PhyAug: Physics-Directed Data Augmentation for Deep Sensing Model Transfer in Cyber-Physical Systems

Wenjie Luo[*]
Singtel Cognitive and AI Lab for Enterprises
Nanyang Technological University
Singapore

Zhenyu Yan[*]
Singtel Cognitive and AI Lab for Enterprises
Nanyang Technological University
Singapore

Qun Song[*]
ERI@N, Interdisciplinary Graduate School
Nanyang Technological University
Singapore

Rui Tan[*]
Singtel Cognitive and AI Lab for Enterprises
Nanyang Technological University
Singapore

ABSTRACT

Run-time domain shifts from training-phase domains are common in sensing systems designed with deep learning. The shifts can be caused by sensor characteristic variations and/or discrepancies between the design-phase model and the actual model of the sensed physical process. To address these issues, existing transfer learning techniques require substantial target-domain data and thus incur high post-deployment overhead. This paper proposes to exploit the first principle governing the domain shift to reduce the demand on target-domain data. Specifically, our proposed approach called PhyAug uses the first principle fitted with few labeled or unlabeled source/target-domain data pairs to transform the existing source-domain training data into augmented data for updating the deep neural networks. In two case studies of keyword spotting and DeepSpeech2-based automatic speech recognition, with 5-second unlabeled data collected from the target microphones, PhyAug recovers the recognition accuracy losses due to microphone characteristic variations by 37% to 72%. In a case study of seismic source localization with TDoA fingerprints, by exploiting the first principle of signal propagation in uneven media, PhyAug only requires 3% to 8% of labeled TDoA measurements required by the vanilla fingerprinting approach in achieving the same localization accuracy.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; • **Computing methodologies** → *Neural networks*; • **Hardware** → *Sensor applications and deployments*.

KEYWORDS

Cyber-physical system, neural networks, data augmentation, domain adaptation

ACM Reference Format:
Wenjie Luo, Zhenyu Yan, Qun Song, and Rui Tan. 2021. PhyAug: Physics-Directed Data Augmentation for Deep Sensing Model Transfer in Cyber-Physical Systems. In *The 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021) (IPSN '21), May 18–21, 2021, Nashville, TN, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3412382.3458255

1 INTRODUCTION

Recent advances of deep learning have attracted great interest of applying it in various embedded sensing systems. The deep neural networks (DNNs), albeit capable of capturing sophisticated patterns, require significant amounts of labeled training data to realize the capability. A sensing DNN trained on a design dataset is often observed run-time performance degradations, due to *domain shifts* [12]. The shifts are generally caused by the deviations of the sensor characteristics and/or the monitored process dynamics of the real deployments from those captured by the design dataset.

[*] Also with School of Computer Science and Engineering, Nanyang Technological University, Singapore.

IPSN '21, May 18–21, 2021, Nashville, TN, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8098-0/21/05…$15.00
https://doi.org/10.1145/3412382.3458255

Transfer learning [17] has received increasing attention for addressing domain shifts. It is a cluster of approaches aiming at storing knowledge learned from one task and applying it to a different but related task. Under the transfer learning scheme, ideally, with little new training data, we can transfer a DNN trained from the *source domain* (i.e., the design dataset) to the *target domain* (i.e., the sensing data from the real deployment). However, the prevalent transfer learning techniques, such as *freeze-and-train* [21] and *domain adaptation* [17], require substantial training data collected in the target domain. The freeze-and-train approach retrains a number of selected layers of a DNN with new target-domain samples to implement the model transfer. Domain adaptation often needs to train a new DNN to transform the target-domain inference data back to the source domain. For instance, the Mic2Mic [11] trains a cycle-consistent generative adversarial network (CycleGAN) to perform the translation between two microphones that have their own hardware characteristics. However, the training of CycleGAN requires about 20 minutes of microphone recording from both domains for a keyword spotting task [11]. In summary, although the prevalent transfer learning techniques reduce the demands on the target-domain training data in comparison with learning from scratch in the target domain, they still need substantial target-domain data to implement the model transfer.

In the cyber-physical sensing applications, both the monitored physical processes and the sensing apparatus are often governed by certain first principles. In this paper, we investigate the approach to exploit such first principles as a form of prior knowledge to reduce the demand on target-domain data for model transfer, vis-à-vis the aforementioned *physics-regardless* transfer learning techniques [11, 17, 21]. Recent studies attempt to incorporate prior knowledge in the form of commonsense [28] or physical laws [23, 24] to increase the learning efficiency. The presentation of the prior knowledge to learning algorithms is the core problem of *physics-constrained machine learning*. In [23], the law of free fall is incorporated into the loss function of learning the heights of a tossed pillow in a video. In [24], fluid dynamics equations are incorporated into the loss function of training DNNs for real-time fluid flow simulations. However, these physics-constrained machine learning approaches [23, 24] propose new DNN architectures and/or training algorithms; they are not designed to exploit first principles in transferring existing DNNs to address the domain shift problems.

Nevertheless, the improved learning efficiency of the physics-constrained machine learning encourages exploiting first principles to address domain shifts more efficiently. To this end, we propose a new approach called ***phy**sics-directed data **aug**mentation* (PhyAug). Specifically, we use a minimum amount of data collected from the target domain to estimate the parameters of the first principle governing the domain shift process and then use the parametric first principle to generate augmented target-domain training data. Finally, the augmented target-domain data samples are used to transfer or retrain the source-domain DNN. PhyAug has the following two key features. First, different from the conventional data augmentations that apply unguided *ad hoc* perturbations (e.g., noise injection) and transformations (e.g., scaling, rotation, etc) on existing training data to improve the DNNs' robustness against variations, PhyAug augments the training data strategically by following first principles to transfer DNNs. Second, PhyAug uses augmented data to represent the domain shifts and thus requires no modifications to the legacy DNN architectures and training algorithms. This makes PhyAug readily applicable once the data augmentation is completed. In contrast, recently proposed domain adaptation approaches based on adversarial learning [1, 11, 14, 25] update the DNNs under new adversarial training architectures that need extensive hyperparameter optimization and even application-specific redesigns. Such needs largely weaken their readiness, especially when the original DNNs are sophisticated such as the DeepSpeech2 [15] for automatic speech recognition.

In this paper, we apply PhyAug to three case studies and quantify the performance gains compared with other transfer learning approaches. The data and code of the case studies can be found in [10]. The first and the second case studies aim at adapting DNNs for keyword spotting (KWS) and automatic speech recognition (ASR) respectively to individual deployed microphones. The domain shifts are mainly from the microphone's hardware characteristics. Our tests show that the microphone can lead to 15% to 35% absolute accuracy drops, depending on the microphone quality. Instead of collecting training data using the target microphone, PhyAug uses a smartphone to play a 5-second white noise and then estimates the frequency response curve of the microphone based on its received noise data. Then, using the estimated curve, the existing samples in the factory training dataset are transformed into new training data samples, which are used to transfer the DNN to the target domain of the microphone by a retraining process. Experiment results show that PhyAug recovers the microphone-induced accuracy loss by 53%-72% and 37%-70% in KWS and ASR, respectively. PhyAug also outperforms the existing approaches including FADA [14] that is a domain adaptation approach based on adversarial learning and Mic2Mic [11] and CDA [12] that are designed specifically to address microphone heterogeneity. Note that KWS and ASR differ significantly in DNN model depth and complexity.

The third case study is seismic event localization. In the source domain where the density of the signal propagation medium is spatially homogeneous, the problem of estimating the event location based on the time differences of arrival

(TDoAs) of seismic signals received by geographically distributed sensors can follow a multilateration formulation. We aim to adapt to the target domain with an unknown and uneven medium that distorts the TDoAs. Thus, different from the sensor-induced domain shifts in the first and second case studies, the domain shift in this case study is from the variation of the sensed process. PhyAug estimates the signal propagation slowness model of the medium using a small amount of labeled TDoA data and then generates extensive TDoA data with simulated events to train a DNN for event localization. Results show that PhyAug only requires 3% to 8% of the real labeled TDoA data required by the physics-regardless vanilla approach in achieving the same event localization accuracy.

The main contribution of this paper is the proposed approach of using the first principle fitted with a small amount of source- and target-domain data to extensively augment the target-domain data for model transfer. This approach is more efficient than the physics-regardless transfer learning in terms of target-domain data sampling complexity. The applicability of PhyAug is contingent on the availability of the parametric first principle. While the context of cyber-physical systems provides abundant opportunities, the task of pinpointing useful and parametric first principles can be challenging in practice. Fortunately, this task is a one-time effort. Once the first principle for a specific application is identified, the model transfer processes of all the application instances benefit. For instance, by applying PhyAug with a microphone's frequency response curve as the parametric first principle, we can avoid the process of collecting substantial training data from each individual microphone for adapting ASR models.

The remainder of this paper is organized as follows. §2 overviews the PhyAug approach and reviews related work. §3, §4, and §5 present the three case studies. §6 discusses several issues. §7 concludes this paper.

2 APPROACH OVERVIEW & RELATED WORK

In this section, §2.1 overviews the PhyAug approach. §2.2 reviews the related studies and explains their relationships with and differences from PhyAug. §2.3 discusses the research methodology adopted in this paper.

2.1 Approach Overview

Fig. 1 illustrates PhyAug's workflow using a simple example, where the DNN performs a two-class classification task based on two-dimensional (2D) data samples and the first principle governing the domain shift is a nonlinear polynomial transform. Such transform can be used to characterize camera lens distortion [19]. To simplify the discussion, this example considers class-independent domain shift, i.e., the transform is identical across all the classes. Note that PhyAug can deal with class-dependent domain shifts, which will be discussed later. As illustrated in the upper part of Fig. 1, the general transfer learning approaches regardless of the first principles need to draw substantial data samples from both the source and target domains. Then, they apply *domain shift learning* techniques to update the existing source-domain DNN or construct a prefix DNN [11] to address the domain shift. Extensive data collection in the target domain often incurs undesirable overhead in practice.

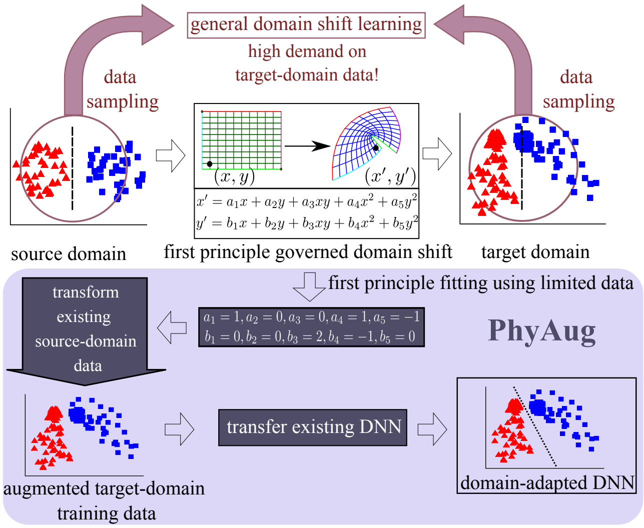

general domain shift learning
high demand on target-domain data!
data sampling
data sampling
(x, y)
(x', y')
$x' = a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2$
$y' = b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2$
source domain
first principle governed domain shift
target domain
first principle fitting using limited data
transform existing source-domain data
$a_1 = 1, a_2 = 0, a_3 = 0, a_4 = 1, a_5 = -1$
$b_1 = 0, b_2 = 0, b_3 = 2, b_4 = -1, b_5 = 0$
PhyAug
transfer existing DNN
augmented target-domain training data
domain-adapted DNN

Figure 1: PhyAug workflow.

Differently, as shown in the lower part of Fig. 1, PhyAug applies the following four steps to avoid extensive data collection in the target domain. ❶ The system designer identifies the parametric first principle governing the domain shift. For the current example, the parametric first principle is $x' = a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2$ and $y' = b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2$, where (x, y) and (x', y') are a pair of data samples in the source and target domains, respectively, and a_i, b_i are unknown parameters. ❷ A small amount of unlabeled data pairs are drawn from the source and target domains. The drawn data pairs are used to estimate the parameters of the first principle. For this example, if the domain shift is perturbation-free, the minimum number of data pairs needed is the number of unknown parameters of the polynomial transform. If the domain shift is also affected by other unmodeled perturbations, more data pairs can be drawn to improve the accuracy of estimating the parameters under a least squares formulation. If the domain shift is class-dependent, the data pair sampling and parameter estimation should be performed for each class separately. ❸ All the existing source-domain training data samples are transformed

Table 1: Categorization, used techniques, and requirements of various solutions to address domain shifts.

Category	Used technique	Solution	Applications in publication	Requirements				
				Source domain label	**Target domain label**	**Paired label data**	**First principle**	**Target-domain data volume***
Domain adaptation (Model transfer)	Adversarial learning	FADA [14]	computer vision	✔	✔	✔	–	▮▮▮▮
		ADDA [25]	computer vision	–	–	–	–	▮▮▮▮▮▮▮▮▮▮
		TransAct [1]	activity sensing	–	–	–	–	▮▮▮▮▮▮
		Mic2Mic [11]	voice sensing	–	–	–	–	▮▮▮▮▮▮▮▮
	Meta learning	MetaSense[6]	voice & motion	✔	✔	–	–	▮▮▮
	Data augmentation	**PhyAug**	voice sensing	–	–	–	✔	▮
			event localization	–	✔	–	✔	▮▮
Model robustness	Data augmentation	CDA [12]	voice and activity sensing	–	–	–	✔	▮▮▮▮▮▮▮▮▮

* The bars represent oracle scales partially based on the reported numbers in respective publications. Fully comparable scales are difficult to obtain because the solutions are designed for different applications. PhyAug is compared with FADA, Mic2Mic, and CDA in the evaluation sections of this paper. Reasons for excluding other approaches from the comparison will be discussed in the case studies.

to the target domain using the fitted first principle, forming an augmented training dataset in the target domain. ❹ With the augmented training dataset, various techniques can be employed to transfer the existing DNN built in the source domain to the target domain. For instance, we can retrain the DNN with the augmented data. The retraining can use the existing DNN as the starting point to speed up the process. For instance, for the DeepSpeech2 [15] which is a large-scale ASR model used in §4, the retraining only requires a half of training time compared with the training from scratch using the augmented data.

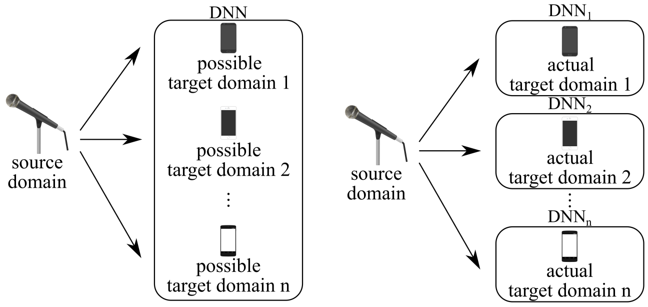
For sensing DNN design, the source domain is in general the design dataset. In such case, the source domain cannot be excited any more for data pair sampling in both domains simultaneously. However, we can recreate the excitation to collect the corresponding target-domain samples. For instance, we can use a speaker to play voice samples in the source-domain dataset and collect the corresponding samples from a target-domain microphone. Similarly, we can use a computer monitor to display image samples in the source-domain dataset and collect the corresponding samples from a target-domain camera that may have optical distortions.

2.2 Related Work

The applications of deep learning in embedded sensing systems have obtained superior inference accuracy compared with heuristics and conventional machine learning. Various approaches have been proposed to address the domain shift problems in embedded sensing [1, 6, 11, 12] and image recognition [14, 25]. Table 1 summarizes the categorization, used techniques, and requirements of these approaches. In what follows, we discuss the important details of these approaches and their differences from PhyAug.

■ **Domain adaptation:** Few-shot Adversarial Domain Adaptation (FADA) [14] transfers the model with limited amount of target-domain training data. It uses the *supervised adversarial learning* technique to find a shared subspace of the data distributions in the source and target domains. FADA requires labeled and paired data samples from both the source and target domains. Adversarial Discriminative Domain Adaptation (ADDA) [25] uses *unsupervised adversarial learning* to learn a feature encoder for the target domain. Although ADDA requires neither class labels nor data pairing, it demands substantial unlabeled target-domain data. TransAct in [1] considers sensor heterogeneity in human activity recognition and uses unsupervised adversarial learning to learn stochastic features for both domains. It requires hundreds of unlabeled target-domain data samples. Mic2Mic [11] applies CycleGAN, which is also an adversarial learning technique, to map the target-domain audio recorded by a microphone "in the wild" back to the source-domain microphone for which the DNN is trained. Mic2Mic requires about 20 minutes of speech recording from both microphones, which represents an overhead. Moreover, it can only perform one-to-one translations. Our experiment results in §3 and §4 show that CycleGAN performs unsatisfactorily when the source domains are publicly available speech datasets that are collected using numerous microphones in diverse environments.

PhyAug is a domain adaptation approach. Compared with ADDA [25], TransAct [1], and Mic2Mic [11] that are based on unsupervised adversarial learning and thus require substantial target-domain training data, PhyAug exploits the first principle governing the domain shift to reduce the demand on target-domain data. Although FADA [14] aims at reducing the demand of target-domain data, it requires extensive other



(a) Data augmentation for model robustness (e.g.,[12]). **(b) Data augmentation for model transfer (PhyAug).**

Figure 2: Different purposes of data augmentation illustrated using voice sensing. Note that the source domain may contain many microphones used to collect training samples.

information such as class labels in both domains. In contrast, PhyAug can operate with unlabeled data (cf. §3 and §4). Different from Mic2Mic [11] that requires the source domain to be a single microphone, PhyAug admits a source-domain dataset collected via many (and even unknown) microphones in the KWS and ASR case studies. This makes PhyAug practical since the datasets used to drive the design of DNNs for real-world applications often consist of recordings from diverse sources.

MetaSense [6] uses data collected from multiple source domains to train a base model that can adapt to a target domain related to the source domains. However, it requires substantial training data from both domains and class labels from each source domain. For voice sensing, MetaSense cannot use a source-domain dataset collected via many unlabeled microphones. But PhyAug can.

■ **Model robustness via data augmentation:** Data augmentation has been widely adopted for enhancing model robustness. As illustrated in Fig. 2a, a conventional scheme presumes a number of domain shifts (e.g., scaling, rotation, noise injection, etc) and follows them to generate augmented training samples. Then, the original and the augmented data samples are used to train a single DNN. During the serving phase, this DNN remains robust to the domain shift resembling the presumption. However, should the actual domain shift be out of the presumption, the robustness is lost. The CDA approach proposed in [12] follows the conventional data augmentation scheme to mitigate the impact of sensor heterogeneity on DNN's accuracy. Specifically, it estimates the probability distribution of sensors' heterogeneity characteristics from a *heterogeneity dataset* and then uses the characteristics sampled from the estimated distribution to generate augmented training data. As the dataset needs to cover heterogeneity characteristics, its collection in practice incurs a considerable overhead. Specifically, the heterogeneity dataset used in [12] consists of 2-hour recordings of 20 different microphones placed equidistant from an audio speaker. If the characteristic of a microphone "in the wild" is out of the estimated characteristic distribution (i.e., a missed catch), the enhanced DNN may not perform well. Since CDA uses sensor characteristics, we view it as an approach directed by first principles.

Different from CDA's objective of enhancing model robustness, PhyAug uses data augmentation to transfer a model to a specific target domain. Fig. 2b illustrates this in the context of voice sensing, where microphones' unique characteristics create domains. PhyAug constructs a dedicated DNN for each target domain. Thus, PhyAug is free of the missed catch problem faced by CDA.

2.3 Methodology

As this paper proposes PhyAug which is a domain adaptation approach, it is desirable to show PhyAug's applicability to multiple applications and its scalability to address different levels of pattern sophistication. Therefore, we apply PhyAug to three applications, i.e., KWS, ASR, and seismic event localization. Although KWS and ASR are two specific human voice sensing tasks, they have significantly different complexities. Different from KWS and ASR whose domain shift is mainly caused by sensor heterogeneity, the seismic event localization concerns about the domain shift caused by variations of the monitored physical process. For each case study, we also compare PhyAug with multiple existing approaches to show the advantages and performance gains of PhyAug.

3 CASE STUDY 1: KEYWORD SPOTTING

Human voice sensing is important for human-computer interactions in many Internet of Things (IoT) applications. At present, the DNN for a specific human voice sensing task is often trained based on a *standard dataset.* However, as IoT microphones are often of small form factors and low cost, their recordings often suffer degraded and varied voice qualities. In addition, the environment that an IoT microphone resides in can also affect its recording. For instance, the echo patterns in indoor spaces of different sizes can be distinct. Such run-time variations may be poorly captured by the standard dataset. As a result, the DNN yields reduced accuracy after the deployment. We apply PhyAug to address this domain shift problem. Specifically, we start from a swift process of profiling the IoT microphone's frequency response curve (FRC) with the help of a smartphone. Then, we use the FRC to transform the standard dataset. Finally, we retrain the DNN using the transformed dataset to obtain a personalized DNN for the IoT microphone.

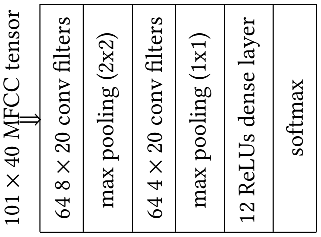


Figure 3: CNN structure used in case study.

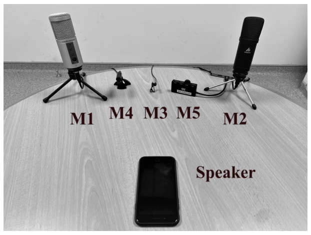


Figure 4: Microphones & experiment setup.

In this paper, we consider two human voice sensing functions: KWS and ASR. Most intelligent virtual assistant systems implement both functions. For instance, a virtual assistant often uses a low-power co-processor to perform KWS at all times. Once a designated keyword (e.g., "Hey Siri") is detected, the virtual assistant will activate the main processor to execute the more sophisticated ASR. In this section, we focus on KWS. §4 will focus on ASR. The results show that, with a 5-second smartphone-assisted FRC profiling process, we can recover a significant portion of accuracy loss caused by the domain shifts.

In the case studies of KWS (§3) and ASR (§4), **source domain** is the standard dataset originally used by the DNN; **target domain** is the dataset of voice samples captured by a specific deployed microphone; **first principle** is the microphone's FRC induced by the microphone hardware and its ambient environment.

3.1 Problem Description

We conduct a set of preliminary experiments to investigate the impact of diverse microphones on the KWS accuracy. Based on the results, we state the problem that we aim to address.

3.1.1 Standard dataset and DNN. We use Google Speech Commands Dataset [26] as the standard dataset in this case study. It contains 65,000 one-second utterances of 30 keywords collected from thousands of people. Audio files are sampled at 16 kilo samples per second (ksps). We pre-process the voice samples as follows. First, we apply a low-pass filter (LPF) with a cutoff frequency of 4 kHz on each voice sample, because human voice's frequency band ranges from approximately 0.3 kHz to 3.4 kHz. Then, for each filtered voice sample, we generate 40-dimensional Mel-Frequency Cepstral Coefficients (MFCC) frames using 30-millisecond window size and 10-millisecond window shift. The z-score normalization is applied on each MFCC frame. Eventually, each voice sample is converted to a 101×40 MFCC tensor. The dataset is randomly split into training, validation, and testing sets following an 8:1:1 ratio.

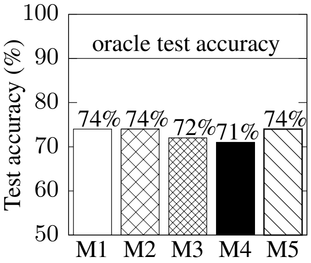


Figure 5: KWS accuracy on microphones. Horizontal line is accuracy on standard dataset.

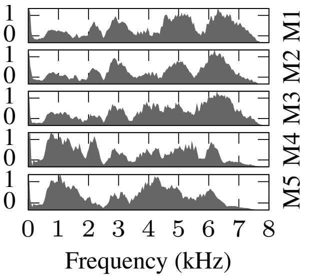


Figure 6: The five microphones' FRCs. The y-axis of each sub-figure is normalized amplitude.

We implement a CNN to recognize 10 keywords, i.e., "yes", "no", "left", "right", "up", "down", "stop", "go", "on", and "off". We also add two more classes to represent *silence* and *unknown keyword*. Fig. 3 shows the structure of the CNN. It achieves 90% test accuracy, which is similar to that in [29] and referred to as the *oracle test accuracy*.

3.1.2 Impact of microphone on KWS performance. In this section, we demonstrate that the CNN has performance degradation as a result of microphone heterogeneity. We test the CNN on samples captured by five different microphones named M1, M2, M3, M4, and M5 as shown in Fig. 4 that have list prices from high ($80) to low ($3.5). M1 and M2 are two high-end desktop cardioid condenser microphones, supporting sampling rates of 192 ksps at 24-bit depth and 48 ksps at 16-bit depth, effective frequency responses of [30 Hz, 16 kHz] and [30 Hz, 15 kHz], respectively. M3 is a portable clip-on microphone with an effective frequency response range of [20 Hz, 16 kHz]. M4 and M5 are two low-cost mini microphones without detailed specifications. Fig. 4 shows the placement of the microphones. For fair comparison and result reproducibility, we use an Apple iPhone 7 to play the original samples of the test dataset through its loudspeaker, with all microphones placed at equal distances away.

The samples recorded by each microphone are fed into the KWS CNN for inference. Fig. 5 shows the test accuracy for each microphone. Compared with the oracle test accuracy of 90%, there are 14% to 19% absolute accuracy drops due to domain shifts. By inspecting the spectrograms of the original test sample and the corresponding ones captured by the microphones, we can observe the differences. This explains the distinct accuracy drops among microphones. From the above experiment results, the research questions addressed in this case study are as follows. First, how to profile the characteristics of individual microphones with low overhead? Second, how to exploit the profile of a particular microphone to recover KWS's accuracy?

3.2 PhyAug for Keyword Spotting

PhyAug for KWS consists of two procedures: *fast microphone profiling* and *model transfer via data augmentation.*

3.2.1 Fast microphone profiling. A microphone can be characterized by its frequency response consisting of magnitude and phase. We only consider the magnitude component, because the information of a voice signal is largely represented by the energy distribution over frequencies, with little/no impact from the phase of the voice signal in the time domain. Let $X(f)$ and $Y(f)$ denote the frequency-domain representations of the considered microphone's input and output. The FRC to characterize the microphone is $H(f) = \frac{|Y(f)|}{|X(f)|}$, where $|\cdot|$ represents the magnitude.

We propose a fast microphone profiling approach that estimates $H(f)$ in a short time. It can be performed through a factory calibration process or by the user after the microphone is deployed. Specifically, a loudspeaker placed close to the target microphone emits a band-limited acoustic white noise $n(t)$ for a certain time duration. The frequency band of the white noise generator is set to be the band that we desire to profile. Meanwhile, the target microphone records the received acoustic signal $y_n(t)$. Thus, the FRC is estimated as $H(f) = \frac{|\mathcal{F}[y_n(t)]|}{|\mathcal{F}[n(t)]|}$, where $\mathcal{F}[\cdot]$ represents the Fourier transform. As the white noise $n(t)$ has a nearly constant power spectral density (PSD), this approach profiles the microphone's response at all frequencies in the given band.

In our experiments, we use the iPhone 7 shown in Fig. 4 to emit the white noise. We set the frequency band of the noise generator to be $[0, 8\,\text{kHz}]$, which is the Nyquist frequency of the microphone. Fig. 6 shows the measured FRCs of the five microphones used in our experiments. Each FRC is normalized to $[0, 1]$. We can see that the microphones exhibit distinct FRCs. In addition, we observe that the two low-end microphones M4 and M5 have lower sensitivities to the higher frequency band, i.e., 5 kHz to 8 kHz, compared with the microphones M1, M2, and M3.

3.2.2 Model transfer via data augmentation. We augment training samples in the target microphone's domain by transforming the original training samples using FRC. The procedure for transforming a sample $x(t)$ is as follows: (1) Apply the pre-processing LPF on $x(t)$ to produce $x'(t)$; (2) Conduct short-time Fourier transform using 30-millisecond sliding windows with an offset of 10 milliseconds on $x'(t)$ to produce 101 Fourier frames, i.e., $X_i(f)$, $i = 1, 2, \ldots 100$; (3) Multiply the magnitude of each Fourier frame with the FRC to produce $|Y_i(f)| = H(f) \cdot |X_i(f)|$; (4) Generate the MFCC frame from each PSD $|Y_i(f)|^2$; (5) Concatenate all 101 MFCC frames to form the MFCC tensor. Lastly, PhyAug retrains the CNN with augmented data samples for the microphone. Note that we use the pre-trained CNN as the starting point of the retraining process.

3.3 Performance Evaluation

3.3.1 Alternative approaches. Our performance evaluation employs the following alternative approaches.

■ **Data calibration:** At run time, it uses the measured FRC to convert the target-domain data back to the source-domain data and then applies the pre-trained CNN on the converted data. Specifically, let $Y_i(f)$ denote the ith Fourier frame after the microphone applies the LPF and short-time Fourier transform on the captured raw data. Then, it estimates the corresponding source-domain PSD as $|X_i(f)|^2 = \left(\frac{|Y_i(f)|}{H(f)}\right)^2$ and generates the MFCC frame from $|X_i(f)|^2$. The MFCC tensor concatenated from the MFCC frames over time is fed to the pre-trained CNN.

■ **Conventional data augmentation (CDA) [12]:** This alternative captures the essence of the approach in [12] following the conventional data augmentation scheme illustrated in Fig. 2a. Specifically, one out of the five microphones, e.g., M1, is designated as the testing microphone. The remaining four, e.g., M2 to M5, are used to generate a *heterogeneity dataset* [12]. The *heterogeneity generator* [12] is constructed as follows. For each microphone in the heterogeneity dataset, FRC is measured multiple times with the fast profiling process. At any frequency f, the FRC value is modeled by a Gaussian distribution. A Gaussian mixture is formed by the four heterogeneity-dataset microphones' Gaussian distributions with equal weights. The Gaussian mixtures for all frequencies form the heterogeneity generator. Then, each source-domain training sample is transformed by an FRC sampled from the heterogeneity generator into an augmented sample. Lastly, the DNN is retrained with the augmented training samples and tested with the samples captured by the testing microphone.

■ **CycleGAN (essence of [11]):** Mic2Mic [11] trains a CycleGAN using unlabeled and unpaired data samples collected from two microphones A and B. Then, CycleGAN can translate a sample captured by A to the domain of B, or vice versa. Following [11], we train a CycleGAN to translate the samples captured by a target microphone to the source domain of Google Speech Commands Dataset. Same as [11], the training of a CycleGAN for a target microphone uses 15 minutes data collected by the microphone. We train five CycleGANs for the five microphones, respectively. To measure the test accuracy, a test sample collected by a microphone is converted by the corresponding CycleGAN back to the source domain and then fed into the pre-trained CNN.

Compared with PhyAug that requires a single 5-second profiling data collection process for each microphone, CDA

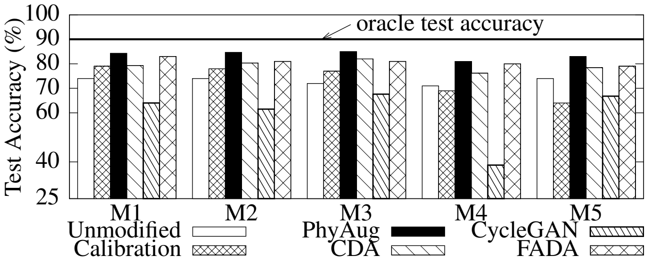


Figure 7: KWS test accuracy using various approaches on tested microphones. Compared with the unmodified baseline, PhyAug recovers the accuracy losses by 64%, 67%, 72%, 53%, and 56% respectively for the five microphones toward the oracle test accuracy.

repeats the profiling process many times for each heterogeneity microphone to construct the heterogeneity generator; the training of CycleGAN requires 15 minutes of data collected from each target microphone. Thus, both alternative approaches have higher overhead.

■ **FADA [14]**: It trains a feature encoder and classifier in the source domain. Then, it combines source-domain and target-domain data to train a domain-class discriminator. Finally, the weights of the feature encoder and classifier are updated to the target domain through adversarial learning using the domain-class discriminator. To apply FADA for KWS, we follow the architecture in [14] and modify the KWS model in Fig. 3 by adding a fully-connected layer before the last dense layer. Thus, the model has a feature encoder (CNN layers) and a classifier (fully-connected layers).

We exclude the MetaSense, ADDA and TransAct reviewed in §2.2 from the baselines for the following reasons. MetaSense cannot be applied to a source-domain dataset collected via many unlabeled microphones. We obtain unsatisfactory results for ADDA in the adversarial training with hours' target-domain training data and extensive hyperparameter tuning. We suspect that the amount of target-domain training data is still insufficient for ADDA. Note that PhyAug only requires five seconds' unlabeled target-domain data as shown shortly. TransAct is customized for activity recognition that differs from human voice sensing.

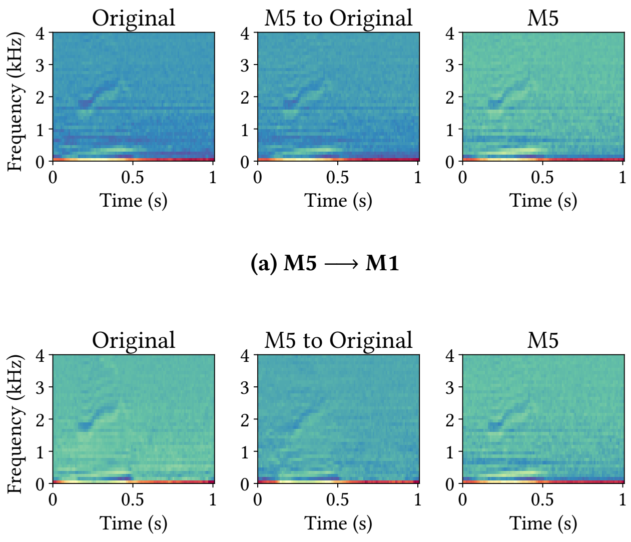
3.3.2 Evaluation results. We apply PhyAug and the alternatives for the five microphones in Fig. 4. The test accuracies are shown in Fig. 7. The bars labeled "unmodified" are the results from Fig. 5, for which no domain adaptation technique is applied. We include them as the baseline. The results are explained in detail as follows.

■ **Data calibration:** It brings test accuracy improvements for M1, M2, and M3. The average test accuracy gain is about 4%. For the cheap microphones M4 and M5, it results in test accuracy deteriorations. The reason is as follows. Its back mapping uses the reciprocal of the measured FRC (i.e., $1/H(f)$), which contains large elements due to the near-zero elements of $H(f)$. The larger noises produced by the low-end microphones M4 and M5 are further amplified by the large elements of $1/H(f)$, resulting in performance deteriorations. Thus, although this approach may bring performance improvements, it is susceptible to noises.

■ **PhyAug:** The black bars in Fig. 7 show PhyAug's results. Compared with the unmodified baseline, PhyAug recovers the test accuracy losses by 64%, 67%, 72%, 53%, and 56% for the five microphones. PhyAug cannot fully recover the test accuracy losses. This is because PhyAug only addresses the deterministic distortions due to microphones; it does not address the other stochastic factors such as the environmental noises and the microphones' thermal noises.

■ **CDA:** It recovers certain test accuracy losses for all microphones. This is because for any target microphone, there is at least one heterogeneity dataset microphone giving a similar FRC as the target microphone. Specifically, from Fig. 6, M1, M2, and M3 exhibit similar FRCs; M4 and M5 exhibit similar FRCs (i.e., they have good responses in lower frequencies). However, PhyAug consistently outperforms CDA. In addition, CDA introduces larger overhead than PhyAug as discussed in §3.3.1.

■ **CycleGAN:** It leads to test accuracy deteriorations for all five target microphones. Although CycleGAN is effective in translating the domain of a microphone to that of another microphone, which is the basis of Mic2Mic [11], it is *ineffective* in translating a certain microphone to the source domain of a dataset that consists of recordings captured by many microphones. We illustrate this using an example. First, we train a CycleGAN to translate M5 to M1. The first and the third columns of Fig. 8a shows the spectrograms captured by M1 and M5 for the same sample played by the smartphone in the setup shown in Fig. 4. We can see that there are discernible differences. The mid column shows the output of the CycleGAN, which is very similar to the first column. This result suggests that CycleGAN is effective for device-to-device domain translation and provides a minimal validation of Mic2Mic [11]. Then, we apply the same approach to train a different CycleGAN to translate M5 to the domain of Google Speech Commands Dataset. Fig. 8b shows the results. The third column is the spectrogram captured by M5 when a dataset sample shown in the first column is played by the smartphone in the setup shown in Fig. 4. The mid column is the CycleGAN's translation result, which has discernible differences from the first column, suggesting the ineffectiveness of CycleGAN. An intuitive explanation is that the CycleGAN shown with samples captured by many microphones during the training phase is confused and caters into no single microphone. Due to the discrepancy between



(a) M5 ⟶ M1

(b) M5 ⟶ Original

Figure 8: CycleGAN translation results (mid column). (a) Translation from M5 to M1. High similarity between first and second columns shows effectiveness of CycleGAN. (b) Translation from M5 to the domain of Google Speech Commands Dataset. Dissimilarity between first and second columns shows ineffectiveness of CycleGAN.

CycleGAN's output and the dataset, the pre-trained CNN fed with CycleGAN's outputs yields low test accuracy.

■ **FADA:** When we set the number of labeled target-domain samples per class (LTS/C) to 10 for FADA training, it recovers the accuracy loss for the five microphones by 56%, 38%, 47%, 47%, and 37%, respectively, as shown in Fig. 7. The performance of FADA increases with LTS/C. When we increase LTS/C to 20, PhyAug still outperforms FADA. Note that PhyAug requires a single unlabeled target-domain sample only. In addition, from our experience, FADA is sensitive to hyperparameter setting.

■ **Required noise emission time for microphone profiling:** In the previous experiments, the microphone profiling uses a 5-minute noise emission time. We conduct experiments to investigate the impact of shorter noise emission durations on the performance of PhyAug. The results show that the FRCs of a certain microphone measured with various noise emission durations down to five seconds are very similar. The corresponding test accuracies of PhyAug are also similar. (The detailed results are omitted here due to space constraint.) Thus, a noise emission time of five seconds is sufficient. This shows that PhyAug incurs little overhead.

3.4 Application Considerations

From the above results, PhyAug is desirable for KWS on virtual assistant systems. We envisage that more home IoT devices (e.g., smart lights, smart kitchen appliances, etc.) will support KWS. To apply PhyAug, the appliance manufacturer can offer the fast microphone profiling function as a mobile app and the model transfer function as a cloud service. Thus, the end user can use the mobile app to obtain the FRC, transmit it to the cloud service, and receive the customized KWS DNN. As the KWS DNN is not very deep and PhyAug is a one-time effort for each device, the model retraining performed in the cloud is an acceptable overhead to trade for better KWS accuracy over the entire device lifetime.

4 CASE STUDY 2: SPEECH RECOGNITION

ASR models often have performance degradation after deployments. This section shows the impact of various microphone models on ASR and how PhyAug is applied to recover the accuracy loss.

4.1 Impact of Microphone on ASR

We use LibriSpeech [18] as the standard dataset in this case study. It contains approximately 1,000 hours of English speech corpus sampled at 16 ksps. Each sample is an utterance for four to five seconds. We use an implementation [15] of Baidu DeepSpeech2, which is a DNN-based end-to-end ASR system exceeding the accuracy of Amazon Mechanical Turk human workers on several benchmarks. The used DeepSpeech2 model is pre-trained with LibriSpeech training dataset and achieves 8.25% word error rate (WER) on LibriSpeech test dataset. This 8.25% WER is referred to as *oracle WER*. Note that the input to DeepSpeech2 is the spectrogram of a LibriSpeech sample, which is constructed from the Fourier frames using 20-millisecond window size and 10-millisecond window shift.

DeepSpeech2 has 11 hidden layers with 86.6 million weights. It is far more complicated than the KWS CNN. Specifically, DeepSpeech2 is 175 times larger than the KWS CNN in terms of the weight amount. All the existing studies (e.g., Mic2Mic [11], MetaSense [6], and CDA [12]) that aimed at addressing domain shift problems in voice sensing only focused on simple tasks like KWS and did not attempt a sophisticated model such as DeepSpeech2.

We test the performance of the pre-trained DeepSpeech2 on the five microphones M1 to M5 used in §3. We follow the same test methodology as presented in §3.1.2. In Fig. 9, the histograms labeled "unmodified" represent the WERs of the pre-trained DeepSpeech2 on the test samples recorded by the five microphones. The horizontal line in the figure represents the *oracle WER*. We can see that the microphones introduce about 15% to 35% WER increases. In particular,

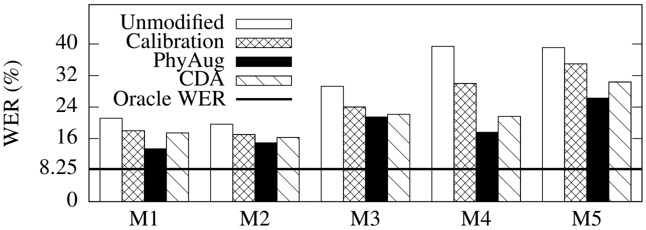


Figure 9: WERs using various approaches on tested microphones. Compared with the unmodified baseline, PhyAug reduces WER by 60%, 41%, 37%, 70%, and 42% respectively for the five microphones toward the oracle WER. As CycleGAN gives very high WERs (about 90%), it is not shown here.

the two low-end microphones M4 and M5 incur the highest WER increases. This result is consistent with the intuition. From the above test results, this section investigates whether PhyAug described in §3 for KWS is also effective for ASR. Different from the KWS CNN that takes MFCC tensors as the input, DeepSpeech2 takes the spectrograms as the input. Thus, in this case study, PhyAug does not need to convert spectrograms to MFCC tensors in the data augmentation.

4.2 Performance Evaluation

4.2.1 Comparison with alternative approaches. We use data calibration, CDA [12], and CycleGAN (i.e., essence of [11]) described in §3.3.1 as the baselines. FADA [14] cannot be readily applied to DeepSpeech2, because FADA requires class labels while DeepSpeech2 performs audio-to-text conversion without the concept of class labels. Differently, PhyAug and the three used baselines transform data without needing class labels.

■ **Data calibration:** Its results are shown by the histograms labeled "calibration" in Fig. 9. Compared with the unmodified baseline, this approach reduces some WERs.

■ **PhyAug:** Among all tested approaches, PhyAug achieves the lowest WERs for all microphones. Compared with the unmodified baseline, PhyAug reduces WER by 60%, 41%, 37%, 70%, and 42%, respectively, for the five microphones toward the oracle WER.

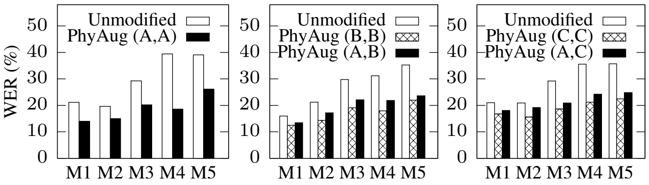
■ **CDA [12]:** It performs better than the data calibration approach but worse than PhyAug. As PhyAug is directed by the target microphone's actual characteristics, it outperforms CDA that is based on the *predicted* characteristics that may be inaccurate.

■ **CycleGAN:** We record a 3.5-hour speech dataset and use it to train a CycleGAN to translate samples captured by a target microphone to the source domain of LibriSpeech dataset. Unfortunately, DeepSpeech2's WERs on the data translated by CycleGAN from the microphones' samples are higher than 90%, indicating CycleGAN's inefficacy. We are unable to make it effective after extensive attempts. We also try to train the CycleGAN to perform M5-to-M1 domain translation following the design of Mic2Mic in [11]. The resulting WER is 65%. Although this result is better than 90%, it is still unsatisfactory. The reason for CycleGAN's inefficacy for ASR is as follows. Unlike the KWS task studied in Mic2Mic [11] and §3 of this paper, which discriminates a few target classes only, end-to-end ASR is much more complicated. CycleGAN may require much more training samples beyond we use to achieve good performance, rendering it too demanding and unattractive in practice.

4.2.2 Impact of various factors on PhyAug. We evaluate the impact of the following three factors on PhyAug: the indoor location of the microphone, the distance between the microphone and the sound source, and the environment type. We adopt an evaluation methodology as follows. When we evaluate the impact of a factor, the remaining two factors are fixed. For a certain factor, let X and Y denote two different settings of the factor. We use PhyAug(X,Y) to denote the experiment in which the microphone profiling is performed under the setting X and then the transferred model is tested under the setting Y. Thus, PhyAug(X,X) evaluates *in situ* performance; PhyAug(X,Y) evaluates the sensitivity to the factor.

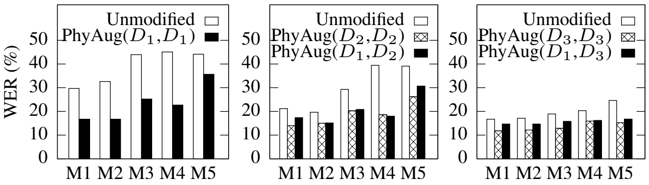
■ **Impact of microphone location:** Microphones at different locations of an indoor space may be subject to different acoustic reverberation effects. We set up experiments at three spots, namely, A, B, and C, in a $7 \times 4\,\text{m}^2$ meeting room. Spot B is located at the room center; Spots A and C are located at two sides of B, about 1 m apart from B along the room's long dimension. The phone and five microphones are set up in the same way as Fig. 4. Fig. 10 shows the results of the unmodified baseline approach tested at three spots, as well as PhyAug's *in situ* performance and location sensitivity. PhyAug's *in situ* WERs (i.e., PhyAug(A,A), PhyAug(B,B), PhyAug(C,C)) are consistently lower than those of the unmodified baseline. The WERs of PhyAug(A,B) and PhyAug(A,C) are slightly higher than PhyAug(B,B) and PhyAug(C,C), respectively. This shows that location affects the performance of a certain ASR model transferred by PhyAug, but not much. Thus, PhyAug for DeepSpeech2 is insensitive to the locations in a certain space.

■ **Impact of microphone-speaker distance:** The distance affects the signal-to-noise ratio (SNR) received by the microphone and thus ASR performance. With the setup at the aforementioned Spot C, we vary the distance between the microphones and the iPhone 7 used to play test samples to be 75 cm, 45 cm, and 15 cm (referred to as D_1, D_2, and D_3). Fig. 11 shows the results. The unmodified baseline's WERs become lower when the microphone-speaker distance



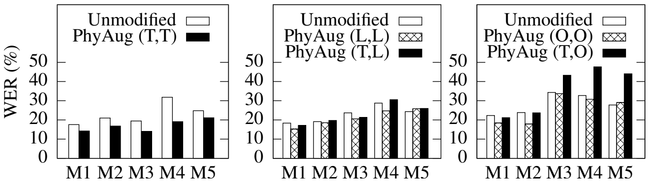
(a) Spot A **(b) Spot B** **(c) Spot C**

Figure 10: PhyAug's *in situ* performance and location sensitivity evaluated at three spots in a $7 \times 4\,\mathrm{m}^2$ meeting room.



(a) 75 **cm** (D_1) **(b)** 45 **cm** (D_2) **(c)** 15 **cm** (D_3)

Figure 11: PhyAug's *in situ* performance and microphone-speaker distance sensitivity evaluated with three distances.



(a) Tutorial room **(b) Lecture theater** **(c) Open area**

Figure 12: PhyAug's *in situ* performance and environment sensitivity evaluated in three types of environment, namely, small tutorial room (T), large lecture theater (L), and outdoor open area (O).

is shorter, due to the increased SNR. PhyAug's *in situ* WERs (i.e., PhyAug(D_1,D_1), PhyAug(D_2,D_2), and PhyAug(D_3,D_3)) are consistently lower than those of the unmodified baseline. The performance gain is better exhibited when the distances are longer. This suggests that *in situ* PhyAug improves the resilience of DeepSpeech2 against weak signals. In most cases, the WERs of PhyAug(D_1,D_2) and PhyAug(D_1,D_3) are slightly higher than those of PhyAug(D_2,D_2) and PhyAug(D_3,D_3), respectively. This shows that the microphone-speaker distance affects the performance of a certain model transferred by PhyAug, but not much. Thus, PhyAug for DeepSpeech2 is insensitive to the microphone-speaker distance.

Another related factor is the speaker's azimuth with respect to the microphone that can affect the quality of the recorded signal due to the microphone's polar-pattern characteristic. For a certain microphone, the different azimuths of the speaker create multiple target domains. If the speaker's azimuth can be sensed (e.g., by a microphone array), PhyAug can be applied. However, as the five microphones used in this paper lacks speaker azimuth sensing capability, we skip the application of PhyAug to address the domain shifts caused by the speaker's azimuth.

■ **Impact of environment:** Different types of environments in general have distinct acoustic reverberation profiles, which may affect the microphone's signal reception. We deploy our experiment setup in three distinct types of environments: a small tutorial room (T), a large lecture theatre (L), and an outdoor open area (O). Fig. 12 shows the results. The unmodified baseline approach has similar results in T and L. Its WERs become higher in O, because O has a higher level of background noise. PhyAug's *in situ* WERs in T, i.e., PhyAug(T,T), are consistently lower than those of the unmodified baseline. PhyAug(L,L) and PhyAug(O,O) reduce WERs compared with the unmodified baseline, except for the low-quality microphone M5. As M5 has higher noise levels, the microphone profiling process may not generate fidelity FRCs for M5, leading to increased WERs. As shown in Figs. 12b and 12c, the WERs of PhyAug(T,L) and PhyAug(T,O) are higher than those of the unmodified baseline. The above results show that PhyAug for DeepSpeech2 may have degraded performance on low-quality microphones. In addition, PhyAug for DeepSpeech2 is sensitive to various environments.

4.3 Application Considerations

Use scenarios: The results in §4.2 show that PhyAug is sensitive to the type of environment because the microphone profiling additionally captures the acoustic reverberation profile of the environment. Thus, PhyAug suits ASR systems deployed at fixed locations, such as residential and in-car voice assistance systems, as well as minutes transcription systems installed in meeting rooms. PhyAug can also be applied to the *ad hoc* deployment of ASR and automatic language translation for a multilingual environment.

PhyAug and continuous learning (CL): An ASR system can be improved via CL that gradually adapts the ASR model to the speaker and/or the environment when exposed to a continuous data stream for a long period of time. PhyAug is complementary to CL since PhyAug is applied once. Jointly applying PhyAug and CL can maximize the ASR system's quality of service.

5 CASE STUDY 3: SEISMIC SOURCE LOCALIZATION

Estimating the location of a seismic event source using distributed sensors finds applications in earthquake detection

[5], volcano monitoring [27], footstep localization [13], and fall detection [4]. TDoA-based localization approaches have been widely employed in these applications. The TDoA measurement of a sensor is the difference between the time instants at which the signal from the same event arrives at the sensor and a reference sensor. In the source domain where the medium density is spatially homogeneous, the seismic signal propagation velocity is also spatially homogeneous. To address measurement noises, the TDoA-based multilateration problem is often solved under a least squares formulation. However, in practice, the medium density is often spatially heterogeneous. This case study aims to deal with the target domain where the medium density is unknown and uneven. For instance, the density of the magma beneath an active volcano varies with depth. As such, seismologists need a *slowness model* that depicts the seismic wave propagation speeds at different depths before hypocenter estimation can be performed [9]. In footstep localization and fall detection, the heterogeneity of the floor materials affects the seismic wave propagation speed and degrades the performance of the simplistic multilateration formulation. Unfortunately, directly measuring the slowness model is tedious or even unfeasible in many cases.

To cope with heterogeneous media, the fingerprinting approach can be employed. Specifically, when a seismic event with a known location is triggered, the TDoA measurements by the sensors form a fingerprint of the known location. With the fingerprints of many locations, a seismic event with an unknown location can be localized by comparing the sensors' TDoA measurements with the fingerprints. The fingerprints can be collected by triggering controlled events at different locations, e.g., controlled explosions in seismology [8] and hammer excitations in structure health monitoring [7]. Under the fingerprinting approach, determining the location of an event source can be formulated as a classification problem, in which the fingerprint is the input data and the corresponding location is the class label. To achieve a high localization accuracy, a laborious blanket process of fingerprinting many/all locations is generally required. In this case study, we show that by exploiting the first principle of seismic wave propagation in an uneven medium, we can significantly reduce the amount of fingerprints and achieves a certain level of localization accuracy. Note that, from Appendix A, even with homogeneous medium, the fingerprinting approach outperforms the least squares approach in terms of response time, while offering comparable localization accuracy.

In this case study, **source domain** is the homogeneous medium for seismic signals; **target domain** is the heterogeneous medium for seismic signals; **first principle** is the slowness model characterizing seismic signal propagations in heterogeneous media.

5.1 Problem Description

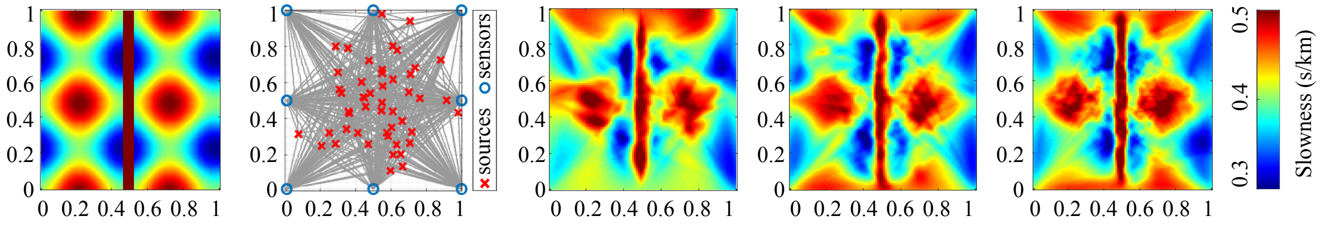
Consider a 2D field divided into $W_1 \times W_2$ grids, where W_1 and W_2 are integers. Thus, the field has a total of $N = W_1 \cdot W_2$ grids. Each grid is associated with a slowness value in seconds per kilometer (s/km), which is the reciprocal of the seismic wave propagation speed in the grid. We assume that the slowness at any position in a grid is a constant, while the slownesses in different grids can be distinct. Thus, the slowness model is a matrix (denoted by $\mathrm{S} \in \mathbb{R}^{W_1 \times W_2}$) with the grids' slowness values as the elements. In this case study, we adopt a slowness model from [2] as shown in Fig. 13(a), which is a $1 \times 1\,\text{km}^2$ square field with a wavy pattern and a barrier stripe in the middle. The pattern and the barrier create challenges to the event localization and will also better exhibit the effectiveness of PhyAug in addressing heterogeneous medium.

There are a total of M seismic sensors deployed in the field. When there is an event occurring in the field, the propagation path of the seismic wave front from the event source to any sensor follows a straight ray path. For instance, Fig. 13(b) shows the ray paths for the eight sensors considered in this case study. Note that this case study can be also extended to address the refraction of the seismic wave at the boundary of any two grids by using a ray tracing algorithm [9] to determine the signal propagation path. In Fig. 13(b), the deployment of the sensors at the field boundary is consistent with the practices of floor event monitoring [13] and volcano activity monitoring [27]. The seismic event locations follow a Gaussian distribution centered at the field center.

In what follows, we model the seismic signal propagation process for the lth event. For the mth sensor, denote the propagation ray path by $p_{l,m}$; denote the *ray tracing* matrix by $\mathbf{A}_{l,m} \in \mathbb{R}^{W_1 \times W_2}$, where its (i, j)th element is the length of $p_{l,m}$ in the (i, j)th grid. If $p_{l,m}$ does not go through the (i, j)th grid, the corresponding element of $\mathbf{A}_{l,m}$ is zero. Let $\mathbf{a}_{l,m} \in \mathbb{R}^{1 \times N}$ denote a row vector flattened from $\mathbf{A}_{l,m}$ in a row-wise way. Therefore, the ray tracing matrix for all sensors in the lth event, denoted by $\mathbf{A}_l \in \mathbb{R}^{M \times N}$, is $\mathbf{A}_l = [\mathbf{a}_{l,1}; \mathbf{a}_{l,2}; \ldots; \mathbf{a}_{l,M}]$. Let $t_{l,m}$ denote the time for the seismic wave front to propagate from the lth event's source to the mth sensor. Denote $\mathbf{t}_l = [t_{l,1}; t_{l,2}; \ldots; t_{l,M}] \in \mathbb{R}^{M \times 1}$. Let $\mathbf{s} \in \mathbf{R}^{N \times 1}$ denote a column vector transposed from the row vector that is the row-wise flattening of the slowness model S. Thus, the first principle governing the propagation times is

$$\mathbf{t}_l = \mathbf{A}_l \mathbf{s}. \tag{1}$$

Note that the flattened slowness model $\mathbf{s}$ is identical for all events. Denote by $\widetilde{\mathbf{t}}_l = [\widetilde{t}_{l,1}; \widetilde{t}_{l,2}; \ldots; \widetilde{t}_{l,M}]$ the measurements of the propagation times. We assume $\widetilde{\mathbf{t}}_l = \mathbf{t}_l + \boldsymbol{\epsilon}$, where the measurement noise $\boldsymbol{\epsilon} \in \mathbb{R}^M$ is a random variable following an M-dimensional Gaussian distribution $\mathcal{N}\left(\mathbf{0}_M, \sigma_\epsilon^2 \mathbf{I}_M\right)$. In



(a) True slowness model (b) Events simulation (c) $L = 25$ (d) $L = 50$ (e) $L = 100$

Figure 13: The $1 \times 1\,\mathbf{km}^2$ 2D field considered in the seismic source localization case study. (a) The ground-truth slowness model with 100×100 grids. (b) Seismic event source locations and their ray paths to sensors. (c)-(e) The estimated slowness models with 25, 50, and 100 seismic events that occur at random positions in the 2D field as the training samples, respectively.

the numerical experiments, we set $\sigma_\epsilon = \xi \cdot \bar{\mathbf{t}}_l$, where ξ is called *noise level* and $\bar{\mathbf{t}}_l$ is the average value of the elements in $\mathbf{t}_l$. In the evaluation experiments, the default noise level is $\xi = 2\%$.

In the TDoA-based fingerprinting approach, a target-domain training data sample consists of the position of the triggered event as the label and the TDoA measurements as the feature. Specifically, if the first sensor is chosen to be the reference, the feature of the lth event is

$$\mathbf{f}_l = [\widetilde{t}_{l,2} - \widetilde{t}_{l,1}; \widetilde{t}_{l,3} - \widetilde{t}_{l,1}; \ldots ; \widetilde{t}_{l,M} - \widetilde{t}_{l,1}] \in \mathbf{R}^{(M-1)\times 1}. \quad (2)$$

A support vector machine (SVM) or DNN can be trained based on a training dataset and then used to localize an event at run time. The research questions addressed in this case study are as follows. First, how to exploit the first principle in Eq. (1) to augment the training dataset? Second, to what extent the demand on actual training data samples can be reduced by applying PhyAug?

5.2 PhyAug for Seismic Source Localization

To use the first principle in Eq. (1) to augment the training dataset, the flattened slowness model $\mathbf{s}$ needs to be estimated using some training data samples. This tomography problem can be solved by the Bayesian Algebraic Reconstruction Technique (BART) or Least Squares with QR-factorization (LSQR) algorithm [16]. In this work, we apply BART to generate an estimated slowness model denoted by $\hat{\mathbf{s}}$ based on a total of L training samples collected by triggering events with known positions in the field. The details of BART are omitted here due to space constraint and can be found in [20]. Figs. 13(c)-(e) show $\hat{\mathbf{s}}$ when $L = 25$, $L = 50$, and $L = 100$, respectively. We can see that when more seismic events are used, the $\hat{\mathbf{s}}$ is closer to the ground truth shown in Fig. 13(a). The above tomography process uses L labeled target-domain data samples. Thus, PhyAug for this case study requires target-domain class labels as indicated in Table 1. As PhyAug can significantly reduce the amount of needed target-domain data samples as shown shortly, the related overhead is largely mitigated.

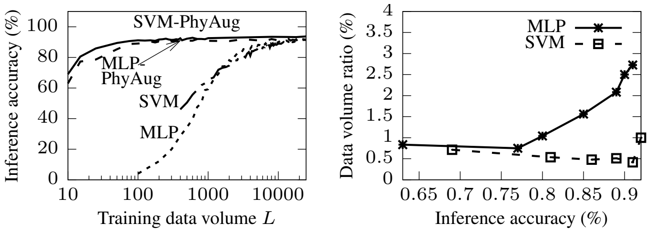
With the estimated slowness model $\hat{\mathbf{s}}$, we can generate a large amount of augmented fingerprints to extend the training dataset. Specifically, to generate the xth augmented fingerprint denoted by $\mathbf{t}_x$, we randomly and uniformly draw a position from the 2D field as the event source location and then compute the ray tracing matrix $\mathbf{A}_x$ and the fingerprint $\mathbf{t}_x = \mathbf{A}_x \hat{\mathbf{s}}$. Lastly, the SVM or DNN is trained using the extended training dataset consisting of the L genuine training samples and X augmented training samples.

With the above approach, we can generate any number of augmented training samples. In this case study, we adopt the following approach to decide the volume of augmented training samples. Initially, we set $X = 100 \times N$, where N is the number of grids, and train the SVM/DNN with the augmented training dataset. We double the volume of the augmented training samples (i.e., $X = 2 \times X$) until the validation accuracy of the trained SVM/DNN saturates.

5.3 Performance Evaluation

We use both SVM and multilayer perceptron (MLP) for fingerprint-based source localization. We implement SVM using LIBSVM 3.24 [3]. It uses radial basis function kernel with two configurable parameters C and γ. During training, we apply grid search to optimize the settings of C and γ. In addition, we construct a 5-layer MLP. The numbers of neurons in the layers are M, 1024, 1024, 512, and N, respectively. For training, a 0.2 dropout rate is used between any two hidden layers to prevent overfitting. We use cross-entropy as the loss function at the output layer as the training feedback.

5.3.1 Advantages brought by PhyAug to SVM/MLP-based fingerprinting approach. We set $N = 20 \times 20$. We use the grid-wise inference accuracy as the evaluation metric. Fig. 14a shows the inference accuracy of SVM and MLP, without and with PhyAug, versus the training data volume L. First, we discuss the results of SVM and MLP without PhyAug. We can



(a) Inference accuracy vs. training data volume for SVM and MLP with or without PhyAug.

(b) Ratio of training data volumes with and without PhyAug vs. required inference accuracy.

Figure 14: Impact of PhyAug on SVM/MLP-based fingerprinting approaches (number of grids: 400; $\xi = 2\%$).

see that, the inference accuracy of SVM and MLP increases with L. When more than 8,000 training samples are provided, SVM and MLP achieve more than 92% inference accuracy. When less than 11,000 training samples are provided, SVM outperforms MLP; otherwise, MLP outperforms SVM. This observation is consistent with the general understanding that deep learning with "big data" outperforms the traditional machine learning approaches. Second, we discuss the results of SVM and MLP with PhyAug. The inference accuracy of SVM-PhyAug and MLP-PhyAug also increases with L. With more training samples, the estimated slowness model $\hat{\mathbf{s}}$ is more accurate. As a result, the augmented data samples will be of higher quality, thus helping the SVM/MLP achieve higher test accuracy. From Fig. 14a, we can see that PhyAug boosts the inference accuracy of SVM and MLP when the training data volume is limited. Fig. 14b shows the ratio of the training data volumes required by a classifier with/without PhyAug to achieve a specified inference accuracy. With PhyAug, only less than 3% training samples are needed. This shows that PhyAug is very effective in reducing the demand on training data.

5.3.2 Impact of noise level. The noise level of TDoA data affects the accuracy of $\hat{\mathbf{s}}$. Our evaluation results in Appendix B show that PhyAug requires 1% to 8% of actual training data required by SVM or MLP when ξ increases from 0 to 8%.

5.3.3 Summary. Different from the KWS and ASR case studies that use PhyAug to recover recognition accuracy loss mainly caused by sensor hardware characteristics, this case study uses PhyAug to reduce the demand for actual training data in dealing with the complexity of the sensed physical process. Although this case study is primarily based on numerical experiments, the results provide baseline understanding on the advantages brought by PhyAug.

6 DISCUSSIONS

The three case studies have demonstrated the advantages of exploiting the first principles in dealing with domain shifts that are often experienced by deployed sensing systems. Pinpointing the useful first principles can be challenging in practice and requires separate studies/experimentation for different applications. For the applications that lack useful first principles, we may fall back to the existing physics-regardless transfer learning approaches. However, the fallback option should not discourage us from being discerning on the exploitable first principles in the pursuit of advancing and customizing deep learning-based sensing in the domain of physics-rich cyber-physical systems. In what follows, we briefly mention several other sensing tasks that PhyAug may be applicable to, which are also interesting for future work.

■ Polynomial transforms can describe the optical distortions of the camera lens that may be introduced purposely (e.g., fisheye lens) [19]. Visual sensing applications can adapt to varied optical distortions to improve DNN performance.

■ Room impulse response (RIR) describes indoor audio processes. Smart voice-based appliances can exploit RIR as the first principle for effective adaptations to the deployment environments. Acoustic-based indoor localization with deep learning [22] can exploit RIR to reduce target-domain training data sampling complexity.

■ Computational fluid dynamics (CFD) describes the thermal processes in indoor spaces (e.g., data centers). A trained deep reinforcement learning-based environment condition controller can adapt to new spaces with CFD models and a few target-domain data samples in each new space.

7 CONCLUSION

This paper described PhyAug, an efficient data augmentation approach to deal with domain shifts governed by first principles. We presented the applications of PhyAug to three case studies of keyword spotting, automatic speech recognition, and seismic event localization. They have distinct task objectives and require deep models with quite different architectures and scales. The extensive and comparative experiments showed that PhyAug can recover significant portions of accuracy losses caused by sensors' characteristics and reduce target-domain training data sampling complexity in dealing with the domain shifts caused by the variations of the dynamics of the sensed physical process.

ACKNOWLEDGMENTS

This research was conducted at Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU), which is a collaboration between Singapore Telecommunications Limited (Singtel) and Nanyang Technological University (NTU) that is funded by the Singapore Government through the

Industry Alignment Fund - Industry Collaboration Projects Grant.

REFERENCES

[1] Ali Akbari and Roozbeh Jafari. 2019. Transferring activity recognition models for new wearable sensors with deep generative domain adaptation. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. 85–96.

[2] Michael J Bianco and Peter Gerstoft. 2018. Travel time tomography with adaptive dictionaries. *IEEE Transactions on Computational Imaging* 4, 4 (2018), 499–511.

[3] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 1–27.

[4] Jose Clemente, WenZhan Song, Maria Valero, Fangyu Li, and Xiangyang Liy. 2019. Indoor person identification and fall detection through non-intrusive floor seismic sensing. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 417–424.

[5] Matthew Faulkner, Michael Olson, Rishi Chandy, Jonathan Krause, K Mani Chandy, and Andreas Krause. 2011. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 13–24.

[6] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. 2019. MetaSense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems (SenSys)*. 110–123.

[7] Gregory Hackmann, Fei Sun, Nestor Castaneda, Chenyang Lu, and Shirley Dyke. 2012. A holistic approach to decentralized structural damage localization using wireless sensor networks. *Computer Communications* 36, 1 (2012), 29–41.

[8] Kevin Hupp, Brandon Schmandt, Eric Kiser, Steven M Hansen, and Alan Levander. 2016. A controlled source seismic attenuation study of the crust beneath Mount St. Helens with a dense array. In *American Geophysical Union (AGU) Fall Meeting Abstracts*, Vol. 2016. V33E–3170.

[9] Jonathan M Lees and Robert S Crosson. 1991. Bayesian Art versus Conjugate Gradientf Methods in Tomographic Seismic Imaging: An Application at Mount St. Helens, Washington. *Lecture Notes-Monograph Series* (1991), 186–208.

[10] Wenjie Luo. 2021. PhyAug Dataverse. https://researchdata.ntu.edu.sg/dataverse/phyaug/.

[11] Akhil Mathur, Anton Isopoussu, Fahim Kawsar, Nadia Berthouze, and Nicholas D Lane. 2019. Mic2mic: using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems. In *Proceedings of the 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 169–180.

[12] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Velickovic, Leonid Joffe, Nicholas D Lane, Fahim Kawsar, and Pietro Lió. 2018. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 200–211.

[13] Mostafa Mirshekari, Shijia Pan, Jonathon Fagert, Eve M Schooler, Pei Zhang, and Hae Young Noh. 2018. Occupant localization using footstep-induced structural vibration. *Mechanical Systems and Signal Processing* 112 (2018), 77–97.

[14] Saeid Motiian, Quinn Jones, Seyed Iranmanesh, and Gianfranco Doretto. 2017. Few-shot adversarial domain adaptation. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*. 6670–6680.

[15] Sean Naren. 2020. deepspeech.pytorch. https://github.com/SeanNaren/deepspeech.pytorch.

[16] Jorge Nocedal and Stephen Wright. 2006. *Numerical optimization*. Springer Science & Business Media.

[17] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2009), 1345–1359.

[18] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5206–5210.

[19] Matthias Pohl, Michael Schaeferling, Gundolf Kiefer, Plamen Petrow, Egmont Woitzel, and Frank Papenfuß. 2014. Leveraging polynomial approximation for non-linear image transformations in real time. *Computers & Electrical Engineering* 40, 4 (2014), 1146–1157.

[20] Clive D Rodgers. 2000. *Inverse methods for atmospheric sounding: theory and practice*. Vol. 2. World scientific.

[21] Dipanjan Sarkar, Raghav Bali, and Tamoghna Ghosh. 2018. *Hands-On Transfer Learning with Python*. Packt Publishing.

[22] Qun Song, Chaojie Gu, and Rui Tan. 2018. Deep room recognition using inaudible echos. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 3 (2018), 1–28.

[23] Russell Stewart and Stefano Ermon. 2017. Label-free supervision of neural networks with physics and domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[24] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering* 361 (2020), 112732.

[25] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7167–7176.

[26] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (2018).

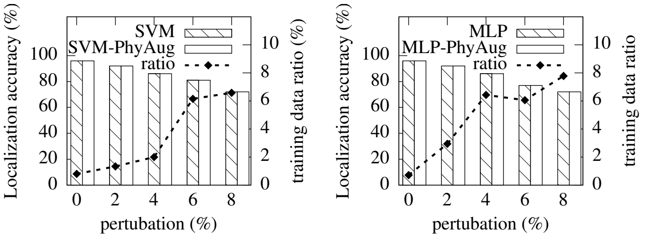
[27] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. 2006. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*. 381–396.

[28] Chengcheng Yu, Xiaobai Liu, and Song-Chun Zhu. 2017. Single-Image 3D Scene Parsing Using Geometric Commonsense. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 4655–4661.

[29] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. 2017. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128* (2017).

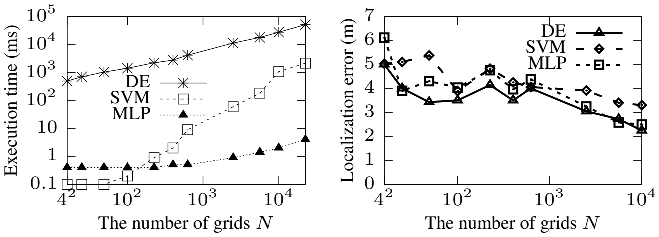
A SVM/DNN VS. LEAST SQUARES METHOD FOR SEISMIC SOURCE LOCALIZATION

The estimated slowness model $\hat{\mathbf{s}}$ can be directly used to estimate the source location at run time by a least squares method. In the least squares method, we apply differential evolution (DE), which is a population-based metaheuristic search algorithm, to perform grid-granular search and iteratively improve a candidate solution p with $\|\mathbf{f} - f\left(\mathbf{A}_p \hat{\mathbf{s}}\right)\|_{\ell_2}^2$ as the error metric. In the above error metric, the $\mathbf{f}$ is the feature vector of TDoA measurements given by Eq. (2) for the run-time event; the $\mathbf{A}_p$ is the ray tracing matrix of the candidate



(a) Classifier is SVM. **(b) Classifier is MLP.**

Figure 16: Impact of TDoA measurement noise level on PhyAug's effectiveness.



(a) Inference time vs. the number of grids (both the x- and y-axes are in log scale). **(b) Localization error vs. the number of grids (the x-axis is in log scale).**

Figure 15: Performance comparison of differential evolution (DE), SVM, and MLP.

position p; the $f(\cdot)$ is a function converting the seismic propagation times to the feature vector of TDoA measurements. Fig. 15 compares the performance of DE, SVM, and MLP in terms of average execution time over 100 events. In the evaluation, we increase the number of grids N for finer inference granularity. Fig.15a shows the execution time versus N. From a regression analysis on the results, DE has a time complexity of $O(N^{0.45})$. It's execution time is several orders of SVM and MLP. For instance, when $N = 22500$, DE's execution time is 50.73 s, which is about 23x and 12,500x longer than SVM's and MLP's, respectively. The long response delays make DE unsuitable for a range of time-critical applications such as earthquake early warning [5]. From Fig. 15a, the execution time of ML is within 10 ms when N is up to 22,500. Fig. 15b shows the average localization error in terms of Euclidean distance versus N. We can see that the three approaches give comparable localization accuracy. From the above results, SVM and MLP are superior to DE due primarily to response times.

B IMPACT OF NOISE LEVEL ON PHYAUG FOR SEISMIC SOURCE LOCALIZATION

This appendix contains experiment results on the impact of noise level ξ on the performance of PhyAug for seismic source localization. As defined in §5.1, the TDoA measurement contains a random noise following $\mathcal{N}\left(\mathbf{0}_M, (\xi\bar{\mathbf{t}}_l)^2\mathbf{I}_M\right)$. The histograms in Fig. 16 show the grid-wise localization accuracy of SVM, MLP, and their PhyAug-assisted variants when ξ increases from 0% to 8%. The dashed curve in Figs. 16a and 16b shows the ratio between the volumes of actual training data required by SVM/MLP with and without PhyAug. The SVM/MLP approach uses the same amount of training data for all ξ settings, whereas we adjust the amount of the actual training data used for the PhyAug-assisted variant to achieve the same grid-wise localization accuracy as the SVM/MLP approach. From the figure, the localization accuracy decreases with ξ. This is consistent with intuition because larger noise levels lead to more classification errors. In addition, the ratio of the actual training data amounts required by SVM/MLP with and without PhyAug increases with ξ. For example, MLP-PhyAug only requires about 1% of the training data needed by MLP without PhyAug to achieve the same 96% accuracy when $\xi = 0\%$; this ratio increases to about 8% to achieve the same 77% accuracy when $\xi = 8\%$. This is because PhyAug needs more actual training data to estimate a good slowness model when the noise level is higher. Nevertheless, PhyAug reduces the demand for actual training data by a factor of more than 10 when ξ is up to 8%.