Mälardalen University
School of Innovation Design and Engineering
Västerås, Sweden

Master of Science in Computer Science - 60 ECTS

MULTICHANNEL NOISE REDUCTION WITH NEURAL NETWORKS FOR HEARING AIDS

Harry Setiawan Hamjaya
hha24001@student.mdh.se

Examiner: Abu Naser Masud
Mälardalen University, Västerås, Sweden

Supervisors: Antonio Cicchetti
Mälardalen University, Västerås, Sweden

Company supervisor: Anu Huttunen
Huawei Technologies Oy, Tampere, Finland

September 18, 2025

This thesis was conducted within the Erasmus Mundus Joint Master Degree Programme on the Engineering of Data-intensive Intelligent Software Systems (EDISS)

Abstract

Hearing aids are essential assistive devices that improve communication for individuals with hearing loss, yet their performance in noisy, real-world environments remains limited. This thesis investigates multichannel speech enhancement for real-time noise reduction using neural networks, with the goal of improving speech intelligibility and clarity in hearing aid applications. The central research question addressed is how a multichannel enhancement algorithm can be effectively designed and implemented under the real-time constraints.

The motivation for this work lies in the unmet needs of hearing aid users, where traditional linear methods such as beamforming and Wiener filtering struggle in dynamic, multi-source acoustic conditions. To tackle these challenges, this thesis adapts the Joint Non-linear Filtering (JNF) model with a unidirectional LSTM architecture, enabling causal, frame-wise inference compatible with real-time processing. The system is trained on a large multichannel dataset augmented with realistic environmental noise and optimized using a hybrid between time and frequency loss function combining perceptual quality (PESQ) objectives.

The results demonstrate that the proposed approach achieves competitive performance relative to state-of-the-art methods while meeting real-time requirements. Notably, the introduced sliding-window inference strategy balances accuracy with memory efficiency, making deployment on resource-constrained devices feasible. The findings confirm that neural-based multichannel processing can significantly enhance speech intelligibility in hearing aid scenarios. Beyond hearing aids, the framework shows potential for broader applications in communication systems and real-time audio enhancement tasks. Overall, this work contributes a practical and scalable solution that bridges the gap between high-performance research models and real-world deployment.

Table of Contents

1 Introduction

Understanding speech in a noisy environment is one of the challenges human beings have faced on various occasions. One of the well-known classic problems is a conversation at a cocktail party, which requires one to extract a target speech that interferes with background noise, other speakers, and/or reverberation [1]. However, even in normal everyday situations, this can be a challenge for some people. For example, having a conversation in a crowded restaurant or in a noisy factory environment can be particularly difficult.

This challenge is especially pronounced for individuals with sensorineural hearing loss, a condition that typically cannot be cured through surgical or pharmacological means [2]. Instead, such hearing impairments are most effectively addressed through technical amplification using hearing aids. Modern hearing aids have undergone significant technological advancement over the past decades, including improved signal processing, directional microphones, and open fitting strategies that reduce occlusion effects and enhance sound quality. These improvements have significantly increased user satisfaction, particularly in complex acoustic environments.

The importance of early hearing aid intervention is also well-supported by recent research. Delayed use of hearing aids can lead to auditory deprivation, social isolation, cognitive decline, and even dementia [2]. Hearing aids are not merely assistive devices but tools for maintaining quality of life, especially among the elderly population. Despite this, adoption rates remain low often due to stigma, cost, or lack of awareness highlighting the ongoing need for better hearing aid education and accessibility.

In addition to personal hearing assistance, the underlying technologies used in hearing aids are applicable to broader domains such as telecommunication systems and automatic speech recognition systems that operate in noisy environments. The technical solution lies in algorithms for speech enhancement that reduce background noise while preserving intelligibility.

The classification of speech processing algorithms often depends on the number of microphones (or channels) used in audio data acquisition. In single-channel algorithms, tempo-spectral features are the primary basis for processing. In contrast, multi-channel algorithms can leverage spatial information, since audio signals travel different paths to each microphone depending on their source direction. This spatial diversity is exploited in methods such as the delay-and-sum beamformer [3], which aligns signals in time across channels to suppress uncorrelated noise. Another technique is the Minimum Variance Distortionless Response (MVDR) beamformer [3], which minimizes noise variance while maintaining the integrity of the target signal. These and other traditional linear spatial filters are commonly followed by single-channel post-filters to further enhance noise reduction. The multichannel Wiener filter combines MVDR beamforming and a Wiener post-filter [4]. However, these methods often rely on the assumption of Gaussian-distributed noise.

More recent work has explored nonlinear methods such as the Joint Non-linear Filter (JNF), which performs joint spatial and tempo-spectral processing for more effective speech enhancement in non-Gaussian noise environments [5]. Tesch et al. [6] evaluated several DNN-based JNF models (F-JNF, T-JNF, FT-JNF), revealing that integrating spectral and spatial information enhances spatial selectivity and performance. However, their simulation-based noise generation and use of bidirectional layers limited the model's applicability to real-time scenarios.

In this work, we focus specifically on real-time speech enhancement using a frame-wise version of the FT-JNF model [7], designed without bidirectional layers to support causal processing. Instead of using simulation-based noise, we use recorded background noise captured in a laboratory setting. The central challenge and novelty of this work lies in implementing frame-wise prediction using time-series audio data, enabling performance comparable to offline methods while maintaining real-time feasibility.

Importantly, the proposed model achieved notable improvements in the speech enhancement task, with higher PESQ scores (perceptual quality) and consistent gains in SDR/SI-SDR (signal-to-distortion ratios) compared to the state-of-the-art baseline [6]. These results confirm that careful loss design and architectural adjustments allow real-time systems to approach the performance of more computationally expensive offline models, making the approach especially promising for hearing aid applications.

2 Background

Speech enhancement seeks to improve the quality and intelligibility of degraded speech signals by suppressing noise, reverberation, and interfering speakers while preserving the target speech. In multi-microphone (multi-channel) settings, spatial diversity enables the exploitation of inter-channel time, phase, and level differences to separate sources and attenuate noise more effectively than single-channel techniques [3, 8, 9, 10]. Beyond communication and ASR, enhanced spatial processing supports assistive listening and hearing aids, where constraints on latency, power, and form factor motivate robust, low-complexity methods [2, 11]. Recent surveys and monographs provide broad perspectives on algorithmic families ranging from classical statistical estimators to modern machine learning approaches [12, 13, 14, 15].

2.1 Short-Time Fourier Transform (STFT)

A fundamental step in speech enhancement is the choice of signal representation. Because speech signals are inherently non-stationary, they must be analyzed in a way that captures both their temporal evolution and spectral content. The short-time Fourier transform (STFT) provides such a framework, allowing signals to be expressed in the joint time-frequency domain, which is widely used in both classical and neural-based speech processing.

The short-time Fourier transform (STFT) is a fundamental tool in speech and audio signal processing, as it allows non-stationary signals such as speech to be represented jointly in time and frequency domains [12, 16, 17]. Since speech characteristics vary rapidly, the STFT provides a sequence of quasi-stationary frames that enable effective modeling of speech structures, noise statistics, and spatial correlations across multiple microphones.

2.1.1 Continuous-Time STFT

In the continuous domain, the STFT of a signal $X(t,f)$ is defined as

$$X(t,f) = \int_{-\infty}^{\infty} x(u)\, h^*(u-t)\, e^{-j2\pi f u}\, du, \tag{1}$$

where $h^*(\cdot)$ is the conjugated analysis window, t is the time-shift parameter, and f is the frequency variable. Here, u is the integration variable that spans the entire signal domain, while the window function $h(u-t)$ restricts the analysis to a neighborhood around the time instant t. This way, the STFT captures the local spectral content of $x(t)$ around time t by focusing only on the portion of the signal where the window has significant support [16].

2.1.2 Discrete-Time STFT

For sampled signals $x[n]$, the discrete-time STFT is defined as

$$X(m,k) = \sum_{n=0}^{N-1} x[n+mR]\, h[n]\, e^{-j(2\pi/N)kn}, \tag{2}$$

where $h[n]$ is the analysis window of length N, k is the frequency bin index ($0 \leq k < N$), m is the frame index, and R is the hop size (frame shift). This formulation enables practical implementation of the STFT in digital systems [17].

2.1.3 Inverse STFT (iSTFT)

The time-domain signal can be reconstructed from its STFT using the inverse short-time Fourier transform (iSTFT), typically implemented via the *overlap-add* method. The reconstructed signal is given by

$$\hat{x}[n] = \frac{\sum_m h[n-mR] \cdot \hat{x}_m[n]}{\sum_m h[n-mR]}, \tag{3}$$

where $\hat{x}_m[n]$ is the time-domain representation of the m-th frame, obtained by taking the inverse discrete Fourier transform (IDFT) of the STFT coefficients $X(k,m)$

$$\hat{x}_m[n] = \frac{1}{N}\sum_{k=0}^{N-1} X(k,m)e^{j(2\pi/N)kn}. \tag{4}$$

Here, $X(k,m)$ denotes the short-time Fourier transform coefficient at frequency bin k and frame index m, N is the DFT size (typically equal to the window length), R is the hop size, and $h[n]$ is the analysis window used in the forward STFT (assumed to be the same as the synthesis window in this case). The denominator in (3) serves as a normalization factor to compensate for overlapping windows. When the window $h[n]$ and hop size R satisfy the *constant overlap-add (COLA)* condition, perfect reconstruction can be achieved, i.e., $\hat{x}[n] = x[n]$ (up to a delay) [18].

The Short-Time Fourier Transform (STFT) is widely used in speech and audio processing because it shows how the frequency content of a signal changes over time. In speech enhancement, it helps separate clean speech from noise by analyzing the time-frequency (TF) representation [12]. For source separation, it uses differences in location and frequency across microphones to isolate individual sounds [14]. The STFT is also key for feature extraction, as spectrograms and mel-spectrograms derived from it are commonly used in speech recognition and deep learning [16]. It further supports time-frequency filtering, such as designing masks to reduce noise, reverberation, or focus on specific directions (beamforming) [17, 18]. Overall, the STFT provides a simple yet powerful way to analyze both time and frequency, making it essential in modern audio systems.

2.2 Multichannel Acoustic Propagation and Problem Formulation

Multi-channel speech enhancement exploits arrays of microphones to create spatially selective filters that can focus on a target direction-of-arrival (DoA), attenuate both diffuse and directional interference, and reduce reverberation.

A common processing framework is the STFT domain, where spatial filtering (beamforming) is combined with statistical post-processing, such as Wiener or MMSE estimators, or mask-based weighting strategies [8, 11, 12, 19].

Let $x_m(t)$ denote the signal captured by the m-th microphone at time t, and let M represent the number of microphones. In the STFT domain, the multichannel observation vector is modeled as

$$\mathbf{X}(t,f) = \mathbf{D}(f)S(t,f) + \mathbf{N}(t,f), \tag{5}$$

where $S(t,f)$ is the target speech STFT, $\mathbf{D}(f) = [D_1(f),\ldots,D_M(f)]^T$ is the steering vector that captures the acoustic transfer from the source to each microphone, and $\mathbf{N}(t,f) = [N_1(t,f),\ldots,N_M(t,f)]^T$ represents the additive noise and interference.

A linear spatial filter $\mathbf{W}(f) \in \mathbb{C}^M$ then produces an enhanced output

$$Y(t,f) = \mathbf{W}^H(f)\mathbf{X}(t,f), \tag{6}$$

where $(\cdot)^H$ denotes Hermitian transpose. This formulation provides the foundation for a wide range of spatial filtering strategies, from classical beamforming to modern neural approaches.

Working in the STFT domain provides several practical benefits. It enables frequency-dependent beamforming, where filters $\mathbf{W}(f)$ are optimized for each frequency bin to maximize the target-to-noise ratio. It also facilitates statistical estimation methods, such as the Wiener filter or the minimum mean-square error (MMSE) estimator, which rely on estimating the power spectral densities of speech and noise [8, 20]. Additionally, the STFT domain allows seamless integration with modern deep learning approaches that operate on magnitude or log-magnitude spectrograms, often combined with phase reconstruction techniques to recover high-quality time-domain signals [21, 22].

2.3 Traditional Linear Spatial Filtering

Traditional linear spatial filters, also known as beamformers, linearly combine microphone signals to emphasize a target source while attenuating unwanted components. These methods rely on geometric or statistical assumptions and provide interpretable solutions with low complexity. Three representative approaches are described below.

2.3.1 Delay-and-Sum Beamformer

The delay-and-sum (DS) beamformer is the most fundamental spatial filter. It addresses time differences of arrival (TDOAs) by applying suitable delays or phase shifts to each microphone signal before coherent summation. Assuming the target DoA is known, the DS beamformer is easy to implement and incurs very low latency, making it widely used in acoustics and telephony [3].

The enhanced signal is obtained by

$$y(t) = \frac{1}{M} \sum_{m=1}^{M} x_m\big(t - \delta_m\big) \tag{7}$$

in time domain and

$$Y(t,f) = \frac{1}{M} \sum_{m=1}^{M} e^{-j2\pi f \delta_m} X_m(t,f) \tag{8}$$

in frequency domain, where δ_m compensates for the propagation delay relative to the target DoA. While simple and robust, its performance degrades in reverberant environments or with inaccurate DoA estimation.

2.3.2 Filter-and-Sum Beamformer

The filter-and-sum (FS) beamformer generalizes the DS approach by assigning a frequency- or time-dependent filter to each microphone input. These filters can be optimized to shape the beam pattern or to suppress noise more effectively than equal weighting [3].

In the frequency domain, the enhanced signal is given by

$$Y(t,f) = \sum_{m=1}^{M} W_m(f) X_m(t,f), \tag{9}$$

where $Y(t,f)$ is the enhanced signal in the time-frequency domain, corresponding to time frame t and frequency f, $W_m(F)$ denotes the complex frequency-dependent filter weight for microphone m, and $X_m(t,f)$ represents the short-time Fourier transform (STFT) of the signal from microphone m. Furthermore, M denotes the total number of microphones in the array, and the summation $\sum_{m=1}^{M}$ indicates the aggregation across all microphone channels. This added flexibility improves robustness over DS but requires more computational effort and design considerations.

2.3.3 MVDR (Minimum Variance Distortionless Response)

The MVDR beamformer represents a more advanced adaptive approach. It minimizes output power while maintaining a distortionless response for the target source. Formally, the MVDR filter $W(f)$ solves

$$\min_{W(f)} W(f)^{\mathsf{H}} \Phi_v(f) W(f) \quad \text{s. t.} \quad W(f)^{\mathsf{H}} D_q(f) = 1, \tag{10}$$

where $\Phi_v(f)$ is the spatial covariance of interference and $D_q(f)$ is the steering vector. The closed-form solution is

$$W(f) = \frac{\Phi_v(f)^{-1} D_q(f)}{D_q(f)^{\mathsf{H}} \Phi_v(f)^{-1} D_q(f)}. \tag{11}$$

Compared with DS and FS, MVDR provides stronger interference suppression and adaptive null steering. However, it depends critically on accurate covariance estimation and can be sensitive to reverberation [6, 23].

2.4 Deep Learning for Spatial Filtering

While classical beamformers rely on analytic design or statistical estimation, deep learning has introduced data-driven approaches that can either complement traditional methods or replace them entirely. Neural architectures have been integrated to estimate masks, predict spatial covariance matrices, or directly learn beamforming filters.

2.4.1 Mask-Based Neural Beamforming

Mask-based neural beamforming combines machine learning with classical signal processing. A neural network predicts time-frequency masks that indicate speech-dominant regions. These masks are used to compute spatial covariance matrices (SCMs) for speech and noise, which are then passed into conventional beamformers such as MVDR [24, 25, 26]. This hybrid strategy leverages the strengths of both domains, providing robust data-driven mask estimation along with the well-established statistical foundations of beamforming.

2.4.2 End-to-End Neural Beamforming

In contrast, end-to-end neural beamforming eliminates the explicit beamforming stage and instead learns the entire enhancement process within a neural network. These models can operate either in the time-frequency or time domain, and are capable of learning complex, adaptive filters that approximate or even surpass the performance of conventional beamformers. Approaches in this domain include differentiable all-neural beamforming in the STFT domain, where spatial filtering is jointly optimized with mask estimation [27]; time-domain architectures that learn response-invariant filters directly from raw waveforms, using directional cues as conditioning signals [27]; methods that co-optimize the front-end filterbank and beamforming components, enabling joint learning of feature representations and spatial selectivity [28]; and neural beamformers trained with specialized loss functions that incorporate array response constraints, facilitating simultaneous speech enhancement and source localization [29]. These end-to-end frameworks have demonstrated strong performance in real-world scenarios and pave the way for unified, fully data-driven enhancement systems.

2.5 Hearing Aids

Hearing aids represent a critical application domain for speech enhancement. They are medical devices designed to compensate for hearing loss by amplifying and clarifying sounds in everyday listening environments. For users, the primary goal is improved communication and quality of life, including social participation, reduced cognitive decline risk, and relief from tinnitus symptoms [2].

Despite these benefits, adoption remains limited due to stigma, unrealistic expectations, and incomplete awareness of device capabilities. From a technical perspective, hearing aids must achieve robust noise reduction and speech preservation under strict constraints on latency, power consumption, and form factor.

Traditional methods such as beamforming [3] and Wiener filtering [4] form the foundation of current devices. However, real-world listening conditions characterized by reverberation, multiple interferers, and non-stationary noise often exceed the limits of these linear models. This motivates more advanced approaches, including nonlinear and neural methods such as Joint Non-linear Filtering (JNF) [6], which will be discussed in detail in Sections 5 and 8.

3 Related Work

Tesch et al. demonstrated that nonlinear spatial filtering can significantly enhance spatial processing capabilities, achieving near-complete suppression of five Gaussian interfering point sources using only two microphones [30]. Building on this, their subsequent work [6] presents a state-of-the-art (SOTA) approach that effectively integrates spatial and tempo-spectral information using a relatively simple neural network architecture utilizing two LSTM layers and a single feed-forward layer. Despite its simplicity, this model [7] outperforms more complex architectures in both speaker extraction and multi-channel speech enhancement tasks.

3.1 Dataset

The dataset used in [6] comprises of 6,000 training, 1,000 validation, and 600 test samples, sampled at 16 kHz, with clean speech drawn from the WSJ0 corpus [31]. In [6], noisy mixtures are generated using *pyroomacoustics* [32] to simulate multi-speaker environments, where the nearest speaker is treated as the target and others as interference. The signal-to-noise ratio (SNR) is not directly controlled but emerges naturally from the simulation setup, which varies the distances between sound sources and the microphone array. The average SNR is -4 dB, with 95% of the samples falling within the range of -9 dB to 2 dB [6].

In [6], they employ a circular microphone array with a 10 cm diameter, configured with between two and five channels. The array is positioned randomly in the xy-plane, at a height of 1.5 m and at least 1 m away from any walls. The target speaker is placed at a fixed orientation relative to the array, with a distance to the microphone ranging from 0.3 m to 1 m. The five interfering sound sources are located within the designated gray area, each at a minimum distance of 1 m from the microphone array as shown in figure 3.1

This method of data processing highlights a key distinction between speaker extraction, as conducted in [6], and the speech enhancement objective pursued in this work. While speaker extraction focuses on isolating a specific speaker based on identity or spatial cues, our goal is to improve overall speech quality and intelligibility in noisy environments. The dataset used in our experiments is described in detail in Section 6.1. Further information on the data preprocessing pipeline is provided in Section 6.2.

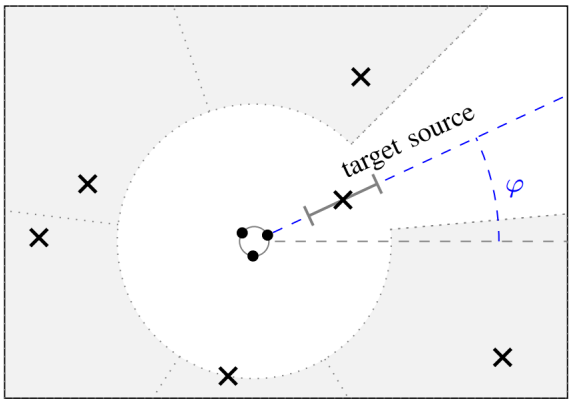

Figure 3.1: Pyroomacoustic simulation setup used in [6]. A circular microphone array with a diameter of 10 cm is placed randomly in the xy-plane at a height of 1.5 m and at least 1 m from any wall. Depending on the configuration, the array consists of two to five microphones. The target speaker is located at a fixed orientation relative to the array, at a distance between 0.3 m and 1 m. Five interfering sources are positioned within the gray region shown, each at least 1 m from the array. This setup is used to generate realistic multichannel noisy mixtures for evaluating speech enhancement models.

3.2 Model Architecture

In [6], they propose several alternative network architectures to facilitate a comprehensive analysis. Their base architecture is illustrated in Figure 3.2. As shown in the lower portion of the figure, the network comprises of just three layers: two bidirectional long short-term memory (LSTM) layers, followed by a

feed-forward (FF) layer. LSTM layers [33] are widely used for modeling sequential data due to their ability to capture temporal dependencies.

As shown in Figure 3.2, the input features represent channel information that are structured along two sequence dimensions. The sequence dimension is determined by secondary data representing frequency in wide-band representations and time in narrow-band representations. Furthermore, their proposed model [6] combines all three sources of information combining spatial information with temporal and spectral information. The architecture depicted in Figure 3.2 is adapted from the work of Li and Horaud [34]. However, in their implementation, only temporal features were utilized, despite employing a similar network structure. To maintain result comparability, the base architecture and total number of parameters are kept unchanged. Instead, they modify the data arrangement at the location indicated by the circled "2". In the FT-JNF model [7], wide-band data is first fed into the initial LSTM layer. The resulting features are then restructured into a narrow-band format before being passed to the second LSTM layer. This design enables FT-JNF to effectively leverage all three sources of information spatial, spectral, and temporal within a consistent architectural framework.

This model architecture highlights a key distinction between offline processing, as used in [6], and the real-time objective pursued in this work. In the previous study [6], the deep learning model is applied to speaker extraction in an offline setting, where the entire audio signal is available for processing at the same time. In contrast, our goal is to develop an architecture capable of operating in real time. The model must support online, frame-by-frame inference with minimal latency. It is designed for causal processing, making it suitable for live applications. The detailed architecture is described in Section 6.3. Additionally, the loss function used during training is outlined in Section 6.4.

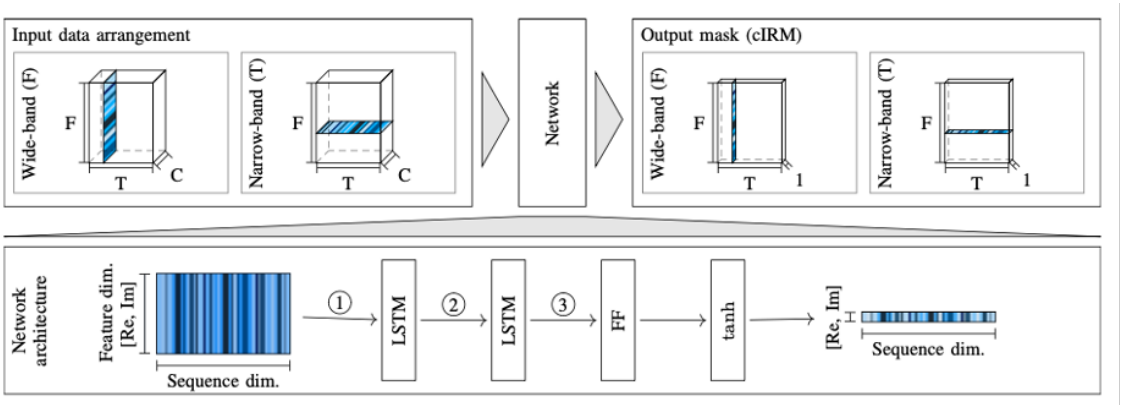

Figure 3.2: System architecture of the FT-JNF model adapted from [6]. The input features encode multi-channel information and are organized along two sequence dimensions: frequency (wide-band representation) and time (narrow-band representation). In contrast to the original work of Li and Horaud [34], which only exploited temporal features, the FT-JNF design restructures the data flow (as indicated at the circled "2") so that wide-band input is first processed by the initial LSTM layer, then reformatted into narrow-band features before being passed to the second LSTM layer. This restructuring enables the model to jointly capture spatial, spectral, and temporal information within the same architecture, while keeping the overall parameter count comparable to baseline designs.

3.3 Loss Function and Evaluation Metrics

In the baseline work by [6], the training objective is based on a time-frequency domain loss function originally proposed by Tolooshams et al. [35]. This loss jointly optimizes the model in both the time and frequency domains to improve reconstruction fidelity across different signal representations. The loss is defined as

$$l = \sum_{n=1}^{N} \left(\alpha \|s_n - \hat{s}_n\|_1 + \|S_n - \hat{S}_n\|_1 \right), \tag{12}$$

where s_n and $\hat{s}_n$ denote the true and estimated speech signals in the time domain for the n-th audio sample, respectively, and S_n and $\hat{S}_n$ represent the corresponding true and estimated speech signals in the frequency

domain. The term $\|\cdot\|_1$ denotes the ℓ_1 norm, computed as the sum of absolute differences between the true and estimated values across all samples. The scalar weighting factor α balances the contribution of the time-domain and frequency-domain loss components, and N is the number of audio samples in the batch (or dataset), ensuring the loss is aggregated over all available instances.

In [6], the value of α is set to 10 to ensure that the time-domain and frequency-domain components contribute similarly to the overall loss, compensating for differences in scale between the two domains. This combined loss encourages the model to preserve both time-frame and spectral structure, which are critical for perceptually plausible speech enhancement.

To evaluate the effectiveness of the trained models, [6] two widely adopted objective metrics are employed: **Perceptual Objective Listening Quality Analysis (POLQA)** and **Extended Short-Time Objective Intelligibility (ESTOI)**. These metrics provide complementary assessments POLQA focuses on perceived speech quality, while ESTOI emphasizes intelligibility both of which are essential for real-world applications.

POLQA (ITU-T P.863) [36] is an advanced standard for predicting subjective speech quality. It improves upon the earlier PESQ metric by supporting wideband and super-wideband signals and accounting for time-varying distortions such as time scaling and spectral smearing. By aligning the reference and processed signals in both time and frequency, POLQA computes a perceptual degradation score that maps to a Mean Opinion Score (MOS) between 1 (bad) and 5 (excellent), closely approximating human listening test results.

ESTOI [37], on the other hand, is designed to predict speech intelligibility in noisy and enhanced conditions. It operates by comparing the temporal envelopes of the clean and enhanced signals across 15 critical frequency bands and short time intervals (30–480 ms). The resulting correlation-based score ranges from 0 (completely unintelligible) to 1 (perfectly intelligible) and has been shown to correlate strongly with human intelligibility scores, particularly in noise suppression and hearing assistance systems.

In [6], the FT-JNF model [7] achieves state-of-the-art performance, consistently outperforming all baseline methods by at least **0.22 in POLQA score** and **0.04 in ESTOI score**. These improvements highlight the effectiveness of the joint time-frequency loss in producing enhanced speech that is both perceptually natural and intelligible. The success of this approach serves as a strong foundation for our work, which will be discussed in Section 6.

4 Problem Formulation

The primary purpose of this thesis is to research, design, and implement a multichannel speech enhancement system capable of real-time noise reduction. The system is specifically targeted for integration into modern hearing aids, which require robust performance in complex and noisy acoustic environments. The goal is to improve speech clarity and intelligibility for hearing aid users, while effectively suppressing background noise.

Hearing aids operate in real-time and are constrained by size, power consumption, and processing capacity. Therefore, this research will explore the challenges of implementing advanced light noise reduction algorithms using frame-wise (i.e., streaming) processing rather than full-audio batch inference. The system must operate with low latency and high efficiency, while also adapting to the non-stationary and multi-source nature of real-world acoustic scenes.

This thesis seeks to answer the central research question:

"How can a multichannel speech enhancement algorithm be effectively designed and implemented to perform real-time noise reduction?"

From this core question, two sub-questions arise:

- **Q.1:** What are the most effective strategies for reducing noise from multichannel input?
- **Q.2:** How can speech enhancement models be adapted or restructured to operate on a frame-wise basis suitable for real-time applications like hearing aids?

The motivation for this research lies in the unmet needs of hearing aid users. Hearing loss, particularly sensorineural hearing loss, is a chronic condition that limits communication ability and can contribute to social isolation and cognitive decline. While hearing aids are the primary rehabilitative solution, adoption remains low due in part to limitations in real-world performance. One key challenge is speech perception in noisy environments, where traditional denoising algorithms often fail to maintain intelligibility and spatial awareness.

Modern digital hearing aids rely on real-time digital signal processing (DSP) to perform noise reduction, directional filtering, and adaptive gain control. However, existing linear models like beamformers and Wiener filters are insufficient in dynamic, reverberant conditions with multiple competing sound sources. This thesis aims to explore advanced, potentially nonlinear multichannel algorithms that can enhance speech intelligibility in such environments, particularly under real-time constraints.

To address the research questions, the following objectives are established:

1. Research and implement an algorithm capable of leveraging spatial information from multiple microphones to suppress background noise effectively in hearing aid scenarios.
2. Implement real-time frame-wise processing to simulate real-time behavior suitable for hearing aid hardware constraints.
3. Compare the proposed system's performance against the baseline method in terms of speech intelligibility, and signal to distortion ratio.

By achieving these objectives, the thesis aims to contribute to the field of hearing aid signal processing with a practical and scalable speech enhancement solution that balances denoising performance with real-time constraints. The expected outcome is a prototype real-time model capable of improving communication quality for individuals with hearing loss in realistic, noisy environments.

5 Methodology

This section outlines the scientific methods used to achieve the research objectives presented in Section 4. Each method has been carefully selected based on its appropriateness for addressing the specific research questions and the practical implementation challenges associated with real-time multichannel speech enhancement in hearing aid applications. This methodology is particularly well-suited for this work, as it combines iterative algorithmic development with empirical validation, ensuring that the proposed systems are not only theoretically sound but also functionally effective in realistic scenarios.

The research process integrates three key components: (1) algorithmic development, guided by models from statistical signal processing and deep learning; (2) real-time implementation strategies, including researching and developing the model to run in frame-wise operation; and (3) comprehensive performance evaluation using objective metrics. These stages follow a cyclical and iterative research paradigm, allowing for continuous refinement based on empirical feedback consistent with best practices in audio signal processing and human-centered technology development [13, 38].

By grounding the methodology in established theoretical frameworks and validated experimental protocols, this work ensures scientific rigor while maintaining a strong focus on practical applicability essential for advancing hearing assistive technologies.

Goal 1: Research and Develop a Multichannel Noise Reduction Algorithm

To design a robust noise reduction algorithm that leverages spatial information from multiple microphones, we adopted a LSTM-based model for the signal processing approach. Specifically, we investigated and modified Joint Non-linear Filtering (JNF) [6, 7] for our use case. These methods are well-suited for hearing aid applications because they utilize spatial cues to suppress directional noise sources without degrading the target speech signal. Furthermore, JNF model was chose because of the simplicity.

The development process began with the creation of a multichannel noisy dataset by joining both clean speech and environmental noise. After the data processing, we trained the model with train and validation sets, where the split of the data was 80% and 10% respectively, to ensure the model is trained and validated properly during the batch training. We also kept 10% of the data for testing to ensure the model robustness.

Training was then conducted with careful tuning of both the model architecture and the training parameters. To meet real-world deployment requirements, especially real-time operation, the model is designed to operate without input normalization. Additionally, it does not depend on future time steps (i.e., it is non-bidirectional), ensuring causal inference. In addition, various configurations were tested, including different batch sizes and training data volumes, to identify the optimal setup for performance and stability. These configurations are detailed in Section 8.

Model optimization was guided by a multi-metric evaluation framework, assessing performance across several key indicators. The choice of metrics and their role in evaluating model effectiveness are further elaborated in Section 6.5. The chosen methods are grounded in spatial signal processing theory and provide a strong foundation for enhancing speech intelligibility in realistic speaker extraction task.

Goal 2: Implement Real-Time Frame-Wise Processing

To transition from offline to real-time processing, we implemented a frame-wise system architecture. This involves processing the input signal in short, overlapping frames based on the hop variables using a short-time Fourier transform (STFT), followed by spectral enhancement and resynthesis using the inverse STFT. This block-based method reflects how real-time hearing aid processors typically function.

The implementation was done using Python and PyTorch for prototyping, with the potential for future migration to C++ or embedded frameworks for hardware deployment. Frame-wise inference was chosen over full-utterance processing because hearing aids require continuous processing with strict latency constraints. Batch methods are unsuitable in such contexts due to their reliance on future context, which is unavailable in real-time.

Following the completion of training and validation, the model was rigorously evaluated under real-time inference conditions to align with the core objectives of this thesis. A comprehensive testing framework was developed to assess performance across both offline and causal, frame-wise inference modes, enabling a direct comparison between idealized and practical deployment scenarios. The offline benchmark processes the entire audio signal at once, assuming full access to whole samples, thereby serving as an upper

performance bound. In contrast, several causal inference strategies were implemented to simulate real-time operation, each designed to balance temporal context, computational efficiency, and memory constraints. These include frame-by-frame processing, which minimizes latency but lacks sufficient context; the additional frame method, which incrementally builds input history to improve contextual modeling at the cost of growing memory usage; sliding window processing with zero-padding, which maintains fixed input size but risks performance degradation due to artificial padded signal in early frames; and a hybrid approach that combines the strengths of both initially accumulating context and then switching to a sliding window once the full input length is reached. This final strategy ensures stable, low-latency inference with bounded memory consumption, making it particularly suitable for long-duration, real-time deployment on resource-constrained platforms. These methods are detailed in Section 8.5.

Goal 3: Evaluate Performance

To assess the effectiveness of the proposed speech enhancement system, we conducted objective evaluations using a set of well-established metrics. These metrics provide quantitative performance measures and have been shown to correlate more reliably with human perception than traditional signal-to-noise ratio (SNR), particularly in the context of speech separation and enhancement. Specifically, we employed the Perceptual Evaluation of Speech Quality (PESQ) [39], Signal-to-Distortion Ratio (SDR) [40], Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) [41], and Short-Time Objective Intelligibility (STOI) [42], all of which are widely used benchmarks in the speech processing community. These metrics are further explained in Section 6.5.

Comparisons were made against baseline method, the Joint Non-linear Filtering (JNF) model proposed by Tesch et al. [6], to demonstrate the relative improvement achieved by the proposed approach. In addition to offline batch evaluation, we performed frame-wise real-time testing using a low-latency inference pipeline that emulates continuous audio input and output. This setup allows us to validate system behavior under live streaming conditions, including latency, stability, and memory usage.

6 Description of work

This section outlines the materials and methodologies used to achieve the objectives of this thesis. The selection of datasets, model design choices, and evaluation strategies were guided by both qualitative and quantitative approaches, tailored to the specific goals of each stage.

Model development was primarily informed by qualitative analysis, involving the comparison of various architectural and preprocessing options, with careful attention to their practical limitations and implementation challenges. In contrast, the evaluation phase relied on quantitative methods, using objective metrics and numerical results to systematically assess and compare model performance.

Although this work is using the architecture and methodology presented in [6, 7], it introduces several key advancements. These include modifications to the dataset, preprocessing pipeline, model structure, and training objectives collectively reflecting a shift from speaker separation in simulated environments to real-time speech enhancement under realistic, real-world acoustic conditions.

6.1 Dataset

The dataset used in [6] comprises 3 sec audio from 6,000 training, 1,000 validation, and 600 test samples, with the total of 6 hours 20 minutes, sampled at 16 kHz, with clean speech drawn from the WSJ0 corpus [31]. The data is partitioned into 80% training, 10% validation, and 10% testing subsets. In contrast, our dataset consists of approximately 18,000 samples of length of 10s (total duration of 50 hours) at a higher sampling rate of 25 kHz. Crucially, we augment clean speech with 400 real-world noise recordings that are played back and captured in a laboratory setting resembling the Multichannel anechoic chamber Wilska in Aalto University [43] using the Compass plug-in [44]. The noise scenarios include for example offices, streets, and public transport, enhancing ecological validity and generalization to real-life conditions.

6.2 Dataset Preprocessing

In [6], noisy mixtures are generated using *pyroomacoustics* [32] to simulate multi-speaker environments, where the nearest speaker is treated as the target and others as interference. Our approach diverges significantly: instead of simulated interference, we mix clean speech with real-world noise recordings at varying signal-to-noise ratios (SNRs). This is achieved through controlled amplification of noise, producing more authentic and perceptually relevant noisy conditions. This shift eliminates the domain gap between training and deployment, improving robustness in practical applications.

6.3 Model Architecture

Although our model retains the core structure of two recurrent layers followed by a feed-forward layer in [6] (see Figure 3.2), we employ a *unidirectional* LSTM, in contrast to the *bidirectional* LSTM used in [6]. This architectural choice is essential for real-time operation, as bidirectional models require future input frames which is unavailable in live processing. Our unidirectional design ensures causality, enabling frame-wise inference that processes audio in real time. This makes the model deployable in latency-sensitive applications such as hearing aids or voice communication systems.

6.4 Loss Function

The original work [6] employs a composite loss combining time-domain and frequency-domain l (Mean Absolute Error) losses (See Eq. 12). In this thesis, we extend the loss function by incorporating the Perceptual Evaluation of Speech Quality (PESQ) [39] as an additional training objective. PESQ correlates strongly with human perception of speech quality, guiding the model to preserve naturalness and intelligibility after noise reduction. This addition is particularly beneficial under real-world noise conditions, where purely signal-metric losses may produce perceptually unnatural outputs, usually called as artifcats. Mathematically, our loss function (L) could be written as

$$L = \sum_{n=1}^{N} (\beta l_n + P_n), \tag{13}$$

where l_n represents the time- and frequency-domain loss component for the n -th audio sample as detailed in Eq. 12, P_n denotes the PESQ-based loss for the n -th audio sample that provides a perceptually motivated measure of speech quality, and β is a scalar weighting factor used to balance the contributions of the l and P loss terms. The summation is performed over N audio samples in the batch (or dataset), ensuring that the total loss reflects the model's performance across all instances in the training set.

6.5 Evaluation

The primary objective of model training is to iteratively improve the accuracy of predictions. To support this goal, a set of quantitative evaluation metrics is established to assess the model's performance. These metrics serve as a critical guide during development, enabling the systematic comparison of different model configurations and providing insight into whether modifications lead to meaningful improvements.

6.5.1 Perceptual Evaluation of Speech Quality (PESQ)

The Perceptual Evaluation of Speech Quality (PESQ), standardized as ITU-T P.862 [39], is an objective metric designed to predict the perceived quality of speech signals as judged by human listeners. It functions by aligning the enhanced (processed) signal with the clean reference signal in both time and frequency domains, accounting for delays and distortions introduced during processing. A mean opinion score (MOS)-like value is then computed, ranging from 1 (bad) to 4.5 (excellent), reflecting the overall perceptual quality.

PESQ effectively captures impairments caused by noise, filtering, compression, and other signal degradations, making it a widely used benchmark in speech enhancement and telecommunication applications. Its strong correlation with subjective listening tests makes it particularly valuable for evaluating systems intended for real-world deployment, such as hearing aids.

6.5.2 Signal-to-Distortion Ratio (SDR)

The Signal-to-Distortion Ratio (SDR) provides a global measure of the fidelity of the reconstructed speech signal [40]. It is defined as the ratio between the power of the target (clean) speech component and the total power of all distortions, including residual noise, interfering sources, and processing artifacts. A higher SDR value indicates better reconstruction quality and less signal degradation.

SDR is commonly used in audio source separation and speech enhancement benchmarks due to its ability to reflect overall signal integrity. However, it is sensitive to absolute scaling and time alignment, which can affect its reliability in certain scenarios, particularly when post-processing gain is applied.

6.5.3 Scale-Invariant Signal-to-Distortion Ratio (SI-SDR)

To address the limitations of SDR with respect to amplitude scaling, the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) was adopted [41]. SI-SDR removes dependency on absolute signal magnitude by projecting the estimated signal onto the direction of the reference signal before computing the error. This makes it invariant to linear scaling, which is especially important in hearing aid applications where automatic gain control (AGC) or user-specific volume adjustments are common.

Mathematically, the scale-invariant signal-to-distortion ratio (SI-SDR) is defined as

$$\text{SI-SDR} = 10\log_{10}\left(\frac{\|\theta s\|^2}{\|\hat{s}-\theta s\|^2}\right), \tag{14}$$

where s is the target speech signal, $\hat{s}$ is the enhanced estimate, and

$$\theta = \frac{\langle \hat{s}, s\rangle}{\|s\|^2} \tag{15}$$

is the optimal scaling factor. Here, $\langle\cdot,\cdot\rangle$ denotes the inner product between two signals, while $\|\cdot\|$ denotes the ℓ_2 norm (i.e., the Euclidean length of a vector). By focusing on the waveform shape rather than its absolute amplitude, SI-SDR provides a more robust and perceptually relevant assessment of enhancement performance.

6.5.4 Short-Time Objective Intelligibility (STOI)

The Short-Time Objective Intelligibility (STOI) metric is used to predict the speech intelligibility of enhanced signals [42]. It operates in the short-time Fourier transform (STFT) domain, computing the temporal envelope correlation between the clean and processed speech signals across multiple frequency bands (typically 150–4000 Hz), which are critical for human speech understanding.

STOI scores range from 0 to 1, with values closer to 1 indicating higher intelligibility. The metric has been shown to correlate strongly with word recognition rates in noisy environments, making it highly relevant for hearing assistance devices where preserving speech clarity is paramount. While STOI does not capture perceptual quality directly, it serves as a crucial indicator of functional performance in real-world listening conditions.

6.6 Real-Time Applicability

While [6] primarily addresses *speaker separation* in simulated environments, our work targets *speech enhancement* under real-world acoustic conditions. This change in objective directly influences both data design and model development. A major advantage of our approach is its compatibility with real-time inference. By ensuring causal processing through a unidirectional architecture, the model relies only on past and present frames, making it suitable for hearing aid devices.

To achieve this, we employ a causal, unidirectional architecture that processes audio using only past and present input frames, thereby ensuring temporal causality. Furthermore, we introduce an enhanced frame-wise processing strategy that maintains continuity of input frames while minimizing processing latency. These frameworks are detailed in Section 8.5 and Section 9.4. They enable realistic evaluation of real-time performance and support future deployment.

7 Ethical Considerations

The development of this thesis involved the use of audio data, primarily consisting of speech recordings and real-world background noise, which raised important ethical considerations regarding data privacy and confidentiality [45]. The speech data used in this work is based on the AISHELL-1 corpus [46], an open-source Mandarin speech dataset recorded by 400 speakers from various accent regions in China [46]. The original recordings were collected in a quiet indoor environment. While the dataset is publicly available for academic use under the Apache License 2.0, the version employed in this study was re-recorded in a laboratory setting through playback and re-capture, simulating realistic acoustic conditions. This processed data is treated with sensitivity due to its potential to preserve speaker-specific characteristics, despite being derived from a public source. To ensure ethical compliance, access to the dataset was strictly limited to authorized personnel within Huawei's secure network infrastructure. The data is stored and processed only on secured, company-controlled devices, with robust cybersecurity measures in place to protect individual privacy.

Similarly, the outcomes of this research including the developed model and associated results are treated as confidential and will be accessible solely to authorized parties within Huawei. This ensures the responsible development and controlled use of the technology, minimizing the risk of unauthorized or unintended applications. By maintaining strict data governance, the project adheres to high standards of ethical research practice.

This work has meaningful societal implications, particularly in the area of speech enhancement for noise reduction. By improving the clarity of speech in noisy environments, the model can contribute to enhanced hearing assistance features in Huawei's audio devices. This aligns with the United Nations Sustainable Development Goals (SDGs) [47], specifically **Goal 3: Good Health and Well-being** and **Goal 10: Reduced Inequalities**, by supporting accessible and inclusive communication technologies for individuals with hearing difficulties or those in underserved communities.

8 Experiments

This section outlines the experimental procedure in a step-by-step manner, detailing the key decisions made throughout its development. It covers the setup of the programming environment, the data engineering tasks required to prepare the input data, the choices involved in model training, and the evaluation methodology based on selected performance metrics.

8.1 Chosen Language and Libraries

As in the previous work [7], Python was selected as the main programming language for conducting the experiments in this thesis. This choice was motivated by Python's extensive ecosystem of libraries, particularly in the domain of machine learning especially in speech and signal processing. Its widespread adoption in both academic research and industrial applications enhances the relevance and reproducibility of the results across different contexts.

Key libraries such as `Pandas` and `NumPy` were employed for data manipulation, preprocessing, and result analysis, providing robust tools for handling structured data efficiently. On the other hand `Librosa` was used for the speech input and output, while `torchmetrics.audio` was the primary library for the model evaluation functions.

The Joint Non-linear Filter model [7] was implemented using `PyTorch Lightning`, a specialized library built on top of the PyTorch framework. This library offers high-level abstractions tailored for simplifying the implementation of neural network. By leveraging PyTorch's flexibility and computational efficiency, `PyTorch Lightning` facilitates rapid prototyping, training, and evaluation of deep learning models for audio data.

8.2 System Preparation

To support the development, testing, and refinement of the models, a controlled and reproducible Python environment was established. This environment ensures consistency across experimental runs, enabling reliable version control of both dependencies and source code.

To maintain compatibility and reproducibility, certain libraries were pinned to specific versions, avoiding potential issues arising from breaking changes in newer releases. Python 3.12.7 and PyTorch 2.6.0 were used in this work similarly to the previous work [7].

Given the scale of the dataset and the computational demands of training deep learning models, high-performance computing resources were required. To this end, Huawei provided the advanced computing infrastructure for the research and development purpose.

8.3 Data Engineering

Data engineering plays a crucial role in the development of any machine learning model, and its importance cannot be overstated in the context of this thesis. The quality, structure, and relevance of the input data directly influence the performance and generalization capability of the model. This section describes the comprehensive process of preparing the raw dataset for use with the Joint Non-linear Filter (JNF) model [7].

The data engineering pipeline is structured into two main stages: *Data Cleaning*, and *Feature Preprocessing*. Each stage is designed to ensure that the final dataset is accurate, meaningful, and optimally formatted for training and evaluation.

For all data manipulation tasks, the `librosa` library was used extensively, leveraging its powerful `stft` features for efficient handling of audio-type data. These tools enabled streamlined operations such as filtering, transformation, aggregation, and formatting, which are essential when working with large-scale datasets. Proper organization of the data into a model-ready format ensures that the JNF receives high-quality, temporal spectral features to achieving reliable and accurate predictions.

8.3.1 Data Cleaning

Data cleaning is the foundational step in any data preprocessing pipeline. It involves inspecting and validating the dataset to ensure data integrity, consistency, and completeness before proceeding to more advanced

processing stages. This step is essential to prevent biases, errors, or inefficiencies from propagating into the model training phase.

The step in this process was to assess the completeness of each speech audio and noise audio respectively. Missing values can significantly impair model performance by introducing uncertainty or distorting statistical relationships. A detailed examination of the dataset identified that 7.31% of the speech samples (1,309 data points) were not containing any speech and were subsequently removed before further processing.

8.3.2 Preprocessing

In the context of speech enhancement, preprocessing plays a pivotal role in transforming raw audio signals into representations that effectively expose the underlying structure of speech and noise components. Since deep learning models operate on numerical inputs, converting time-domain audio waveforms into informative spectral features is essential for enabling the model to distinguish between target speech and background interference.

Similarly to the previous work [7], the Short-Time Fourier Transform (STFT) was employed as the preprocessing step. The STFT decomposes the input audio signal into its time-frequency components by applying a sliding window across the waveform and computing the Fourier transform within each window (See Eq. 2). This yields a complex-valued spectrogram that captures both the magnitude and phase of frequency content over time, providing a rich representation well-suited for modeling non-stationary signals like speech.

The raw audio recordings were first segmented into overlapping frames using a Hann window of 240 samples with a hop size of 120 samples, striking a balance between temporal resolution and spectral smoothness. The resulting STFT spectra were then used to compute the magnitude spectrogram, which serves as the main input to the model. In some experiments, the logarithmic scale was applied to compress the dynamic range and better align with human perceptual sensitivity to loudness.

Additionally, the phase information extracted from the STFT was preserved for use during the signal reconstruction stage, where the inverse STFT (iSTFT) is applied to convert enhanced spectral features back into the time domain. By leveraging the STFT as the foundation of the feature engineering pipeline [7], this approach ensures that the model receives a detailed and interpretable representation of the speech signal in both time and frequency, which is crucial for effective noise suppression and speech fidelity preservation.

8.3.3 Dataset Preparation and Data Loader Construction

After completing the preprocessing, the next essential step is to partition the dataset into distinct subsets for training, validation, and testing. This three-fold split ensures a rigorous evaluation framework: the training set is used to optimize model parameters, the validation set supports hyperparameter tuning and early stopping during training, and the test set provides an unbiased assessment of the final model's generalization performance on completely unseen data.

The dataset was divided into three non-overlapping subsets using a index-based split to prevent data leakage. Specifically, the majority of the data (approximately 80%) was allocated to the training set, while 10% was reserved for validation and the remaining 10% for testing. This distribution ensures sufficient data for model learning while maintaining representative and independent evaluation sets.

To guarantee reproducibility across experiments, a fixed random seed was set before splitting the data. Although the split is deterministic, it was performed in a specific order such that the future changes could not unintentionally influence the model performance. Following the split, dedicated `DataLoader` objects were created for each subset using PyTorch's data loading module [7]. These data loaders are a core component of the training pipeline, responsible for efficient batch generation, parallel data loading, and memory management. The `training data loader` was configured to shuffle the samples at the beginning of each epoch to prevent the model from learning artificial patterns based on input order, thereby improving generalization and reducing overfitting. The `validation and test data loaders`, on the other hand, were set up without shuffling to ensure consistent and repeatable evaluation results across epochs and experiments.

By structuring the data pipeline in this way, the model receives well-organized, efficiently loaded spectral inputs throughout training and evaluation. This setup not only supports robust optimization but also enables a fair and comprehensive assessment of the model's performance in realistic speech enhancement scenarios.

8.4 Model Training and Tuning

After completing the data engineering pipeline for the full dataset, the next phase involved constructing and preparing the Joint Non-linear Filter (JNF) model for training. To streamline development and enable flexible customization, the `PyTorch Lightning` library was selected as the primary framework [7]. This library offers a robust `Trainer` object that automates and accelerates the training workflow. Additionally, it integrates well with the `DataLoader` for the Pytorch's data loading module, that separates the data clearly between training, validation and testing.

The JNF model [6, 7] was built using PyTorch's built-in neural network implementation. The model was customized by modifying the loss function, as described in [35]. We began with a set of initial hyperparameters, including the choice of loss function and a bidirectional model configuration. Additionally, several data preprocessing settings were established, such as data simulation methods, normalization techniques, and specifications for data size and sequence length. These parameters were subsequently refined through iterative experimentation to better align with the temporal structure and complexity of the speech enhancement task.

With the model configured, a `Trainer` object was set up to manage the training process [7]. This component is central to the training pipeline, handling core operations such as epoch iteration, batch processing, GPU acceleration, gradient computation, and checkpointing. It also ensures efficient data loading via integration with the previously defined data loaders, enabling smooth interaction between the model and the training/validation datasets. The maximum number of training epochs was set to 250 [6] . This value was determined empirically through preliminary experiments conducted without early stopping and on a smaller subset of the data. Beyond this point, the validation loss began to plateau or increase, indicating diminishing returns and the onset of overfitting.

8.5 Model Testing

After completing the training and validation phases, we conducted a comprehensive evaluation of the model under real-time inference conditions as part of the main objective of this Thesis. To thoroughly assess performance in practical deployment scenarios, we designed and implemented several testing strategies to evaluate both offline and causal, frame-wise inference modes.

1. **Whole Audio Processing (Offline Benchmark)**

   This approach processes the entire audio signal at once, without any frame-wise decomposition. Also referred to as offline inference, it assumes full access to the input sequence during processing, as illustrated in Figure 8.1. While not suitable for real-time applications, this method serves as a performance upper bound, providing a benchmark for evaluating the quality of causal, streaming alternatives.

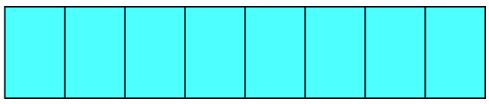

Figure 8.1: Whole Audio Processing

2. **Frame-Wise Testing Strategies**

   To evaluate real-time applicability, we tested several causal, frame-by-frame inference methods.

   (a) **Frame-by-Frame Processing**

   In this minimal approach, each incoming audio frame is processed independently, with no context from previous frames retained beyond the internal model state (see Figure 8.2). While computationally efficient, this method lacks sufficient temporal context, which can degrade performance, especially for models relying on long-term dependencies.

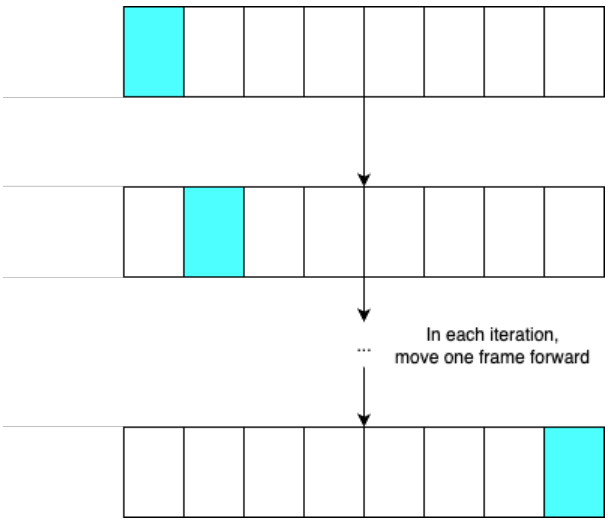

Figure 8.2: Frame-by-Frame Processing

   (b) **Additional Frame Method**

   This strategy incrementally builds the input by appending each new frame to the previous sequence, increasing the input length by one frame per step (Figure 8.3). The process starts with a single frame and continues until the full audio is processed. The output from each step is stored to maintain alignment with the growing input. This method preserves temporal context and mimics training conditions more closely, particularly benefiting time-domain feature learning. However, it increases memory and computational costs over time.

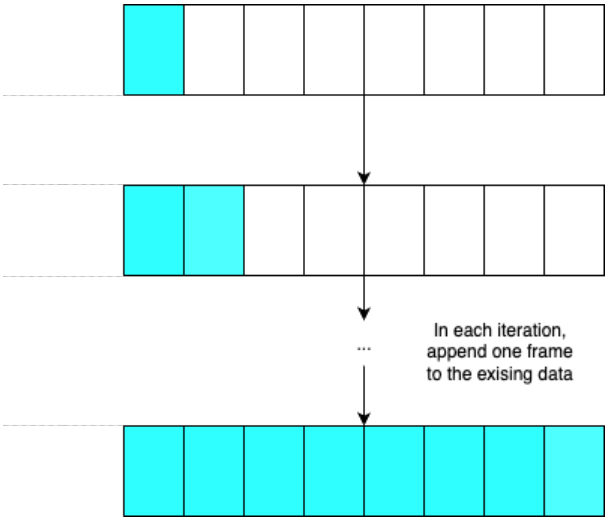

Figure 8.3: Additional Frame Processing

(c) **Sliding Window with Zero Padding**

To maintain a fixed input size (matching the training configuration), this method applies a sliding window approach with zero-padding for early frames (Figure 8.4). Initially, missing past frames are replaced with zeros, and as more data arrives, real frames gradually replace the padding. While this stabilizes memory usage, the artificial padding, especially in the initial frames that can distort the signal and degrade model performance, despite being an improvement over basic frame-by-frame processing.

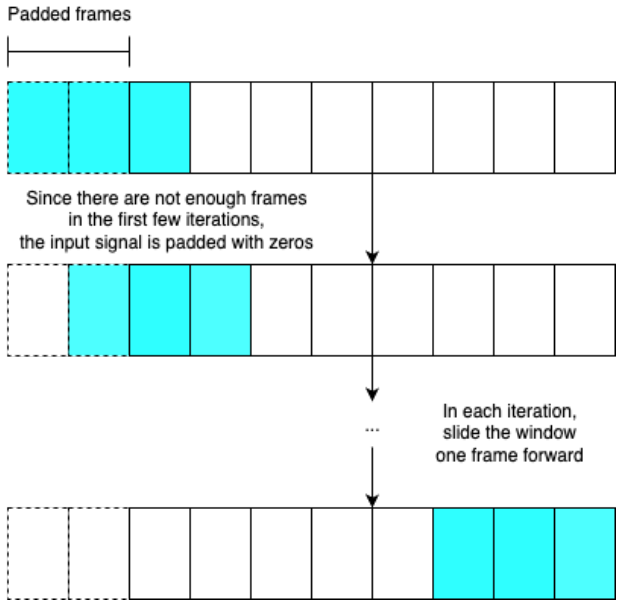

Figure 8.4: Sliding Window with Padded Frame

(d) **Additional Frame Followed by Sliding Window**

This hybrid method combines the strengths of the previous two approaches. Initially, the *additional frame* strategy is used to build up context until the input reaches the fixed length used during training. Once this threshold is reached, a sliding window is applied: for each new frame, the oldest frame is discarded, maintaining a constant input size (Figure 8.5). This approach ensures robust early predictions while capping memory usage, making it ideal for long-duration, real-time deployment on resource-constrained devices.

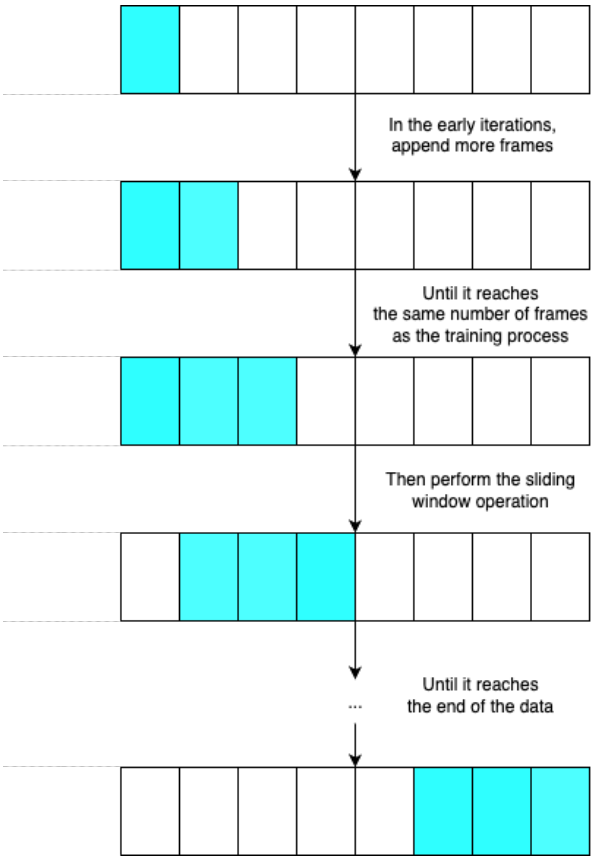

Figure 8.5: Hybrid Processing: Additional Frame Followed by Sliding Window

9 Results

This section presents the experimental results evaluating the performance and limitations of the modified Joint Non-linear Filter (JNF) model [7]. Quantitative metrics are computed by comparing the model's predictions to ground truth on the test set, with results summarized in tables to highlight behavior across different configurations. Besides that, subjective listening is done by the author during the testing to ensure the quantitative metric and the quality of the output audio.

Training involved iterative experiments focused on hyperparameter tuning, including the loss function, l weighting factor, input normalization, simulated data type, bidirectional architecture, and input frame length. Performance was assessed by measuring the improvement in noise suppression, defined as the difference between the clean target and the enhanced output.

9.1 Loss Function Optimization

The loss function plays a central role in shaping the model's learning objectives, as it directly determines the trade-off between perceptual speech quality and the degree of noise suppression. To investigate this, we performed two complementary ablation studies:

1. scaling between l and PESQ within our loss function L as explained in Eq. 13, and
2. weighting between the time- and frequency-domain contributions within l as detailed in Eq. 12.

The following subsections summarize the results.

9.1.1 Balancing l and PESQ in L loss

We first evaluated how the relative scaling between l and PESQ influences performance. Each configuration combined a scaled l term [35], representing the mean absolute error (Section 3.3), with a PESQ component [36] (Section 6.5.1). The l term was included to promote accurate reconstruction in both the time and frequency domains, while the PESQ component encouraged the model to produce outputs that align with human-perceived quality.

The scaling factor beta in front of l in Eq. 12 effectively adjusts the relative importance of reconstruction fidelity versus perceptual quality. The impact of this balance is reflected in the experimental results, which are summarized in Table 1. At lower scaling factors (e.g., $L = 25l + P$), the PESQ term dominated, leading to reduced l-driven accuracy and noticeably lower SDR performance. Conversely, very high scaling factors (e.g., $L = 500l + P$) placed excessive emphasis on minimizing absolute error, limiting improvements in perceptual metrics such as PESQ and STOI.

A clear performance peak emerged when our loss function $L = 200l + P$ configuration as referred in Section 8.4, which achieved the highest scores across three of the four evaluation metrics Δ PESQ, Δ SiSDR, and Δ SDR and the second-highest score in Δ STOI. This suggests that a scaling factor of 200 strikes an optimal balance: the l term is sufficiently weighted to ensure precise signal reconstruction, while the PESQ term still contributes meaningfully to perceptual improvements.

Furthermore, configurations close to this optimum (e.g., $L = 225l + P$ and $L = 250l + P$) also produced strong results, indicating a plateau region around the optimal scaling. However, moving further away from the optimum in either direction led to consistent performance degradation. For example, scaling factors below 100 suffered from weaker objective SDR improvements, while factors above 300 exhibited a gradual decline in PESQ gains.

In summary, the loss function experiment highlights the sensitivity of performance to the relative weighting of the l and PESQ components. The chosen $L = 200l + P$ formulation offers a robust compromise between perceptual quality and noise suppression, and was therefore selected as the primary training loss for the final JNF configuration.

Table 1: Evaluation of different loss function configurations on speech enhancement performance. The first row corresponds to the baseline loss function (L) [35] from the reference implementation [6], while subsequent rows evaluate hybrid loss functions combining a scaled l loss and PESQ-based loss [36]. The columns report the average improvement (Δ) in key speech quality metrics. All values are computed on the test set. The best result across each metric is highlighted in **bold**. The configuration $200l + P$ achieves the highest scores in all metrics, indicating the best performance.

Loss Function (L)	**Δ PESQ**	**Δ SiSDR**	**Δ SDR**	**Δ STOI**
l [6]	0.162551155	9.681057144	7.013664673	0.016737901
$25l + P$	0.118554508	4.799198917	3.620645337	0.006887415
$40l + P$	0.219031123	8.09461037	5.836518184	0.021244384
$50l + P$	0.231794181	9.515879106	6.530330248	0.025234174
$75l + P$	0.163666282	8.281006489	5.45888124	0.015699776
$100l + P$	0.185635867	9.051536257	6.484702233	0.021822053
$150l + P$	0.141623696	8.798429114	5.727203318	0.026941689
$200l + P$	**0.251403012**	**9.918974122**	**7.142367393**	**0.031007841**
$225l + P$	0.224951106	9.577326223	6.660023683	0.030005388
$250l + P$	0.214543373	9.603371051	6.766007248	0.030583394
$275l + P$	0.220662231	9.797966937	7.050779017	0.026261687
$300l + P$	0.212605895	9.47127902	6.666971335	0.026113993
$350l + P$	0.189376933	9.572006899	6.516678759	0.029518639
$500l + P$	0.189661323	9.613787022	6.722937608	0.031008782

9.1.2 Time–Frequency Weighting in l Loss

The l loss term (see Eq. 12), representing the mean absolute error, is a core component of our training objective, as it penalizes deviations between the enhanced and clean signals. In our formulation, l incorporates contributions from both the time domain and the frequency domain to ensure that the model learns to reproduce not only accurate waveforms but also faithful spectral characteristics. Following the method in [6], the combined l loss is expressed as shown in Equation 12.

In [6], a value of $\alpha = 3$ was used to balance the contributions of the time and frequency domains. However, the ideal weighting may vary depending on the noise characteristics and signal properties of the dataset. To investigate this, we performed an ablation study, evaluating models trained with different values of α. The performance was assessed using the same loss function ($200l + P$), as summarized in Table 2.

The results reveal that $\alpha = 4$ achieves the highest scores across all reported metrics. This adjustment not only improved ΔPESQ but also yielded consistent gains in ΔSiSDR, ΔSDR, and ΔSTOI. For extremely high α values (e.g., $\alpha = 100$) or extremely low ones (e.g., $\alpha = 0.1$), performance degraded noticeably. Excessive emphasis on the time domain risks neglecting spectral accuracy, while too little weight on the time domain impairs waveform fidelity and reduces intelligibility. The extreme case of $\alpha = 0.01$ led to an outright failure in ΔSiSDR (negative improvement), indicating that time-domain information is essential for effective denoising in our setup. Based on these findings, we adopt $\alpha = 4$ in the final Joint Non-linear Filter configuration, as it provides a stable and empirically validated trade-off between temporal and spectral reconstruction accuracy.

Table 2: Ablation study on the weighting factor α in the combined time-frequency l loss. Each model was trained using the hybrid loss $200l + P$ and evaluated on the same test set. Columns report the average improvement (Δ) over a no-processing baseline in key speech quality metrics. The best result across all metrics is achieved at $\alpha = 4$, and these values are highlighted in **bold**.

α	Δ PESQ	Δ SiSDR	Δ SDR	Δ STOI
4	**0.251403012**	**9.918974122**	**7.142367393**	**0.031007841**
10	0.203706242	9.618928577	7.13258073	0.026522422
1	0.198978874	9.402112198	6.595115214	0.027914107
100	0.139415268	9.229543365	6.502659836	0.024073986
0.1	0.179831827	8.901499218	6.070412406	0.023885948
0.01	0.136318258	-2.296120191	4.251218084	0.022252169

9.2 Dataset Optimization

Following the selection of the loss function, we investigated the necessity of two preprocessing components used in the state-of-the-art approach [6], namely artificial data simulation and input normalization. While these steps are known to improve robustness in low-data scenarios, they also increase pipeline complexity and training time. Our objective was to assess whether they could be removed without reducing the performance.

9.2.1 Effect of Dataset Size

Initial experiments were conducted with only 194 samples of length of 3 s (total duration nearly 10 minutes) and the full preprocessing pipeline (simulation and normalization) in place. This configuration yielded poor results, with unstable convergence and low perceptual quality. However, scaling the dataset to 3866 samples of length of 3 s (total duration 3 hours and 13 minutes) led to substantial improvements in all metrics, particularly in ΔPESQ and ΔSiSDR as presented in Table 3. This suggests that data quantity is a primary driver of robustness, even before considering the removal of preprocessing steps.

9.2.2 Impact of Removing Normalization

When normalization was removed while keeping simulation and dataset size constant at 3,866 samples, PESQ dropped around 30% as shown in Table 3. This decline reflects the challenge of training on unnormalized data, where the model must learn to handle a wider range of input amplitudes and dynamic ranges. Such variability can delay learning and lead to suboptimal perceptual quality in low to medium-sized datasets, even if other objective metrics such as SDR improve slightly.

9.2.3 Simulation vs. Real Data

Interestingly, when normalization was retained but simulation removed with 3,866 samples, ΔPESQ improved over the baseline with simulation as depicted in Table 3. This implies that overly synthetic data may not perfectly match the statistical properties of real-world signals, and in some cases, it may be beneficial to rely solely on diverse real recordings provided there is enough variability in the dataset.

9.2.4 Large-Scale Training without Preprocessing

The most significant finding emerged when both preprocessing steps were removed and the dataset was expanded to 7732 samples of length of 3 s (total duration 6 hours and 26 minutes). Under these conditions, the model achieved the best results across *all* metrics, including more than a fourfold increase in ΔPESQ compared to the small-data baseline as detailed in Table 3. This demonstrates that with sufficient volume and diversity of training data, the model can learn robust and generalizable features directly from raw inputs without the need for handcrafted normalization or simulated augmentation. The results from Table 3 support the conclusion that preprocessing is most valuable in small-data regimes, while in large-data regimes it becomes less critical, enabling a more direct and maintainable training pipeline.

Table 3: Evaluation of the impact of data simulation, feature normalization, and dataset size on speech enhancement performance. The table compares different preprocessing and data generation configurations. **Simulation** indicates whether synthetic noisy-clean speech pairs were generated; **Normalization** refers to mean-variance normalization applied to input features. **Dataset Size** denotes the number of training samples. The remaining columns show the improvement (Δ) in key metrics. The first row uses a small simulated and normalized dataset as a reference point. Subsequent rows explore ablations in simulation, normalization, and scaling of data size. The best result in each metric is highlighted in **bold**. The final row (no simulation, no normalization, largest dataset) achieves the top performance across all metrics, suggesting that larger real data and avoiding aggressive preprocessing can be beneficial when sufficient data is available.

Simulation	Normalization	Dataset Size	Δ PESQ	Δ SiSDR	Δ SDR	Δ STOI
Yes	Yes	194	0.25140301	9.91897412	7.14236739	0.03100784
Yes	Yes	3866	0.62339540	12.2530696	9.00130070	0.09483265
No	Yes	3866	0.83433435	7.3381653	9.10900313	0.08821156
No	No	3866	0.56607393	9.42322416	11.1725061	0.07061303
No	**No**	**7732**	**1.08178604**	**11.6506281**	**12.3585569**	**0.14453324**

9.2.5 Bidirectional vs. Unidirectional Architectures

A related ablation study was conducted on the LSTM architecture. The reference model [6] employs bidirectional layers to leverage both past and future context, boosting reconstruction quality. However, such architectures are *non-causal* and thus incompatible with real-time streaming applications, where only past and present information is available at inference time. To address this limitation, we evaluated a unidirectional version of the model, training it with the largest available dataset around 17,986 samples (consist of 3 sec audio each samples, with the total of 6 hours 26 minutes) to compensate for the loss of future context.

The results, summarized in Table 4, confirm that removing bidirectionality reduces performance to all metrics when it is trained with the same dataset. It is clear that the improvements in SDR and STOI drop significantly, though PESQ remains relatively stable. However, when the unidirectional model is trained on a larger dataset of 17,986 samples, performance improves across all metrics, with the most notable gains in PESQ [36]. Furthermore, the drop is particularly evident in ΔSDR and ΔSTOI, indicating degraded noise reduction quality compared to bidirectional model. Nevertheless, we opted for the unidirectional architecture in the final deployment to satisfy real-time latency requirements, accepting the trade-off of slightly reduced quality in exchange for practical deployability.

Beyond perceptual and separation performance, we also profiled both architectures using THOP to quantify their computational demands (Table 5). The bidirectional model incurs **2.56 times more Multiply-Accumulate Operations (MACs)** (605.4G vs. 236.6G) and **2.56 times more parameters** (1.194M vs. 466.2K). This is a direct consequence of maintaining dual hidden state trajectories and doubling the recurrent layer capacity. The overhead translates directly into higher memory requirements, longer inference latency, and increased power consumption.

Therefore, despite the bidirectional model's superior quality, we selected the **unidirectional variant trained on the larger dataset** for final deployment. This decision embodies a deliberate and quantified trade-off: we sacrifice in SDR, SiSDR, and STOI, in exchange for **real-time causal operation**, **61% reduction in computation**, and **61% fewer parameters** enabling deployment on resource-constrained devices

without buffering or lookahead.

Table 4: Comparison of unidirectional and bidirectional model configurations in terms of speech enhancement performance. The **Bidirectional** column indicates whether the model processes the input sequence in both forward and backward directions (Yes) or only forward (No). The remaining columns show the improvement (Δ) in key metrics. All models were trained using the same architecture and loss function ($200l + P$), with the exception of the recurrent directionality. The **Dataset Size** reflects the number of training samples used for each configuration. Performance is evaluated using improvements (Δ) over a no-processing baseline in key metrics. The chosen method is highlighted in **bold**.

Bidirectional	**Dataset Size**	**Δ PESQ**	**Δ Si SDR**	**Δ SDR**	**Δ STOI**
Yes	7732	1.081786036	11.65062809	12.35855685	0.144533239
No	7732	1.064351401	5.5928884607	6.639585339	0.087944407
No	**17986**	**1.129459209**	**7.747693917**	**8.729436295**	**0.088295742**

Table 5: Comparison of unidirectional and bidirectional model configurations in terms of computational complexity and parameter count, as measured by the THOP profiler. The **Bidirectional** column indicates whether the model processes the input sequence in both forward and backward directions (Yes) or only in the forward direction (No). The **MACs** column reports the total number of Multiply-Accumulate Operations (in billions, G), and the **Params** column reports the total number of trainable parameters (in thousands, K or millions, M). All models share the same base architecture and were profiled under identical input conditions; the only variable is the directionality of recurrent or sequential processing. Computational efficiency is a key consideration, lower MACs and fewer parameters typically imply faster inference and reduced memory footprint. The most efficient configuration is highlighted in **bold**.

Bidirectional	**MACs**	**Params**
Yes	605.371G	1.194M
No	**236.598G**	**466.178K**

9.2.6 Effect of Data Length

Building on the previous optimization results, the next step investigates how the choice of input audio segment length influences speech enhancement performance. In supervised training of speech enhancement models, the segment duration determines the amount of temporal context available to the network. Longer segments provide richer information about speech continuity and noise patterns, which can improve enhancement quality. However, they also increase memory requirements and training time. Shorter segments reduce computational cost but may limit the model's ability to capture long-range dependencies, which are particularly important in speech sequences containing continuous or slowly varying noise events.

The state-of-the-art model proposed in [6] employs 3-second segments, representing a trade-off between temporal coverage and computational feasibility. To assess whether this configuration is optimal, we conducted a systematic evaluation across multiple input lengths ranging from 0.2 seconds to 3 seconds. The results are summarized in Table 6. All experiments were performed with identical hyperparameters (Table 7), i.e., same loss, architecture, and training setup, with no bidirectionality, normalization, or additional data simulation. The only factor varied was the segment length.

As shown in Table 6, very short segments (0.2–0.4 seconds) lead to substantial drops in all performance metrics, with ΔPESQ decreasing by more than 60% compared to the 3-second baseline. This decline reflects the model's limited temporal context, which reduces its ability to suppress noise effectively and often produces perceptually noticeable artifacts. Increasing the segment length to 1 second yields a marked improvement, especially in PESQ, indicating that even modest additional context enables better enhancement.

At 2 seconds, the performance approaches that of 3 seconds, though the 3-second configuration consistently yields the highest results across all four metrics (ΔPESQ, ΔSiSDR, ΔSDR, ΔSTOI) as shown in Table 6. This suggests diminishing returns beyond 2 seconds, but confirms that the full 3-second setting still provides the most effective balance of contextual modeling and enhancement quality.

In summary, longer data segments substantially benefit performance, and the 3-second length remains optimal within the tested range. This choice therefore forms the basis for the Joint Non-linear Filter (JNF) training configuration.

Table 6: Ablation study on the effect of input audio segment length during training and inference. The **Audio Length** column indicates the duration (in seconds) of the signal chunks used for processing. All models were trained using the same architecture and loss function ($200l + P$), with identical hyperparameters such as unidirectional model, with no data simulation and normalization. Performance is measured as the improvement (Δ) in key speech quality metrics. Results show a clear trend of increasing performance with longer segments, with the best scores achieved at **3 seconds**, highlighted in **bold**. This suggests that longer temporal context enables the model to better capture speech dynamics and noise patterns, leading to more effective denoising and enhanced perceptual quality. Shorter segments (e.g., 0.2–0.4 sec) result in significantly lower performance, likely due to insufficient contextual information.

Audio Length	**Δ PESQ**	**Δ Si SDR**	**Δ SDR**	**Δ STOI**
0.2 sec	0.407387902	6.855344088	7.972177439	0.055434321
0.4 sec	0.633248314	7.540423454	8.423096134	0.076802934
1 sec	1.053406482	7.693254794	8.803876188	0.082682878
2 sec	1.060674543	7.537191244	8.693831517	0.085378469
3 sec	**1.129459209**	**7.747693917**	**8.729436295**	**0.088295742**

9.3 Final Model Configuration

The final hyperparameter configuration was selected through a series of ablation studies that evaluated the impact of key architectural and training choices on validation performance, generalization, and training stability. Each hyperparameter was systematically tuned based on its contribution to speech enhancement quality, as measured by objective metrics such as PESQ, SiSDR, SDR, and STOI. The resulting configuration, summarized in Table 7, represents the optimal setup that maximizes perceptual quality under our experimental conditions.

The loss function combines a scaled time-frequency l loss and a perceptual PESQ-based loss ($200l + P$) to balance waveform accuracy and speech intelligibility. The l component uses a weighting factor of $\alpha = 4$, meaning the time-domain loss is weighted four times more than the frequency-domain loss, as determined in Table 2. The model uses a unidirectional architecture, which despite training on a larger dataset provided better PESQ compared to bidirectional variants. No synthetic data simulation was applied, and input features were used without normalization, as both were found to degrade generalization or add unnecessary complexity. Input audio segments of 3 seconds were chosen to provide sufficient temporal context for effective noise modeling and speech structure preservation, as shown in Table 6. This final configuration achieves the best trade-off between performance, robustness, and practical deployment considerations.

Table 7: Final hyperparameter configuration of the modified Joint Non-linear Filter (JNF) model, selected based on comprehensive ablation studies and validation performance.

Hyperparameter	**Value**
Loss function	$200l + P$ Loss
l function	4 Time-domain Loss + Frequency-domain Loss
Bidirectional Value	None
Data simulation	None
Data normalization	None
Data length	3 Sec

9.4 Real Time Operation

A central objective of this work is the design of a model capable of performing real-time, frame-wise inference with minimal latency, while maintaining enhancement quality comparable to offline processing. Such a capability is critical for practical deployment in scenarios such as live communications, hearing assistive devices, and embedded IoT platforms. After completing model training and hyperparameter tuning, we systematically evaluated multiple real-time inference strategies to identify a solution that balances accuracy, computational cost, and memory efficiency. The evaluation was conducted on test sequences of 10-second duration. The five strategies investigated are outlined in Section 8.5, with quantitative results provided in Table 8.

The `whole audio` method (Figure 8.1), which processes the complete signal in a single pass, serves as the performance upper bound. While this yields the best possible metric scores by providing the model with full temporal context, it is inherently non-causal and unsuitable for real-time operation.

The `frame-by-frame` method (Figure 8.2) represents the opposite extreme, operating strictly on individual frames without any temporal overlap. Although computationally efficient, its inability to exploit temporal dependencies results in a sharp degradation across all evaluation metrics. As shown in Table 8, ΔPESQ drops by more than 70% relative to the baseline), confirming that isolated frames are insufficient for high-quality speech reconstruction.

The `additional frame` method method (Figure 8.3) offers a near-perfect match to the offline `whole audio` baseline by progressively appending each new frame to the existing buffer, thereby continuously increasing the temporal context available to the model. However, the approach has a fundamental scalability issue: the memory usage grows linearly with processing time. In long-form streaming applications, such as multi-hour conference recordings, this would require storing and reprocessing the entire history at every inference step, quickly exhausting the resources of embedded hardware.

The `sliding window with padding` method (Figure 8.4) mitigates the memory issue by maintaining a fixed-length buffer that advances through the audio stream. When insufficient context is available (e.g.,

at the start of a recording), the missing frames are zero-padded to match the expected input length. While this keeps memory usage constant, the reliance on zero-padding for initial frames leads to a noticeable drop in perceptual quality, especially in early segments, as reflected in the reduced ΔPESQ and ΔSDR scores as defined in Table 8.

To address these limitations, we propose a hybrid approach, `additional frame with sliding window` (Figure 8.5), which integrates the strengths of both strategies:

1. **Initial phase:** When the accumulated audio is shorter than the model's input length (3 seconds), the `additional frame` strategy is applied. This ensures the model has the maximum available context from the start, avoiding the quality degradation caused by zero-padding.

2. **Steady-state phase:** Once the buffer reaches the target length, the system switches to a `sliding window` mode, discarding the oldest frame as each new one arrives. This keeps memory usage constant while preserving a continuous, context-rich input for the model.

This hybrid method not only stabilizes performance across time but also ensures that early and late segments of a stream are processed under similar conditions, improving perceptual consistency. Quantitatively, the approach achieves performance metrics close to the `additional frame` method while avoiding its unbounded memory growth. Notably, it even outperforms the `whole audio` method in ΔSiSDR and ΔSDR as exhibited in Table 8, likely due to the consistent frame segmentation that better aligns with the model's training conditions.

These results demonstrate that the `additional frame with sliding window` method provides an effective and practical solution for real-time deployment, delivering high-quality speech enhancement under strict latency and memory constraints.

Table 8: Evaluation of different inference strategies for speech enhancement. The first row (*Whole Audio*) serves as the baseline, processing the full signal at once. Subsequent rows evaluate chunked or windowed approaches for practical deployment. The evaluation was conducted on test sequences of 10-second duration. The columns report the improvement (Δ) in key speech quality metrics. All results are computed on the test set with the hyperparameter as described in Table 7. The best result in each metric is highlighted in **bold**. The *Additional Frame with Sliding Window* method (bolded row) achieves the best trade-off across metrics and is selected as the final inference strategy.

Type of Test	**Δ PESQ**	**Δ Si SDR**	**Δ SDR**	**Δ STOI**
Whole Audio	1.129459209	7.747693917	8.729436295	0.088295742
Frame by Frame	0.325592561	1.051973248	3.252265246	0.022529206
Additional Frame	1.129457313	7.747685493	8.729433328	0.088295708
Sliding Window with Padding	0.746132352	6.982236842	7.485668785	0.084381609
Additional Frame with Sliding Window	**1.082061826**	**8.109883735**	**9.133770969**	**0.081332392**

10 Discussion

The results demonstrate that the proposed Joint Non-linear Filter (JNF) model, when optimized with a $200l + P$ loss function and an l factor $\alpha = 4$, provides significant improvements over the state-of-the-art baseline [6]. The achieved performance gains across multiple evaluation metrics indicate that careful loss function tuning is a key factor in maximizing model performance for both time-domain and frequency-domain reconstruction. In particular, the PESQ improvement highlights better perceptual quality, which is essential for practical audio enhancement tasks.

The experiments further reveal that dataset size and preprocessing strategies have a substantial impact on model robustness. While normalization was found to improve performance on small datasets, its absence could be compensated by increasing the dataset size substantially. Moreover, the best overall results were obtained without normalization or simulation, but only when training on a significantly larger dataset. This suggests that large-scale data can reduce the dependency on specific preprocessing techniques, aligning with trends observed in deep learning where model generalization improves with data scale.

The decision to use a non-bidirectional LSTM architecture was motivated by the requirement for real-time frame-wise processing, despite a measurable decline in performance metrics compared to the bidirectional model. This trade-off highlights a common engineering constraint: predicting with unseen data. Similar reasoning influenced the choice of a 3-second audio length, which provided the best balance between model performance and memory usage.

In the testing phase, the proposed `Additional Frame with Sliding Window` method emerged as a viable compromise between prediction accuracy and memory efficiency. Although the `Additional Frame` technique achieved the highest similarity to the `Whole Audio` method, it was impractical for embedded systems due to its linear memory growth consumption. By combining `Additional Frame` and `Sliding Window` strategies, we were able to retain high predictive accuracy while ensuring bounded memory usage, thus meeting the primary goal of enabling real-time processing on resource-constrained devices.

With respect to the first research question, **What are the most effective strategies for reducing noise from multichannel input?**, the findings show that loss function design and dataset scale are central to effective noise suppression. The optimized loss configuration not only improved quantitative metrics but also enhanced perceptual quality, indicating better handling of noise distortions across channels. Furthermore, the results reveal that large-scale data training reduces reliance on preprocessing methods such as normalization, thereby improving robustness in noisy multichannel environments.

Regarding to the second research question, **How can speech enhancement models be adapted or restructured to operate on a frame-wise basis suitable for real-time applications like hearing aids?**, the study highlights the modified Joint Non-linear Filter (JNF) model. The decision to employ a causal, non-bidirectional LSTM ensured real-time, frame-wise prediction at the expense of some performance compared to bidirectional models. The choice of a 3-second audio length optimized the trade-off between accuracy and memory usage. Most notably, the proposed `Additional Frame with Sliding Window` inference strategy successfully balanced prediction accuracy with bounded memory consumption, enabling real-time deployment on resource-constrained devices. These adaptations directly support the goal of practical use in applications such as hearing aids, where latency and memory efficiency are critical.

The study successfully met its primary research objectives: the model was optimized for real-time inference, achieved competitive performance relative to state-of-the-art approaches, and introduced a novel inference strategy that effectively balances computational efficiency with output quality. Nevertheless, certain limitations persist. The model's reliance on a large training dataset to compensate for the removal of input normalization may hinder its applicability in data-constrained environments. Furthermore, while the non-bidirectional, causal architecture enables low-latency processing suitable for real-time deployment, the observed performance gap compared to bidirectional models indicates room for improvement. This motivates future exploration into more efficient causal architectures that maintain high fidelity without compromising speed.

In the broader context of speech and signal processing, the proposed approach shows strong potential for generalization to other real-time audio enhancement tasks such as noise suppression in communication systems, voice assistants, or hearing aids, where low latency and high-quality output are critical. A natural next step involves deploying the model on embedded systems, which would require implementation in efficient programming languages such as `C` or `C++`. Additionally, although the current architecture uses only two LSTM layers (with 256 and 128 hidden units, respectively), further model compression such as reducing hidden unit counts or applying quantization and pruning techniques could enhance suitability for

resource-limited devices.

Overall, this work not only refines the hyperparameter configuration and architectural design of the JNF model [6, 7] but also contributes a practical and deployable inference framework, effectively bridging the gap between high-performance research models and real-world applicability.

11 Conclusions

This thesis investigated a multichannel speech enhancement system designed for real-time noise reduction, with a focus on hearing aid applications. The proposed approach was based on a unidirectional LSTM Joint Non-linear Filter (JNF) model, optimized through targeted hyperparameter tuning and evaluated against the state-of-the-art baseline [6].

The most important findings are as follows. First, the combination of the $200l + P$ loss function and l factor $\alpha = 4$ produced the best overall performance across perceptual (PESQ) and distortion-based (SDR, SI-SDR) metrics, indicating that joint optimization for perceptual quality and reconstruction accuracy is crucial for practical audio enhancement. Second, dataset size was found to strongly influence robustness: while normalization improved results for small datasets, large-scale training data reduced the need for preprocessing, achieving the best results without normalization or simulation. Third, replacing the bidirectional LSTM with a unidirectional architecture enabled real-time frame-wise operation essential for hearing aids at the cost of a modest performance reduction. Finally, the proposed `Additional Frame with Sliding Window` inference method successfully balanced accuracy and memory usage, making it suitable for resource-constrained embedded systems.

The research objectives outlined in Section 4 were achieved: the system operates in real-time, maintains competitive performance compared to the state-of-the-art, and introduces a novel, memory-efficient inference strategy. The results confirm that the approach is well-suited for hearing aid scenarios, and potentially generalizable to other real-time audio enhancement tasks such as telecommunication noise suppression or live broadcast processing.

Nevertheless, some limitations remain. The dependency on large datasets for optimal performance may hinder adoption in domains with limited data. Additionally, while the unidirectional LSTM meets latency constraints, the observed performance gap to bidirectional models suggests opportunities for exploring lightweight causal transformer or convolutional architectures. Model size is another consideration despite using only two LSTM layers (256 and 128 hidden units), further compression through pruning, quantization, or reduced layer sizes could improve deployability.

12 Future Work

Future work should focus on porting the model to embedded hardware, requiring implementation in `C` or `C++` with optimized memory and computational efficiency. Exploration of model compression techniques, such as weight pruning or low-rank approximations, could further reduce computational load without significantly impacting performance. Additional research into adaptive frame sizing and context management could also improve efficiency under varying acoustic conditions. Finally, subjective listening tests with target users, including hearing-impaired participants, will be critical for validating perceptual improvements and guiding further refinements.

In conclusion, this work advances the state of real-time multichannel speech enhancement by providing a technically sound, practically feasible, and experimentally validated approach that addresses the unique constraints of hearing aid applications, while offering a foundation for further optimization and deployment in real-world devices.

References

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] U. Hoppe and G. Hesse, "Hearing aids: indications, technology, adaptation, and quality control," *GMS Current Topics in Otorhinolaryngology, Head and Neck Surgery*, vol. 16, p. Doc08, 2017.

[3] P. Vary and R. Martin, *Digital speech transmission: enhancement, coding and error concealment*. Chichester, England; Hoboken, NJ: John Wiley, 2006.

[4] K. U. Simmer, J. Bitzer, and C. Marro, "Post-filtering techniques," in *Microphone Arrays: Signal Processing Techniques and Applications*, M. Brandstein and D. Ward, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 39–60.

[5] R. C. Hendriks, R. Heusdens, U. Kjems, and J. Jensen, "On optimal multichannel mean-squared error estimators for speech enhancement," *IEEE Signal Processing Letters*, vol. 16, pp. 885–888, 2009.

[6] K. Tesch and T. Gerkmann, "Insights into deep non-linear filters for improved multi-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 563–575, 2023.

[7] K. Tesch, "Deep non-linear filters for multi-channel speech enhancement and separation," 2024, accessed: Feb 01, 2025. [Online]. Available: https://github.com/sp-uhh

[8] H. L. V. Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. John Wiley & Sons, 2004.

[9] J. Benesty, S. Makino, and M. M. Sondhi, *Speech Enhancement*. Springer, 2008.

[10] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. Springer Science & Business Media, 2001.

[11] S. Doclo, W. Kellermann, S. Makino, and S. E. Nordholm, "Multichannel signal enhancement algorithms for assisted listening devices: Exploiting spatial diversity using multiple microphones," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 18–30, 2015.

[12] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC press, 2013.

[13] D. Wang and J. Chen, "Computational auditory scene analysis for speech separation and enhancement," *IEEE Signal Processing Magazine*, vol. 38, no. 1, pp. 109–118, 2021.

[14] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. John Wiley & Sons, 2018.

[15] S. Makino, *Audio Source Separation*. Springer, 2018.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, 1999.

[17] J. B. Allen and L. R. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

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

[19] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[20] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[21] D. S. Williamson, Y. Wang, and D. L. Wang, "Complex ratio masking for monaural speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4368–4372.

[22] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[23] K. Tan, Z.-Q. Wang, and D. Wang, "Neural spectrospatial filtering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 605–621, 2022.

[24] T. Ochiai, M. Delcroix, T. Nakatani, and S. Araki, "Mask-based neural beamforming for moving speakers with self-attention-based tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 835–848, 2023.

[25] Y. Masuyama, M. Togami, and T. Komatsu, "Multichannel loss function for supervised speech source separation by mask-based beamforming," in *Proceedings of Interspeech*, Graz, Austria, 2019, pp. 2708–2712.

[26] C. Boeddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, "Exploring practical aspects of neural mask-based beamforming for far-field speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 6697–6701.

[27] R. Gu, S.-X. Zhang, Y. Zou, and D. Yu, "Towards unified all-neural beamforming for time and frequency domain speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, no. 99, pp. 1–14, January 2022.

[28] S. Cornell, M. Pariente, F. Grondin, and S. Squartini, "Learning filterbanks for end-to-end acoustic beamforming," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Singapore: IEEE, May 2022, pp. 536–540.

[29] H. Chang, Y. Hsu, and M. R. Bai, "Deep beamforming for speech enhancement and speaker localization with an array response - aware loss function," -, Oct 2023.

[30] K. Tesch and T. Gerkmann, "Nonlinear spatial filtering in multichannel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1795–1805, 2021.

[31] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Complete LDC93S6A," Linguistic Data Consortium, Philadelphia, May 2007, catalog No. LDC93S6A.

[32] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Calgary, Canada: IEEE, 2018, pp. 351–355.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] X. Li and R. Horaud, "Multichannel speech enhancement based on time-frequency masking using subband long short-term memory," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2019, pp. 298–302.

[35] B. Tolooshams, R. Giri, A. H. Song, U. Isik, and A. Krishnaswamy, "Channel-attention dense u-net for multichannel speech enhancement," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 836–840.

[36] ITU, "Itu-t recommendation p.863: Perceptual objective listening quality analysis (polqa)," International Telecommunication Union, Geneva, Switzerland, Tech. Rep., 2011.

[37] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2002–2015, 2016.

[38] P. C. Loizou, *Speech enhancement: theory and practice*, 2nd ed. CRC press, 2013.

[39] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq) - a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. Salt Lake City, UT, USA: IEEE, 2001, pp. 749–752.

[40] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[41] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 626–630.

[42] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noise reduction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2108–2118, 2011.

[43] Aalto University, "Aalto acoustics lab facilities," https://www.aalto.fi/en/aalto-acoustics-lab/aalto-acoustics-lab-facilities, 2023, accessed: 2025-04-05.

[44] L. McCormack and A. Politis, "Compass: Plug-in suite for parametric spatial audio processing," 2024, accessed: Apr. 05, 2025. [Online]. Available: https://leomccormack.github.io/sparta-site/docs/plugins/compass-suite/

[45] European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," pp. 1–88, 2016, accessed: Apr. 5, 2025. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[46] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Oriental COCOSDA 2017*, 2017, p. Submitted.

[47] U. Nations, "The 17 goals," 2015, accessed: Oct 26, 2023. [Online]. Available: https://sdgs.un.org/goals