assists in eliminating the informational redundancy of LSF vectors and learning compact (essential) representation. The third is a multilayer perceptron (MLP). Its architecture is the same as the SAE. However, unlike SAE, the training process of an MLP is based on a backpropagation algorithm (Rojas and Rojas, 1996). By comparing with the performance of an MLP, we would like to know if the "pre-training" plus "fine-tuning" operations enable the SAE to learn more robust features from the noise-corrupted data. To make a fair comparison, we also use cluster-specific training for ELMs and MLPs. There are 16 clusters in this experiment. Since all the resultant networks are trained to minimize the MSE between the target and generated outputs, we measure the MSE between the clean LSFs and their corrected LSFs under different input SNR levels across all the noise types. The sensitivity of MSE to large

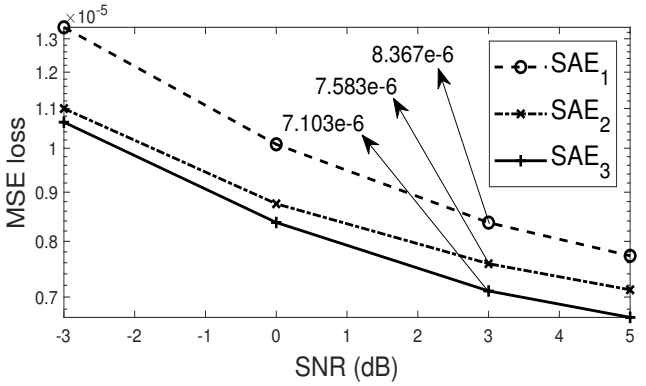

Figure 12: The MSE loss of the LP gain correction from SAE_1, SAE_2, and SAE_3.

deviations is ideal for capturing the instability in the LSF correction. Fig. 11 shows the average MSE results across all the clusters. We get that the SAE has the lowest MSE at all the SNR conditions. For instance, when the input SNR level is 3 dB, the ELM yields an average MSE of 0.087. On the other hand, the average MSE of SAE is reduced by around 6%. This experimental result demonstrates the SAE encoder-decoder structure's usefulness in learning essential LSF representation from the redundant neighboring frames. Moreover, the SAE outperforms the MLP, suggesting that combining pre-training and fine-tuning allows the SAE to get a more robust generalization ability.

4.5. Effectiveness of LP Gain Correction Design

In Section 3.5, we proposed an indirect way to cluster LP gains, i.e., the LSF clustering result guides the grouping of LP gain vectors. In addition, we integrated Hamming weighting factors into the conventional MSE-based loss function to prevent the SAE from overfitting less pertinent features at the boundary frames. This subsection validates the effectiveness of the above two strategies. Three kinds of LP gain correction designs are considered. One is that we directly apply the LBG algorithm to split the LP gain vectors into 16 clusters. For each cluster, an SAE with a standard MSE-based loss function

$$\widetilde{\mathscr{L}}_{\text{tune}} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{\zeta=1}^{2q+1}\left(\mathscr{g}_i^{(\zeta)} - \mathbb{g}_i^{(\zeta)}\right)^2 \tag{21}$$

is trained for LP gain correction. Let us denote this design as SAE_1. The second is adopting indirect clustering to group LP gain vectors. When training SAE, the loss function is the same as (21). It is denoted as SAE_2. The third is adopting both the indirect strategy and the Hamming-weighted loss function, as stated in Section 3.5. Denote it as SAE_3. During the test, for each input SNR level across all the noise types, we measure the difference between the clean and corrected LP gains derived from these designs. Fig. 12 depicts the average MSE results across all the clusters. It can be observed that compared to the direct clustering strategy, the indirect one helps lift the correction performance. For example, at the input SNR level of 3 dB, the MSE loss of SAE_2 is only 7.58×10^{-6}, while that of SAE_1 is 8.37×10^{-6}. It is because LP gain, as an energy-related parameter, is highly sensitive to noise, particularly under low-SNR conditions. Its values can fluctuate or become biased, resulting in unstable clustering behavior. The LSFs, as the speech resonant characteristics, are relatively less affected by the noise energy. With the Hamming-weighted loss function, the correction performance is further improved, e.g., the MSE loss is reduced to 7.1×10^{-6} for SAE_3 at the same SNR level. The experimental result demonstrates the usefulness of the proposed loss function.

4.6. Ablation Study

In this subsection, we investigate the contribution of key modules (i.e., LSA-based pre-cleaning, SAE-based LP envelope and gain correction, and HNM analysis-synthesis structure) in the proposed enhancement system. An ablation study is conducted with three kinds of system module configurations. One is directly applying HNM to reconstruct speech from noisy utterances, denoted as noisyHNM. The second is applying HNM to reconstruct speech from pre-cleaned utterances using the LSA method, denoted as LsaHNM. The third is the proposed system HNM_SE, which

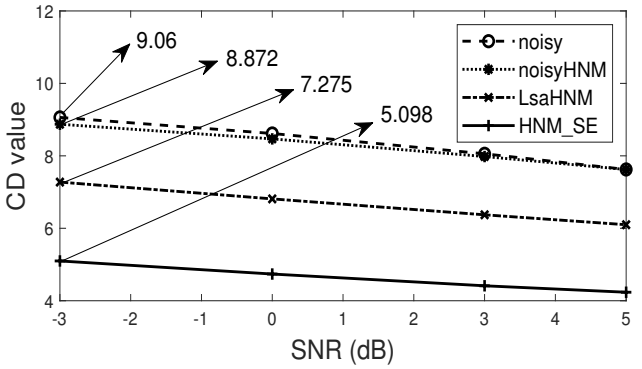

Figure 13: The CD values of the enhancement system with different module configurations.

combines the LSA-based pre-cleaning operation, SAE-based LP parameter recovery, and HNM. We want to know whether these modules really help recover the clean harmonic structure from noisy utterances. As the CD metric reflects changes in speech's periodic components (i.e., harmonics), it is used to evaluate the difference between the enhanced and clean signals. Fig. 13 depicts the associated CD results under various input SNR levels across all the noise types. We can see that with the inclusion of successive modules, the performance of the enhancement system progressively improves. For example, when the input SNR level is -3 dB, applying HNM to noisy speeches can slightly decrease the distance, i.e., from 9.06 to 8.872 for noisyHNM. It is because the reconstructed harmonic structure from the HNM framework does not have any background noise components. However, the improvement remains modest due to inaccurate pitch detection in noisy speech, which makes the recovered harmonic structure deviate from the true one. With the pre-cleaning operation, the cepstrum distance is further reduced to 7.275. This improvement can be attributed to the LSA method's ability: it not only roughly recovers the distorted spectral envelope but also provides a spectrum with more prominent harmonic components, allowing the system to detect pitch more accurately. After adding the SAE-based LP parameter correction modules, the distance for HNM_SE is only 5.098. As the only difference between LsaHNM and HNM_SE is the inclusion of the SAE module, the performance gain observed when transitioning from LsaHNM to HNM_SE, i.e., an extra $(7.275-5.098)/7.275 \approx 30\%$ reduction in the cepstrum distance, explicitly reflects the isolated SAE's effect on recovering LP parameters. To sum up, all the modules in the proposed system contribute to the harmonic structure recovery in noisy speeches.

5. Discussion

This section first discusses the training time complexity of the proposed speech enhancement system. Then, its inference time is evaluated and compared with that of existing works.

5.1. Training Time Complexity

We analyze the training time complexity of the proposed scheme. Recall that it consists of five modules: (1) pre-cleaning noisy speech with LSA method, (2) clustering LSFs from precleaned consecutive frames via LBG algorithm, (3) SAE training for LP envelope correction, (4) SAE training for LP gain correction, and (5) HNM for enhanced speech reconstruction. As the pre-cleaning operation and the HNM do not have any training process, we exclude them from this analysis. LBG-based clustering is conducted on the bundled LSF vectors. Let N_{LSF} be the total number of input vectors. As mentioned before, the dimensionality of each vector is $D_{\text{vec}} = P \times (2q+1)$. Hence, the time complexity of LBG is $O(E_{\text{LBG}} \times N_{\text{LSF}} \times \kappa \times D_{\text{vec}})$, where E_{LBG} is the number of iterations to terminate the training and κ is the number of clusters. For each cluster, an SAE is trained to correct LSFs. Let $N_{\mathscr{k}}$ be the number of LSF vectors in the $\mathscr{k}$-th cluster. $E_{\text{PT}}^{(\mathscr{k})}$ and $E_{\text{FT}}^{(\mathscr{k})}$ are the epochs for pre-training and fine-tuning, respectively. There are L_{num} layers in a SAE and the layer ℓ has H_ℓ neurons. The associated training time complexity across all the clusters is $O\Big(\sum_{\mathscr{k}=1}^{\kappa}\Big(E_{\text{PT}}^{(\mathscr{k})}+E_{\text{FT}}^{(\mathscr{k})}\Big)\times N_{\mathscr{k}}\times\sum_{\ell=1}^{L_{\text{num}}-1}H_\ell\times H_{\ell+1}\Big)$. Similarly, in LP gain correction, the associated training time complexity across all the clusters is $O\Big(\sum_{\mathscr{k}=1}^{\kappa}\Big(\hat{E}_{\text{PT}}^{(\mathscr{k})}+\hat{E}_{\text{FT}}^{(\mathscr{k})}\Big)\times N_{\mathscr{k}}\times\sum_{\hat{\ell}=1}^{\hat{L}_{\text{num}}-1}H_{\hat{\ell}}\times H_{\hat{\ell}+1}\Big)$, where $\hat{E}_{\text{PT}}^{(\mathscr{k})}$ and $\hat{E}_{\text{FT}}^{(\mathscr{k})}$ are

the associated epochs for pre-training and fine-tuning, respectively. Each SAE has $\hat{L}_{\text{num}}$ layers and the layer $\hat{\ell}$ has $H_{\hat{\ell}}$ neurons. To sum up, the training time complexity of the proposed speech enhancement system is

$$O(E_{\text{LBG}}\times N_{\text{LSF}}\times\kappa\times D_{\text{vec}}) + O\Big(\sum_{k=1}^{\kappa}\Big(E_{\text{PT}}^{(k)}+E_{\text{FT}}^{(k)}\Big)\times N_k\times\sum_{\ell=1}^{L_{\text{num}}-1}H_\ell\times H_{\ell+1}\Big)$$
$$+O\Big(\sum_{k=1}^{\kappa}\Big(\hat{E}_{\text{PT}}^{(k)}+\hat{E}_{\text{FT}}^{(k)}\Big)\times N_k\times\sum_{\hat{\ell}=1}^{\hat{L}_{\text{num}}-1}H_{\hat{\ell}}\times H_{\hat{\ell}+1}\Big). \tag{22}$$

Please note that the proposed framework adopts the cluster-specific training. Since each cluster is processed independently, it is possible to perform the subsequent training of an SAE for each cluster in a distributed or parallelized way. In that case, the time complexity in (22) can be further reduced to

$$O(E_{\text{LBG}}\times N_{\text{LSF}}\times\kappa\times D_{\text{vec}}) + O\Big(\max\Big(\Big(E_{\text{PT}}^{(k)}+E_{\text{FT}}^{(k)}\Big)\times N_k\Big)\times\sum_{\ell=1}^{L_{\text{num}}-1}H_\ell\times H_{\ell+1}\Big)$$
$$+O\Big(\max\Big(\Big(\hat{E}_{\text{PT}}^{(k)}+\hat{E}_{\text{FT}}^{(k)}\Big)\times N_k\Big)\times\sum_{\hat{\ell}=1}^{\hat{L}_{\text{num}}-1}H_{\hat{\ell}}\times H_{\hat{\ell}+1}\Big). \tag{23}$$

5.2. Inference Time Evaluation

We perform simulations to evaluate the inference time of our scheme. The test set consists of 30 noisy speeches. That is, 30 clean utterances (from the TIMIT Corpus) are mixed with white noise (from the NOISEX-92 database) at the input SNR level of 5dB. The proposed system, HNM_SE, is trained with the same settings and model configurations as in Section 4. For each noisy speech, HNM_SE is used to enhance it 10 times. We measure the average inference time for the enhancing process. The average inference time for the existing works, FNN and CNN, is also measured for comparison. The Matlab simulations[5] are run on an Intel Core i7-10750H processor. We get that the average time values are 3.34s for HNM_SE, 1.95s for FNN, and 2.51s for CNN. The longer inference time of HNM_SE, i.e., extra 1.39s compared to FNN and 0.83s compared to CNN, is due to its inherent design. First, the proposed system relies on the LSA method to pre-clean the noisy speech. It introduces an unavoidable initial delay. Second, as shown in Section 2.1, pitch detection in HNM_SE requires an exhaustive evaluation of all pitch candidates by calculating their matching errors. This operation is computationally intensive. Please be reminded that although FNN and CNN run faster, both methods neglect the effect of harmonic structure on perceptual speech quality. From Table 3, HNM_SE's PESQ score is 0.3 and 0.24 higher than FNN's and CNN's across all the noise conditions, respectively. A PESQ improvement exceeding 0.2 usually indicates a perceivable enhancement in speech quality.

6. Conclusion

This paper presented a speech enhancement system that combines the HNM analysis–synthesis framework with SAE-based spectral envelope correction. We applied the HNM to reconstruct the harmonic structure based on acoustic parameters and remove background noise components overlapped with the structure. As the acoustic parameters (LP envelope and gain) extracted from the noise-corrupted spectrum are greatly distorted, a clustering coupled with the SAE-based correction process was proposed for their restoration. Through the exploration of inter-frame and intra-frame perceptual speech characteristics, the SAEs build a robust mapping relationship between the noise-free and distorted parameters. In particular, we designed a Hamming-weighted loss function to apply a higher weight to the central frame while gradually reducing it towards the edges when training SAEs to correct the gain. Unlike end-to-end deep-learning-based enhancement methods that primarily aim to minimize the overall difference between predicted and clean signals under mathematical mapping criteria, the proposed system reconstructs clean speech by explicitly leveraging perceptually relevant acoustic cues with learning-based correction to refine specific speech characteristics, thereby improving perceptual quality. Moreover, the time complexity of the system's training process was discussed, and its inference time was evaluated. From a practical perspective, the proposed system introduces extra inference latency compared to end-to-end deep neural networks due to the inclusion of noise pre-cleaning and acoustic parameter extraction. However, this increased computational cost leads to a clear improvement in perceptual quality, as evidenced by a PESQ gain exceeding 0.2. Experiments demonstrate the effectiveness of our scheme in terms of objective and subjective metrics.

[5]Here, all the inference simulations are conducted on MATLAB rather than the Python platform. It is because the HNM analysis-synthesis framework is implemented in MATLAB. For a fair comparison, all the associated neural networks in this test are recoded in MATLAB format accordingly.

There are several potential directions for our future work. First, the V/UV mixing function in the current study is based on spectral flatness measurement. It may not accurately capture the energy distribution between voiced and unvoiced components within each sub-band. Therefore, we will find a better V/UV mixing method with an energy-sensitive criterion, e.g., locally normalized cross-correlation in (Erro, Sainz, Navas and Hernáez, 2011b). Second, the proposed enhancement system can be simplified to strengthen its applicability for real-time applications. For example, lightweight convolutional neural networks with 1D convolution operation (Luo and Mesgarani, 2019) may be used to replace the SAEs. Third, as phase correction is able to enhance perceptual quality, we plan to explore attention-based mechanisms (Saleem, Gunawan, Dhahbi and Bourouis, 2024; Li, Liu and Zhou, 2025) for phase recovery and investigate their lightweight implementations to reduce computational load.

Funding information

This research/project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. The work at the University of Alberta was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Project RES0048688 and Project RES0054326, and by Alberta Innovates under Project RES0053965.

CRediT authorship contribution statement

Wenhao Lu: Conceptualization, Methodology, Software, Writing - Original Draft. **Zhenya Zang:** Software, Investigation. **Feng Qin:** Methodology, Writing - Review and Editing. **Xia Dong:** Validation. **Jie Han:** Software, Writing - Review and Editing. **Zuozhou Pan:** Investigation, Validation, Writing - Review and Editing. **Yiping Ke:** Writing - Review and Editing.

References

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., 2006. Greedy layer-wise training of deep networks. Advances in neural information processing systems 19.

Chan, C.F., Yu, E., 1996. Improving pitch estimation for efficient multiband excitation coding of speech. Electronics Letters 32, 870–872.

Chao, R., Yu, C., Fu, S.W., Lu, X., Tsao, Y., 2022. Perceptual contrast stretching on target feature for speech enhancement. Proc. of INTERSPEECH .

Chen, C., Hu, Y., Zou, H., Sun, L., Chng, E.S., 2023. Unsupervised noise adaptation using data simulation, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1–5.

Chen, C., Zhang, P., 2024. Trnet: Two-level refinement network leveraging speech enhancement for noise robust speech emotion recognition. arXiv preprint arXiv:2404.12979 .

Cui, Z., Zhang, S., Chen, Y., Gao, Y., Deng, C., Feng, J., 2023. Semi-supervised speech enhancement based on speech purity, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1–5.

Dean, D., Sridharan, S., Vogt, R., Mason, M., 2010. The qut-noise-timit corpus for evaluation of voice activity detection algorithms, in: Proceedings of the 11th annual conference of the international speech communication association, International Speech Communication Association. pp. 3110–3113.

Durbin, J., 1960. Estimation of parameters in time-series regression models. Journal of the Royal Statistical Society Series B: Statistical Methodology 22, 139–153.

Ephraim, Y., Malah, D., 1985. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE transactions on acoustics, speech, and signal processing 33, 443–445.

Erro, D., Sainz, I., Navas, E., Hernáez, I., 2011a. Hnm-based mfcc+ f0 extractor applied to statistical speech synthesis, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 4728–4731.

Erro, D., Sainz, I., Navas, E., Hernáez, I., 2011b. Improved hnm-based vocoder for statistical synthesizers., in: Interspeech, pp. 1809–1812.

Ferreira, A.J., 2007. Static features in real-time recognition of isolated vowels at high pitch. The Journal of the Acoustical Society of America 122, 2389–2404.

Flanagan, J.L., Cherry, L., 1969. Excitation of vocal-tract synthesizers. The journal of the Acoustical Society of America 45, 764–769.

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., 1988. Getting started with the darpa timit cd-rom: An acoustic phonetic continuous speech database. National Institute of Standards and Technology (NIST), Gaithersburgh, MD 107, 16.

Green, T., Hilkhuysen, G., Huckvale, M., Rosen, S., Brookes, M., Moore, A., Naylor, P., Lightburn, L., Xue, W., 2022. Speech recognition with a hearing-aid processing scheme combining beamforming with mask-informed speech enhancement. Trends in Hearing 26, 23312165211068629.

Griffin, D.W., Lim, J.S., 1988. Multiband excitation vocoder. IEEE Transactions on acoustics, speech, and signal processing 36, 1223–1235.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. science 313, 504–507.

Hsu, Y., Lee, Y., Bai, M.R., 2022. Learning-based personal speech enhancement for teleconferencing by exploiting spatial-spectral features, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 8787–8791.

Huang, Q., Bao, C., Wang, X., 2017. Improved codebook-based speech enhancement based on mbe model., in: Interspeech, pp. 3627–3631.

Huang, Q., Bao, C., Wang, X., Xiang, Y., 2020. Speech enhancement method based on multi-band excitation model. Applied Acoustics 163, 107236.

Kirton-Wingate, J., Ahmed, S., Gogate, M., Tsao, Y., Hussain, A., 2023. Towards individualised speech enhancement: An snr preference learning system for multi-modal hearing aids, in: 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), IEEE. pp. 1–5.

Kitawaki, N., Nagabuchi, H., Itoh, K., 1988. Objective quality evaluation for low-bit-rate speech coding systems. IEEE Journal on Selected Areas in Communications 6, 242–248.

Li, M., Liu, Y., Zhou, L., 2025. Deconformer-senet: An efficient deformable conformer speech enhancement network. Digital Signal Processing 156, 104787.

Linde, Y., Buzo, A., Gray, R., 1980. An algorithm for vector quantizer design. IEEE Transactions on communications 28, 84–95.

Liu, D., Smaragdis, P., Kim, M., 2014. Experiments on deep learning for speech denoising, in: Fifteenth Annual Conference of the International Speech Communication Association.

Loizou, P.C., 2007. Speech enhancement: theory and practice. CRC press.

Lu, Y.J., Wang, Z.Q., Watanabe, S., Richard, A., Yu, C., Tsao, Y., 2022. Conditional diffusion probabilistic model for speech enhancement, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Ieee. pp. 7402–7406.

Luo, Y., Mesgarani, N., 2019. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. IEEE/ACM transactions on audio, speech, and language processing 27, 1256–1266.

Park, S.R., Lee, J., 2016. A fully convolutional neural network for speech enhancement. arXiv preprint arXiv:1609.07132 .

Piczak, K.J., 2015. Esc: Dataset for environmental sound classification, in: Proceedings of the 23rd ACM international conference on Multimedia, pp. 1015–1018.

Ping, H., Yafeng, W., 2022. Single-channel speech enhancement using improved progressive deep neural network and masking-based harmonic regeneration. Speech Communication 145, 36–46.

Rabiner, L., Juang, B.H., 1993. Fundamentals of speech recognition. Prentice-Hall, Inc.

Rao, S., Pearlman, W.A., 1996. Analysis of linear prediction, coding, and spectral estimation from subbands. IEEE Transactions on Information Theory 42, 1160–1178.

Recommendation, I.T., 2001. Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. Rec. ITU-T P. 862 .

Richter, J., Welker, S., Lemercier, J.M., Lay, B., Gerkmann, T., 2023. Speech enhancement and dereverberation with diffusion-based generative models. IEEE/ACM Transactions on Audio, Speech, and Language Processing 31, 2351–2364.

Rojas, R., Rojas, R., 1996. The backpropagation algorithm. Neural networks: a systematic introduction , 149–182.

Saleem, N., Gunawan, T.S., Dhahbi, S., Bourouis, S., 2024. Time domain speech enhancement with cnn and time-attention transformer. Digital Signal Processing 147, 104408.

Samui, S., Chakrabarti, I., Ghosh, S.K., 2019. Time–frequency masking based supervised speech enhancement framework using fuzzy deep belief network. Applied Soft Computing 74, 583–602.

Shin, W., Lee, B.H., Kim, J.S., Park, H.J., Han, S.W., 2023. Metricgan-okd: multi-metric optimization of metricgan via online knowledge distillation for speech enhancement, in: International Conference on Machine Learning, PMLR. pp. 31521–31538.

Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J., 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech, in: 2010 IEEE international conference on acoustics, speech and signal processing, IEEE. pp. 4214–4217.

Tan, K., Zhang, X., Wang, D., 2021. Deep learning based real-time speech enhancement for dual-microphone mobile phones. IEEE/ACM transactions on audio, speech, and language processing 29, 1853–1863.

Trinh, V.A., Braun, S., 2022. Unsupervised speech enhancement with speech recognition embedding and disentanglement losses, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 391–395.

Umargono, E., Suseno, J.E., Gunawan, S.V., 2020. K-means clustering optimization using the elbow method and early centroid determination based on mean and median formula, in: The 2nd international seminar on science and technology (ISSTEC 2019), Atlantis Press. pp. 121–129.

Varga, A., Steeneken, H.J., 1993. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. Speech communication 12, 247–251.

Wang, D., Lim, J., 1982. The unimportance of phase in speech enhancement. IEEE Transactions on Acoustics, Speech, and Signal Processing 30, 679–681.

Wang, H., Wang, D., 2023. Cross-domain diffusion based speech enhancement for very noisy speech, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 1–5.

Welker, S., Richter, J., Gerkmann, T., 2022. Speech enhancement with score-based generative models in the complex stft domain, in: Interspeech 2022, Int Speech Commun Assoc. ISCA. pp. 2928–2932.

Xu, Y., Du, J., Dai, L.R., Lee, C.H., 2014. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM transactions on audio, speech, and language processing 23, 7–19.

Yu, E.W., Chan, C.F., 1999. Harmonic+ noise coding using improved v/uv mixing and efficient spectral quantization, in: 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258), IEEE. pp. 477–480.