SELF-SUPERVISED SPEECH DENOISING WITH MULTI-MODAL DATA

Yu-Lin Wei & Rajalaxmi Rajagopalan & Romit Roy Choudhury
Department of Electrical & Computer Engineering
University of Illinois, Urbana-Champaign
{yulinlw2,rr30,croy}@illinois.edu

Bashima Islam
Department of Electrical & Computer Engineering
Worcester Polytechnic Institute
bislam@wpi.edu

ABSTRACT

We consider the problem of speech enhancement in earphones. While microphones are classical speech sensors, motion sensors embedded in modern earphones also pick up faint components of the user's speech. While this faint motion data has generally been ignored, we show that they can serve as a pathway for self-supervised speech enhancement. Our proposed model is an iterative framework in which the motion data offers a hint to the microphone (in the form of an estimated posterior); the microphone SNR improves from the hint, which then helps the motion data to refine it's next hint. Results show that this alternating self-supervision converges even in the presence of strong ambient noise, and the performance is comparable to supervised Denoisers. When small amount of training data is available, our model outperforms the same Denoisers.

1 INTRODUCTION

Speech enhancement/denoising is of growing interest in the context of modern earphones. Since users are speaking to their earphones in public environments, the SNR of the recorded speech is often weak. This is not only due to heavy ambient interference but also because users tend to speak softer in presence of others nearby. A rich body of work has investigated the general speech denoising problem, however, a modest amount of clean data is still needed to train personalized Denoisers (Schwartz, 2022). Eliminating the need for clean data can relieve users from separately training their earphones. This paper identifies an opportunity for self-supervised speech enhancement through multi-modal sensing, obviating the need to collect noise-free speech data.

Today's earphones include inertial measurement units (IMUs) that sense motion, vibration, and orientation with a sasmpling rate of ≈ 400 Hz. IMUs help with sensing user activities such as jogging, or for detecting when the user has worn the earphone (so audio can be automatically played or paused). Interestingly, when users speak, we find that IMUs can also pick up faint vibrations from the speech signals. These vibrations essentially conduct from the throat to the skull (Jabra, 2022), becoming faint and distorted when they finally arrive near the ear. Nonetheless, these faint and low bandwidth IMU signals *are un-interfered by background noise* Blue et al. (2013). This is in sharp contrast to a microphone that senses the user's speech at full bandwidth of 44 kHz but is heavily corrupted by background noise (see Figure 1). While Denoisers can suppress some of the noise, they need a modest amount of clean data, especially when the noise is non-stationary (Wang and Chen, 2018).

This paper asks: *can the faint but noise-free IMU signal facilitate a self-supervised approach to speech denoising?* In fact, for any signal denoising task, is information from a second sensing modality as effective as having clean training data with a single modality?

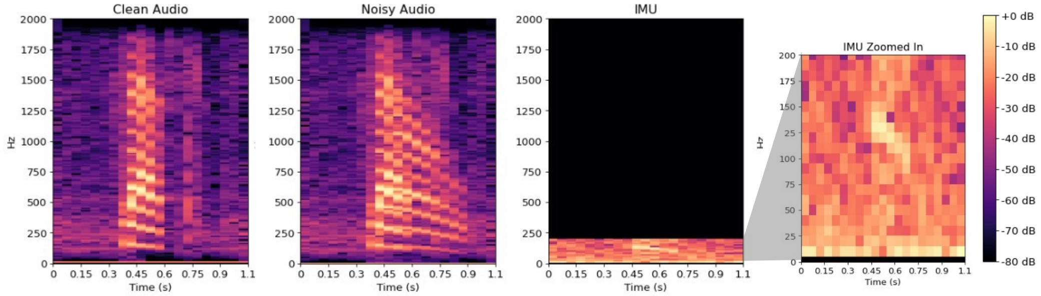

Figure 1: (a) Microphone recording without interference, (b) Microphone recording with interference, (c) IMU recording from earphone. (d) Zoomed in view of IMU signal between $[0, 200]$ Hz.

We propose AlterNet, a two-stage architecture that develops a cooperation between the IMU and the microphone, so each modality can teach and learn from the other. The two stages correspond to a **Translator** and a **Denoiser** that operate on the Short Time Fourier Transform (STFT) of the microphone and IMU data. Briefly, the Translator up-samples the distorted IMU signal to higher-resolution audio, crudely localizing the user's speech in the STFT domain. This localization is extremely crude since the Translator has no clean speech that it can optimize towards; it must use the noisy microphone signal as it's reference. Nonetheless, this crudely localized speech now serves as a reference to the Denoiser, allowing it to slightly improve the speech SNR in the microphone's recording. This slightly enhanced speech then serves as a new reference to the Translator, which localizes the speech slightly better. The iteration converges to an SNR-enhanced speech signal at the output of the Denoiser. Importantly, the alternating iteration is free from clean training data – the corrupt data in the two modalities help each other out of their corruption.

AlterNet inherits the Expectation Maximization (EM) framework and combines it with deep learning. The microphone data is modeled as a mixture of the user's speech S and the background interference B; for every time-frequency (t, f) bin in the STFT, we model the mixture's composition using a hidden random variable $Z_{t,f} = \{S, B\}$. Denoising can be viewed as first estimating $p(Z_{t,f} = S)$, $\forall t, f$, and then using the probabilities as a 2D mask to filter out speech from the noisy microphone data. Unfortunately, given the non-stationary behavior of S and B, EM is unable to estimate $Z_{t,f}$.

The IMU data on the other hand predominantly contains the same user's speech. Although the speech is non-stationary, it presents an opportunity to learn the user's voice fingerprint; this fingerprint can pose as a proxy for the initial posterior $Z^{(1)}_{t,f}$. Learning the fingerprint is not easy either because there is no clean user-speech to train against. Hence, our Translator uses the corrupt microphone data as the reference. The hope is that even a crude voice fingerprint can help to improve the audio SNR, and that SNR-improved audio can be used as a reference to train the Denoiser. A slightly better Denoiser offers a better reference to the Translator, which then generates a better posterior $Z^{(2)}_{t,f}$.

Our surprise in the paper arises from how the very faint and distorted IMU data, which learns a very crude fingerprint (or posterior), can still guide the AlterNet architecture to convergence. While this is an empirical example of success, we believe the core idea could lead to more general ideas of multi-modal self-supervision. Our future work is focused on understanding this generalization.

Summary of Results: With help from 7 volunteers, we gathered IMU and microphone data from earphones and injected interference from a public audio dataset (speech and noise) into the microphone data stream. The *self-supervised* AlterNet model is trained on this unclean dataset (at varying SINR levels). We evaluate the final denoised signal using two metrics: word error rate (WER) from an automatic speech recognizer (ASR) (Yu and Deng, 2016) and scale-invariant signal-to-noise ratio (SI-SNR). Results show that in terms of WER (KWS-35), *self-supervised* AlterNet is comparable with the *supervised audio Denoiser* (trained with clean voice data), achieving less than 5% difference. When we allow AlterNet to also train on clean signals, *supervised* AlterNet exceeds *self-supervised* AlterNet by 16%. In closing, we find that IMU extends one of two advantages — we can either choose to improve denoising performance or relieve the user from collecting clean voice data.

2 Self-supervision with Expectation Maximization (EM)

2.1 Problem Statement

We consider two input streams: a high-resolution audio signal X_m from the microphone and a low-resolution surface-vibration signal X_u from the IMU. Since all recordings are from on-the-fly everyday environments, X_m is composed of three parts: the target speech signal, $X_{m,s}$, the background interfering signals from nearby people, $X_{m,b}$, and the hardware/ambient noises, $X_{m,n}$. Thus, $X_m = X_{m,s} + X_{m,b} + X_{m,n}$. We assume no knowledge of $X_{m,s}$, $X_{m,b}$, or $X_{m,n}$.

The IMU signal X_u consists of two parts: the speech vibration from the user, $X_{u,s}$, and the hardware noise, $X_{u,n}$. We assume no knowledge of either $X_{u,s}$ or $X_{u,n}$. Also, since the vibrations are essentially an outcome of the user's speech, $X_{u,s}$ is a non-linear, low-dimensional projection of $X_{m,s}$, i.e., $X_{u,s} = f(X_{m,s})$. This projection is expected to differ across users, depending on each user's bone, muscle, and tissue conduction properties. Using X_m and X_u as input, `AlterNet`'s output is expected to be a denoised high-resolution audio signal $\hat{X}_{m,s}$, containing only the target user's speech.

2.2 EM background

`AlterNet` inherits ideas from Expectation Maximization (EM) (Moon, 1996). Briefly, recall that EM assumes the observed samples $X = \{x_{1:N}\}$ are drawn from a mixture of K latent distributions $z_{1:K}$. Each distribution z_k is described by parameters θ_k. The samples $X = \{x_1, x_2, ..., x_N\}$ and the number of distributions K are known, and the goal is to estimate: (1) the probability of each sample x_n belonging to a distribution z_k, denoted as Z, and (2) the parameters θ_k for each distribution.

The EM algorithm estimates (1) and (2) iteratively. In the first **expectation step (E-step)**, we estimate Z by computing the posterior distribution of assignment, $p(Z|X, \theta)$, across all the samples X. In the second **maximization step (M-step)**, the algorithm estimates θ by maximizing the expected likelihood $log\, p(X, Z|\theta)$. Combining (1) and (2), EM iteratively optimizes θ with

$$\theta^{(i+1)} = \arg\max_{\theta} \mathbb{E}_{Z\sim p(Z|X,\theta^{(i)})} \left[log\, p(X|Z, \theta) + log\, p(Z|\theta)\right] \tag{1}$$

where $\theta^{(i)}$ denotes the estimate of θ at the i-th iteration. EM must begin with initial values of $\theta_{1:K}$ and the priors $p(z_k)$ for each latent distribution, and the algorithm's convergence is known to be sensitive to this initialization. A comprehensive discussion on EM is available in (LinEM, 2012).

2.3 Modeling Self-Supervision using EM

In mapping `AlterNet` to EM, we have the noisy microphone and IMU as the input data $X = \{X_m, X_u\}$. We denote a time-frequency (TF) bin of the speech X_m as $X_m(t, f)$. The latent variable $Z_m(t, f)$ tracks whether $X_m(t, f)$ belongs to the target speech S or not. The probability $p(Z_m(t, f) = S)$ models the fraction of speech S in that TF-bin. Thus, $Z_m(t, f)$ is essentially a mask that filters out the user's speech from the STFT of X_m.

Our goal is to learn the posterior distribution of $Z_m(t, f)$, however, given that we do not know the parameters θ of the sources S and B, the posterior cannot be calculated. This is where we use the IMU data X_u to construct candidate posterior distributions; the optimal posterior is the one that best matches X_m. The `Translator` learns θ_T to compute this posterior.

Figure 2 attempts to visualize the idea. The green TF-bins on the left are the unpolluted (but heavily aliased) speech signals in X_u. The `Translator` learns to "un-alias" X_u to construct a complete STFT ($Z_m(t, f) = S$ shown in the middle) that best matches X_m (shown on the right). Since X_m is heavily polluted by interference marked in red, the match is very crude at this point. Said differently, the `Translator` empirically learns a crude posterior by minimizing a loss between Z_m and X_m.

Given this posterior, we can now maximize the expected likelihood by training a `Denoiser` with parameter θ_D. The `Denoiser` maximizes the joint probability, $p(X_m(t, f), Z_m(t, f) = S \mid \theta_D)$.

2.4 Learning the Posterior Distribution

To learn the posterior, we design the `Translator` as an auto-encoder that takes X_u as the input and outputs the posterior $p(Z_m(t, f) = S)$. Two observations guide the `Translator` design:

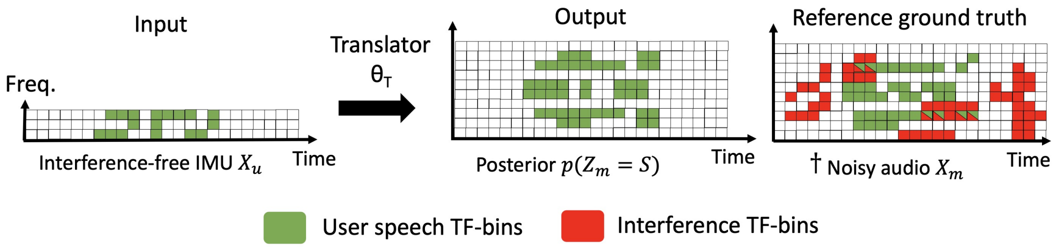

Figure 2: Translator diagram for estimating the posterior of the target speech. † Translator optimizes for the denoised audio as a reference starting from the second cycle.

(a) The Translator does not take the noisy audio X_m as the input because we intend to force the Translator to model only the target speech posterior $Z_m(t,f) = S$. If X_m is given as the input, the output may get biased by the interference.
(b) The ground truth to train the Translator is the denoised signal from the previous iteration $\hat{X}^{i-1}_{m,s}$. Intuitively, if the target speech has high amplitude in a T-F bin, $p(Z_m(t,f) = S)$ will be assigned a proportionally higher probability.

For the first iteration, we use the noisy speech signal X_m as the ground truth. Given the interference is drawn from uncorrelated sources (discussed more in Section 2.6), the posterior begins to localize the speech TF bins. Needless to say, the correlation is very weak at this point. To map the Translator output to a probability $p(Z_m(t,f) = S)$, we apply a Sigmoid function to the Translator's output. We denote this posterior as a mask $M(t,f)$. When $M(t,f)$ is perfectly estimated, the target speech can be retrieved by $X_m \odot M$, where $\odot$ denotes the element-wise product. Thus, Stage 1 in AlterNet can be formally expressed as follows:

$$\theta^i_T = \underset{\theta_T}{argmax} \prod_j \prod_{t,f} p(Z^j_m(t,f) = S \mid X_u,\, \hat{X}^{i-1}_{m,s},\, \theta_T) \tag{2}$$

$$M^{j,i}(t,f) = p(Z^j_m(t,f) = S|X, \theta^i_T) \approx Sigmoid\left(f_{\theta^i_T}(X_u)\right) \tag{3}$$

where $f_{\theta^i_T}(X_u)$ is the converged Translator after the i^{th} iteration, and $j = \{1, 2, \ldots, N\}$ is the sample index. Note that we do not explicitly model the distribution of background interference B because the IMU signal X_u does not capture B. We revisit this point in Section 2.6 and present Translator/Denoiser architecture in Section 3.

2.5 Learning to Maximize the Expected Log Likelihood

We design a Denoiser to maximize the expected likelihood, $E_z\left[p(X, Z|\theta)\right]$. The Denoiser takes $X = \{X_m, X_u\}$ as input and outputs a denoised speech, $\hat{X}_{m,s}$. To train the Denoiser, we use the ground truth as $X_m \odot M$, which is essentially the noisy microphone recording masked by the posterior. The Denoiser is essentially maximizing an *approximate expected likelihood* as follows:

$$\theta^i_D = \underset{\theta_D}{argmax}\; M^i log\, p(X_m \odot M^i, X_u \mid M^i, \theta_D) \tag{4}$$

The objective is an approximate of the expected likelihood because the expectation is over a single term, $p(Z_m(t,f) = S)$. Since we cannot model/learn the interference, we do not have a term corresponding to $p(Z_m(t,f) = B)$.

Finally, since the Denoiser also accepts X_u as an input, we learn a latent embedding of X_m and train this embedding against X_u. Hence the complete loss function for the Denoiser is composed of both the masked audio $X_m \odot M$ and the IMU data X_u (more details in Section 3.2). Once the Denoiser θ^i_D has converged in the i^{th} iteration, its output $\hat{X}^i_{m,s}$ becomes the reference signal for training the next iteration of the Translator, θ^{i+1}_T. The iterations continue till convergence.

2.6 Discussion on Interference

(1) The EM algorithm models the mixture as K distributions parameterized by $\theta_{1:K}$. In AlterNet we only learn the parameters of the user's voice and ignore the background interference in the

expected likelihood. This is because the interference in our application is drawn from diverse on-the-fly environments (e.g., other people's speech, music, everyday activity noise, diffused noise, etc.) and modeling this distribution is intractable. This means that in our expected likelihood, the term corresponding to $p(Z_m(t, f) = B)$ is assumed to be constant, essentially pretending that the interference is drawn from a uniform distribution. If the interference happened to be another person's speech, or some tractable distribution, various other techniques become relevant, including source separation networks (such as Conv-TASNET (Luo and Mesgarani, 2019) and others).

(2) AlterNet assumes the interference across the training samples to be uncorrelated (or weakly correlated at best). If the interference were correlated, the Translator could easily overfit the interference distribution in the training data. This would affect the posterior in the first cycle, further misguiding the downstream convergence (given that EM is known to be sensitive to its initial posterior estimation). In AlterNet, the IMU data and the uncorrelated interference alleviate this concern.

3 Network Architecture

Translator design: Figure 3 shows the proposed network architecture, with the *Translator* on top and the Denoiser below it. The Translator's input is the vibration signal X_u at 400 Hz; the output is the posterior estimation M. Since M needs to be at 4 or 16 kHz, the Translator's task can be viewed as super-resolution. This large up-sampling factor, from 400 Hz to 16 KHz, is prone to overfitting with a conventional auto-encoder. Hence, we design the network as a guided autoencoder to inherit earlier successes in (Lai et al., 2017).

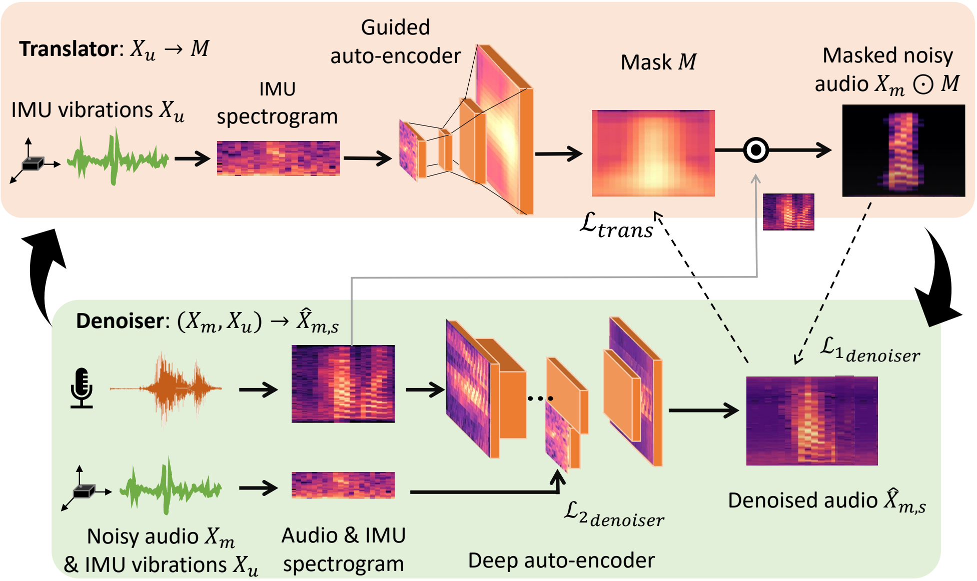
Translator: $X_u \rightarrow M$
Guided auto-encoder
IMU vibrations X_u
IMU spectrogram
Mask M
Masked noisy audio $X_m \odot M$
$\mathcal{L}_{trans}$
Denoiser: $(X_m, X_u) \rightarrow \hat{X}_{m,s}$
$\mathcal{L}_{1_{denoiser}}$
$\mathcal{L}_{2_{denoiser}}$
Denoised audio $\hat{X}_{m,s}$
Noisy audio X_m & IMU vibrations X_u
Audio & IMU spectrogram
Deep auto-encoder

Figure 3: Proposed AlterNet architecture composed of a Translator on top and Denoiser at the bottom, using each other's output as the reference for minimizing the loss function.

The idea is to up-sample the signal in multiple stages, each stage with a small up-scaling factor and a corresponding stage loss. Using a 3-stage decoder, we up-sample the STFT of X_u from 400 Hz to 800, 3200, and finally 16 KHz. The final loss is regularized by the individual stage losses to curb the decoder from overfitting.

Denoiser design: The Denoiser's input is both X_m and X_u and the output is the denoised signal $\hat{X}_{m,s}$. The lack of clean data $X_{m,s}$ precludes an end-to-end network that maps (X_m, X_u) to $\hat{X}_{m,s}$. However, we know that a consistent mapping exists between audio and IMU, i.e., $X_u = f_{imu}(X_{m,s})$, dictated by the bone channel that conducts the throat's vibration. To leverage this, we design an auto-encoder (AE) using only the microphone recording X_m as input, and forcing part of the latent space to match the IMU signal X_u.

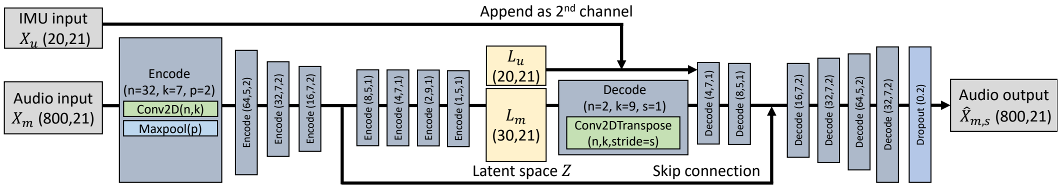

Figure 4: Denoiser architecture: The audio is encoded into a latent space, one part of which mimics the IMU and the other parts are representations of high-frequency speech signals and interference.

We design the AE's latent space as $L = \{L_u, L_m\}$ (see Fig. 4) and force L_u to match the IMU data X_u (loss terms reported in the next section). The remaining $L_m = L \setminus L_u$ is allocated to represent the "gap" between audio and IMU. This gap arises because the IMU only picks-up (aliased) lower-frequencies of the user's voice and is unable to sense the higher-frequency voice harmonics, and neither the interference signal. Hence, we model $L_m = \{L_s^{(hi)}, L_b^{(all)}\}$, where $L_s^{(hi)}$ is a representation of the target's *high* frequency components, and $L_b^{(all)}$ is a highly compressed representation of *all* the background interference. Assuming the interference is uncorrelated to the target user's speech, we add a loss term between $L_b^{(all)}$ and the IMU signal L_u to enforce the average contrast. We also add another loss term between $L_s^{(hi)}$ and L_u to enforce their correlation. Finally, the decoder uses only $\{L_u, L_s^{(hi)}\}$ to reconstruct the denoised audio signal, $\hat{X}_{m,s}$, and trains it against the `Translator`'s output, $X \odot M$.

The `Denoiser` is almost ready, except for one small detail. To utilize the IMU data X_u during test time as well, we concatenate X_u as a second channel, alongside L_u. The 2 channels serve as the first layer of the decoder. To match the dimensions, L_m progresses through one additional layer of decoding. Although we design L_u to match X_u, it's important to concatenate X_u because it does not contain any background interference. Since L_u and X_u both represents the low-frequency target signal, subsequent layers will learn from both modalities.

3.1 Training

The `Translator` begins by training against the noisy audio X_m. After $N_t = 25$ epochs, we freeze the `Translator` and use its output (i.e., the masked audio $X_m \odot M$) to train the `Denoiser` for the next $N_d = 75$ epochs. We denote $(N_t + N_d)$ epochs as one training cycle. We then start the next cycle by freezing the `Denoiser` and using the denoised signal $\hat{X}_{m,s}$ *from the previous cycle* to train the `Translator`. The iteration is performed for $C = 3$ cycles.

Fig. 5 shows snapshots from the start and end of the training process. The first vertical column in Fig. 5 plots the spectrogram of clean target speech $X_{m,s}$ on top, and the interfered audio X_m at the bottom. The top of the second column shows the `Translator`'s mask after the first training cycle; evidently, IMU offers only a crude map M at this time. The bottom of the second column plots the `Denoiser`'s output when it has been trained using the masked audio, $X_m \odot M$. The top of the third column shows the mask after the last training cycle — the improvement is visible. The `Translator` converges well because the interference is uncorrelated, preventing the `Translator` from overfitting to the interference. Finally, the bottom of column 3 shows the denoised audio $\hat{X}_{m,s}$ using the final mask; this is `AlterNet`'s final output.

3.2 Loss functions

Translator's loss function: Aggressive up-sampling is prone to overfitting, so the Translator incorporates a loss function at each stage of the guided auto-encoder. The final loss is a convex combination of Mean Absolute Error (MAE):

$$\mathcal{L}_{trans} = \mathbb{E}_{x \sim p(x)} \frac{\sum_{i=1}^{n} w_i ||D_{-1}(x)_i - T(x)_i||_1}{\sum_{i=1}^{n} w_i} \tag{5}$$

where n is the number of scale-up stages; w_i is the weight for stage i; $D_{-1}(x)_i$ is the Denoiser's output, down-sampled to match stage i; and $T(x)_i$ is the Translator's output after stage i.

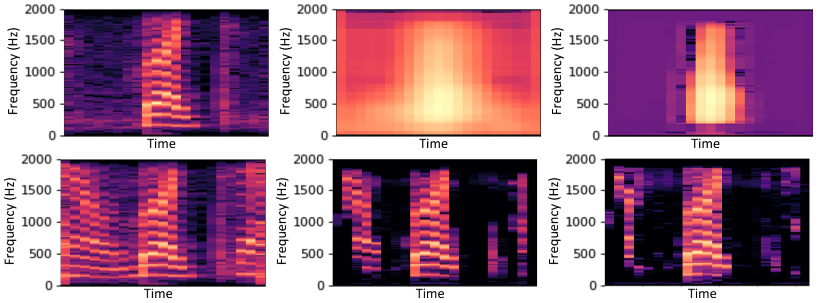

Figure 5: (Column 1) Spectrogram of clean target signal $X_{m,s}$ on top, and the noisy microphone signal X_m at the bottom. (Column 2) Translator's mask after the first cycle on top, and the Denoiser's output after the first cycle at the bottom. (Column 3) Translator's mask after the *last* cycle on top, and the Denoiser's output after the *last* cycle at the bottom.

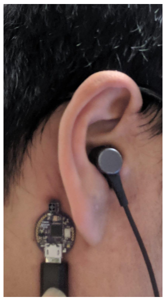

Figure 6: IMU placement for data collection

The Denoiser's loss function is composed of three terms: $\mathcal{L}_{denoiser} = \mathcal{L}_H + \lambda_1 * \mathcal{L}_L + \lambda_2 * \mathcal{L}_C$ where $\mathcal{L}_H$ denotes the *audio reconstruction loss*; $\mathcal{L}_L$ is the *IMU loss* from the latent space; $\mathcal{L}_C$ is the *correlation loss*, and λ is the weighing scalar. The loss functions are defined as:

$$\mathcal{L}_H = \mathbb{E}_{x \sim p(x)} ||T(x) - D(x)||_1 \qquad \mathcal{L}_L = \mathbb{E}_{x \sim p(x)} ||L_u - X_u||_1 \tag{6}$$

$$\mathcal{L}_C = \mathbb{E}_{x \sim p(x)} \Big[\sum_{i,j} \big| \, \rho_{corr}\big(X_u(i), L_b^{(all)}(j)\big) \, \big| \; - \; \sum_{i,k} \big| \, \rho_{corr}\big(X_u(i), L_s^{(hi)}(k)\big) \, \big| \Big] \tag{7}$$

The *Correlation loss* $\mathcal{L}_C$ aims to capture the uncorrelated relationship between the IMU signal X_u and the interference embedding $L_b^{(all)}$, as well as the correlation between the IMU X_u and the high frequency components of the speech, $L_s^{(hi)}$. The negative sign for the second term indicates that higher correlation reduces the loss function (and vice versa for the first term). In the equation, i, j, k are the indices of the dimensions of X_u, $L_b^{(all)}$, and $L_s^{(hi)}$. We calculate the absolute value of correlation coefficients to account for harmonic behaviors in the speech signals.

4 Experiments and Results

Dataset Construction. We recruit 10 volunteers and ask them to wear normal earphones and a separate IMU (Fem, 2022) near their ears – Figure 6 shows the set-up. A separate IMU is needed since today's earphones do not make the raw IMU data accessible. Each volunteer speaks 39 different keywords prescribed by the Google's Speech Command dataset (Warden, 2018), as well as wake words like Google, Siri, Bixby, and Alexa — each word is repeated 10 times. The measurements are performed in a room and serve as $X_{m,s}$. The earphone's microphone samples the audio at 44.1 KHz and we sub-sample to 16 KHz which is standard for state-of-the-art speech algorithms Rybakov et al. (2020); Baevski et al. (2020). The IMU is sampled at 400 Hz. *We have published the dataset on GitHub anonymously* (IMU, 2023b). For 3 users, their raw data achieved poor ASR performance; we removed this data and utilized the 7 remaining users. To the best of our knowledge, this is the first speech dataset composed of synchronized audio and IMU vibrations from the ear location.

To synthesize background interference $X_{m,b}$, we randomly draw audio samples from Google's speech command dataset (Warden, 2018), containing voices of $2,618$ human speakers. Unless specified otherwise, we synthesize the mixture X_m at 5 dB SIR. The IMU signal needs no synthesis, so we automatically have X_u. The total dataset $\langle X_m, X_u \rangle$ is now ready and extends over 1000 hours.

Performance Metrics. (i) Scale-invariant SNR (SI-SNR) is a natural metric to assess speech enhancement. However, issues appear when evaluating self-supervised approaches with SI-SNR, and particularly with multi-modal data. Observe that we no longer have access to clean speech $X_{m,s}$; with self-supervised, the reference becomes $R = (X_{m,s} + X_{m,n})$ since the microphone recordings were performed in everyday environments. Now, since $\hat{X}_{m,s}$ is reconstructing from IMU, it is possible to correctly remove $X_{m,n}$ in many TF-bins. This implies $\hat{X}_{m,s}$ could be closer to $X_{m,s}$ but further from R. To mitigate this we report a range of SI-SNR, where the upper bound of the range is computed by identifying TF-bins that AlterNet has suppressed, and when these bins contain noise below a small threshold, the noise is added back to $\hat{X}_{m,s}$. The actual SI-SNR is expected to lie in this range.

(ii) We also report the *word recognition accuracy* of the denoised signal, using Google's Key Word Spotting Classifier (KWS) with 10 and 35 classes, denoted as KWS10 and KWS35, respectively (Rybakov et al., 2020; Goo, 2022). KWS does not suffer from the SI-SNR issues discussed above.

Models for Comparison.

(1) Unprocessed:	The raw audio without denoising
(2) Supervised Denoiser:	A recent speech enhancement model (Park and Lee, 2016) trained on clean speech; 216K parameters.
(3) Supervised AlterNet:	Our proposed model trained on clean speech; 60K parameters.
(4) Self-Supervised AlterNet:	Our proposed iterative model in Figure 3; 180K parameters.

We publish code in GitHub (IMU, 2023b), and post samples of denoised audio here (IMU, 2023a).

4.1 Overall performance

Table 1 reports comparative results across all metrics and models. Unsurprisingly, *supervised AlterNet* substantively outperforms all models across all metrics. *Self-supervised AlterNet* is comparable to *Supervised denoiser* with negligible performance loss. This distills the contribution of AlterNet to speech enhancement as follows: we can either choose to obtain higher performance gain while requiring the user to provide clean speech data or relieve the user from the data collection burden at the cost of sacrificing that same performance gain. We also observe that for 4 KHz audio, *Self-supervised AlterNet* is as good as (sometimes even better) the *Supervised denoiser* as it is easier to upsample the $400Hz$ IMU signal to 4 KHz rather than 16 KHz.

Table 1: Performance comparison across models and metrics. Note that SI-SNR for self-supervised AlterNet is reported as a range since the ground truth speech is unavailable in such approaches.

Sample rate	Models	SI-SNR (dB)	Acc.(%) KWS10	Acc.(%) KWS35
16KHz	Supervised Denoiser Gain	6.29 ± 1.78	21.76 ± 7.49	18.65 ± 7.69
	Supervised AlterNet Gain	5.57 ± 1.30	27.46 ± 7.30	30.45 ± 6.11
	Self-supervised AlterNet Gain	$[0.49, 4.34]$	15.20 ± 7.24	14.09 ± 8.37
4KHz	Supervised Denoiser Gain	3.0 ± 0.77	16.48 ± 6.81	15.10 ± 7.17
	Supervised AlterNet Gain	4.6 ± 2.52	19.78 ± 6.95	17.86 ± 7.22
	Self-supervised AlterNet Gain	$[4.0, 6.47]$	15.21 ± 7.78	13.91 ± 9.40

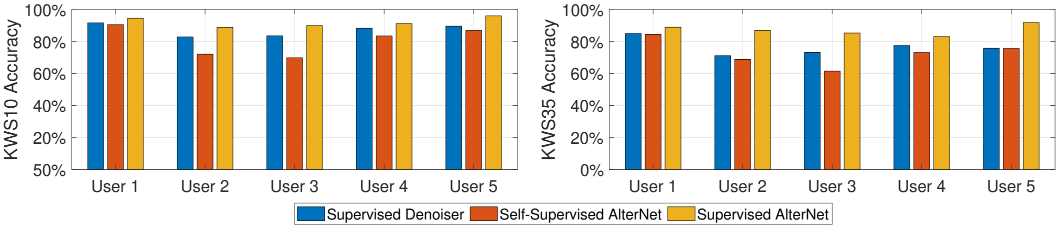

Figure 7: Performance across different users for (a) KWS 10, (b) KWS 35 accuracy.

4.2 Ablation Study

Variation across users: Figure 7 plots the SI-SNR and KWS accuracy across 5 random users from our dataset. *Supervised AlterNet* consistently benefits from both the IMU and the clean-data supervision while the *Supervised Denoiser* and *Self-supervised AlterNet* are mostly comparable. We observe that users (e.g., user 3) with a higher pitch experience less gain from the IMU as it fails to capture the high frequencies; the *Supervised Denoiser* avails an advantage for these scenarios.

Varying mix of clean and interfered data: In the average case of earphone applications, users will speak in a combination of silent and noisy environments. Thus, evaluating the Self-supervised

AlterNet's performance in a mixed scenario is crucial as it has no access to a clean signal. Figure 8 shows that the gain of *Self-supervised AlterNet* over *Supervised Denoiser* is not affected by the fraction of clean signals in both the train and test dataset.

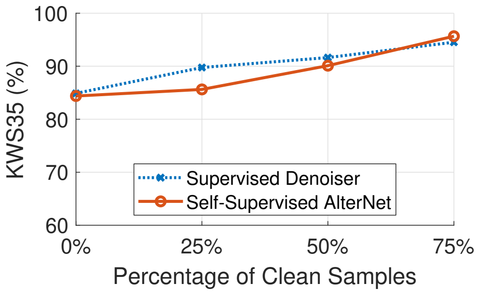

Figure 8: AlterNet offers gain with an increasing percentage of clean data in train set.

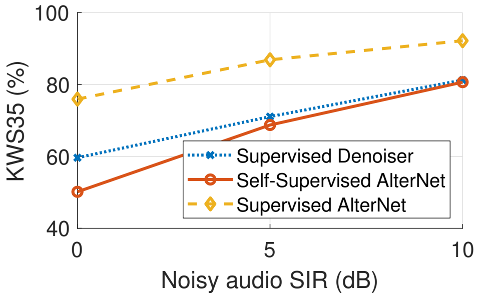

Figure 9: Performance versus KWS35 regimes.

Varying interference: Figure 9 plots SI-SNR against varying SIR (Signal to Interference Ratio) of the training/testing data. The contribution of IMU grows as the SIR drops since the additional IMU modality becomes more valuable under more noisy environments. This explains why *Self-supervised AlterNet* outperforms *supervised Denoiser* at low SIRs but worsens at higher SIRs where the penalty of self-supervision offsets the gain from IMU's guidance.

5 Related Work

Audio and Motion modalities for Speech Enhancement: The most closely related work is SEANet Tagliasacchi et al. (2020) that uses both audio and IMU through a wave-to-wave fully convolutional generator and discriminator architecture. SEANet assumes the availability of clean speech, placing the onus of clean data on the user. Another recent work Wang et al. (2021) uses "alias unfolding" to reconstruct user speech from low resolution IMU motion signals. This work uses only the IMU signal to reconstruct/classify a given set of keywords, using knowledge of phonemes. Although the speech enhancement task is different, our Translator inherently performs "anti-aliasing" to up-sample the IMU signal to speech, and does not need clean phoneme data to train the network.

Multi-modality learning: Multi-modal deep learning is becoming popular for various speech-related applications. A particularly growing trend is in audio-visual speech enhancement and source separation Hou et al. (2018); Gabbay et al. (2017; 2018); Ephrat et al. (2018); Afouras et al. (2018); Lu et al. (2019) where the audio and visual modalities are acquired together. However, these approaches have thus far relied on clean training data. We believe AlterNet–style approaches can be built atop the existing creative ideas.

Self-supervision: Several works Chen et al. (2021); Cheng et al. (2021) have incorporated self-supervision for audio processing tasks . Authors in Wang et al. (2020) learn a latent representation of a limited set of clean speech and use noisy speech to share a latent representation with the clean examples (reducing the burden of clean data). This bears similarity to our paper, but AlterNet relaxes the assumption on the initial clean set (via the guidance from the second IMU modality). Another work, MixIT Wisdom et al. (2021), presents great gains in both speech separation and enhancement. Self-Supervised gain of AlterNetat 0 dB SINR setting is comparable to MixIT gain for source separation (10 dB on anechoic setting and 4 dB in reverberant setting). However, MixIT assumes both the target audio and interference audio to be speech (same distribution), as a result, it fits more in the indoor conversation scenario. Moreover, MixIT takes the full noisy sentence as the input, which might not be possible for keyword spotting scenarios like audio assistant.

6 Conclusion

Learning from diverse sources of unlabelled everyday data remains a desirable property of deep learning. This paper shows possibilities in the specific context of speech enhancement with multi-modal data (microphone and IMU). The core idea allows each modality to build upon the other, cooperatively extracting the latent patterns from the noisy, unlabelled data. There still remains a performance gap from purely supervised techniques, however, the core ideas of iterative learning between modalities offers promise. We intend to continue expanding on this idea of alternating learning, and explore their generalization to other modalities and applications beyond speech enhancement.

REFERENCES

Femtobeacon-atsamr21e-lwmesh-20190615-brochure., 2022. URL `https://downloads.femto.io/FemtoBeacon-ATSAMR21E-LWMesh-20190615-Brochure.pdf`.

google-research/kws_streaming at master, 2022. URL `https://github.com/google-research/google-research/tree/master/kws_streaming`.

`AlterNet` demo, 2023a. URL `https://alter-net.github.io/AlterNet/`.

`AlterNet` dataset, 2023b. URL `https://github.com/Alter-Net/Dataset`.

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

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Misty Blue, Maranda McBride, Rachel Weatherless, and Tomasz Letowski. Impact of a bone conduction communication channel on multichannel communication system effectiveness. *Hum. Factors*, 55(2):346–355, April 2013.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv preprint arXiv:2110.13900*, 2021.

Ying Cheng, Mengyu He, Jiashuo Yu, and Rui Feng. Improving multimodal speech enhancement by incorporating self-supervised and curriculum learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4285–4289. IEEE, 2021.

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

Aviv Gabbay, Asaph Shamir, and Shmuel Peleg. Visual speech enhancement. *arXiv preprint arXiv:1711.08789*, 2017.

Aviv Gabbay, Ariel Ephrat, Tavi Halperin, and Shmuel Peleg. Seeing through noise: Visually driven speaker separation and enhancement. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 3051–3055. IEEE, 2018.

Jen-Cheng Hou, Syu-Siang Wang, Ying-Hui Lai, Yu Tsao, Hsiu-Wen Chang, and Hsin-Min Wang. Audio-visual speech enhancement using multimodal deep convolutional neural networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2):117–128, 2018.

Jabra. Commercial earphones equipped with imu, 2022. URL `https://www.jabra.com/business/office-headsets/jabra-motion#6630-900-105`.

Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

LinEM. An introduction to expectation-maximization, 2012. URL `https://courses.csail.mit.edu/6.867/wiki/images/b/b5/Em_tutorial.pdf`.

Rui Lu, Zhiyao Duan, and Changshui Zhang. Audio–visual deep clustering for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(11):1697–1712, 2019.

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

Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

Se Rim Park and Jinwon Lee. A fully convolutional neural network for speech enhancement. *arXiv preprint arXiv:1609.07132*, 2016.

Oleg Rybakov, Natasha Kononenko, Niranjan Subrahmanya, Mirko Visontai, and Stella Laurenzo. Streaming keyword spotting on mobile devices. *arXiv preprint arXiv:2005.06720*, 2020.

Eric Schwartz. Alexa will automatically adjust volume to be heard when it's loud, 2022. URL `https://voicebot.ai/2021/09/02/alexa-will-automatically-adjust-volume-to-be-heard-when-its-loud/`.

Marco Tagliasacchi, Yunpeng Li, Karolis Misiunas, and Dominik Roblek. Seanet: A multi-modal speech enhancement network. *arXiv preprint arXiv:2009.02095*, 2020.

DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018.

Tianshi Wang, Shuochao Yao, Shengzhong Liu, Jinyang Li, Dongxin Liu, Huajie Shao, Ruijie Wang, and Tarek Abdelzaher. 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):1–29, 2021.

Yu-Che Wang, Shrikant Venkataramani, and Paris Smaragdis. Self-supervised learning for speech enhancement. *arXiv preprint arXiv:2006.10388*, 2020.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Scott Wisdom, Aren Jansen, Ron J Weiss, Hakan Erdogan, and John Hershey. Sparse, efficient, and semantic mixit: Taming in-the-wild unsupervised sound separation. 2021.

Dong Yu and Li Deng. *Automatic speech recognition*, volume 1. Springer, 2016.