AccCall: Enhancing Real-Time Phone Call Quality with Smartphone's Built-in Accelerometer

LEI WANG, Soochow University University, China
XINGWEI WANG, Soochow University, China and Beijing University of Technology, China
XI ZHANG*, Macquarie University, China
XIAOLEI MA, Soochow University, China
YU ZHANG, Macquarie University, Australia
FUSANG ZHANG, Beihang University, China and Inspur Computing Technology Pty Ltd, China
TAO GU, Macquarie University, Australia
HAIPENG DAI*, Nanjing University, China

Speech enhancement can greatly improve the user experience during phone calls in low signal-to-noise ratio (SNR) scenarios. In this paper, we propose a low-cost, energy-efficient, and environment-independent speech enhancement system, namely AccCall, that improves phone call quality using the smartphone's built-in accelerometer. However, a significant gap remains between the underlying insight and its practical applications, as several critical challenges should be addressed, including efficiency of speech enhancement in cross-user scenario, adaptive system triggering to reduce energy consumption, and lightweight deployment for real-time processing. To this end, we first design Acc-Aided Network (AccNet), a cross-modal deep learning model inherently capable of cross-user generalization through three key components, including cross-modal fusion module, accelerometer-aided (abbreviated as acc-aided) mask generator, the unified loss function. Second, we adopt a machine learning-based approach instead of deep learning to achieve high accuracy in distinguishing call activity states followed by adaptive system triggering, ensuring lower energy consumption and efficient deployment on mobile platforms. Finally, we propose a knowledge-distillation-driven structured pruning framework that optimizes model efficiency while preserving performance. Extensive experiments with 20 participants have been conducted under a user-independent scenario. The results show that AccCall achieves excellent and reliable adaptive triggering performance, and enables substantial real-time improvements in SISDR, SISNR, STOI, PESQ, and WER, demonstrating the superiority of our system in enhancing speech quality and intelligibility for phone calls.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Accelerometer sensing, Speech enhancement.

*Corresponding authors.

Authors' Contact Information: Lei Wang, Soochow University University, China, wanglei@suda.edu.cn; Xingwei Wang, Soochow University, China and Beijing University of Technology, Beijing, China, xw.wang@emails.bjut.edu.cn; Xi Zhang, Macquarie University, China, zaibuer@gmail.com; Xiaolei Ma, Soochow University, China, 20235227057@stu.suda.edu.cn; Yu Zhang, Macquarie University, Australia, y.zhang@mq.edu.au; Fusang Zhang, Beihang University, Beijing, China and Inspur Computing Technology Pty Ltd, Beijing, China, zhangfusang@buaa.edu.cn; Tao Gu, Macquarie University, Australia, tao.gu@mq.edu.au; Haipeng Dai, Nanjing University, China, haipengdai@nju.edu.cn.

ACM 2474-9567/2025/9-ART133
https://doi.org/10.1145/3749463

ACM Reference Format:
Lei Wang, Xingwei Wang, Xi Zhang, Xiaolei Ma, Yu Zhang, Fusang Zhang, Tao Gu, and Haipeng Dai. 2025. AccCall: Enhancing Real-Time Phone Call Quality with Smartphone's Built-in Accelerometer . *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 3, Article 133 (September 2025), 33 pages. https://doi.org/10.1145/3749463

1 Introduction

Phone call, as an efficient communication mode, is one of the most widely used forms of voice communication today. Compared to voice input, such as voice assistants [19] or voice messages [47], phone call has the advantages of stronger interactivity and faster response times. Especially in emergency situations, complex discussions, or when quick feedback is needed, phone calls offer substantial advantages. According to a 2021 report by Statista [54], an estimated 13.5 billion phone calls were made worldwide each day. However, in real-world environments such as train stations, construction sites, or streets, users are often surrounded by noise. In these settings, phone calls are frequently overwhelmed by various sounds, including music, machinery noise, conversations, and car horns, etc. Moreover, the user may speak softly to avoid disturbing others or revealing private information in some public environments [57, 77]. In such cases, the signals received by the microphone may have a low SNR, making it difficult for the receiver to clearly understand the conversation's content [42]. Consequently, speech enhancement is a critical necessity for phone calls. Note that the hands-on mode of phone calls is generally preferred due to its superior privacy, clearer sound quality, and lower energy consumption compared to the hands-free mode [21]. Therefore, our work focuses on enabling speech enhancement specifically during hands-on calls.

Compared to single-modality enhancement methods, which are limited to scenarios with up to two mixed speeches [36, 58], multi-modality enhancement leverages correlated information across multiple modalities to achieve speech enhancement in scenarios with more complex mixtures [10, 40], which represents a more advanced form of speech enhancement. Therefore, we direct our research toward multi-modality speech enhancement scheme for phone call scenarios. Recent studies have demonstrated that ultrasound signals captured by built-in microphone, reflecting lip movements, can assist in the recovery of clean speech [10, 58, 78]. Although this modality can be involved for phone calls, it has two significant drawbacks: Firstly, ultrasound must be actively emitted by the smartphone's built-in speaker, leading to significant additional power consumption [65]. Secondly, ultrasound waves are highly susceptible to interference from wind currents, which limits their effectiveness in outdoor environments. In this paper, we propose a low-cost, energy-efficient, and environment-independent speech enhancement system that improves phone call quality using the smartphone's built-in accelerometer. The key insight lies in the following two aspects. Firstly, compared to a pair of speakers and microphones, the smartphone's built-in accelerometer is low-power [26] and robust to airflow (as discussed in Section 3.2), making it a promising alternative modality better suited for phone call scenarios. Secondly, speech-induced vibration patterns, including both vocal vibrations and skin movements, are transmitted to the built-in accelerometer through facial bones and skin via physical contact between the cheek and the phone, ensuring that the accelerometer readings are closely correlated with speech rather than background noise. We envision a scenario where, whether indoors or outdoors, the recipient can hear the user's speech clearly without interference from environmental noise or airflow when the user makes a phone call using our APP, as illustrated in Fig. 1.

While promising, there remains a significant gap between the underlying insight that utilizes the built-in smartphone accelerometer as an auxiliary modality for speech enhancement and its practical applications, as several critical challenges must be addressed. The first challenge is to enable speech enhancement in cross-user scenario. We aim for the accelerometer aided speech enhancement model to be effective not only for users seen in the training dataset but also to exhibit strong generalization to unseen users. A straightforward approach is to train

(a) Making a phone call in an outdoor environment. (b) Interface of our APP

Fig. 1. Example of an application scenario for AccCall.

the model using as many speaker samples as possible [18, 62]. However, this method is labor-intensive and time-consuming. To address this challenge, we design AccNet, a cross-modal deep learning model inherently capable of cross-user generalization through three key components. The cross-modal fusion module integrates speech and accelerometer features, extracting shared and modality-specific representations while dynamically adjusting feature contributions to ensure speaker-independent learning. The acc-aided mask generator leverages cross-modal representations to capture user-independent motion patterns, enabling effective target speech separation across different speakers. Additionally, the unified loss function jointly optimizes speech reconstruction and refines the acc-aided mask by embedding speaker-specific knowledge, enhancing separation accuracy while preserving generalization. Together, these components eliminate the need for extensive speaker-dependent training, ensuring robust speech enhancement across diverse users and acoustic environments.

The second challenge is to efficiently trigger the speech enhancement system to minimize battery drain and computation, especially on resource-constrained mobile devices. A common strategy is to rely on "normal phone mode" or proximity sensors to infer whether the phone is held close to the user's face. However, these methods are often unreliable in real-world scenarios. Normal phone mode does not reliably imply face contact. Users frequently move the phone away during a call to check messages, read content, or take photos, all while the device remains in call mode. Some users habitually maintain a 2 to 3 cm gap between the phone and their face. While this distance does not affect call quality, it can significantly reduce the effectiveness of our speech enhancement system. Proximity sensors are also unreliable in practice. Their binary output lacks the precision to capture nuanced device positioning. Moreover, their performance is easily disrupted by high ambient light or incidental obstructions such as fingers or clothing [60, 64]. To overcome these limitations, we propose an adaptive trigger that relies solely on accelerometer data. The system uses a lightweight machine learning model to recognize a series of motion patterns associated with phone-to-cheek contact, enabling accurate detection of relevant call states without depending on unreliable indicators or additional sensors. To ensure the model remains both efficient and effective, we design a two-stage feature selection strategy. First, we evaluate the individual relevance of each candidate feature using a mutual information scoring method and retain the most informative ones. Then, we apply an iterative feature elimination process, recursively removing less significant features while validating model performance through cross-validation. This procedure ensures that the final model is not only accurate but also lightweight enough for real-time deployment on mobile devices.

The third challenge is to enable efficient deployment of the heavyweight deep neural network (DNN) model on resource-constrained mobile platforms. In real-time phone call scenarios, mobile devices are sensitive to the size and computational demands of deployed models. However, the original DNN model consists of a larger

Table 1. Comparison of speech enhancement methods

Method	Modality Setting	Privacy Protection	COTS Device	Airflow Resistance	Prior Knowledge	Speakers	Energy Efficiency
AccNet	**Audio+ Accelerom-eter**	**Yes**	**Yes**	**Yes**	**No Need**	**Unlimited**	**Yes**
Studies [1, 41, 43, 46, 52, 67]	Audio	Yes	Unknown	Yes	Need	Unknown	Unknown
Studies [34, 38, 42, 71, 75]	Audio	Yes	Unknown	Yes	No Need	≤ 2	Unknown
Studies [2, 11, 53]	Audio+ Visual	No	Yes	Yes	No Need	Unlimited	No
Studies [32, 40]	Audio+ mmWave	Yes	No	Yes	No Need	Unlimited	Unknown
Study [50]	Audio+ Lidar	Yes	No	Yes	No Need	Unlimited	Unknown
Studies [10, 58, 78]	Audio+ Ultrasound	Yes	Yes	No	No Need	Unlimited	No

number of parameters and Floating Point Operations Per Second (FLOPs), making it difficult to meet the real-time computing requirements. To address this challenge, we introduce a knowledge-distillation-driven structured pruning framework that optimizes model efficiency while preserving performance. Our method leverages a teacher-student schema to assess channel importance by analyzing activation similarity between clean and noisy speech data. By selectively retaining channels that contribute most to clean speech reconstruction, our approach enables a task-aware, data-driven pruning process, significantly reducing computational complexity without compromising speech enhancement quality.

Overall, the contributions of our work are summarized as follows:

• We envision a promising and practical scenario for enhancing speech quality in phone calls, allowing users to make calls indoors or outdoors without disruption from interfering noise or airflow, as illustrated in Fig. 1(a). We believe this approach will significantly advance speech enhancement, relying solely on low-power built-in IMU sensor for commercial mobile devices.

• We carefully address three key challenges to bridge the gap between underlying insights and practical applications, which includes designing a cross-modal deep learning model inherently capable of cross-user generalization, adopting a machine learning-based approach for adaptive system triggering, and introducing a knowledge-distillation-driven structured pruning framework to enhance model efficiency.

• We successfully implement AccCall on three types of Android smartphones and conduct extensive experiments with 20 volunteers in a cross-user scenario. The results show that AccCall achieves excellent and reliable adaptive triggering performance, and enables substantial real-time improvements in SISDR, SISNR, STOI, PESQ, and WER, demonstrating the superiority of our system in enhancing speech quality and intelligibility for phone calls.

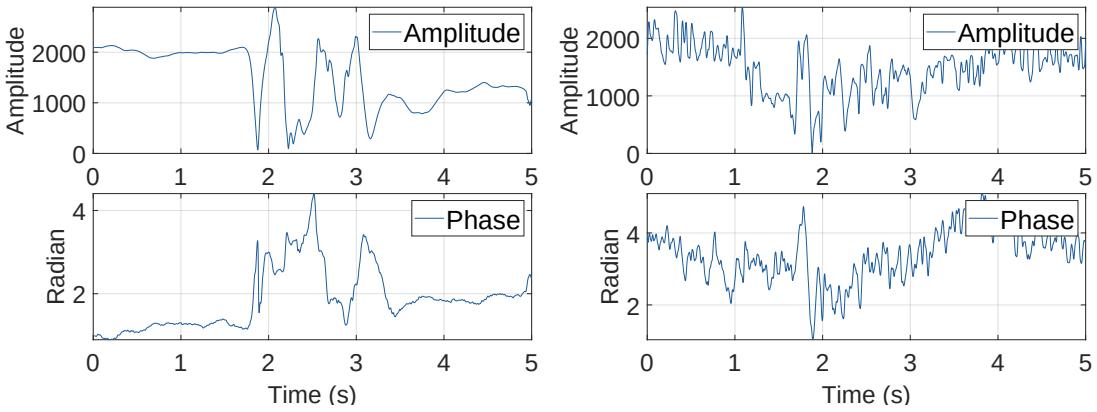
2 Related Work

Audio-only Speech Enhancement: For single-modality method, various filter-based approaches, including Kalman [41, 52], Wiener [1, 67], and adaptive filtering [43, 46], focus on analyzing the statistical characteristics of signals to eliminate the noise. However, these approaches rely on prior knowledge, such as the characteristics of

encountered noise or clean speech, which significantly limits their feasibility in real-world scenarios, as illustrated in Tab. 1. In recent years, deep learning technologies have substantially facilitated advancements in speech enhancement [34, 38, 42, 71, 75]. They use neural network models to learn the mapping relationship between noisy speech and clean speech, producing output that closely approximates the clean speech with noisy input. While these methods can be applied to speech enhancement for handset mode phone calls, they rely solely on audio information, which limits their performance in complex noisy environments, particularly when there are more than two speakers [36, 58] or when the SNR is significantly low [10].

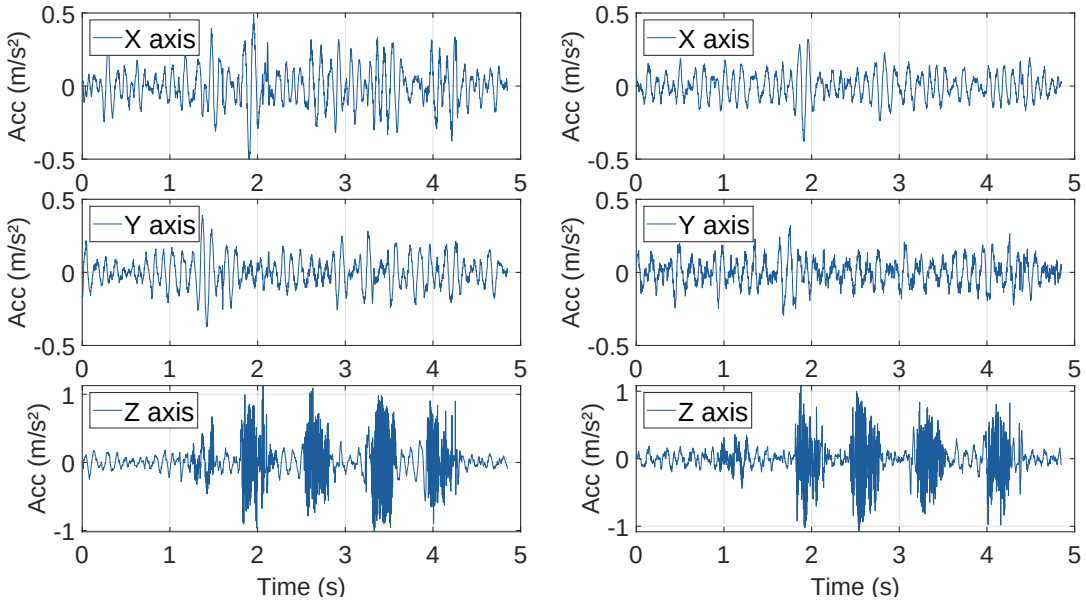
Multi-Modality Speech Enhancement: For multi-modality approaches, various methods utilize different sensors to provide complementary modalities. For example, some studies explore using cameras to capture speakers' facial movements for audio-visual speech enhancement [2, 11, 53], as shown in Tab. 1. However, they must contend with complex visual features, posing a significant burden on smartphones. In addition, the requirements for adequate lighting and privacy concerns restrict its practical usability [3]. Other approaches require additional hardware, such as mmWave [32, 40] and Lidar [50], to function as sensing modalities. While external hardware can indeed enhance speech quality on for phone calls, such solutions suffer from two significant drawbacks: additional cost and lack of portability. As a result, these approaches are often impractical for real-world applications. A practical approach is to leverage speech-related information from the smartphone's microphone as an auxiliary modality. Recent studies focused on leveraging ultrasound signals captured by built-in microphone regarding lip movements to assist in the recovery of clean speech [10, 58, 78]. However, ultrasound is highly affected by air currents in the environment. For example, in windy outdoor conditions, the waveform of ultrasound undergoes significant disturbances, which severely impacts its real-life use. Moreover, ultrasound requires the active emission of signals from speakers, leading to energy loss beyond signal processing [65], which is not conducive to long-duration calls. In contrast, we shift our focus to another built-in sensor: the accelerometer. The accelerometer not only offers lower power consumption [26], but is also unaffected by air currents, making it more suitable for everyday use. We develop a novel, lightweight, and adaptive speech enhancement system based on the built-in accelerometer, specifically designed for cross-user scenarios. Extensive experiments show that our system can significantly enhance real-time phone call quality in low-SNR noisy environments.

Accelerometer Sensing: Accelerometer sensing has been explored in previous studies related to audio topics. Laporte *et al.* [27] explores the use of inertial sensors in earables, including accelerometer, to recognize both verbal and non-verbal gestures. JawSense [22] detects muscle vibrations produced during unvoiced speech to identify unvoiced phonemes. Several studies [6, 51, 55, 72] focus on eavesdropping speech information using motion sensors, including accelerometers, on smartphones or VR headsets. However, they only detect audio events or semantic information instead of reconstructing full-spectrum audio. Additionally, the accelerometer plugged in the head-mounted wearables [18] or earbuds [62] have been integrated with the audio modality to enhance speech quality. However, these systems rely on dedicated equipments, which increase deployment costs. Moreover, they utilize only bone-conduction vibrations of sound as an auxiliary modality. In contrast, speech enhancement in phone call scenarios must be implemented on commercial mobile phones without any hardware modifications. Additionally, the signal captured by the accelerometer, which results from the physical contact between the cheek and the phone, comprises a mixture of vibrations originating from both the bones and the skin. Whether this mixed signal can effectively serve as an auxiliary modality remains unverified. Through a preliminary study, we have demonstrated a strong correlation between the smartphone's built-in accelerometer readings and articulatory gestures. Upon this finding, and by addressing key deployment challenges, including user independence, adaptive system triggering, and running efficiency, we have successfully implemented speech enhancement for phone calls using the smartphone's built-in accelerometer.



(a) Ultrasonic variation in a windless environment (b) Ultrasonic variation in a windy environment

Fig. 2. The impact of air flow on the ultrasound.



(a) Accelerometer readings in a windless environment (b) Accelerometer readings in a windy environment

Fig. 3. The impact of air flow on the accelerometer.

3 Preliminary Study

3.1 Relationship between Speech and Articulatory Gesture

Articulatory gestures are the coordinated movements of speech organs during speech production, determining phoneme formation and acoustic properties [8]. Different phonemes require specific gestures, such as /p/, which involves lip closure and sudden air release, and /s/, which requires the tongue tip near the upper teeth. These gestures distinguish phonemes and enable listeners to differentiate them. Additionally, articulatory gestures

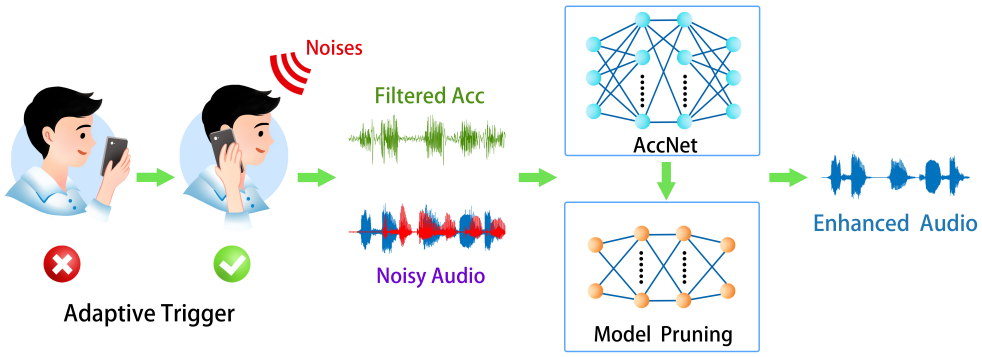


Fig. 4. Overview of AccCall.

influence emotional and semantic expression in speech [7, 9], with more intense gestures indicating anger and softer gestures signaling gentleness. The precision of these gestures is crucial for speech clarity and comprehension. In sum, articulatory gestures are integral to speech production, meaning transmission, and emotional expression.

3.2 Advantage of Accelerometer in Sensing Articulatory Gesture

The coordinated movement of articulators can be captured by a smartphone's microphone via ultrasound signals, aiding in speech enhancement [10, 58, 78]. However, ultrasound signals require active emission, leading to high power consumption, and are vulnerable to interference, especially in outdoor environments with wind. In contrast, the smartphone's built-in accelerometer overcomes these issues. When the phone is held close to the ear, facial movements, including jaw, tongue, and lip motions, as well as vocal cord vibrations, are transmitted to the accelerometer, enabling effective detection of vocal gestures during phone calls.

To verify feasibility, we invite a volunteer to use a Samsung Galaxy S10 smartphone to speak the same sentence in both an indoor environment and an outdoor environment with wind, enabling a comparison of the stability between the two modalities. As shown in Fig. 2(a) and Fig. 3(a), in the indoor environment, both ultrasonic waveform and accelerometer Z-axis readings display distinct patterns. However, in the outdoor environment, the ultrasonic waveform becomes irregular due to its sensitivity to airflow, while the accelerometer waveform maintains a stable pattern consistent with that observed indoors, as shown in Fig. 2(b) and Fig. 3(b). In addition, we observe that, compared to the Z-axis, the readings in the X and Y axes do not exhibit a distinct pattern. This is because when the phone is placed against the face, speech-induced vibrations are primarily manifested in the direction perpendicular to the face, resulting in a significantly higher SNR in the Z-axis compared to the other two axes. Therefore, our system uses only the accelerometer readings in the Z-axis to reduce noise interference. The above results demonstrate the accelerometer's robustness to environmental changes and its strong correlation with articulatory gestures, as reflected in the waveform patterns.

Next, we measure the power consumption of the two modalities during data collection. Measured using Android Debug Bridge (ADB) in wireless debugging mode via a Wi-Fi connection, the energy required for 5-minute ultrasound data collection reaches 59.83 mAh, while the accelerometer consumes only 6.43 mAh, further demonstrating the accelerometer's superior energy efficiency. Overall, accelerometer signals can be regarded as a ideal complementary modality for speech enhancement during phone calls.

4 System Overview

Fig. 4 gives an overview of AccCall. In this study, we develop a speech enhancement system for a common scenario—phone calls—by integrating features from smartphone's built-in accelerometer readings and noisy speech. To achieve this, the system undergoes the four main steps.

(i) Adaptive triggering (Section 5.1). To trigger the speech enhancement system avoid unnecessarily consuming battery and computational resources, we introduce an adaptive triggering scheme that accurately detects call activity states based on accelerometer data, enabling or disabling speech enhancement accordingly. A key aspect of this scheme is selecting the appropriate features. To achieve this, we design and implement a two-stage feature selection strategy, incorporating a mutual information scoring method and an iterative feature elimination strategy.

(ii) Multi-modal speech enhancement (Section 5.2). To enable robust speech enhancement on mobile devices, our system first process both accelerometer and audio signals to extract reliable input features. A low-order high-pass Biquad filter with a tuned cutoff frequency removes low-frequency motion artifacts from accelerometer readings, isolating speech-induced vibrations. Audio features are normalized to ensure consistent scaling across different recording conditions. The processed features are then fed into AccNet, a lightweight multi-modal architecture comprising three core components: a cross-modal fusion module, acc-aided mask generator, and unified loss function, all designed to achieve strong cross-user generalization and ensure stable performance across diverse conditions.

(iii) Model pruning (Section 5.3). To facilitate the efficient deployment of heavyweight DNN models on resource-constrained mobile platforms, we introduce a knowledge-distillation-driven structured pruning framework that enhances model efficiency while maintaining performance. Our approach employs a teacher-student schema to evaluate channel importance by analyzing activation similarity between clean and noisy speech data.

5 System Design

5.1 Adaptive Triggering Scheme

In this section, we present an adaptive trigger scheme that determines call activity states based on accelerometer data, enabling or disabling speech enhancement accordingly. This mechanism ensures effective enhancement while minimizing computational and energy overhead. To ensure effective performance on battery-powered smartphones, we adopt a machine learning-based approach, which offers lower computational cost and data requirements compared to deep learning. Deep models typically demand large-scale annotated datasets and substantial energy resources [4, 63], making them less suitable for mobile deployment. Our method leverages accelerometer signals to detect a sequence of hand and arm movements that typically precede phone-to-face contact. In addition, we observe a distinctive spike in the accelerometer readings at the moment of contact. This empirical signature provides a strong indication of actual face-to-phone contact and serves as a reliable cue to complement the motion patterns. These insights inform the design of our machine learning features, improving the accuracy of phone-to-face contact detection.

Feature Design: Fig. 5 shows the accelerometer data across three call activity states: State 1 (Call Initiation), State 2 (Active Call), and State 3 (Call Termination). State 1 refers to the transition where the phone is raised from an idle or handheld position to near the face, while State 3 represents the transition where the phone is moved away from the face and returned to a resting position. In contrast, State 2 corresponds to the active call state, where the phone remains near the user's face with minimal movement. The adaptive trigger must accurately detect State 1 and State 3 using accelerometer data to activate or deactivate speech enhancement as needed. To achieve this, we extract two primary types of features: jerk-based and gravity-derived features. Jerk measures the rate of change in acceleration, effectively capturing the sharp, short-duration movements characteristic of State 1

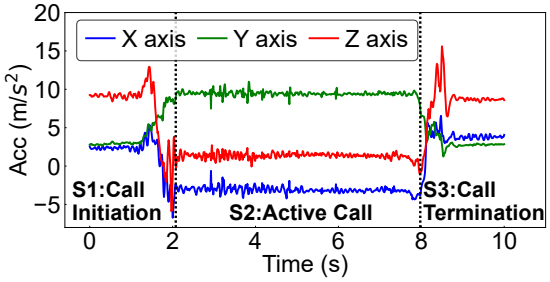


Fig. 5. Accelerometer changes across call states.

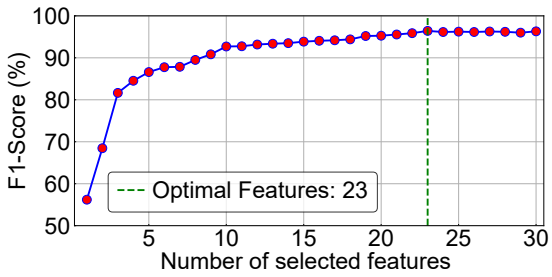
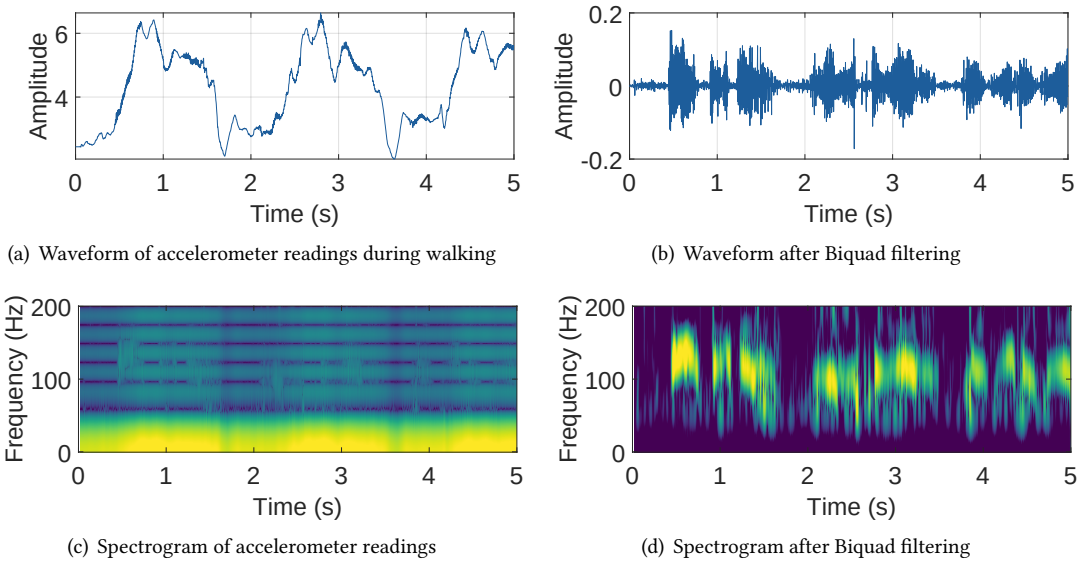


Fig. 6. F1-Score for optimal feature selection.

and State 3. For a single axis (e.g., X-axis) of the accelerometer, jerk at time t is computed as: $J(t) = \frac{a(t))-a(t-1))}{\Delta t}$, where $a(t)$ is the acceleration at time t and Δt is the time difference between consecutive acceleration readings. In addition, gravity-derived features capture variations in the gravity component of acceleration to estimate phone orientation changes. During state transitions, such as in State 1 and State 3, the phone orientation undergoes noticeable shifts between handheld and near-face positions. These movements cause measurable changes in the gravity vector along different axes, allowing the system to detect transitions. We estimate gravity acceleration directly from the accelerometer data, which consists of both gravity and linear acceleration components. The total acceleration can be expressed as $a(t) = g(t) + m(t)$, where $g(t)$ represents the gravity acceleration, and $m(t)$ represents linear acceleration. We apply an exponentially weighted moving average (EWMA) filter to estimate gravitational acceleration. The gravity acceleration at time t is calculated as: $g(t) = \alpha \cdot g(t-1) + (1-\alpha) \cdot a(t)$, where α is the smoothing factor controlling responsiveness. A larger α (close to 1) emphasizes historical data, resulting in slower gravity component changes, ideal for steady motion. A smaller α (close to 0) responds more quickly to current changes, better for dynamic motion. For our implementation, we set $\alpha = 0.2$ to prioritize responsiveness, allowing the filter to quickly adapt to rapid acceleration changes. To improve the robustness of the adaptive trigger, we further extract statistical metrics such as minimum, maximum, standard deviation, range, and root mean square (RMS) from both jerk and attitude-based features. Additionally, we incorporate general time-domain and frequency-domain features. These features are then combined to construct the initial feature set.

Feature Selection: The adaptive trigger is designed for smartphone deployment, where computational efficiency is a key concern. To minimize resource consumption while maintaining robust performance, we adopt a two-stage feature selection process. In the first step, we perform coarse selection using the mutual information filter method, ranking features based on their contribution to classification performance. The top 30 features with the highest scores are retained. This step ensures that only the most informative features are preserved from the initial feature set. In the second stage, we apply fine-grained selection using recursive feature elimination (RFE) with a random forest estimator. RFE iteratively removes the least important features based on their contribution to model performance through cross-validation. The use of a random forest estimator is particularly beneficial, as it inherently captures both linear and non-linear feature dependencies. F1-score is used as the evaluation metric for feature selection as it balances precision and recall, which is crucial for reliable detection. Fig. 6 shows how detection performance changes with the number of selected features. The F1-score increases as more features are included, reaching its peak at 23 features. Beyond this point, additional features offer minimal improvements. This result suggests that selecting only the top 23 features is sufficient to achieve high detection performance while reducing computational overhead.



(a) Waveform of accelerometer readings during walking

(b) Waveform after Biquad filtering

(c) Spectrogram of accelerometer readings

(d) Spectrogram after Biquad filtering

Fig. 7. Performance of high-pass Biquad filtering.

Real-Time Implementation: To integrate the adaptive trigger into a smartphone, we implemented a real-time monitoring mechanism that processes only accelerometer data. The phone's attitude is estimated from these readings, eliminating the need for additional sensors and reducing resource consumption. The system analyzes accelerometer data in sliding windows to detect state transitions and control speech enhancement activation. We set the window size to 1 second to effectively capture transient movements associated with call initiation and termination, ensuring accurate state detection. The step size is set to 0.02 seconds, balancing responsiveness and energy efficiency. The adaptive trigger activates as soon as the user answers a call and continuously monitors phone usage states. It activates speech enhancement upon detecting S1 and deactivates it when S3 is detected.

5.2 Multi-Modal Speech Enhancement

This section describes the data preprocessing steps and the design of AccNet. Accelerometer and audio inputs are first processed to extract reliable features. Following preprocessing, AccNet applies a cross-modal fusion module that integrates speech and accelerometer signals, capturing both shared and modality-specific for robust speaker-independent representation. The acc-aided mask generator enhances separation by leveraging the robust representation from the fusion model, ensuring consistency across users. Finally, the loss function jointly optimizes both speech reconstruction performance and generalization to unseen speakers. Together, these components enable AccNet to achieve strong cross-user generalization and stable performance in diverse conditions.

5.2.1 Preprocessing for Multi-Modal Inputs. Due to the high sensitivity of accelerometers to motion, when a person speaks, motion artifacts such as walking, shaking the head, or nodding can cause the speech vibrations to be overwhelmed, as shown in Fig. 7(a). As depicted in Fig. 7(c), the frequency of body movements is mainly concentrated below 50 Hz, while the fundamental frequency range of speech is above 50 Hz [44]. Therefore, we consider using a high-pass filter to eliminate low-frequency components. Commonly used high-pass filters, such as high-pass FIR filters [37], high-pass Butterworth filters [25], and high-pass IIR filters [59], are typically high-order designs that require complex-valued coefficients, consisting of both real and imaginary parts, to simultaneously control the amplitude and phase of the signal for effective high-pass performance, which demands

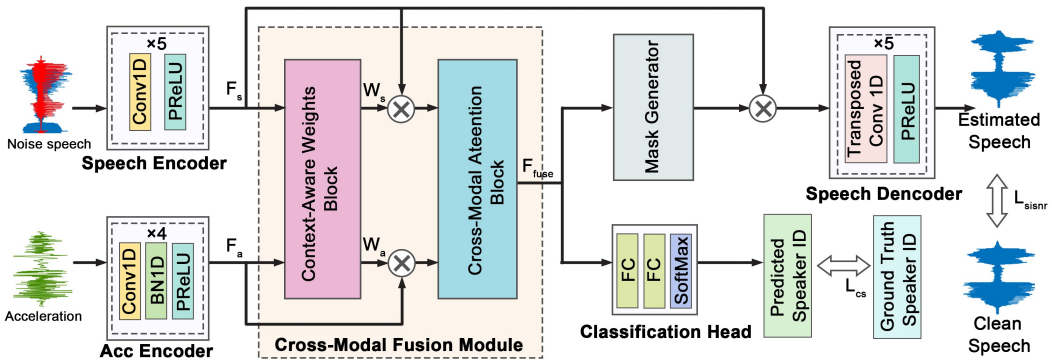


Fig. 8. Overall architecture of AccNet.

significant computational resources. Due to the lightweight requirements for mobile devices, we opt for a low-order high-pass Biquad filter with real-valued coefficients, which provides an accurate cutoff frequency while being computationally efficient compared to high-order filters. We set the cutoff frequency to 50 Hz. As shown in Fig. 7(d), after filtering, most low-frequency components are well attenuated, while the speech-induced pattern is significantly purified. Accordingly, we observe that the speech-induced vibration pattern is significantly highlighted in the time domain, as illustrated in Fig. 7(b).

For the speech stream, input normalization is essential to ensure stable convergence and consistent performance during model training and inference [75]. While accelerometer data is standardized through high-pass filtering, speech signals also require normalization to maintain consistency across varying input conditions. We apply min-max normalization to scale the speech signal within a fixed range, ensuring stable feature distributions during training and inference. The normalized speech signals and filtered accelerometer data serve as inputs to the AccNet, which fully leverages the complementary strengths of both modalities for speech enhancement.

5.2.2 AccNet Structure Design. The core idea behind AccNet is to leverage accelerometer data as a complementary modality to audio signals, taking advantage of its exclusive correlation with the target user's vocal activity. Unlike audio signals, which are often contaminated by environmental noise or interference from other speakers, accelerometer data remains unaffected by such disturbances, capturing motion patterns linked to speech production. AccNet is designed to fully exploit this modality, enhancing its ability to separate target speech. This speaker-independent separation mechanism ensures robust performance across different users, even in challenging acoustic conditions.

Fig. 8 illustrates the overall architecture of AccNet. The network processes data from two modalities: the speech signal (D_s) and the accelerometer signal (D_a). Each modality is passed through a dedicated encoder module to extract modality-specific feature representations, denoted as F_s for the speech signal and F_a for the accelerometer signal. Inspired by established architectures in speech and signal processing [5, 30, 39], the speech encoder consists of a five-layer convolutional network (Conv1D) with Parametric Rectified Linear Unit (PReLU) activation [16], designed to capture both the temporal and spectral characteristics of the speech signal. The accelerometer encoder employs a four-layer Conv1D network with Batch Normalization (BN1D) and PReLU activation to effectively extract motion-related features while maintaining training stability.

The encoded features F_s and F_a are then fed into the cross-modal fusion module, which integrates the complementary information from both modalities to produce a fused feature representation F_{fuse}. This fused feature is

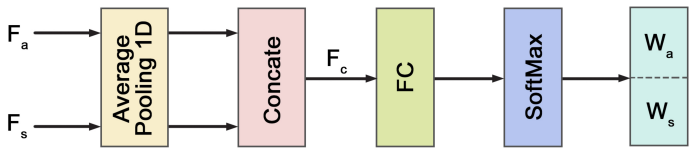


Fig. 9. Structure of context-aware weights block.

subsequently used for two prediction tasks: speech enhancement and speaker identification. For speech enhancement, F_{fuse} is passed through a mask generator module to generate an acc-aided mask. This mask is then applied to F_s to suppress noise and interference, enhancing the target speech. The enhanced speech features are then passed to the decoder to reconstruct the clean speech signal. The decoder, designed to complement the encoder, reverses its transformations to restore the original temporal resolution. It consists of five layers of transposed 1D convolution (Transposed Conv1D), each followed by PReLU activation. This architecture gradually upsamples the features back to the temporal resolution of the original signal while refining the reconstructed speech.

For speaker identification, F_{fuse} is input into a classification head to predict the speaker identity. This auxiliary task improves the model's ability to focus on the target user's speech and enhances generalization to unseen speakers, further improving the overall performance of speech enhancement.

5.2.3 Cross-Modal Fusion Module. The cross-modal fusion module is composed of two key components: the context-aware weights block and the cross-modal attention block, each addressing specific challenges in integrating speech and accelerometer features. This design extracts both modality-specific and shared representations, ensuring effective feature integration while preserving speaker-independent characteristics. By dynamically adjusting feature contributions based on contextual reliability and leveraging cross-modal interactions, this module enhances the robustness of speech enhancement across different users and noisy environments.

The context-aware weights block introduces a dynamic weighting mechanism to mitigate the noise vulnerabilities of each modality. Speech features (F_s) are often affected by acoustic noise from the environment and other speakers, while accelerometer features (F_a) are prone to motion noise caused by unintended user movements. To achieve effective integration, the block computes global representations and dynamically adjusts their contributions based on contextual reliability, as illustrated in Fig. 9. The mechanism of the context-aware weights block can be expressed as:

$$W_s, W_a = \text{Softmax}(\text{Proj}(\text{Concat}(\text{AvgPool}(F_s), \text{AvgPool}(F_a)))), \tag{1}$$

where $\text{AvgPool}(\cdot)$ computes global representations P_s and P_a for the speech and accelerometer features. The concatenated features $F_c = \text{Concat}(P_s, P_a)$ is projected by $\text{Proj}(\cdot)$ to produce the contextual weights W_s and W_a. The computed weights W_s and W_a are applied to the original features as: $F'_s = W_s \cdot F_s$, $F'_a = W_a \cdot F_a$. The weighted features reflect the contextual reliability of each modality. This mechanism ensures robust and reliable integration of speech and accelerometer features, enabling the model to effectively address the noise challenges inherent to each modality.

The cross-modal attention block fuses speech and accelerometer modalities by leveraging a Query-Key-Value attention mechanism [69], as illustrated in Fig. 10. The inputs to this block are the weighted speech features (F'_s) and weighted accelerometer features (F'_a). In this framework, F'_a is projected through a fully connected layer to produce the Query (Q_a), while F'_s is projected to generate the Key (K_s) and Value (V_s). This configuration enables the accelerometer features to guide the selection and weighting of relevant speech features. The attention mechanism computes the relevance of the speech features with respect to the accelerometer features through a

scaled dot product between Q_a and K_s:

$$F_{\mathrm{m}} = \mathrm{Softmax}(Q_a \cdot K_s^T) \cdot V_s, \tag{2}$$

where $\mathrm{Softmax}(Q_a \cdot K_s^T)$ represents the normalized attention scores. The resulting attention-modulated speech features (F_{m}) emphasize the most relevant components of the speech representation based on contextual guidance. Finally, F_{m} are passed through an output projection layer, which produces the final fused representation (F_{fuse}). This cross-modal interaction enables robust speaker-independent feature extraction, as the attention mechanism adapts feature selection based on the relationship between the two modalities rather than speaker identity.

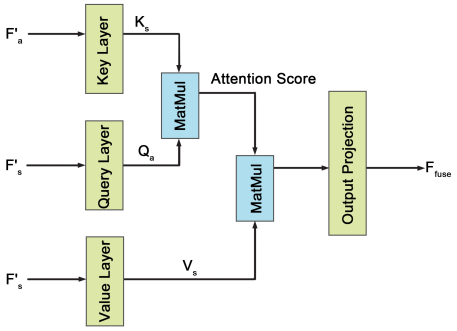


Fig. 10. Structure of cross-modal attention block.

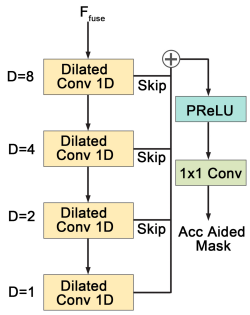


Fig. 11. Structure of mask generator.

5.2.4 Acc-Aided Mask Design. As described in Section 5.2.2, the fused feature representation (F_{fuse}), which integrates complementary information from both accelerometer and speech modalities, is fed into the mask generator to estimate the acc-aided mask. Prior studies [73] have demonstrated that mask-based methods outperform direct speech prediction in speech enhancement tasks. By leveraging the strengths of both modalities, the acc-aided mask selectively enhances target speech components while suppressing noise and interference. Unlike conventional masks generated solely from speech features [35, 56], the acc-aided mask benefits from cross-modal representations, allowing it to capture user-independent motion patterns associated with vocal activity. This characteristic ensures consistent performance across different speakers, enhancing the model's generalization capability and maintaining high separation accuracy even for unseen speakers.

The proposed mask generator, illustrated in Fig. 11, is based on the temporal convolutional network (TCN) architecture [29], which is well-suited for capturing long-range dependencies in sequence modeling tasks. The generator consists of a stack of 1-D dilated convolutional blocks, where the dilation factors follow an exponential progression, $D = 2^i$ ($i = 0, 1, 2, 3$). This design enables the model to effectively capture both short-term and long-term temporal dependencies, allowing for a comprehensive representation of local and global patterns in the speech signal. The concept is analogous to multi-scale processing in computer vision, where features at different receptive field sizes are combined to capture information across various spatial scales [49]. To further enhance training efficiency and stability, each dilated convolutional block includes skip connections, which facilitate gradient flow and prevent degradation during deep network training [17].

Following the dilated convolutional layers, the outputs are processed through a pointwise convolution with a PReLU activation function to generate the acc-aided mask. This final mask aligns with the temporal resolution and feature dimensions of the input speech representation, ensuring precise separation. By leveraging acc-aided information, the mask generator reinforces speaker-independent feature separation, contributing to strong generalization in cross-user scenarios.

5.2.5 Loss Function Design. The proposed loss function for AccNet consists of two components: the speech reconstruction loss and the speaker recognition-enhanced loss.

Speech Reconstruction Loss: For the speech enhancement task, we adopt a time-domain loss function, the Scale-Invariant Signal-to-Noise Ratio (SISNR) loss, to evaluate the quality of the reconstructed speech. Unlike traditional mean squared error (MSE), which focuses solely on magnitude differences and fails to capture phase information, SISNR directly measures the alignment between the estimated speech signal and the ground truth clean speech signal in the time domain. The negative SISNR loss (L_{sisnr}) is defined as:

$$L_{sisnr} = -\frac{1}{B}\sum_{b=1}^{B} 20 \cdot \log_{10}\left(\frac{\|t_b\|}{\|p_{s_b} - t_b\| + \epsilon}\right), \tag{3}$$

where $t_b = \frac{\langle p_{s_b}, g_{s_b}\rangle}{\|g_{s_b}\|^2+\epsilon} g_{s_b}$ is the projection of the estimated speech signal p_{s_b} for batch b onto the ground truth clean speech signal g_{s_b}, representing the aligned component of the estimated signal with the clean speech. Both p_{s_b} and g_{s_b} are zero-mean signals, where $p_{s_b} = p_{s_b} - \mu_{p_s}$ and $g_{s_b} = g_{s_b} - \mu_{g_s}$ with μ_{p_s} and μ_{g_s} denoting their respective means. The small constant ϵ is introduced to ensure numerical stability, preventing division by zero in the computation of projections and ratios.

By removing dependencies on the magnitude scaling of the speech signal, SISNR ensures the model focuses on the relative structure and quality of the signal rather than absolute amplitude, making it more robust to variations in loudness. Additionally, SISNR incorporates both amplitude and phase information, leading to a more perceptually relevant optimization process, critical for high-quality speech reconstruction in complex and noisy environments.

Speaker Recognition-Enhanced Loss: The speaker recognition-enhanced loss is designed to enhance the model's capacity to extract speaker-specific features, thereby facilitating more effective separation of the target user's speech from that of other speakers. The underlying insight is that incorporating speaker information helps the model refine the acc-aided mask. This refined mask improves the overall performance of the speech separation process. As evidenced by prior research [10, 24, 74], knowing which speech belongs to the target speaker allows the model to better adapt to the unique characteristics of the target user, thereby improving separation performance.

To achieve this, the speaker recognition-enhanced loss is introduced to optimize the speaker identification task. The fused feature F_{fuse} is passed through a classification head module to predict the speaker identity. This module consists of two fully connected layers followed by a softmax function, which produces a probability distribution p_i over all possible speakers. The speaker recognition loss is formulated as the cross-entropy loss L_{ce}:

$$L_{\text{ce}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} g_{i,j} \cdot \log(p_{i,j}), \tag{4}$$

where N is the total number of samples in the batch, C is the total number of possible speakers, $g_{i,j}$ represents the ground truth label for sample i as a one-hot encoded vector where $g_{i,j} = 1$ if the j-th speaker is the correct speaker for the i-th sample and $g_{i,j} = 0$ otherwise, and $p_{i,j}$ denotes the predicted probability for the j-th speaker in the i-th sample, output by the softmax function.

Unified Loss Function for Joint Optimization: To jointly optimize speech enhancement and speaker identification, we combine the two loss components into a unified loss function:

$$L = \alpha \cdot L_{sisnr} + \beta \cdot L_{ce}, \tag{5}$$

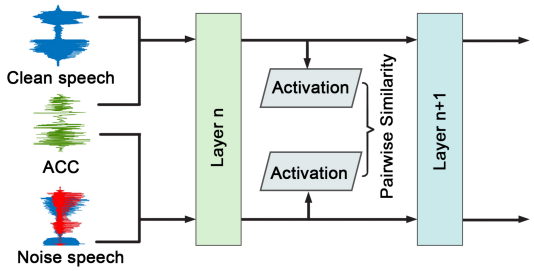


Fig. 12. Channel importance calculation via activation similarity.

where α and β are hyperparameters controlling the contributions of each component. During training, L_{ce} updates the parameters of the cross-modal module, embedding speaker-specific knowledge into F_{fuse}. This enriched representation enhances the effectiveness of L_{sisnr} in reconstructing clean speech. By combining these losses, the model achieves a unified learning process that promotes synergistic interaction between the accelerometer and speech modalities, leading to robust speech enhancement.

5.3 Task-Aware Knowledge Distillation Pruning

In real-time phone call scenarios, mobile devices are sensitive to model size and computational demands. The original AccNet, consisting of 6.6 M parameters and requiring 2.8 G FLOPs, poses significant challenges for efficient deployment on mobile platforms. First, its high computational cost increases battery consumption. Second, it introduces latency, disrupting real-time communication. For instance, on a Samsung Galaxy S10, inference takes 122 ms, causing noticeable delays that degrade user experience. To address these challenges, we apply model pruning to reduce computational overhead, ensuring efficient operation on mobile devices.

Model pruning methods are generally categorized into two main approaches: unstructured pruning and structured pruning. Unstructured pruning removes individual weights identified as less important, resulting in a sparse model. However, sparse models often face compatibility issues with hardware acceleration on most mobile devices, limiting their practical deployment [12, 15]. In contrast, structured pruning removes entire channels, filters, or layers, resulting in dense models with modified architectures. This approach is better suited for mobile hardware platforms, offering improved compatibility.

A common structured pruning method evaluates channel importance based on the L_2-norm of their weights, where channels with smaller L_2-norm are considered less important and pruned accordingly [31]. Specifically, channel importance is computed as: $I_c = \|W_c\|_2 = \sqrt{\sum_k w_{c,k}^2}$, where W_c represents the weights associated with channel c and k denotes the number of filters in the layer. While L_2-norm pruning effectively reduces model size and computational complexity, it considers only weight magnitudes, ignoring their interaction with input data. This static, weight-centric approach often results in substantial accuracy degradation, particularly in tasks like speech enhancement, where effective data-driven feature extraction is crucial.

To address the limitations of L_2-norm pruning, prior studies on knowledge distillation (KD) have demonstrated the effectiveness of leveraging intermediate representations and domain-specific knowledge to guide pruning process [33, 66]. Inspired by these insights, we propose a KD-based structured pruning method that employs a teacher-student schema to evaluate channel importance by analyzing the similarity between intermediate activations of clean and noisy speech data. Unlike static weight-based methods (e.g., L_2-norm pruning), our approach incorporates task-specific information, retaining channels that contribute more to clean speech reconstruction.

The core principle of our method is that channels with high activation similarity between noisy and clean speech data contribute more to clean speech reconstruction. Fig. 12 provides a schematic illustration of this

approach. The input consists of three modalities: accelerometer data, noisy speech, and its corresponding clean speech data, forming paired inputs. These pairs are fed into the AccNet model, and activations are extracted at intermediate layers. To evaluate the contribution of each channel, we compute the pairwise cosine similarity between activations from clean and noisy speech data. Specifically, for each intermediate layer, the importance of each channel is defined as the average cosine similarity between its activations for clean speech and noisy speech across the dataset:

$$I_c = \frac{1}{N}\sum_{n=1}^{N} \mathrm{Cos}(A^c_{\mathrm{m},n}, A^c_{\mathrm{c},n}) = \frac{1}{N}\sum_{n=1}^{N} \frac{\sum_{t=1}^{T} A^c_{m,n}[t] \cdot A^c_{c,n}[t]}{\sqrt{\sum_{t=1}^{T} \left(A^c_{m,n}[t]\right)^2} \cdot \sqrt{\sum_{t=1}^{T} \left(A^c_{c,n}[t]\right)^2}}, \tag{6}$$

where $A^c_{\mathrm{m},n}[t]$ and $A^c_{\mathrm{c},n}[t]$ denote the activations of channel c at time step t for the n-th noisy and clean speech inputs, respectively. $\mathrm{Cos}(\cdot)$ is cosine similarity function, where $\mathrm{Cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$. T represents the temporal length of the activation, and N is the total number of data samples in the dataset.

Channels with lower average similarity scores are considered less relevant for clean speech reconstruction and are pruned. For each intermediate layer, the pruning process is guided by a threshold defined by the pruning percentage P, as follow:

$$C_{\mathrm{p}} = \{c \mid I_c \leq \mathrm{Percentile}(\{I_c\}, P)\}, \tag{7}$$

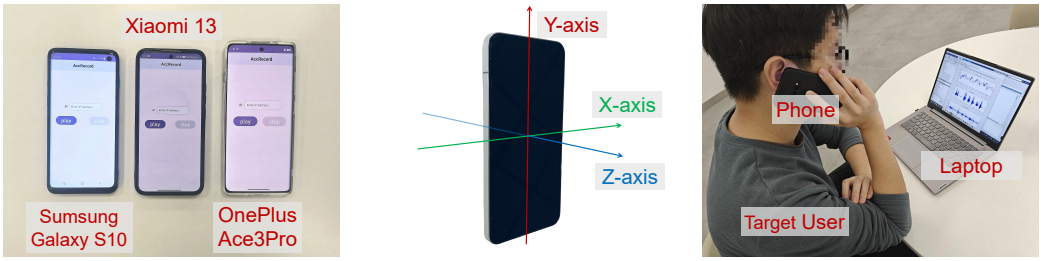
where C_{p} represents the set of pruned channels, P denotes the percentage of channels to prune, and $\mathrm{Percentile}(\{I_c\}, P)$ indicates the threshold corresponding to the P-th percentile of channel importance scores. Once the channels to be pruned are identified, we utilize the open-source library Torch-Pruning [68] to perform structured pruning. This library automatically adjusts the model architecture based on the selected channel set C_{p} for each layer, ensuring an efficient and seamless pruning process.

To mitigate performance degradation, we fine-tune the model for 5 epochs after pruning, allowing it to adapt to the reduced architecture and regain accuracy. By leveraging intermediate activations to assess channel importance, our approach integrates task-specific optimization into the pruning process. This method ensures that pruning decisions are guided by the interaction between the model and input data, resulting in a task-driven and effective reduction of computational complexity.

6 implementation

We have implemented our system on three types of smartphones: the Samsung Galaxy S10 (abbreviated as S10), Xiaomi 13, and OPPO OnePlus Ace 3 Pro (abbreviated as Ace 3 Pro). These devices feature built-in accelerometers and are equipped with Qualcomm SM8150 Snapdragon 2.9 GHz Octa-core, Qualcomm SM8550-AB Snapdragon 3.2 GHz Octa-core, and Qualcomm SM8650-AB Snapdragon 3.3 GHz Octa-core processors, respectively, as shown in Fig. 1(b). For data collection, we develop an app called AccRecord on the mobile phone to capture multi-modal data, including audio and accelerometer readings, with sampling rates of 16 kHz and 400 Hz, respectively, as shown in Fig. 13(a). The collected multi-modal data is then transmitted to a Lenovo ThinkBook 14 via WiFi for further processing in Python 3.11, which includes signal preprocessing, adaptive system triggering, the AccNet framework, and model pruning.

We implement the AccNet model using PyTorch. The detailed encoder and decoder configurations of AccNet, as shown in Fig. 8, are described as follows. The Acc Encoder consists of four Conv1D layers, each with a kernel size of 3, a stride of 1, and channel dimensions of $[16, 32, 64, 128]$. The Speech Encoder begins with an initial Conv1D layer with a kernel size of 80 and a stride of 40, followed by four additional Conv1D layers with a kernel size of 3, a stride of 1, and channel dimensions of $[16, 32, 64, 128]$. The Decoder mirrors this structure but in reverse order, utilizing four Transposed Conv1D layers with a kernel size of 3 and a stride of 1. The final layer employs a Transposed Conv1D with a kernel size of 80 and a stride of 40 for signal reconstruction.



(a) Device type (b) Axes of built-in accelerometer (c) Data collection scenario

Fig. 13. Experimental setup.

We train the AccNet model on an NVIDIA A100 GPU with 80 GB of memory. The model is optimized using a joint loss function with $\alpha = 0.8$ and $\beta = 0.2$ as defined in Eq. 5.2.5. Training is conducted for 30 epochs with a batch size of 32 and a learning rate of 1×10^{-4}. We use the Adam optimizer along with a StepLR scheduler with a step size of 5 to dynamically adjust the learning rate.

7 Evaluation

7.1 Experimental Setup

Data Collecting: We recruit 20 university students, including 4 females and 16 males, among whom 3 are native English-speaking volunteers of Caucasoid descent, and the remaining participants are non-native English-speaking volunteers of Mongoloid descent, with an average age of 24. They are tasked with collecting audio and accelerometer data using an S10. Since the signal from the Z-axis, as shown in Fig. 13(b), of the accelerometer is most sensitive to articulatory movement (introduced in Section 3.2), we focus solely on the Z-axis readings for processing. Before conducting any experiments, we have obtained ethical approval from our institutional review board (IRB). Each volunteer is asked to pronounce 200 sentences, each lasting 5 seconds, from the TIMIT speech corpus [13] in phone call mode within a silent environment, as shown in Fig. 13(c). We generate the noisy speech dataset by synthesizing clean speech with both interfering speech and ambient noise. The interfering speech is sourced from the TIMIT dataset [13], which contains 6,300 sentences from 630 speakers, totaling 3.5 hours of speech. The ambient noise dataset is sourced from AudioSet [14], comprising 3.4 million segments of 5-second sounds from machines, musical instruments, wildlife, and common everyday environmental noises. In detail, the speech collected from each volunteer is mixed with 20 different noise settings. For each noise setting, the number of interfering speeches is uniformly distributed between 1 and 4, and both the interfering speech and ambient noise are different for each clean speech to prevent overfitting. The SNR is uniformly distributed between -5 and 10 dB (average 2.5 dB).

Model Training/Testing: Each training/testing speech sample consists of three parts: clean speech, noisy speech, and the corresponding accelerometer data. All of our experiments are conducted in a cross-user scenario. Unless otherwise specified, each subject participates in all experiments, and we use 4-fold cross-validation to evaluate the system's performance with a user-independent dataset. Specifically, for n subjects, we divide the raw data into 4 sets and repeat the testing process 4 times. In each iteration, data from $\frac{3n}{4}$ subjects are used as the training samples, while data from the remaining $\frac{n}{4}$ subjects serve as the testing samples.

7.2 Evaluation Metrics

We evaluate the performance of AccCall using the following four commonly used metrics:

• SISDR (Scale-Invariant Signal-to-Distortion Ratio) [28]: Signal-to-Distortion Ratio (SDR) [70] is a common metric for evaluating signal quality, representing the ratio of the useful signal to the distortion components in the reconstructed signal. However, SDR is sensitive to variations in signal amplitude. In contrast, SISDR, an improved version of SDR, rescales the target signal's amplitude to maximize its projection onto the estimated signal, thereby eliminating the effect of the scaling factor.

• SISNR [35]: SISNR is an improved version of SNR, designed to eliminate the impact of scaling factor on the SNR. By aligning the estimated signal with the original signal in scale, it reflects only the signal quality rather than the influence of amplitude.

• STOI (Short-Time Objective Intelligibility) [61]: STOI is used to measure the intelligibility of enhanced speech, ranging from 0 to 1, where 1 indicates fully intelligible speech, and 0 indicates completely unintelligible speech.

• PESQ (Perceptual Evaluation of Speech Quality) [48]: PESQ simulates human auditory perception of speech and evaluates quality degradation after transmission or processing, with scores typically ranging from 1 (poor) to 5 (excellent).

• WER (Word Error Rate) [23]: We use a common automatic speech recognition (ASR) model, the Whisper model [45], to convert speech into text in real-world scenarios. Then, we use WER to evaluate the speech intelligibility with AccCall, calculated as:

$$WER = \frac{N_S + N_D + N_I}{N_R}, \tag{8}$$

where N_S denotes the number of incorrectly substituted words in the recognition output, N_D denotes the number of words from the reference text that are missing in the recognition output, N_I denotes the number of extra words in the recognition output, and N_R denotes the total number of words in the reference text. A lower WER indicates higher speech intelligibility.

7.3 Overall Performance

7.3.1 Detection Accuracy of Adaptive Trigger. In this section, we evaluate the effectiveness of the adaptive triggering scheme. Data collection follows the same setup as described in Section 7.1. Each participant is instructed to perform two actions: S1 (call initiation) and S3 (call termination), along with various non-trigger activities and background noise as negative samples to ensure robustness against unintended activations. Each participant contributes 50 samples per class, with each sample lasting 5 seconds. To facilitate precise labeling during data collection, participants annotate their actions by pressing the volume-up button of the smartphone immediately after completing S1 and the volume-down button after completing S3. These annotations serve as ground-truth labels for model training and evaluation. The adaptive trigger classifies three distinct states: S1 labeled as 1, S3 labeled as 2, and non-trigger activities labeled as 0. We evaluate the trigger under the real-world implementation setup specified in Section 5.1. To assess the performance of the adaptive trigger, we adopt precision, recall, and F1-score as evaluation metrics, ensuring a comprehensive analysis of detection accuracy and robustness.

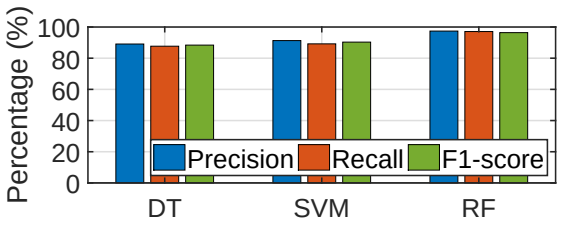


Fig. 14. Adaptive trigger classifier evaluation.

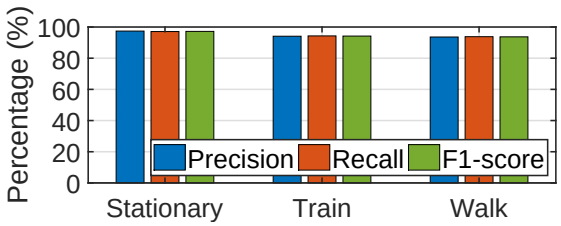


Fig. 15. Adaptive trigger in different scenarios.

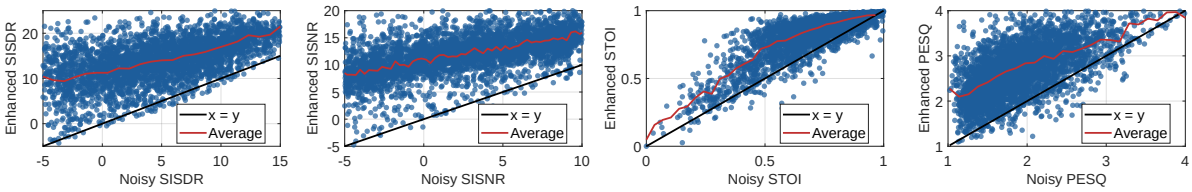
Effectiveness of Selected Features and Classifier Evaluation: In Section 5.1, we employ a two-stage feature selection strategy to identify the top 23 most effective features for adaptive trigger detection. To validate the effectiveness of the selected features, we adopt three different machine learning models, including Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF). The evaluation process follows the same cross-validation setup as described in Section 7.1. Fig. 14 illustrates that all three evaluated models achieve F1-score above 88%, demonstrating the robustness of the selected features. Specifically, DT achieves 89.1% precision, 87.7% recall, and an 88.4% F1-score. SVM obtains 91.3% precision, 89.2% recall, and 90.32% F1-score. The highest performance is observed with RF, which achieves precision of 97.4%, recall of 97.1%, and F1-score of 97.2%. These results confirm the effectiveness of the selected features in distinguishing relevant actions from noise. Furthermore, the superior performance of RF confirms its suitability as the final classifier for the adaptive trigger system. To assess its real-time applicability, we evaluate the inference time on S10, which is measured at 3.65 ms and remains well within the requirement for real-time processing.

Adaptive Trigger Performance in Real-World Scenarios: To evaluate the adaptive trigger system in real-world scenarios, we collect additional data from volunteers in train commuting and walking environments. We then compare its detection performance across three settings: stationary, train commuting, and walking. As shown in Fig. 15, the system achieves 97.4% precision, 97.1% recall, and an F1-score of 97.2% in the stationary environment. In the train commuting environment, performance slightly decreases by 3.3 Δprecision, 2.8 Δrecall and 3 ΔF1-score. The largest drop occurs in the walking scenario, where precision, recall, and F1-score decrease to 93.6%, 93.8%, and 93.7%, corresponding to 3.8 Δprecision, 3.3 Δrecall and 3.5 ΔF1-score. Despite these variations, the system maintains F1-scores above 93% across all environments, demonstrating strong resilience to background noise and motion, ensuring reliable performance in real-world applications.

7.3.2 Performance of AccNet: In this section, we adopt the same training and testing strategies, along with the evaluation metrics outlined in Sections 7.1 and 7.2, to evaluate the performance of AccNet in cross-user scenarios. The scatter plot in Fig. 16 shows the input and output SISDR, SISNR, STOI, and PESQ for each sentence in the testing dataset using AccNet. Taking Fig. 16(a) as an example, the x-axis of a scatter point represents the SISDR of the noisy input, while the y-axis represents the SISDR after applying AccNet. The difference between them is denoted as ΔSISDR. The black line serves as the baseline, indicating no change in SISDR between the input and output data. Similarly, the x-axis of the points on the red line represents the average SISDR of noisy data within a small range of SISDR, while the y-axis represents the average SISDR of the corresponding output data. It is clear that the farther the red line is from the black line, the greater the improvement in ΔSISDR. The results show that when the average SNR of the noisy speech is 2.5 dB, the average values of SISDR, SISNR, STOI, and PESQ are 3.4 dB, 2.62 dB, 0.72, and 1.92, respectively. After applying AccNet, these values increase significantly to 13.32 dB, 11.76 dB, 0.84, and 2.67, with improvements of 9.92 dB, 9.14 dB, 0.12, and 0.75, respectively. This improvement is attributed to the cross-user-driven design of AccNet, which integrates a cross-modal fusion scheme, acc-aided mask design, and user-specific loss function, enabling the model to perform exceptionally well in cross-user scenarios.

Morever, we observe that the lower the quality of the noisy speech, the greater the improvement achieved by AccNet. For example, when the SISDR of the noisy speech is -5 dB, the enhanced speech achieves an average improvement of 15.5 dB, whereas the improvement gradually decreases as the SISDR of the noisy speech increases. This is reasonable because the accelerometer's contribution becomes less significant as the quality of the noisy speech improves. However, even in the best-input scenario with an SISDR of 15 dB, the enhanced speech still achieves an average SISDR improvement of 5.64 dB.

7.3.3 Speech Enhancement Model Comparison. In this section, we evaluate the performance of various speech enhancement models in cross-user scenarios under different noise settings, following the setup described in



(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

Fig. 16. Overall performance of AccNet.

Table 2. Model comparison

Environment	Methods	SISDR (dB)	SISNR (dB)	STOI	PESQ
1S+A	**AccNet**	**14.38**	**12.89**	**0.85**	**2.84**
	VibVoice	10.85	9.67	0.78	2.24
	SEANet	10.56	8.51	0.79	2.36
	SepFormer	11.08	9.44	0.77	2.20
	PHASEN	9.74	8.50	0.76	2.16
2S+A	**AccNet**	**13.18**	**11.59**	**0.83**	**2.65**
	VibVoice	9.72	8.34	0.75	2.12
	SEANet	9.03	6.95	0.76	2.18
	SepFormer	9.57	7.85	0.74	2.07
	PHASEN	7.87	6.74	0.72	2.03
3S+A	**AccNet**	**12.12**	**10.42**	**0.81**	**2.50**
	VibVoice	8.25	6.80	0.72	1.99
	SEANet	7.41	5.17	0.73	2.04
	SepFormer	7.64	5.94	0.70	1.92
	PHASEN	4.89	4.30	0.68	1.84
4S+A	**AccNet**	**10.85**	**9.07**	**0.79**	**2.35**
	VibVoice	6.91	5.23	0.68	1.87
	SEANet	5.66	3.23	0.68	1.90
	SepFormer	5.43	3.82	0.65	1.79
	PHASEN	2.31	1.79	0.62	1.70
2S	**AccNet**	**14.09**	**12.47**	**0.86**	**2.85**
	VibVoice	9.78	8.44	0.76	2.22
	SEANet	8.47	6.18	0.77	2.29
	SepFormer	9.41	7.66	0.76	2.24
	PHASEN	6.70	5.54	0.75	2.22
Average	**AccNet**	**13.32**	**11.76**	**0.84**	**2.67**
	VibVoice	9.76	8.47	0.76	2.12
	SEANet	9.06	6.99	0.77	2.20
	SepFormer	9.64	8.06	0.75	2.07
	PHASEN	7.76	6.79	0.73	2.02
	MetricGAN	6.77	5.42	0.70	1.85

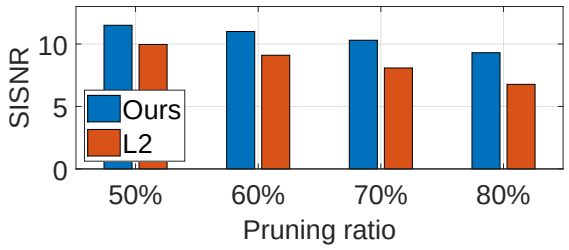


Fig. 17. SISNR comparison of pruning methods.

Section 7.3.2. The testing environment includes five noise settings: "1S+A", "2S+A", "3S+A", "4S+A", and "2S", where "S" represents interfering speech and "A" denotes ambient noise. The SNR of the noisy speech is uniformly distributed within the range of [-5, 10] dB. In detail, the average SNR for the five noise settings is 5 dB, 2.5 dB, 0 dB, -2.5 dB, and 2.5 dB, respectively. We compare AccNet with four state-of-the-art speech enhancement models: VibVoice [18], SEANet [62], SepFormer [56], and PHASEN [76], where VibVoice and SEANet are the accelerometer-based speech enhancement models, while SepFormer and PHASEN are single-modality models. To ensure a fair comparison, we re-implement these four models and train and test them on the AccRecord dataset. For PHASEN and SepFormer, as they are designed only for speech enhancement and not for speech separation, their training is limited to the "1S+A" dataset. For VibVoice and SEANet, we use the same training and testing settings as AccNet.

Tab. 2 presents the testing results. Compared to state-of-the-art speech enhancement methods, AccNet significantly enhances speech quality in multi-speaker and ambient noise environments. AccNet outperforms the single-modality models (i.e., SepFormer, PHASEN) across all four metrics. For example, in the "1S+A" scenario, AccNet achieves average performance of 14.38 SISDR, 12.89 SISNR, 0.85 STOI, and 2.84 PESQ, significantly outperforming SepFormer's performance of 11.8 SISDR, 9.44 SISNR, 0.77 STOI, and 2.20 PESQ. Furthermore, as the number of interfering speakers increases (i.e., lower SNR), the accelerometer sensing modality plays a more critical role, leading to greater performance improvements. AccNet also outperforms the prior acc-aided multi-modality models (i.e., VibVoice and SEANet), primarily due to its cross-user-driven design, which integrates a cross-modal fusion scheme, an acc-aided mask design, and a user-specific loss function, thereby enhancing the model's generalization capability. We observe that the model performs slightly better in the "2S" environment compared to the "2S+A" environment. This is likely because ambient noise exhibits a more complex distribution compared to interfering speech, which increases the model's processing difficulty. In addition, we compare the performance of the pre-trained model MetricGAN+, and the results indicate that it performs significantly worse than other audio-only models, primarily due to its limited generalization capability.

7.3.4 Performance of Model Pruning. In this section, we evaluate the effectiveness of the proposed structured pruning approach in Section 5.3 by comparing it with conventional L_2-norm pruning and assessing its impact on computational efficiency and speech enhancement performance.

Comparison with L2-norm Pruning Approach: We compare our method against L_2-norm pruning approach by analyzing SISNR metrics across different pruning ratios. Fig. 17 shows a significant performance gap between the two approaches, particularly at higher pruning ratio. Notably, at 80% pruning, our method maintains an SISNR of 9.32 dB, which is comparable to the L_2-norm pruning at 50%. This demonstrates the superior ability of our method to preserve performance even under aggressive pruning. Moreover, as the pruning ratio increases, the performance gap between the two methods expands. At 50% pruning, our method outperforms L_2-norm

pruning by 1.56 dB in SISNR. However, at 80% pruning, ΔSISNR increases to 2.55 dB, indicating that our method maintains a more stable SISNR even under higher compression. Comparing both approaches to the original AccNet, which achieves an SISNR of 11.76 dB, we consider a ΔSISNR within 1 dB to be an acceptable trade-off for model compression. Our method stays within this threshold even at 60% pruning, whereas the L_2-norm approach already exceeds this degradation threshold at only 50% pruning. These findings indicate the limitations of L_2-norm pruning, which leads to excessive performance degradation, making it impractical for mobile deployment. In contrast, our approach offers a more effective balance between model efficiency and performance, rendering it more suitable for real-world applications.

Table 3. Impact of model pruning on performance and efficiency

Pruned (%)	SISNR (dB)	Params (M)	FLOPs (G)	Load (ms)	Infer (ms)	Load speed-up	Infer speed-up
Baseline	11.76	6.60	2.80	120.96	122.10	-	-
50%	11.52	2.61	0.93	36.85	35.61	3.28x	2.53x
60%	**11.09**	**2.12**	**0.66**	**27.51**	**21.46**	**4.40x**	**4.20x**
70%	10.35	1.73	0.41	21.84	12.59	5.54x	7.15x

Table 4. Comparison of model efficiency metrics

Model	**AccNet**	VibVoice	SEANet	SepFormer	PHASEN
Params (M)	**2.12**	3.51	8.28	3.25	6.28
Infer (ms)	**21.46**	334	366	117	303
FLOPs (G)	**0.66**	7.55	8.39	2.54	6.88

Trade-off Between Efficiency and Performance: We assess the trade-off between model efficiency and performance by comparing the pruned model to the original AccNet (Baseline). All real-time performance measurements are obtained from the S10 smartphone. "Load" refers to the one-time delay incurred when loading the model into memory, which is primarily influenced by the number of model parameters. "Infer" represents the computational delay per execution and is closely related to the number of FLOPs. As shown in Tab. 3, increasing the pruning ratio significantly reduces both load and inference times, improving computational efficiency. However, beyond 70% pruning, the degradation in speech enhancement performance becomes substantial. At 70% pruning, SISNR drops to 10.35 dB, with 1.31 dB ΔSISNR, exceeding the acceptable threshold of 1.0 dB ΔSISNR for acceptable quality loss. Given this trade-off, we focus on the 50% and 60% pruning settings. At 50% pruning, inference time is reduced to 35.61 ms (2.53× speed-up), while at 60%, inference time further decreases to 21.46 ms (4.20× speed-up). The latter satisfies real-time processing constraints (under 30 ms) while maintaining an acceptable SISNR of 11.09 dB. Based on this trade-off, we select the 60% pruned model as the optimal configuration for deployment. Additionally, we compare our model with prior ones in terms of the number of parameters, FLOPs, and inference time, as shown in Tab. 4. By leveraging model pruning, our model demonstrates a significant advantage.

7.4 Ablation Study

We perform an ablation study to investigate the functionality of various components in AccNet. The testing dataset used here follows the setup described in Section 7.3.2. Tab. 5 summarizes the average experimental results.

Table 5. Ablation study

	SISDR (dB)	SISNR (dB)	STOI	PESQ
AccNet	**13.32**	**11.76**	**0.84**	**2.67**
No Acc Branch	8.76	7.34	0.73	2.12
No Speaker Recognition	12.61	10.97	0.83	2.50
No Cross-Modal Module	10.99	9.84	0.78	2.36
No Mask	11.60	9.96	0.80	2.39

"No Acc Branch" represents the AccNet model without the accelerometer branch. The results indicate that the accelerometer signal significantly enhances model performance, achieving improvements of 4.56 dB ΔSISDR, 4.42 dB ΔSISNR, 0.11 ΔSTOI, and 0.55 ΔPESQ, demonstrating the substantial contribution of the accelerometer modality to speech enhancement.

The "No Cross-Modal Module" experiment replaces the designed cross-modal fusion module with a conventional concatenation operation to combine speech and accelerometer features. This setup evaluates the impact of the proposed fusion module on performance across different evaluation metrics. Removing this module results in a substantial performance reduction, with decreases of 2.33 dB ΔSISDR, 1.92 dB ΔSISNR, 0.06 ΔSTOI, and 0.31 ΔPESQ. These substantial declines highlight the critical role of the cross-modal fusion module in enabling effective feature integration, with its dynamic weighting and attention mechanisms ensuring superior speech enhancement performance compared to a conventional concatenation-based approach.

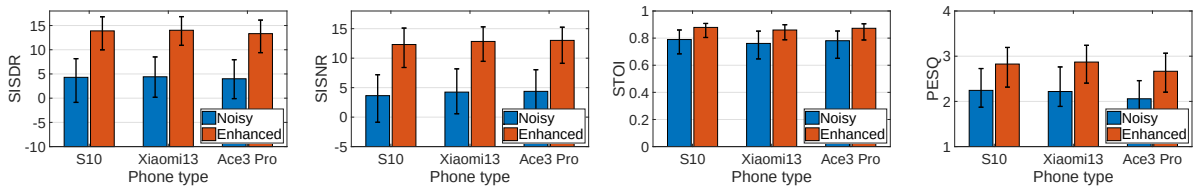
The "No Mask" study evaluates the role of the acc-aided mask in enhancing target speaker separation. In this setup, the fused feature F_{fuse} is fed directly into the speech decoder, bypassing the mask generation process. This adjustment leads to notable performance degradation, dropping by 1.72 dB ΔSISDR and 1.80 dB ΔSISNR, 0.04 ΔSTOI and 0.28 ΔPESQ, respectively. These results emphasise the critical role of the acc-aided mask in isolating target speech effectively and achieving superior speech enhancement compared to directly using fused features.

"No Speaker Recognition" refers to training AccNet without the speaker recognition loss. In this configuration, AccNet achieves average scores of 12.61 SISDR, 10.97 SISNR, 0.83 STOI, and 2.50 PESQ. The absence of speaker-specific knowledge leads to reductions of 0.71 ΔSISDR, 0.79 ΔSISNR, 0.01 ΔSTOI and 0.31 ΔPESQ, highlighting the critical role of this loss in generating precise masks. These findings underscore the importance of integrating speaker-specific information to enhance mask generation and improve speech enhancement for unseen users.

7.5 Evaluation on Impact Factors

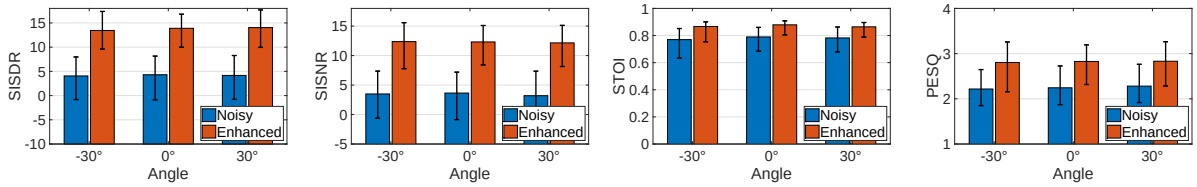
In this section, we evaluate the impact of various factors on system performance. By default, we train the AccNet model using data from 15 volunteers under the same settings as in Section 7.1 and recruit an additional 10 volunteers to participate in all experiments with the "2S" noise setting. Each volunteer speaks 100 sentences, each lasting 5 seconds, from the TIMIT dataset in various scenarios.

7.5.1 Phone Type. Our system is effective across various types of smartphones. We deploy our system on three smartphones—S10, Xiaomi 13, and OnePlus—to evaluate its capability to adapt to different types of devices. Note that the training data is collected exclusively from the S10, while the newly collected data from three different smartphones is used as testing data for comparison. Before collecting data, we configure the accelerometer sampling rate of all smartphones to 400 Hz. As shown in Fig. 18, the average ΔSISDR for the three smartphones is 9.59 dB, 9.61 dB, and 9.31 dB, respectively. The average ΔSISNR is 8.67 dB, 8.61 dB, and 8.65 dB, respectively. The average ΔSTOI is 0.089, 0.099, and 0.092, respectively. The average ΔPESQ is 0.583, 0.651, and 0.608, respectively. These results indicate that the speech enhancement performance among the three smartphones is comparable, largely because accelerometers with the same sampling rate capture fundamentally similar articulatory gesture



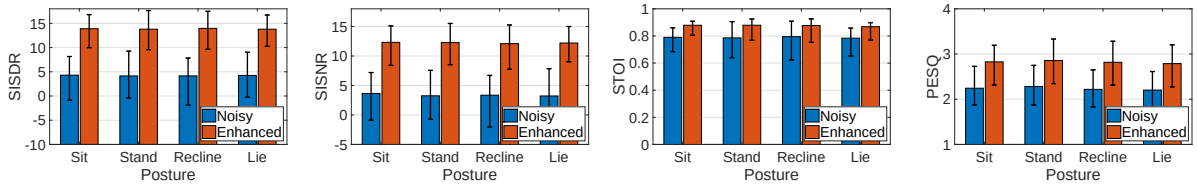
(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

Fig. 18. Impact of different phones.



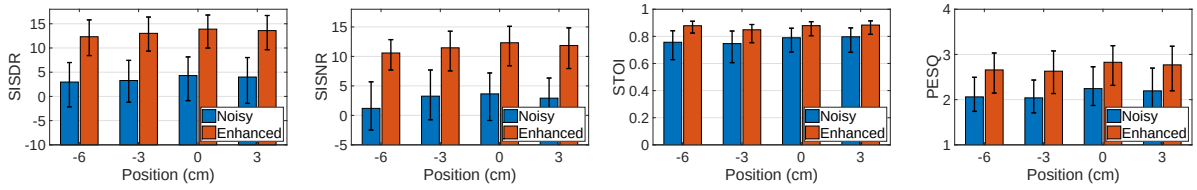
(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

Fig. 19. Impact of different holding angles.



(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

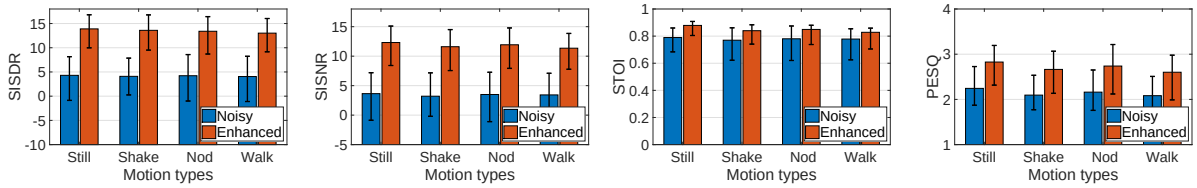
Fig. 20. Impact of different postures.



(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

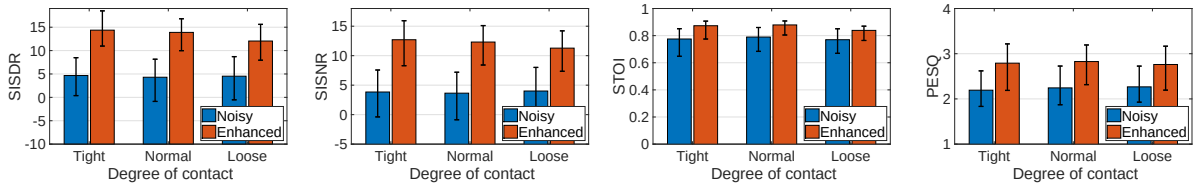
Fig. 21. Impact of different positions.

patterns for the same users, further demonstrating the effectiveness in deploying our system across different types of smartphones.



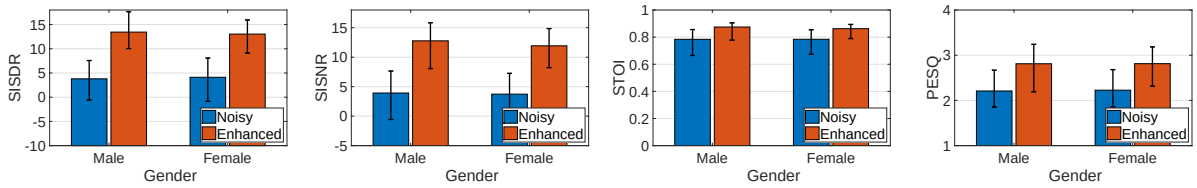
(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

Fig. 22. Impact of different motions.



(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

Fig. 23. Impact of different levels of contact tightness.



(a) SISDR enhancement (b) SISNR enhancement (c) STOI enhancement (d) PESQ enhancement

Fig. 24. Impact of gender.

7.5.2 Holding Angle. Our system remains robust across different holding angles. We instruct volunteers to hold the phone at different angles to examine the impact of angle variations on the system. The reference angle of 0° is defined as the orientation in which the phone's top-to-bottom direction is parallel to the face. Angles measured in the clockwise direction are considered positive, while those in the counterclockwise direction are negative. As shown in Fig. 19, the results indicate that speech enhancement performance at -30° and 30° are comparable to that at the reference angle. For instance, using SISDR as an example, the system improves the performance of noisy speech by 9.39 dB ΔSISDR at -30°, 9.59 dB ΔSISDR at 0°, and 9.88 dB ΔSISDR at 30°, respectively. This suggests that holding angle variations within the common usage range of [-30°, 30°] have minimal impact on the system. This is expected, as variations in holding angle do not impact signal components along the Z-axis.

7.5.3 Posture. Our system achieves comparable performance across different postures. In this section, we ask volunteers to speak in different body postures, including sitting, standing, reclining, and lying. In the sitting posture, volunteers remain upright without resting their back against the chair, whereas in the reclining posture, they lean back against it. Fig. 20 presents the experimental results for both noisy speech and speech enhancement.

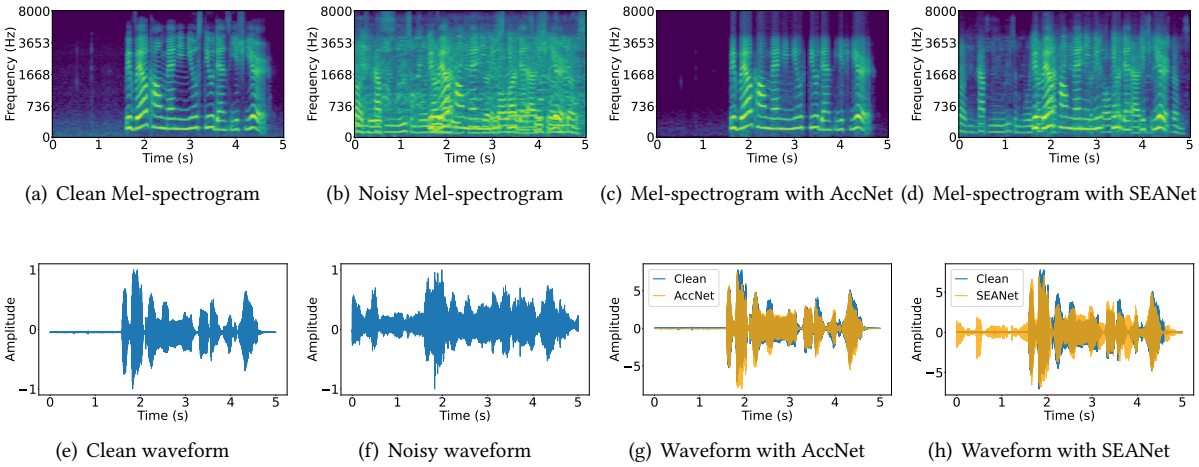
We observe consistently significant gains in SISDR, SISNR, STOI, and PESQ across different postures. Using ΔSISDR as an example metric, our system achieves average scores of 9.59 dB ΔSISDR, 9.66 dB ΔSISDR, 9.81 dB ΔSISDR, and 9.55 dB ΔSISDR for the postures of siting, standing, reclining, and lying, respectively. These results demonstrate that speech enhancement performance remains consistent across all postures. This is reasonable, as changes in posture do not affect the quality of the original speech or the accelerometer readings along the Z-axis, resulting in minimal impact of body posture on the system.

7.5.4 Phone Position. Our system provides comparable speech enhancement performance across different phone positions. We instruct users to hold the phone at various positions along the ear-to-jawline axis to explore the impact of phone placement on system performance. The position where the earpiece speaker contacts the ear is defined as 0 cm, with upward positions assigned positive values and downward positions assigned negative values. We observe that at the 0 cm position, the microphone is closest to the lips, aligning with the highest SISDR of the original noisy speech shown in Fig. 21. Despite the slight differences in original noisy speech, our system demonstrates comparable enhancement performance across the four positions. Using ΔSISDR as an example, the system achieves 9.35 dB ΔSISDR at -6 cm, 9.74 dB ΔSISDR at -3 cm, 9.59 dB ΔSISDR at 0 cm, and 9.58 dB ΔSISDR at 3 cm, respectively. This is reasonable, as the patterns of facial movements and vocal cord vibrations are transmitted via bone conduction to various regions of the cheek, where they are captured by the accelerometer, and remain consistent regardless of the specific position.

7.5.5 Motion Type. Our system can tolerate daily motions of different scales. To evaluate its robustness, we ask volunteers to make phone calls while performing various motions, including remaining still, nodding, shaking the head, and walking to represent typical daily scenarios. As shown in Fig. 22, the system achieves optimal performance when the user remains still. Although slight performance degradation occurs with motion, substantial improvements in SISDR, SISNR, PESQ, and STOI are consistently maintained. Even under the most intense condition, i.e., walking,the system attains an average SISDR gain of 8.96 dB, only 0.63 dB lower than that of the still condition. This robustness is primarily attributed to the integration of a lightweight high-pass Biquad filter in the system design.

7.5.6 Contact Tightness. Our system provides acceptable performance across varying levels of contact tightness between the smartphone and the cheek. We instruct the volunteers to hold the smartphone at different levels of contact tightness with the cheek, including “Tight”, “Normal”, and “Loose”. In the “Normal” scenario, the pressure is typical for a phone call, with “Tight” applying more pressure and “Loose” involving lightly resting the phone on the ear. Note that all levels of contact tightness are defined under the condition that the phone is placed against the cheek. “Loose‘’ denotes the initial light contact between the phone and the user’s cheek, with pressure progressively increased to achieve the “Normal” and “Tight” contact conditions, respectively. As shown in Fig. 23, the results demonstrate an improvement in performance with increasing contact tightness. Using SISDR as an example, the system achieves an average SISDR gain of 7.52 under “Loose”, 9.59 under “Normal”, and 9.70 under “Tight”. We observe that the performance under “Normal” and “Tight” is similar, but noticeably better than under “Loose”. This is reasonable, as under the “Loose” condition, the contact area between the phone and the cheek is much smaller than under “Normal” and “Tight”, reducing the system’s ability to sense speech organ movements. However, even under the “Loose” condition, the system’s performance remains sufficient to meet the requirements for daily use.

7.5.7 Gender. Our system performs slightly worse for female speakers compared to male speakers. We additionally recruit 15 male and 15 female volunteers aged 18-25 to explore the impact of gender on system performance. Fig. 24 presents the experimental results. The results show that the average gains in SISDR, SISNR, STOI, and PESQ for males were 9.65, 8.84, 0.09, and 0.60, respectively, while the average gains for females were 8.92, 8.20, 0.08, and 0.59. The overall performance of females is slightly lower than that of males, primarily due to the higher



(a) Clean Mel-spectrogram (b) Noisy Mel-spectrogram (c) Mel-spectrogram with AccNet (d) Mel-spectrogram with SEANet

(e) Clean waveform (f) Noisy waveform (g) Waveform with AccNet (h) Waveform with SEANet

Fig. 25. Qualitative results.

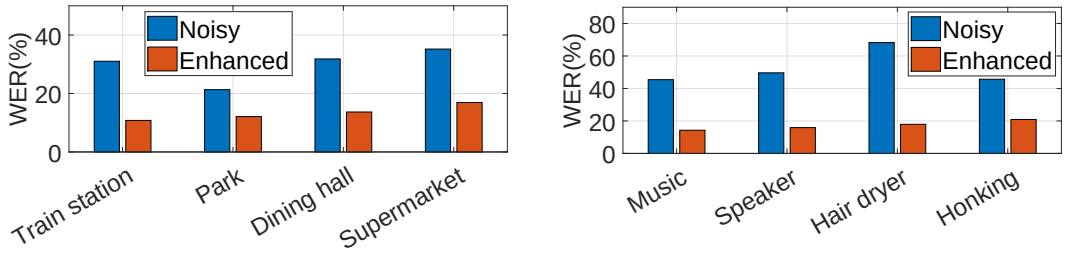
vocal frequencies in females. The smartphone accelerometer we used has a sampling rate of only 400 Hz, which is unable to capture frequency components above 200 Hz. In the future, we could attempt to use smartphones with higher sampling rates to improve performance for females. Although the performance for females is slightly lower, it is still sufficient to meet the requirements of daily scenarios.

7.6 Qualitative Evaluation

Beyond the quantitative results, we provide a qualitative comparison for a sample audio, including time-series waveforms and Mel spectrograms, between different enhancement methods, namely AccNet and SEANet, for the train station scenario, as shown in Fig. 25. We observe that the waveform and Mel-spectrogram patterns of the original clean audio are very distinct, but they become blurred after being mixed with noise. With AccNet, both the waveform and Mel-spectrogram exhibit patterns closely resembling those of the clean signal. In comparison, the results output by SEANet are much blurrier. We attribute this robustness to the design of AccNet, which integrates three key components to improve both separation accuracy and generalization. The cross-modal fusion module combines accelerometer and audio features to learn speaker-independent representations. The acc-aided mask leverages motion cues to support accurate target speech separation under challenging conditions. The unified loss function guides reconstruction while embedding speaker information during training, enabling effective enhancement without speaker-specific tuning. Together, these components contribute to the AccNet's robustness across diverse noisy scenarios.

7.7 Evaluation in Real-World Scenarios

In this section, we evaluate the system's performance under various noisy settings in real-world scenarios. Since clean reference speech is unavailable in these conditions, we use WER as the evaluation metric to assess speech intelligibility. As described in Section 7.2, we first employ the Whisper model [45] to convert the target signal into text and then calculate the WER. For speech enhancement, we use the same pre-trained AccNet model as in Section 7.5 and recruit an additional 10 volunteers to conduct the subsequent experiments. Each volunteer speaks 100 unseen sentences, each lasting 5 seconds, across various scenarios.



(a) WER across different environments

(b) WER across different noise types

Fig. 26. Performance in real-world scenarios.

7.7.1 Different Environments. Our system exhibits strong robustness across a range of real-world environments. To evaluate the practical effectiveness of AccCall, we measured its WER performance in four noisy everyday scenarios: a train station, a park, a dining hall, and a supermarket. As illustrated in Fig. 26(a), AccCall achieves substantial WER reductions, with an average improvement of 16.47% across all environments. The most pronounced gain occurs at the train station—the noisiest setting—where WER drops from 31.03% to 10.79%, marking a 20.24% reduction. In quieter environments such as the park, WER decreases from 21.31% to 12.09% after enhancement. These results underscore AccCall's robustness and consistent ability to improve speech intelligibility under varying levels of background noise.

7.7.2 Different Noise Sources. Our system significantly improves speech intelligibility under diverse noise conditions. To further assess the robustness of AccCall, we evaluate its performance across different noise types, including music, speaker noise, hair dryer noise, and car honking. As depicted in Fig. 26(b), AccCall consistently reduces WER across all noise scenarios, with an average reduction of 35.50%. The greatest improvement occurs in the hair dryer scenario, where WER drops from 68.25% to 17.93%, representing a 50.32% gain. Similarly, in high-noise conditions such as speaker noise and music, WER is reduced by 33.73% and 31.15%, respectively. Even in the challenging car honking scenario, AccCall achieves a 24.78% reduction. These findings demonstrate AccCall's effectiveness in mitigating a wide range of noise types and consistently enhancing speech intelligibility. This is reasonable because AccNet employs accelerometer data as a complementary modality to audio signals for extracting user speech, with the accelerometer data remaining unaffected by noisy inputs.

7.8 System Performance

7.8.1 System Latency. Our system achieves low latency, making it well-suited for real-time speech enhancement in phone calls. We examine the runtime performance of AccCall on three different smartphone: S10, Xiaomi 13, and Ace 3 Pro. Our implementation consists of two parallel threads: the display thread and the speech enhancement thread. For processing audio and accelerometer data, our system utilizes two buffers, each with segment of 1 second of data, updated every 100 ms to ensure real-time speech recovery. Our system includes three blocks: Adaptive trigger, High-pass Biquad filtering (abbreviated as Biquad filtering), and Pruned AccNet. On each phone, we measure the processing time of each block 100 times and calculate the average of these measurements. Tab. 6 summarize the results. Our system achieves average total runtimes of 25.32 ms, 20.30 ms, and 12.00 ms on the S10, Xiaomi 13, and Ace 3 Pro, respectively. Although time delays may vary across phones due to differences in CPU processors, the average total delay consistently remains below the 30 ms delay requirement for voice communication set by the ITU-T standards [20], thereby fully meeting the requirements for real-time processing.

7.8.2 Energy Consumption. Our system is energy-efficient on commercial smartphones. We measure the system's energy consumption using ADB debugging mode. In detail, we perform ten consecutive 5-minute measurements of

Table 6. System Latency

Devices	Adaptive trigger (ms)	Biquad filtering (ms)	Pruned AccNet (ms)	Sum (ms)
S10	3.65	0.21	21.46	25.32
Xiaomi 13	2.74	0.09	17.47	20.30
Ace Pro 3	1.51	0.05	10.44	12.00

the system's energy consumption and compute the average. The results show that for the S10, Xiaomi 13, and Ace Pro 3, the system's average energy consumption is 13.98 mAh, 11.30 mAh, and 7.45 mAh, respectively. Considering that the typical smartphone battery capacity is around 3000 mAh, this energy consumption comfortably meets the requirements for daily use.

8 Limitations and Discussion

Our current implementation for speech enhancement has the following limitations.

High-Frequency Motions: Although the high-pass Biquad filter in our system effectively mitigates the impact of daily motions on accelerometer signals, it remains limited in handling high-frequency motions, such as running and brisk walking. This is primarily because, in addition to low-frequency components, body motions during running also generate high-frequency components that overlap with speech-induced frequencies, rendering a high-pass filter insufficient. In our future work, we will explore more advanced noise reduction schemes to further enhance the system's robustness.

Phone Types: Up to now, our system has been evaluated on only three Android phone models, which is not comprehensive enough. In the future, we need to test our system on a more diverse range of phone types and develop an iOS version to ensure broad compatibility across mainstream smartphone brands.

Expansion to Other Languages: Our speech dataset is collected in English, which may limit the system's speech enhancement capability for other languages due to phoneme differences across languages. According to Section 3.1, articulatory gestures are highly correlated with speech. The articulatory gesture information derived from accelerometer readings serves as an excellent auxiliary modality for speech enhancement, and this underlying principle remains consistent across different languages. To further enhance the system's generalizability, we plan to collect speech data from volunteers covering a diverse range of phonemes in the other languages, such as French, Spanish, and Chinese, and retrain the model, thereby improving its adaptability to multilingual environments.

Usage Modes: The built-in accelerometer can reliably capture the movements of speech organs only when the smartphone is held in contact with the cheek. In contrast, non-contact speech signals are difficult for the accelerometer to capture at a fine-grained level. Therefore, our system is limited to hands-on mode and is not applicable to hands-free mode.

9 Conclusion

This paper presents a lightweight, low-cost, energy-efficient, and environment-independent speech enhancement system for real-time phone calls, leveraging the smartphone's built-in accelerometer. We have addressed three typical challenges in deploying the system. Firstly, we design a cross-modal deep learning model inherently capable of cross-user generalization via three key components, including cross-modal fusion module, acc-aided mask generator, the unified loss function. Next, we employ a machine learning-based approach to accurately distinguish call activities and enable adaptive system triggering, ensuring lower energy consumption and efficient deployment on mobile platforms. Finally, we propose a knowledge-distillation-driven structured pruning

framework to enable real-time processing while maintaining performance. Extensive experiments demonstrate the strong capability of our system in speech enhancement for phone calls, even in real-world scenarios.

Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant No. 62472299). This work is also in part by the National Natural Science Foundation of China under Grant U22A2031. This research is also partly supported by the National Natural Science Foundation of China (No. 62422213, No.62172394), the Beijing Natural Science Foundation (L223034), the Beijing Nova Program (20240484641)

References

[1] M Abd El-Fattah, Moawad Ibrahim Dessouky, Salah Diab, and Fathi Abd El-Samie. 2008. Speech enhancement using an adaptive wiener filtering approach. *Progress In Electromagnetics Research M* 4 (2008), 167–184.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. The conversation: Deep audio-visual speech enhancement. *arXiv preprint arXiv:1804.04121* (2018).

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2019. My lips are concealed: Audio-visual speech enhancement through obstructions. *arXiv preprint arXiv:1907.04975* (2019).

[4] Amazon Web Services. 2024. The Difference Between Machine Learning and Deep Learning. https://aws.amazon.com/compare/the-difference-between-machine-learning-and-deep-learning

[5] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. PMLR, 173–182.

[6] Zhongjie Ba, Tianhang Zheng, Xinyu Zhang, Zhan Qin, Baochun Li, Xue Liu, and Kui Ren. 2020. Learning-based Practical Smartphone Eavesdropping with Built-in Accelerometer.. In *Proceedings of NDSS*.

[7] Peter Birkholz, Lucia Martin, Klaus Willmes, Bernd J Kröger, and Christiane Neuschaefer-Rube. 2015. The contribution of phonation type to the perception of vocal emotions in German: An articulatory synthesis study. *The Journal of the Acoustical Society of America* 137, 3 (2015), 1503–1512.

[8] Catherine P Browman and Louis Goldstein. 1989. Articulatory gestures as phonological units. *Phonology* 6, 2 (1989), 201–251.

[9] Carlos Busso and Shrikanth S Narayanan. 2007. Interrelation between speech and facial gestures in emotional utterances: a single subject study. *IEEE Transactions on Audio, Speech, and Language Processing* 15, 8 (2007), 2331–2347.

[10] Han Ding, Yizhan Wang, Hao Li, Cui Zhao, Ge Wang, Wei Xi, and Jizhong Zhao. 2022. Ultraspeech: Speech enhancement by interaction between ultrasound and speech. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–25.

[11] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. 2018. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619* (2018).

[12] Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574* (2019).

[13] John S Garofolo et al. 1988. DARPA TIMIT acoustic-phonetic speech database. *National Institute of Standards and Technology (NIST)* 15 (1988), 29–50.

[14] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 776–780.

[15] Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems* 28 (2015).

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[18] Lixing He, Haozheng Hou, Shuyao Shi, Xian Shuai, and Zhenyu Yan. 2023. Towards Bone-Conducted Vibration Speech Enhancement on Head-Mounted Wearables. In *Proceedings of ACM MobiSys*.

[19] Matthew B Hoy. 2018. Alexa, Siri, Cortana, and more: an introduction to voice assistants. *Medical reference services quarterly* 37, 1 (2018), 81–88.

[20] RG ITU-T. 2003. Series G: Transmission systems and media digital systems and networks. *Ginebra, Suiza: ITU* (2003).

[21] Dragan Jovanovic, Guillaume Bragard, Dominique Picard, and Sébastien Chauvin. 2015. Mobile telephones: A comparison of radiated power between 3G VoIP calls and 3G VoCS calls. *Journal of Exposure Science & Environmental Epidemiology* 25, 1 (2015), 80–83.

[22] Prerna Khanna, Tanmay Srivastava, Shijia Pan, Shubham Jain, and Phuc Nguyen. 2021. JawSense: recognizing unvoiced sound using a low-cost ear-worn system. In *Proceedings of ACM HotMobile*.

[23] Dietrich Klakow and Jochen Peters. 2002. Testing the correlation of word error rate and perplexity. *Speech Communication* 38, 1-2 (2002), 19–28.

[24] Yuma Koizumi, Kohei Yatabe, Marc Delcroix, Yoshiki Masuyama, and Daiki Takeuchi. 2020. Speech enhancement using self-adaptation and multi-head self-attention. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 181–185.

[25] David Kubanek, Todd Freeborn, Jaroslav Koton, and Norbert Herencsar. 2018. Evaluation of (1+ α) fractional-order approximated Butterworth high-pass and band-pass filter transfer functions. *Elektronika ir Elektrotechnika* 24, 2 (2018), 37–41.

[26] Varun Kumar, Roozbeh Jafari, and Siavash Pourkamali. 2016. Ultra-low power digitally operated tunable MEMS accelerometer. *IEEE Sensors Journal* 16, 24 (2016), 8715–8721.

[27] Matias Laporte, Preety Baglat, Shkurta Gashi, Martin Gjoreski, Silvia Santini, and Marc Langheinrich. 2021. Detecting verbal and non-verbal gestures using earables. In *Proceedings of ACM UbiComp/ISWC*.

[28] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. 2019. SDR–half-baked or well done?. In *Proceedings of IEEE ICASSP*.

[29] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. 2017. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 156–165.

[30] Andong Li, Wenzhe Liu, Chengshi Zheng, Cunhang Fan, and Xiaodong Li. 2021. Two heads are better than one: A two-stage complex spectral mapping approach for monaural speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 1829–1843.

[31] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* (2016).

[32] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenyao Xu, and Kui Ren. 2021. Wavoice: A noise-resistant multi-modal speech recognition system fusing mmwave and audio signals. In *Proceedings of ACM SenSys*.

[33] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*. 2736–2744.

[34] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. 2013. Speech enhancement based on deep denoising autoencoder.. In *Interspeech*, Vol. 2013. 436–440.

[35] Yi Luo and Nima Mesgarani. 2018. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *Proceedings of IEEE ICASSP*.

[36] Yi Luo and Nima Mesgarani. 2019. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing* 27, 8 (2019), 1256–1266.

[37] Sangeeta Mandal, Sakti Prasad Ghoshal, Rajib Kar, and Durbadal Mandal. 2012. Design of optimal linear phase FIR high pass filter using craziness based particle swarm optimization technique. *Journal of King Saud University-Computer and Information Sciences* 24, 1 (2012), 83–92.

[38] Aaron Nicolson and Kuldip K Paliwal. 2019. Deep learning for minimum mean-square error approaches to speech enhancement. *Speech Communication* 111 (2019), 44–55.

[39] Aaron van den Oord. 2016. WaveNet: A Generative Model for Raw Audio. *arXiv preprint arXiv:1609.03499* (2016).

[40] Muhammed Zahid Ozturk, Chenshu Wu, Beibei Wang, Min Wu, and KJ Ray Liu. 2023. Radio SES: mmWave-Based Audioradio Speech Enhancement and Separation System. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), 1333–1347.

[41] K Paliwal and Anjan Basu. 1987. A speech enhancement method based on Kalman filtering. In *Proceedings of IEEE ICASSP*, Vol. 12. 177–180.

[42] Santiago Pascual, Antonio Bonafonte, and Joan Serra. 2017. SEGAN: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452* (2017).

[43] Lalit P Patil, Amiteshwar Bhalavi, Rupesh Dubey, and Mukesh Patidar. 2014. Efficient algorithm for Speech Enhancement using Adaptive filter. *International Journal of Electrical, Electronics and Computer Engineering* 3, 1 (2014), 98.

[44] Alipah Pawi. 2014. *Modelling and extraction of fundamental frequency in speech signals.* Ph. D. Dissertation. Brunel University School of Engineering and Design PhD Theses.

[45] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2003. Robust speech recognition via large-scale weak supervision. In *Proceedings of ICML*.

[46] Pogula Rakesh and T Kishore Kumar. 2015. A novel RLS based adaptive filtering method for speech enhancement. *World Academy of Science, Engineering and Technology, International Journal of Electrical, Computer, Electronics and Communication Engineering* 9, 2 (2015), 624–628.

[47] Ronald E Rice and Douglas E Shook. 2014. Voice messaging, coordination, and communication. In *Intellectual teamwork*. Psychology Press, 341–364.

[48] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. 2001. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *Proceedings of IEEE ICASSP*.
[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 234–241.
[50] Sriram Sami, Yimin Dai, Sean Rui Xiang Tan, Nirupam Roy, and Jun Han. 2020. Spying with your robot vacuum cleaner: eavesdropping via lidar sensors. In *Proceedings of ACM SenSys*. 354–367.
[51] Cong Shi, Xiangyu Xu, Tianfang Zhang, Payton Walker, Yi Wu, Jian Liu, Nitesh Saxena, Yingying Chen, and Jiadi Yu. 2021. Face-Mic: inferring live speech and speaker identity via subtle facial dynamics captured by AR/VR motion sensors. In *Proceedings of ACM MobiCom*.
[52] Stephen So and Kuldip K Paliwal. 2011. Modulation-domain Kalman filtering for single-channel speech enhancement. *Speech Communication* 53, 6 (2011), 818–829.
[53] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6447–6456.
[54] Statista. 2021. . https://www.deadzones.com/2011/05/how-many-cell-phone-calls-are-made-day.html
[55] Weigao Su, Daibo Liu, Taiyuan Zhang, and Hongbo Jiang. 2021. Towards device independent eavesdropping on telephone conversations with built-in accelerometer. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–29.
[56] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. Attention is all you need in speech separation. In *Proceedings of IEEE ICASSP*.
[57] Ke Sun, Chun Yu, Weinan Shi, Lan Liu, and Yuanchun Shi. 2018. Lip-interact: Improving mobile device interaction with silent speech commands. In *Proceedings of ACM UIST*. 581–593.
[58] Ke Sun and Xinyu Zhang. 2021. UltraSE: single-channel speech enhancement using ultrasound. In *Proceedings of ACM MobiCom*.
[59] Shapna Rani Sutradhar, Nazmus Sayadat, Ashfiqur Rahman, Sirajum Munira, AKM Fazlul Haque, and Syed Nazmus Sakib. 2017. IIR based digital filter design and performance analysis. In *Proceedings of IEEE EL-NET)*.
[60] SystematIC Design. 2021. Proximity and Ambient Light Detection. https://systemat-ic.com/cases/proximity-and-ambient-light-detection/
[61] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *Proceedings of IEEE ICASSP*.
[62] Marco Tagliasacchi, Yunpeng Li, Karolis Misiunas, and Dominik Roblek. 2020. SEANet: A multi-modal speech enhancement network. *arXiv preprint arXiv:2009.02095* (2020).
[63] Mohammad Mustafa Taye. 2023. Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers* 12, 5 (2023), 91.
[64] Texas Instruments. 2019. *Optical Proximity Sensing in the Presence of Ambient Light*. Technical Report. Texas Instruments. https://www.ti.com/lit/an/sbau305b/sbau305b.pdf
[65] Mubina Toa and Akeem Whitehead. 2020. Ultrasonic sensing basics. *Dallas: Texas Instruments* (2020), 53–75.
[66] Frederick Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1365–1374.
[67] Navneet Upadhyay and Rahul Kumar Jaiswal. 2016. Single channel speech enhancement: using Wiener filtering with recursive noise estimation. *Procedia Computer Science* 84 (2016), 22–30.
[68] VainF. 2023. Torch-Pruning: Towards Any Structural Pruning. https://github.com/VainF/Torch-Pruning Accessed: February 2025.
[69] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
[70] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. 2006. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing* 14, 4 (2006), 1462–1469.
[71] DeLiang Wang and Jitong Chen. 2018. Supervised speech separation based on deep learning: An overview. *IEEE/ACM transactions on audio, speech, and language processing* 26, 10 (2018), 1702–1726.
[72] Tianshi Wang, Shuochao Yao, Shengzhong Liu, Jinyang Li, Dongxin Liu, Huajie Shao, Ruijie Wang, and Tarek Abdelzaher. 2021. Audio keyword reconstruction from on-device motion sensor signals via neural frequency unfolding. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–29.
[73] Yuxuan Wang, Arun Narayanan, and DeLiang Wang. 2014. On training targets for supervised speech separation. *IEEE/ACM transactions on audio, speech, and language processing* 22, 12 (2014), 1849–1858.
[74] Xiong Xiao, Zhuo Chen, Takuya Yoshioka, Hakan Erdogan, Changliang Liu, Dimitrios Dimitriadis, Jasha Droppo, and Yifan Gong. 2019. Single-channel speech extraction using speaker inventory and attention network. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 86–90.
[75] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. 2014. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM transactions on audio, speech, and language processing* 23, 1 (2014), 7–19.

[76] Dacheng Yin, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. 2020. Phasen: A phase-and-harmonics-aware speech enhancement network. In *Proceedings of AAAI*.

[77] Shang Zeng, Haoran Wan, Shuyu Shi, and Wei Wang. 2023. mSilent: Towards general corpus silent speech recognition using COTS mmWave radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 1 (2023), 1–28.

[78] Qian Zhang, Dong Wang, Run Zhao, Yinggang Yu, and Junjie Shen. 2021. Sensing to hear: Speech enhancement for mobile devices using acoustic signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (2021), 1–30.