Neural-Enhanced Modulation for Spatial Selective Transmission on Low-End IoT Devices

Huangwei Wu, Tingchao Fan, Meng Jin, *Member, IEEE*, Tao Chen, *Member, IEEE*, Xinbing Wang, *Senior Member, IEEE*, and Chenghu Zhou, *Member, IEEE*

***Abstract*—This paper tries to answer a question: "Can we achieve spatial-selective transmission on IoT devices?" A positive answer would enable more secure data transmission among IoT devices. The challenge, however, is how to manipulate signal propagation without relying on beamforming antenna arrays which are usually unavailable on low-end IoT devices. We give an affirmative answer by introducing SpotSound, a novel acoustic communication system that exploits the diversity of multi-path indoors as a natural *beamformer*. By judiciously controlling the way how the information is embedded into the signal, SpotSound can make the signal decodable only when the signal propagates along a certain multipath channel. Since the multipath channel decorrelates rapidly over the distance between receivers, SpotSound can ensure the signal is decodable only at the target position, achieving precise physical isolation. SpotSound is a purely software-based solution that can run on most IoT devices where speakers and microphones are widely used. We implement SpotSound on Raspberry Pi connected with COTS microphone and speaker. Experimental results show that SpotSound could precisely focus its signal on spots with customized sizes ranging from $0.04m^2$ to $0.5m^2$.**

***Index Terms*—IoT security, acoustic, DL-based communication.**

I. Introduction

TODAY most wireless transceivers are omnidirectional sources of electromagnetic waves. Since a wireless channel is a broadcast medium, wireless transmissions suffer more concerns on security and privacy [1], [2], [3], [4], [5]. This concern becomes more serious in the age of the Internet of Things (IoT), where small and low-cost gadgets continuously monitor our vital signs and living environment and share them via wireless channels. Since those low-cost gadgets support only weak encryption or even transmit without encryption [6], their data packets are vulnerable to eavesdropping attack [7]. More importantly, even if the wireless network is encrypted, an eavesdropper can still obtain privacy information by simply observing features of the wireless signal (e.g., infer a device's location based on the received signal strength) [8], [9].

To address this security concern, a potential solution is leveraging beamforming [10], [11], which ensures the signal propagates along a certain direction so that the malicious users at other locations cannot hear it, as shown in Fig. 1 (a). However, beamformer usually faces a critical trade-off between cost and control granularity: low-cost directional antennas form wide beams and thus fail to provide strong security protection. While, antenna arrays with more sophisticated control are costly and bulky, which hinders their deployment on low-end IoT devices. Programming the radio environment through meta-surface [12], [13], [14], [15], [16], as shown in Fig. 1(b), would retain a small form factor for IoT devices. It however comes with even higher infrastructure cost, setting a strong barrier for pervasive deployment.

This paper presents the design and implementation of **SpotSound**, a novel communication system that supports *spatial-selective* transmission without relying on bulky and costly antenna arrays or meta-surfaces (as shown in Fig. 1 (c)). Our idea is to leverage the environment reflectors such as walls, chairs, and tables as a natural "beamformer". Specifically, a transmitted signal will bounce off reflectors indoors and superimpose at the receiver. These reflectors form different physical channels that alter signal transmission in different ways with respect to the receiver's location. Taking a step further, for each location, the associated physical channels essentially form a *location-dependent filter*. By modulating the transmission signal in a way that the embedded information can pass through only the target "filter", we can achieve spatial selective transmission.

However, no existing modulation algorithm can generate signals that satisfy the above objective. Conventional modulation algorithms adopt pre-coding to ensure the modulated signals can be best delivered to the target receiver. However, these schemes fail to support spatial-selective transmission because the pre-coded signals can also be demodulated successfully at other receiver locations as long as the receiver maintains a good link condition with the transmitter. So, to achieve spatial-selective transmission, we need a new modulator that is able to identify high-level, distinctive characteristics of a target channel, uncover how these hidden characteristics impact the signal waveform, and modulate the transmission signal to ensure this signal is decodable only if it passes through the target channel.

Toward the above target, we propose a novel deep learning-based modulation algorithm, which *learns* to generate satisfying signals for any given target channel. In this design, the transmitter is a DL network, which takes as input both the information needed to transmit and a measurement of the target channel, based on which it generates modulated signals. To

Received 5 August 2024; revised 11 December 2025; accepted 20 December 2025; approved by IEEE TRANSACTIONS ON NETWORKING Editor H. Hassanieh. Date of publication 5 January 2026; date of current version 2 February 2026. This work was supported by the National Natural Science Foundation of China under Grant 62272293, Grant 62061146002, Grant 62020106005, Grant 62272301, Grant 61960206002, and Grant 623B2071. *(Huangwei Wu and Tingchao Fan contributed equally to this work.) (Corresponding author: Meng Jin.)*

Huangwei Wu, Tingchao Fan, and Xinbing Wang are with the School of Information Science and Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wuhuangwei@sjtu.edu.cn; kylin_f@sjtu.edu.cn; xwang8@sjtu.edu.cn).

Meng Jin is with the School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jinm@sjtu.edu.cn).

Tao Chen is with Computer Science Department, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: tachen.cs@gmail.com).

Chenghu Zhou is with the Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China (e-mail: zhouch@lreis.ac.cn).

Digital Object Identifier 10.1109/TON.2025.3650658

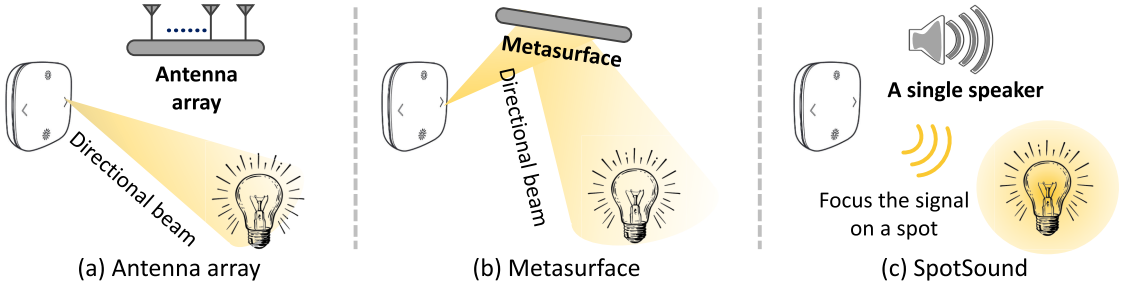


Fig. 1. Spatial selective transmission with beamforming and SpotSound.

force the transmitter to capture the distinctive characteristics of the target channel, we propose a conditional GAN (cGAN) based training model, which involves two discriminators: a *target decoder* and a *non-target decoder*. In the training process, the two decoders respectively check whether the generated signal is decodable on the target channel and the non-target channel. This forces the transmitter to generate signals in a channel-selective manner. By performing training with plenty of different channels, the transmitter can learn a latent mapping from an observed channel to the required signals. Once the mapping is obtained, the transmitter can generate a required signal for any indicated channel.

We further embed the above network into a system that overcomes additional practical challenges, including: i) how to make SpotSound expressive enough to operate on IoT devices; ii) how to make the model learn to deal with practical problems such as packet detection, channel selection, tolerance of channel measurement error, etc. iii) how to achieve pre-transmission channel measurement.

We implement SpotSound on Raspberry pi. Since wireless modules available on IoT devices do not support elaborate enough signal manipulation, we implement SpotSound on acoustic channels using the COTS speaker and microphone. We evaluate its performance in different environments and under different configurations. The results show that SpotSound can always make precise control of signal's coverage, achieving simultaneous transmission secrecy and reliability.

This paper makes the following contributions:

- We for the first time explore the feasibility of achieving spatial-selective transmission by leveraging the spatial-specific channel effect as a natural "beamformer".
- We reveal that deep learning is a particularly good fit to optimize the PHY-layer of wireless communication for objectives that are difficult to achieve with hand-crafted modulation algorithms (e.g., spatial-selective transmission).
- We implement SpotSound on low-end devices and show its ability to precisely focus its signal on spots with customized sizes ranging from $0.04m^2$ to $0.5m^2$.

Compared with the published MobiCom version, we in this version improve both the generalizability and reliability of SpotSound. First, we in Sec. IV-F propose a method to generate signal spots with customized sizes, which provides an on-demand trade-off between transmission reliability and security. The experimental results in Sec. VI-E show that by fine-tuning SpotSound with a specially designed dataset, we can effectively generate spots with different sizes, varying from $0.04m^2$ to $0.5m^2$. Second, to improve the generalizability of SpotSound, we in Sec. IV-D propose a data generator that is able to generate a large number of realistic synthetic CIRs from a small number of real-world measurements. Those synthetic data increase the diversity of the training dataset, which makes SpotSound generalize better to unseen environments. The evaluation results in Sec. VI-C demonstrate that by adding those synthetic CIRs in the training set, the BER of SpotSound in unseen environments decreases by more than 40% compared with the previous version. Third, the previous version of SpotSound is designed based on the assumption of channel reciprocity, which is not always true in practice. To solve this problem, we propose a new channel estimation method that achieves pre-transmission channel estimation without relying on channel reciprocity. The evaluation results in Sec. VI-F3 demonstrate that with the new channel estimation method, SpotSound can precisely focus the signal to the legitimate receiver (Bob) even when the Alice-to-Bob and the Bob-to-Alice channels differ seriously.

II. Threat Model

In our target scenario, the attacker is a malicious receiver (EVE) who tries to eavesdrop on the communication between legitimate transceivers (Alice and Bob). SpotSound's target is to ensure reliable communications between Alice and Bob, making the signal received at EVE resemble noise.

We do not assume Bob has any advantages over EVE in either prior knowledge, hardware, algorithm, or channel quality. Specifically, we assume that EVE has complete knowledge of: i) the communication medium between Alice and Bob (i.e., acoustic); ii) the frequency band for the communication; and even iii) the modulation and demodulation models used by Alice and Bob, based on which it can train an attacking model to decode the signals from Alice.

We also assume that EVE is not staying in the same place with Bob (e.g., their spacing is larger than a certain distance, say >50 cm). This assumption holds in practice since EVE will be exposed if she stays close to Bob. Besides, we also assume EVE is a powerful device that can: i) eavesdrop on any communications between Alice and Bob at any location; ii) record any signal in high fidelity; and iii) process the signal with any advanced signal processing techniques (e.g., train a strong DL network to decode Alice's signal).

III. Background and Intuition

A. Signal's Channel Behavior

In an indoor environment, signals transmitted from a transmitter are usually reflected by multiple surrounding objects, which cause them to traverse different, say P, propagation paths before reconvening at the receiver (as shown in Fig. 2). Each path p will cause an attenuation in signal strength denoted as $\alpha(p)$. If we denote the transmitted signal as $s(t)$, then the received signal $r(t)$ can be expressed as:

$$r(t) = \sum_{p=1}^{P} \alpha(p) \cdot s(t - d_p) + w(t) \quad (1)$$

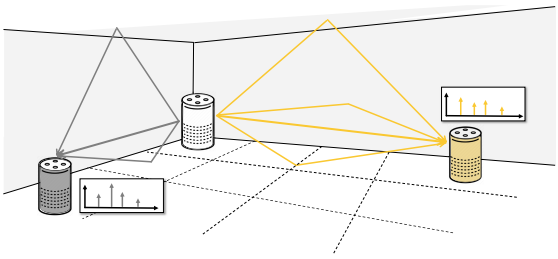


Fig. 2. Multipath reflection.

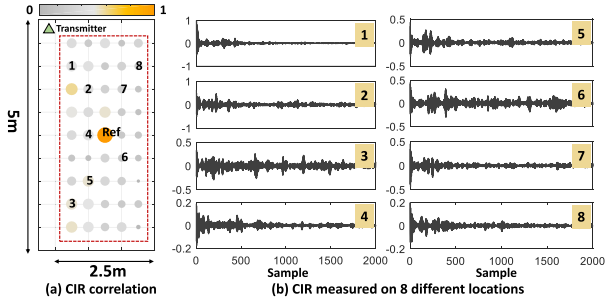


Fig. 3. CIR measured in different environments.

where d_p is the time delay of the p^{th} path, and $w(t)$ is the channel perturbation caused by device distortion and ambient noise. The received signal $r(t)$ can also be written as the temporal convolution of the transmitted signal $s(t)$ and the channel impulse response (CIR) $h(t)$:

$$r(t) = h(t) \otimes s(t) + w(t) \tag{2}$$

where $h(t)$ denotes the amplitude of the multipath signal with the delay of t.

Due to the signal's diverse propagation paths indoors, signals received at different locations will experience quite different channel effects (as shown in Fig. 2). Fig. 3 illustrates the extent of channel diversity measured in a living room. Specifically, we place a speaker at the top left corner of the room and place a microphone at 45 different locations in the room. We measure the CIR on each location and Fig. 3(b) illustrates the CIRs measured on 8 different locations. As can be seen, the 8 CIRs are quite different in both time distribution and amplitude. We further calculate the correlation coefficient between the CIRs measured on a reference location (as marked in Fig. 3(a)) and every other location. Fig. 3(a) visualizes the correlation coefficient of each location. We can see that the CIR varies significantly across locations, and decorrelates quickly with distance. When the distance is >1m, the CIR correlation decreases to less than 0.4.

B. Beamforming

Equation (2) tells that the quality of the received signal is highly determined by the channel it propagates through. Thus, to achieve spatial-selective transmission, a straightforward method is to directly control the signal's propagation behavior, making the signal propagate along a certain direction. There are two typical ways to achieve this target. The first way is to deploy an antenna array on the transmitter and use the beamforming technique to steer the signal's transmission direction, as shown in Fig. 1 (a). The second way is to instead deploy a large array of phase shifters (also known as a meta-surface) in the environment, as shown in Fig. 1 (b). By dynamically shifting the phase of the wireless signal propagating through it, the meta-surface can adaptively configure the channel that the signal propagates through.

However, a problem underlying both methods is that they need to make a difficult trade-off between cost and control granularity. Specifically, sophisticated control of signal propagation typically requires a large number of signal control elements (e.g., antennas or phase shifters). This makes antenna arrays prohibitively bulky and expensive, and they are thus unavailable on IoT devices. An alternative is to use meta-surfaces to offload this burden to the environment. However, this method comes with even higher infrastructure costs, setting a strong barrier to pervasive deployment.

C. Physical Channel as a Beamformer

In this paper, we propose a new approach that achieves spatial-selective transmission by leveraging the environment as a natural beamformer. Specifically, due to the signal's diverse propagation paths indoors, we can actually consider the channel effect as a *location-dependent filter* for the transmitted signal. Signals that propagate toward different locations can be considered as if passing through different filters. So, if the transmitter can predict the signal's transformation on the target channel and elaborately modulate the signal in a way that the embedded information can pass through *only* the filter of that channel, it can achieve spatial-selective transmission.

This new form of "beamforming" provides two benefits:

- It achieves spatial selective transmission by controlling the signal itself, which is much easier to control with high granularity, compared with its propagation behavior.
- Since the channel effect decorrelates rapidly with the distance between two receivers, such an environment-based beamformer can concentrate the signal to a small spot rather than a beam, achieving more precise physical isolation.

However, manually designing a signal modulation algorithm that dynamically generates signals targeting a given channel is an intractable task. It requires the modulator to first identify the distinctive channel characteristics of the target channel, then model how these characteristics in turn determine the features of the required signal, and based on which perform an elaborate manipulation of the transmission signal, making the signal match only the target channel.

We deal with this complexity with deep learning (DL). Specifically, we borrow the idea from cGAN (conditional GAN) [17], a DL framework which has shown great success in generating an image based on a text description of the image. A cGAN model involves a generator that generates images based on the descriptions and a discriminator that tries to identify whether the generated image matches the description. In the training process of cGAN, the generator is optimized to fool the adversarially-trained discriminator into identifying the generated images as satisfying. In this way, the discriminator can guide the generator to capture the visual characteristics contained in the text description and based on which generates corresponding images.

Inspired by cGAN, we propose a novel deep learning-based modulation algorithm, which *learns* to generate satisfying signals for any target channel, as shown in Fig. 4. Our DL module also has a generator (i.e., the transmitter Alice), which takes as input both the data bits and a measurement of the target channel and generates the modulated signals. Since the generated signals need to satisfy two objectives

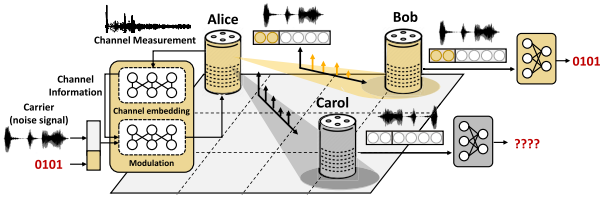


Fig. 4. Intuition in using environment as beamformer.

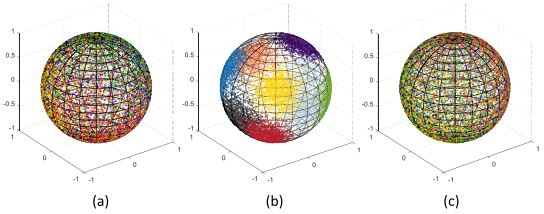


Fig. 5. Feature map of modulation signals: (a) signal generated by Alice; (b) signal received by Bob; (c) signal received by Carol.

simultaneously, i.e., decodable on the target channel and undetectable on other channels, we use two discriminators, which act as the target receiver (denoted as Bob) and a non-target receiver (denoted as Carol). These two discriminators respectively check whether the transmitted signal is decodable on the target channel and other channels. In this way, they can cooperatively guide the generator to generate signals that satisfy the two objectives simultaneously.

The DL network of SpotSound achieves the above objectives through an adversarial training process. Specifically, in our design, Carol's goal is to recover the bits embedded in the signal accurately. Alice and Bob try their best to hide their communication from Carol, and at the same time, boost the performance of their own communication. By maximizing the decoding rate of Bob while minimizing that of Carol, Bob and Carol can guide Alice through back-propagation to learn the following two things:

- An attribute representation of the channel which captures distinctive channel information for signal modulation.
- A latent mapping from the channel information to the modulation signals targeting that channel.

We in Fig. 5 provide a visualization of how Alice achieves spatial-selective transmission. In this case, Alice generates 8 different symbols, which are represented by acoustic signals with different features. Fig. 5 shows the feature maps (which are compressed to a spherical surface) of the generated signals, where points with the same color represent the same symbol. Figs. 5 (a)∼(c) respectively show the feature maps of the original signals, signals received on the target channel, and signals received on a non-target channel. As can be seen, the signal forms 8 non-overlapping clusters only after it passes through the target channel. This indicates that Alice can *embed information into channel-selective features of a signal*, making the information detectable only after the signal passes through the target channel. So an eavesdropper cannot decode the signal even if it can directly obtain the original signal (i.e., the one in Fig. 5 (a)).

We make two important remarks on the above model:

- Our model **can be generalized to unseen channels**. Given an arbitrary measured channel, our model pre-processes the transmission signals so that these signals are only decodable on that specific channel. Note that the key to the pre-processing is the features of the channel which particularly capture the distinctiveness between receivers' locations, but not all the details of the measured CIR (which may be related to other factors such as device noise and ambient interference). Hence, the model does not need to see all the physical environments or channels to generalize. Instead, it only needs to: i) find out the channel features that are really related to the pre-processing process; and ii) learn a mapping between the channel features and the pre-processing operations.
- Our model **can defend unseen eavesdroppers**. Specifically, we do not assume that the eavesdropper will use the same network model as the decoder modules. The role of both Bob and Carol is simply a discriminator which guides Alice to generate signals for an indicated channel. We will show in Secs. VII and VI that signals received on a non-target channel resemble noises (e.g., as in Fig. 5 (c)), which can i) pass the randomness test; and ii) fool an unseen neural network that is much stronger than Bob and Carol.

IV. SpotSound's DL Model

A. Organization and Objectives

Organization. Fig. 6 shows the overall architecture of the model, which involves three parties: Alice, Bob, and Carol. All of them are neural networks, whose parameters are denoted as $\theta_{\mathbf{A}}$, $\theta_{\mathbf{B}}$, and $\theta_{\mathbf{C}}$. In our design, Bob and Carol share the same structure but have different parameters.

Alice takes as inputs the data information X, a real-valued CIR of the target channel $\hat{h}_B$, and a random Gaussian white noise signal S_R. It embeds the information X in the noise signal S_R and outputs the modulation signal S. Specifically, it encodes 4 bits into one symbol, and each symbol x_i is represented as a 16-dimensional one-hot vector. It concatenates x_i with a 300-sample noise signal s_R^i and produces a 600-sample modulation signal s_i. This means a 320 bps bitrate when the microphone's sampling rate is 48KHz. Considering that the multipath effect will create interference between adjacent symbols, to take such inter-symbol interference into consideration, Alice takes as input the whole packet $X = \{x_1, \ldots, x_M\}$ and outputs the modulated signal $S = \{s_1, \ldots, s_M\}$. We set $M = 10$ in our implementation, which is sufficient to capture the inter-symbol interference considering that the delay spread of the multipath signal is usually less than 0.1s (8-symbol length) [18]. The stochasticity introduced by S_R ensures that S remains non-deterministic even with identical X and $\hat{h}_B$, preventing eavesdroppers from predicting the transmission.

The signal S is then transmitted to Bob and Carol through the channels h_B and h_C. The signal received on these two receivers are represented as $R_B = S \otimes h_B + w_B$ and $R_C = S \otimes h_C + w_C$, respectively. After receiving the signals, Bob and Carol process the signals to recover the data X. We represent what they obtained by X_B and X_C, respectively.

Objectives. The objectives of the three parties are as follows. Carol's goal is to minimize the error between X and X_C. Alice and Bob try to minimize the error between X and X_B but also hide their communication from Carol.

We denote Alice's output on data X and random noise S_R as $f_A(\theta_A, X, S_R)$, and denote Bob's and Carol's outputs on signal R_B, R_C as $f_B(\theta_B, R_B)$ and $f_C(\theta_C, R_C)$, respectively. We use L1 distance to measure the error in data bit recovery as $d(X, X_\bullet) = \sum |X(i) - X_\bullet(i)|$ (where $X_\bullet \in \{X_B, X_C\}$). Then the loss function for Carol can be expressed as:

$$L_C(\theta_A, \theta_C) = d(X, f_C(\theta_C, f_A(\theta_A, X, S_R) \otimes h_C + w_C)) \quad (3)$$

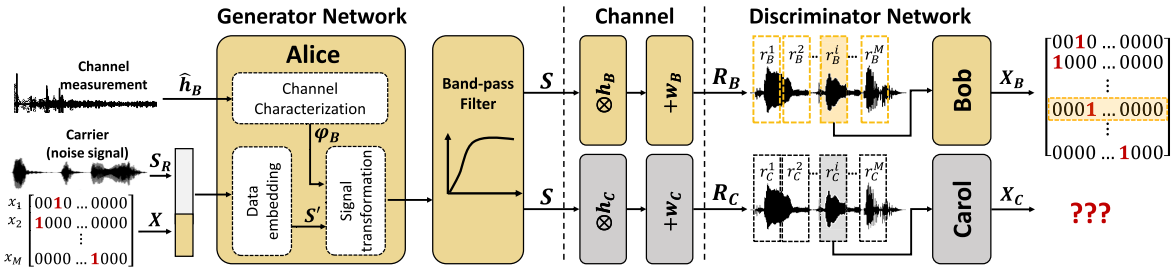


Fig. 6. Architecture of SpotSound's DL model.

Then we obtain the "optimal Carol" by minimizing this loss:

$$O_C(\theta_A) = argmin_{\theta_C} L_C(\theta_A, \theta_C) \tag{4}$$

Similarly, we obtain the loss function for Bob as:

$$L_B(\theta_A, \theta_B) = d(X, f_B(\theta_B, f_A(\theta_A, X, S_R) \otimes h_B + w_B)) \tag{5}$$

We define a loss function for the whole system by combining L_B and the optimal value of L_C:

$$L(\theta_A, \theta_B) = L_B(\theta_A, \theta_B) - L_C(\theta_A, O_C(\theta_A)) \tag{6}$$

By minimizing the above loss function, SpotSound can minimize Bob's decoding error and meanwhile maximize the decoding error of the "optimal Carol".

B. Design Details

Now we introduce the design details of SpotSound's DL model that achieves the above target.

1) Signal Generator: Given the data X, the target channel h_B, and a random noise S_R, Alice's task is to find a function f_A that generates the modulation signal by embedding X into S_R in a way that the information is decodable only on channel h_B. Towards this goal, we decompose the generator into the following three sub-modules (as shown in Fig. 6):

- The *channel characterization module*, which captures distinctive channel characteristics contained in $\hat{h}_B$ with fully-connected (FC) layers. The extracted information φ_B is then used for signal generation.
- The *data embedding module*, which embeds the data information X into the random noise S_R with a deconvolutional network. It discovers channel-sensitive features of a signal, and embeds the information X onto those features, so that the produced signal S' (especially the embedded information) is detectable only on a small proportion of channels. This is analogous to forming a narrow beam which can focus the signal only on a small area.
- The *signal transformation module*, which is a transformation network that applies a transform to S' conditioned by φ_B. This makes the final output signal S perfectly and exclusively fit the target channel h_B. This is analogous to steering the beam to the target position.

Note that although we decompose the transmitter into three sub-modules, we still consider them as a whole in the training process and optimize each sub-module in a way that all the modules can jointly achieve the best end-to-end performance. So that the target of each sub-module can be achieved through back-propagation during the training process.

2) Discriminator: On the receiver side, the received signal is first sliced into symbols (as shown in Fig. 6), then each symbol is processed by the receiver. The core of the receiver is a CNN-based classifier which infers the transmitted data X based on the received signal (i.e., r_B or r_C).

TABLE I
SUBJECTIVE EVALUATION OF AUDIBILITY

Signal	MOS
SpotSound	3.7500 ± 0.7071
White noise (20-23 kHz)	4.1250 ± 0.6409

3) Frequency Selection: To make the modulated signal inaudible to human ears, the transmitter should transmit the signal only on the inaudible bandwidth. Based on a widely accepted consensus that the near-ultrasonic frequency range above 17 kHz is inaudible [19], [20], [21], [22], SpotSound works on the 17-20 kHz band. But how to make Alice learn to modulate the information only on those frequencies? To achieve this, we in the training process add a band-pass filter to the output of the transmitter (as shown in Fig. 6). Only the signals on the 17-20 kHz band can finally arrive at the receiver side. This forces the transmitter to encode information only on the specified frequency band to minimize the training loss. Note that frequency selection is integrated into the end-to-end learning process without any explicit module in the transmitter or any additional loss term in Eq. (6). Moreover, as shown in Sec. VI-G, SpotSound can achieve throughput up to 480 bps with such a bandwidth, which is sufficient to support multiple practical applications [23], such as geofenced connectivity, authentication, and information sharing between devices.

We further conduct a subjective experiment to validate the inaudibility of SpotSound's signal. Specifically, approved by our IRB, we invite 8 volunteers (5 males and 3 females) with ages ranging from 22 to 54 to evaluate whether the SpotSound's signal is audible. In the experiment, the volunteers are engaged in a 10-minute casual conversation, during which we play the SpotSound's signal 1 meter away at a random time within each minute. Volunteers will then answer the question: How much interference was felt during the conversation? *1 - Too distracted to focus, 2 - Noticeable and distracting, 3 - Slight but not distracting, 4 - Uncertain, 5 - Not aware.* In the same way, we have also evaluated white noise on the 20-23 kHz band for comparison. The mean opinion scores (MOS) of users' ratings are shown in Table. I. We can see that the audibility of SpotSound's signal is comparable to that of the white noise above 20 kHz.

C. Model Compression

We in this section compress our DL model to make sure that SpotSound is expressive enough to support real-time processing with limited memory and computational resources.

The complexity of our model mainly comes from the FC layers and the convolution layers. In our design, FC layers are used to capture the long-range dependencies of the signal caused by the multipath effect, which cannot be captured by

TABLE II
MEMORY AND COMPUTATION COST OF THE MODELS

Model	Description	Memory	Computation	Energy
I	LRA+DSC+ pruning	188,467	22.16 MFLOPS	1.06×10^{-4}J
II	LRA+DSC	549,795	26.71 MFLOPS	1.29×10^{-4}J
III	DSC	1,234,743	100.84 MFLOPS	4.89×10^{-4}J

TABLE III
THE MEMORY SIZE AND FLOPS SUPPORTED BY TYPICAL IOT DEVICES

Hardware	Memory Size	MFLOPs
STM32F412	262,144	33
STM32F743	524,288	158.4
Intel Edison	1,073,741,824	4000
Raspberry Pi 4 Model B	8,589,934,592	12288
Qualcomm Snapdragon 800	>268,435,456	18841.6
Nvidia Tegra K1	>268,435,456	18841.6
pixel 1	8,589,934,592	18841.6

convolution layers, and convolution layers are used to extract features. However, the use of FC layers exponentially increases the parameter size and thus the memory cost. Although the convolution layer uses much fewer parameters than the FC layer, it still incurs high computation costs.

We reduce the model's complexity with three techniques. First, we use the *low-rank approximation* (LRA) technique to find a compact representation of the parameter set on the FC layers, with limited loss of information. With this technique, we reduce the parameter size of each FC layer by 62%. Second, we replace the standard convolution layers with depthwise separable convolution (DSC) layers, which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called pointwise convolution. By using DSC layers, we can reduce the computation cost by about 85%, with only a small performance degradation. Last, we adopt *filter pruning* to further compress the model. Specifically, we calculated the L1-norm of weights in each FC and CNN layer and preserved those with the largest L1-norm.

Based on the above techniques, we have proposed three versions of models, each compressed with a different compression strategy. Table II shows the memory cost (i.e., parameter size) and computation cost (i.e., floating point operations, FLOPs) of each model. We also show in Table III the maximum supportable memory size and FLOPs of 7 typical IoT devices. As can be seen, the most lightweight version (Model I) is suitable for all IoT devices. We have also tested the performance of the three models in Sec. VI-D. The results show that Model I incurs only a slight performance degradation compared with the other two models.

We further analyze the energy consumption of the proposed models. Specifically, the power footprint of a model is proportional to the multiply-add operations and the amount of data that needs to be loaded from DRAM, which can be expressed as follows:

$$E = (N_{mul-add} \cdot (E_{mul} + E_{add}) + N_{in} \cdot E_{ram}) \quad (7)$$

where $N_{mul-add}$ is the number of 32-bit-float multiply-add operations associated with the model; E_{mul} and E_{add} are the energy consumption of performing one multiplication and one 32-bit-float-addition, respectively. N_{in} is the input length of the model and E_{ram} is the energy consumption of loading a 32-bit float from DRAM.

Reference [24] shows the power footprint of the above arithmetic and memory operations on a 45nm CMOS process. Based on Eq. (7) and the results shown in [24], we estimate the energy consumption of our three models in generating one 60-bit packet. The results are shown in Table II. As can be seen, the energy consumption of SpotSound keeps lower than 1mJ, which is affordable on most IoT devices.

D. Data Collection and Synthesis

To improve the generalizability of SpotSound and ensure that the trained model can be generalized to unseen environments, we should train the network with a sufficient amount of data collected in various environments. In this section, we first introduce how we build a dataset that contains real-world measurements collected in various indoor environments. Then we introduce a method to generate synthetic data using a DL-based channel synthesis module. This module can generate a large number of realistic synthetic channels from a small number of measured channels. Both the measured data and the synthetic data are used to train the network.

1) Data Collection: Our measured channels are collected from 12 different indoor environments on a university campus, including seminar rooms, cafeterias, offices, and dorms. In each environment, we put a pair of transceivers (denoted as A and B) on 40∼90 pairs of different locations. For each pair of locations, we first measure the realistic channel between A and B. The measurement result is used as Bob's channel layer (i.e., h_B in Fig. 6) in the training process. Then, since in practice, Alice cannot perfectly estimate the target channel, while can only estimate the channel using the method proposed in Sec. V-A. So, we further measure the channel using the proposed method, and the measurement result is used as the input of Alice's signal generator module (i.e., $\hat{h}_B$ in Fig. 6). We collect more than 600 measured CIRs for each pair of locations. We in total collected more than **430,000 measured CIRs** during the data collection process.

2) Data Generation: Although we can collect an extensive dataset that covers various indoor environments, the number of environments involved in the training set is still limited, which may further limit the generalizability of the network. To solve this problem, we train a channel synthesis model which takes measured data as input and generates a large number of synthetic data representing various unseen channel environments. While there are various kinds of generative models (such as GAN and VAE [25], [26], [27]), none of them can generate high-quality data with a small dataset. We solve this problem with a stage-wise training strategy, as shown in Fig. 7. Specifically, we first train a VAE with a small dataset. The trained VAE model can generate a coarse-grained sketch of CIR data. Then, by fine-tuning the generator through adversarial training, it could learn to generate a more realistic, diverse, and fine-grained dataset. In what follows, we describe the two-stage strategy at a high level and in the context of our problem, and we refer the interested reader to [28] and [29] for a more detailed exposition of VAE and GAN.

VAE is best known for its great understanding of data's latent representation, without relying on a large amount of data. In our context, the CIR data we try to generate is mainly determined by the objects in the environment (i.e. large furniture as reflectors), and is characterized by features such as the number of dominant objects, signal's propagation distance on each reflection path, the energy reflected by each object, and etc. Thus we leverage VAE to extract the environment's

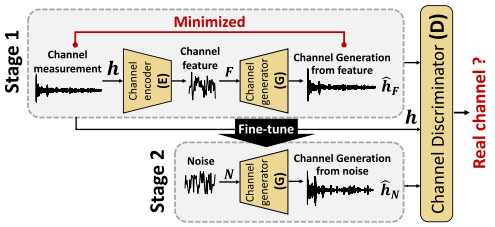


Fig. 7. The channel synthesis module.

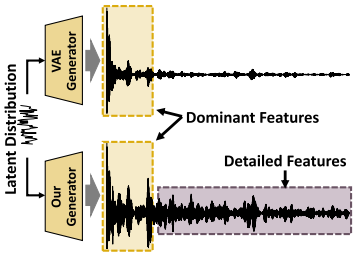


Fig. 8. Comparison between CIR generated w/wo adversarial training.

effect on the channel by learning the distribution of those latent features. Once the underlying distributions are learned, we can draw new samples from them to generate synthetic CIR for unseen environments.

Figure 7 (Stage 1) shows the framework of SpotSound's VAE model, which consists of a channel encoder $E(\cdot)$ and a channel generator $G(\cdot)$. The encoder takes the measured channel h as input and compresses the input channel into a latent feature representation (noted as $F = E(h)$) on a low-dimensional feature space. The channel generator then reconstructs the CIR data from the latent features as $\hat{h}_F = G(F)$. By minimizing the distance between the input CIR h and the reconstructed result $\hat{h}_F$, the VAE model learns a representative low-dimensional distribution of the latent variables.

However, the VAE model captures only the dominant features of the CIR and thus generates only the coarse-grained sketch of CIR data, leaving out the detailed features, as shown in Fig. 8. This on one hand limits the diversity of the generated CIR and, on the other hand, makes the generated data different from the realistic measured CIR. We solve this problem by fine-tuning the generator through adversarial training as shown in Fig. 7 (Stage 2). Specifically, to generate more realistic and diverse measured CIRs, in this process, the generator takes either feature representation or the random noise as input, and learns to generate synthetic CIR data based on either of them. We denote the output of the channel generator on the noise N as $\hat{h}_N = G(N)$ and that on the feature distribution as $\hat{h}_F = G(F)$. The synthetic CIRs are then evaluated by a channel discriminator $D(\cdot)$, which tries to distinguish the synthetic CIRs ($\hat{h}_F$ and $\hat{h}_N$) from the measured CIRs h.

The loss function we use to train the channel synthesis module is shown as follows:

$$\max_{D} \min_{G} L(G, D) \tag{8}$$

where,

$$\begin{aligned} L(G,D) &= \mathcal{L}_G + \mathcal{L}_{D_m}(h) + \mathcal{L}_{D_s}(G(F)) + \mathcal{L}_{D_s}(G(N)) \\ &= \frac{1}{n}\sum_{i=1}^{n} |h(i) - G(F)(i)| \\ &\quad + \mathbb{E}_{p_m(h)} \log(D(h)) \\ &\quad + \mathbb{E}_{p_s(G(F))} \log(1 - D(G(F))) \\ &\quad + \mathbb{E}_{p_s(G(N))} \log(1 - D(G(N))) \end{aligned} \tag{9}$$

In Eq. 9, $\mathcal{L}_G$ is the generation loss, which quantifies the distance between the measured CIRs h and synthetic CIRs $G(F)$. $-\mathcal{L}_{D_m}(h)$ and $-\mathcal{L}_{D_s}(\cdot)$ are the discriminator loss which quantifies the distance between the true label and the classification result of the discriminator. p_m is measured CIR distribution, and p_s is synthetic CIR distribution.

From Eq. 8 and Eq. 9, we can observe that the channel discriminator tries to distinguish between the realistic and the synthetic data by maximizing $\mathcal{L}_{D_m}(h) + \mathcal{L}_{D_s}(G(F)) + \mathcal{L}_{D_s}(G(N))$. The channel generator tries its best to cheat the channel discriminator by minimizing $\mathcal{L}_G$ and $\mathcal{L}_{D_s}(G(F)) + \mathcal{L}_{D_s}(G(N))$. Through this minimax game, the channel generator learns to generate more realistic CIR data.

It is worth noting that, during the training process, we train the channel synthesis module with both the measured CIR and random noise. In the CIR generation process, the channel generator synthesizes the data based only on the noise. Then the whole network of SpotSound is trained on the dataset consisting of both the measured data and the synthetic data.

E. Module Training

Dataset. We collected more than **430,000 measured data** and synthesized more than **1,000,000 synthetic data**. Among the measured CIRs, 30,000 CIRs are used to train the channel synthesis module, and 300,000 CIRs are used to train the whole network. More than 900,000 synthetic CIRs are leveraged in the training process. We also leave 2,000 measured CIRs and 4,000 synthetic CIRs to fine-tune the spot customization module, which we will elaborate on in Sec. IV-F. We leave 98,000 measured CIRs and more than 100,000 synthetic CIRs for evaluation.

Training strategy. In the training process, we alternate the training of Alice and Bob with that of Carol. The training may for example proceed as follows. Alice may initially produce signals that neither Bob nor Carol can decode. By training for a few steps, Alice and Bob may discover a way to communicate that allows Bob to decode Alice's signal, but which is not understood by (the present version of) Carol. In particular, Alice and Bob may discover some trivial ways to hide the information. After a bit of training, however, Carol may start to break this "code". With some more training, Alice and Bob may discover refinements, in particular, some ways that exploit the channel diversity better for information hiding. Carol eventually finds it impossible to decode the signal.

F. Customized Spot Size

We in this section describe how we can generate signal spots with customized sizes to provide a trade-off between transmission reliability and security. Specifically, a smaller spot leads to more secure communications, while a larger spot can be more tolerant of the dynamic scenarios where the legitimate receiver (i.e., a smartwatch worn on the user's arm) moves within a certain range. To meet this requirement, we aim to extend SpotSound to generate spots with different sizes. For example, having trained a default model for a default spot size using a large dataset, we expect to extend it to generate spots of smaller or larger sizes using a small dataset.

To achieve this, one challenge we meet is how the generator network can know the desired spot size. One naive solution

is directly feeding the desired size value to the network. However, it is difficult for a network to build a direct map between the spot size value and the characteristics of the channel toward the target location spot. To solve this problem, we fine-tune the basic model with a specially designed dataset, based on which the model learns to increase or decrease the size of the spot that the transmitter targets.

Recall that in the training process, we divide the collected CIRs into three sets, which are respectively used as the input of the signal generator (denoted as $\hat{h}_{AB}$), and the channel layer of the target receiver (i.e., Bob's channel, denoted as h_{AB}) and the non-target receiver (i.e., Carol's channel, denoted as h_{AC}). To make the transmitter learn to focus the signal on the target spot, when building the training dataset, we make sure that the correlation between $\hat{h}_{AB}$ and h_{AB} is much higher than that between $\hat{h}_{AB}$ and h_{AC}. Here, the boundary between $cor(h_{AB}, \hat{h}_{AB})$ and $cor(h_{AC}, \hat{h}_{AB})$ determines the size of the target spot. Therefore, we can control the size of the generated spot by adjusting the boundary in the training dataset.

We define the lower bound of $cor(h_{AB}, \hat{h}_{AB})$ (denoted as $cor_{min}(h_{AB}, \hat{h}_{AB})$) as the security coefficient (SC) to represent the security degree of the system. To increase the system's security degree, we increase the security coefficient to decrease the size of the system's target spot. Our experimental results in Sec. VI-E tell that when we set $cor(h_{AB}, \hat{h}_{AB}) > 0.9$ and $cor(h_{AC}, \hat{h}_{AB}) < 0.6$, the network can learn to focus the signal on a $0.04m^2$ area. To enlarge the size to $0.25m^2$, we can decrease the lower boundary of $cor(h_{AB}, \hat{h}_{AB})$ to 0.7 and decrease the upper boundary of $cor(h_{AC}, \hat{h}_{AC})$ to 0.4. The above results are consistent with the results in Fig. 3 (a) which describes how the channel correlation changes with the distance between two locations.

Figure 9 illustrates the model fine-tune process, where we employ transfer learning to reduce the number of parameters that need to be learned during the fine-tune process. Specifically, when fine-tuning the generator network (i.e., Alice), we can freeze the channel characterization module and retrain only the data embedding and the signal transformation module. This is because the way to extract the distinctive channel characteristics is common and is unrelated to any specific spot size. Thus the module trained for one spot size can be directly used to generate spots with other sizes. Besides, the receiving module (Bob) is also frozen since the size of the spot is determined only by the transmitter side.

Further, to make the model pay more attention to the change in the boundary, we fine-tune the model using the CIR data that is located around the boundary. For example, if the boundary is set as $cor(h_{AB}, \hat{h}_{AB}) > 0.7$ and $cor(h_{AC}, \hat{h}_{AB}) < 0.5$, we fine-tune the model using the dataset that satisfies $0.7 < cor(h_{AB}, \hat{h}_{AB}) < 0.8$ and $0.4 < cor(h_{AC}, \hat{h}_{AB}) < 0.5$.

V. SpotSound in Practice

With only the core DL model introduced in Sec. IV, SpotSound cannot work in practice because of two unsolved issues: i) pre-transmission channel estimation; and ii) packet detection. This section presents the design enhancement for SpotSound, in order to solve these two issues.

A. Channel Estimation

To estimate the channel (i.e., the CIR), a common practice [30], [31], [32] is to send a known training signal s_T. Then,

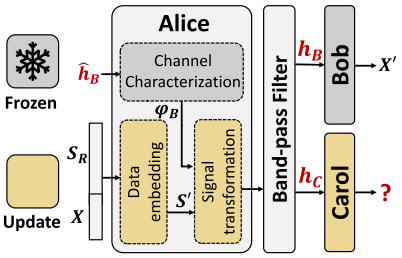


Fig. 9. Model fine-tuning for customized spot size.

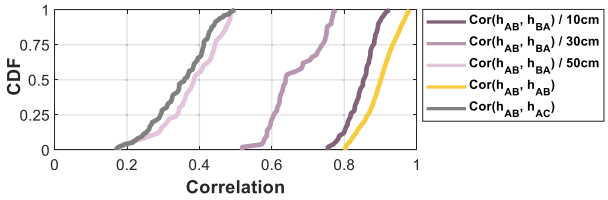


Fig. 10. Channel asymmetry caused by intra-device distances between the speaker and the microphone.

CIR can be derived from the deconvolution of the received signal s'_T and the transmitted training signal s_T. Since the calculation of convolution is nontrivial, the common trick to derive CIR is to convert temporal convolution into multiplication in the frequency domain, followed by an inverse Fourier transform:

$$\hat{h} = \Im^{-1}\{S_T^* S'_T\} \tag{10}$$

where $\Im^{-1}$ denotes the inverse Fourier transform. S'_T is the Fourier transform of the received signal s'_T. S_T^* is the complex conjugate of s_T.

However, there are two challenges in applying this channel estimation design. First, in our case, the task of channel estimation is shifted to the transmitter (i.e., Alice). How can Alice know the channel prior to transmission? Second, transmitting a known signal will expose the transmission process to an eavesdropper. How to make the channel estimation imperceptible to eavesdroppers? We introduce how we solve these problems in the following section.

1) Achieving Pre-Transmission Channel Estimation: To achieve pre-transmission channel estimation, a potential solution is to leverage channel reciprocity–the property that the channel from a node, say Alice, to another node, say Bob, is the same as the channel from Bob to Alice. Based on this phenomenon, Alice can estimate the Alice-to-Bob channel based on Bob's response [33]. This is reasonable when the same antenna is used for both transmitting and receiving (e.g., antennas in RF systems). However, since audio devices use different and spatially separated "antennas" (speaker and microphone) for transmitting and receiving, the distance between the microphone and the speaker on the same device may reduce the correlation between the Alice-to-Bob channel (from Alice's speaker to Bob's microphone, denoted as h_{AB}) and the Bob-to-Alice channel (from Bob's speaker to Alice's microphone, denoted as h_{BA}), making them not perfectly reciprocal.

To experimentally show the effect of this asymmetry, we collected more than 100 pairs of reciprocal CIRs in 10 different environments under different speaker-microphone distances and calculated the correlation between each pair of CIRs. For comparison, except for the correlation between channels ($Cor(h_{AB}, h_{BA})$), we also collected that of the same channel measured at different times ($Cor(h_{AB}, h_{AB})$) and

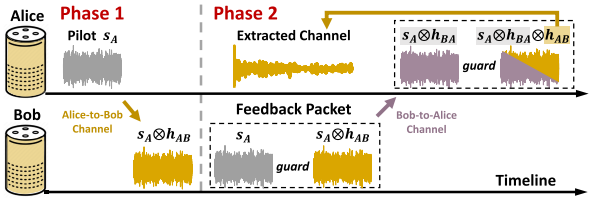


Fig. 11. Illustration of the two-phase proposed channel estimation method.

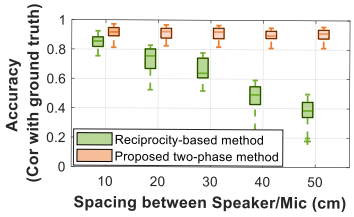


Fig. 12. Comparison between two channel estimation methods.

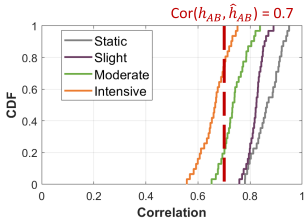


Fig. 13. Channel correlations at different dynamic levels.

different channels ($Cor(h_{AB}, h_{AC})$, where receivers B and C are 50cm away from each other). We can see from Fig. 10 that the average correlation of reciprocal CIRs drops from over 0.8 to no more than 0.4 as the spacing between the speaker and microphone on the same device increases from 10cm to 50cm. Thus, although channel reciprocity holds on small IoT devices (e.g., smart switch or smart speaker), it may not apply to large smart appliances (e.g., smart TV) with non-negligible intra-device distance.

To address the above channel asymmetry problem and make SpotSound generalized to all possible distributions of the speaker and microphone on a device, we propose a two-phase pre-transmission channel estimation method, which can acquire the Alice-to-Bob channel at the Alice side. The proposed method is based on the fact that the distortion of the signal from different channels is multiplicative in the frequency domain. Thus, the Alice-to-Bob channel could be extracted from the division between the pilot signal passing through both the reciprocity channels and that passing through only the Bob-to-Alice channel. Specifically, in the first phase, Alice transmits a pilot signal to Bob. The signal received by Bob can be represented as:

$$s_{AB} = \zeta^{-1}\{S_A \cdot H_{AB}\} \tag{11}$$

where S_A is the Fourier transform of signal s_A sent from Alice, and H_{AB} is the channel frequency response of the Alice-to-Bob channel. Then, as shown in Fig. 11, in the second phase, Bob will send a feedback packet consisting of the original s_A and the received s_{AB} back to Alice. A guard interval is used to mitigate interference between the two parts caused by multipath. The received packet at Alice is:

$$s_{BA} = \xi_A \cdot \delta(t) + \xi_{AB} \cdot \delta(t+\tau) \tag{12}$$

where,

$$\xi_A = \zeta^{-1}\{S_A \cdot H_{BA}\} \tag{13}$$

$$\xi_{AB} = \zeta^{-1}\{S_A \cdot H_{AB} \cdot H_{BA}\} \tag{14}$$

where H_{BA} is the channel frequency response of the Bob-to-Alice channel, $\delta(t)$ is the impulse function and τ is time delay between s_A and s_{AB} in the feedback packet. Denoting ζ as the Fourier transformation, the Alice-to-Bob channel could be derived from Eq. (13) as follows:

$$H_{AB} = \frac{\zeta\{\xi_{AB}\}}{\zeta\{\xi_A\}} = \frac{S_A \cdot H_{AB} \cdot H_{BA}}{S_A \cdot H_{BA}} \tag{15}$$

We compare the reliability of the proposed two-phase method and the method proposed in [33] which leverages channel reciprocity for pre-transmission channel estimation. Specifically, we use those two methods to obtain the Alice-to-Bob channel under different spacing between the microphone and speaker on the same device. We also measure the real Alice-to-Bob channel under different distances as ground truth, and for each method, we calculate the correlation (Cor) between the extracted Alice-to-Bob channel and the real-world measurement as the accuracy of different methods. A higher correlation indicates a higher accuracy in channel estimation. Fig. 12 shows the results. As can be seen, the performance of the reciprocity-based method degrades significantly with the increased speaker-microphone distance. In comparison, the CIR extracted by the two-phase method consistently achieves a higher than 0.8 correlation with the real-world measurement, even when the speaker-microphone distance is 50cm (which is larger than the size of most IoT devices). This verifies its capability to mitigate the channel asymmetry problem.

2) Making Channel Estimation Imperceptible: To make the channel estimation imperceptible to an eavesdropper, we use random white noise as the pilot signal s_A. Note that s_A is a pre-shared secret, known only to the legitimate transceivers (Alice and Bob). Consequently, a legitimate user like Bob can detect the signal via cross-correlation with the known template [34], whereas an eavesdropper lacking this template cannot distinguish the transmission signal from ambient noise. The s_A shared between two devices can be initialized either manually or by leveraging the existing automatic pairing methods for IoT devices [35], [36]. To mitigate the risk of an eavesdropper learning s_A over time, Alice and Bob also update it periodically. Specifically, they can use the old sequence as a seed to generate the new one using a hash function, which can be a public function known to EVE.

Coherence time and channel estimation frequency. Note that the channel estimation process can be triggered periodically instead of being performed for every transmission, even when there's simple human motion or minor environmental changes. This is because the difference between h_{AB} (the real channel during transmission) and $\hat{h}_{AB}$ (the generator network input) during the training tolerates a certain degree of channel variation. For example, when the lower bound of $cor(h_{AB}, \hat{h}_{AB})$ is 0.7 (the default bound for customized spot size), the channel remains coherent as long as the correlation coefficient is greater than this threshold. To demonstrate the coherence time in practical implementation, we measure $cor(h_{AB}, \hat{h}_{AB})$ with the same settings in Sec. VI-F (2) under different dynamic levels. As shown in Fig. 13, a small part of the correlation measurements under moderate dynamic and a large part of the measurements under intensive dynamics fall below 0.7; meanwhile, all the measurements for static level and slight dynamics are higher than 0.7. Our experiments

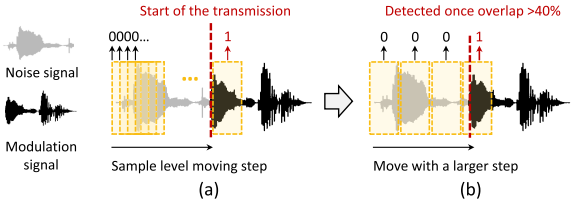


Fig. 14. Packet detection. (a) moving the window sample by sample; (b) moving the window with a long step.

in VI-F (2) also show that a low-frequency channel re-estimation is sufficient for SpotSound to achieve a decent communication performance in a highly dynamic environment.

B. Packet Detection

In a practical communication system, Alice transmits a known preamble at the beginning of each packet [37], [38], so that Bob can locate the packet start and slice the packet into symbols (i.e., 600-sample segments) for further demodulation. This is, however, infeasible in our case since repeatedly transmitting any definite signal (even a noise-like signal as those used in channel estimation) can leave an eavesdropper detecting it by performing self-correlation. So, we need a method to detect the start of a packet without a preamble.

To achieve this, we design a ResNet-based module for packet detection that distinguishes modulated signals from noise. Specifically, the receiver first samples the signal with a moving window, and then feeds the signal in each window to the packet detection module. The module then determines whether the window contains a modulated signal. One way to train this model is to label the training signals that contain a complete symbol as positive and those that contain only ambient noise as negative. However, in this way, the receiver can detect the modulated signal only when the window aligns tightly on one symbol, as shown in Fig. 14 (a). So, to detect the modulated signal, the decoder has to move the window sample by sample, which leads to a high detection delay.

To solve this problem, we propose to train a module that can detect the modulated signal once the window intersects with a symbol, as shown in Fig. 14 (b). In this way, the receiver can locate the packet's start in a coarse-to-fine manner. Specifically, we first move the window with a longer step, as shown in Fig. 14 (b). Once a symbol is detected, the window is moved with finer granularity (e.g., sample by sample) to locate the start of the packet using the decoder module. In the training process, we assign a binary label to each window to indicate whether it contains a symbol or not. A window that overlaps more than 40% with a symbol is set as positive. Note that the packet detector module is trained independently and is not involved in the end-to-end training of SpotSound's DL model. We evaluate this module across 8 different environments, as shown in Fig. 15. All the environments are unseen by SpotSound. The testing locations are marked in the figure. In each environment, we transmit 100 packets and check whether the packet detection module can successfully detect them. As shown in Table. IV, the packet detection accuracy achieves 100% in all the environments.

TABLE IV
PACKET DETECTION ACCURACY IN UNSEEN ENVIRONMENTS

Scene	1	2	3	4	5	6	7	8
Accuracy	100%	100%	100%	100%	100%	100%	100%	100%

VI. EVALUATION

A. Experimental Methodology

Implementation. We implement SpotSound on Raspberry Pi 4 Model B, which is connected with an EDIFIER R1200TII speaker and a TakStar TCM-400 microphone. The frequency range of the acoustic signal is set at 17-20 kHz, and the sampling rate of the microphone is set at 48 kHz. Since Raspberry Pi's onboard sound card does not support audio output, we use UGREEN external USB sound cards instead. SpotSound's DL model is implemented using TensorFlow Lite, and its I/O is implemented using the PyAudio library.

We have implemented three versions of SpotSound, using the models listed in Table II. We compared their performance in Sec. VI-D and selected Model I as the default implementation. The default security coefficient of SpotSound is set at 0.75. We have also trained three versions of EVEs, i.e., one normal version and two strengthened versions. The normal EVE shares the same structure as Bob. The strengthened version ① uses four times the number of feature channels in the convolution layers, compared with Bob. The strengthened version ② further replaces the first two convolution layers of version ① with FC layers. Note that EVE and Carol play different roles in our design. Carol acts as a discriminator in the cGAN model while EVE is the attack model. Carol guides Alice to run information hiding. In training the three EVEs, we freeze Alice, so the trained EVEs are "unseen" to Alice. We use the normal version as the default implementation of EVE. In Sec. VI-H we evaluate SpotSound's performance in resisting all three EVEs.

Experimental setup. The experiments are performed in two kinds of environments: i) an apartment, which has four different rooms, i.e., a 7m×3.8m living room, a 3.3m×2.6m study room, a 3.8m×1.7m enclosed balcony, and a 4.2m×1.8m kitchen; ii) four different office rooms with different sizes. Fig. 15 shows the floor map for the experiment scenes. **Note that all of the testing environments are unseen by SpotSound in the training process.** To thoroughly evaluate the security of SpotSound, we also consider the case where Bob and EVE are located to be symmetric with respect to Alice, as shown in scene 8. The experiments are performed in both static and dynamic scenarios. In dynamic scenarios, users are allowed to move around the testing area.

Metrics. We test SpotSound's performance in terms of transmission secrecy and reliability, which are evaluated with the Bit Error Rate (BER) of EVE and Bob, respectively. SpotSound aims to minimize the BER on Bob, meanwhile, maximize that on EVE. We also evaluate SpotSound's throughput in Sec. VI-G.

Baselines. We compare SpotSound with two baselines that are candidates for spatial selective transmission: i) *Multi-speaker beamforming.* We implemented 2 common layouts of a 6-speaker array, as shown in Fig. 17(a-1) & (a-2), to show beamforming's capability in spatial selective transmission. The spacing between speakers is 0.034 m due to the size restriction of speakers, and the carrier wave is set at 5 kHz, following the principle that the spacing should be lower than half of the wavelength [39]. OOK modulation is adopted for this baseline. ii) *Pre-equalization.* Dividing the modulated signal by the target channel (e.g., h_{AB}) in the frequency domain is also a potential way to achieve spatial selective transmission. Theoretically, it can enhance the signal quality at the legal receiver (Bob) while simultaneously degrading that for an

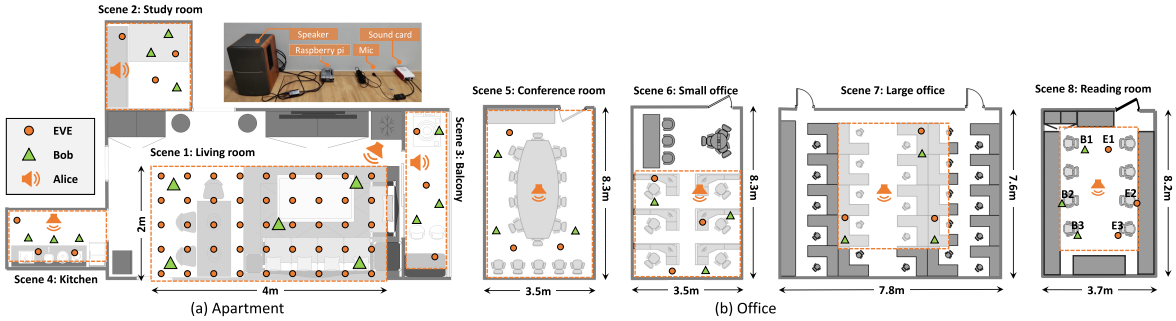


Fig. 15. Experiment settings.

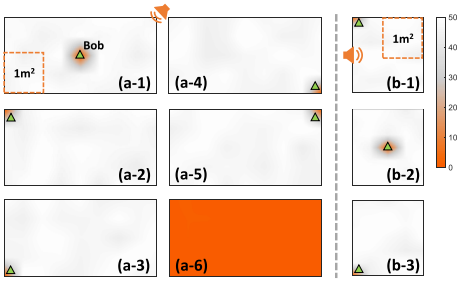


Fig. 16. Overall performance. (a-1)∼(a-5) show EVE's BER tested in scene 1 when Bob is placed at different locations; (a-6) shows Bob's BER tested in scene 1. (b-1)∼(b-3) show EVE's BER tested in scene 2.

illegal receiver (EVE) whose channel is different. To compare this method with SpotSound, we built an acoustic communication system that leverages BPSK modulation operating in 17-20 kHz, as illustrated in Fig. 18 (a).

B. Overall Performance

We start by examining SpotSound's ability to concentrate the signal on a target area. In the experiment, we deploy Alice in the top right corner of a living room and deploy Bob in five different locations in the room, as marked in Fig. 15. At each location, we let Alice transmit signals to Bob in a spatial-selective manner, during which we put EVE at 45 different locations in the environment, and observe whether EVE can decode the transmission between Alice and Bob.

Figures 16(a-1)∼(a-5) show BER of EVE obtained on different locations. A darker color in the heatmap means a lower BER. As can be seen, in most cases, **SpotSound can precisely focus the signal in a less than 0.5m**$\times$**0.5m spot** −− EVE's BER keeps higher than 40% as long as it is located 0.5m away from Bob. EVE cannot decode Alice's signal even in the case where it is located next to Alice.

However, we can also observe that SpotSound's signal control precision slightly decreases when Bob is located quite close to Alice (see Fig. 16(a-1)). This is because SpotSound's spatial-selective transmission needs the support of the multipath environment. When both EVE and Bob are located close to Alice, the difference between their channels becomes marginal, leading to the slight spread of the signal spot shown in Fig. 16(a-5). As a comparison, we also put Bob in the 45 positions and observe Bob's BER. Fig. 16(a-6) shows that since Alice can always concentrate its signal to Bob's location, Bob can always achieve a lower than 1% BER.

We further repeat the above experiment in a multipath-rich study room. In this experiment, we tested EVE's BER on 35 different locations in a 1.6 $\times$ 2.8 area. Figs. 16(b-1)∼(b-3) show the result obtained on three different locations of Bob. As expected, SpotSound can focus the signal on a $< 0.4 \times 0.4$ area in multipath-rich environments.

Comparison with baselines. We further test the performance of the two baseline methods in scene 1. For multi-speaker beamforming, we first measured beampatterns of the two arrays separately in both indoor and outdoor environments. During the measurement, a fixed signal was transmitted, and the received signal power was measured at intervals of 5 degrees, while the distance between transceivers was constant. The measured beam patterns are shown in Figs. 17(b-1) & (b-2). We can see that the main lobes of both patterns match the theoretical ideal values, but overall, the pattern measured in the outdoor environment conforms more closely to the ideal values, while that measured in the indoor environment shows higher side lobes, resulting in degraded directional transmission capability. This attributes to the strong multipath effects indoors. Then, in each scene, we steer the beamforming direction towards Bob's location. Figs. 17(c-1)∼(c-6) show EVE's BER. As can be seen, EVE can decode the signal within a large beam-like region for both layouts and in all directions.

We further test the performance of the pre-equalization method. Figs. 18(b-1) ∼(b-6) show the results. As can be seen, it fails to restrict the decodable range to a small spot as SpotSound does. EVE can still achieve a low BER within a broad region. This is because this method prioritizes optimizing the signal's reception quality at the target location rather than focusing the signal on a specific spot. Thus, the pre-equalized signals can also be decoded successfully at any location with a good link condition to the transmitter.

C. Impact of Training Dataset

We further evaluate how the size of the training set affects the performance of SpotSound. Specifically, we train three networks, respectively with i) 300,000 measured CIRs; ii) 300,000 measured CIRs + 300,000 synthetic CIRs; and iii) 300,000 measured CIRs + 900,000 synthetic CIRs. We repeat the experiments in Sec. VI-B with the network trained with different datasets. Fig. 19 shows the results. As shown in Fig. 19, after the data augmentation, there is a noticeable improvement in SpotSound's performance. Specifically, compared with the model without data augmentation, the two augmented models decrease Bob's BER by 19.6% and 42.4%, respectively. EVE's BER is always higher than 40% for all the three versions.

D. Impact of Model Complexity

We in this section compare the performance of the three models shown in Table II. Specifically, we repeat the

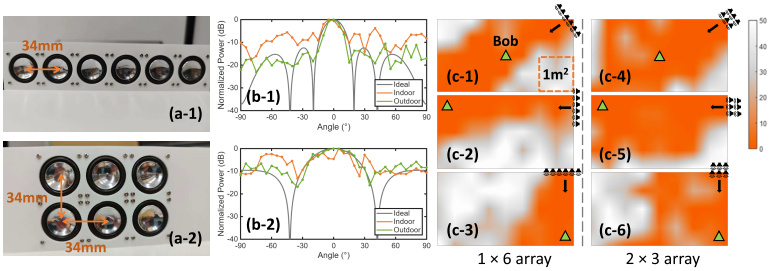


Fig. 17. Performance of multi-speaker beamforming. (a-1) shows the layouts of 1×6 array; (a-2) shows the layout of 2×3 array; (b-1) shows the beam pattern of 1×6 array; (b-2) shows the beam pattern of 2×3 array; (c-1)∼(c-6) show EVE's BER tested in scene 1 with two arrays separately.

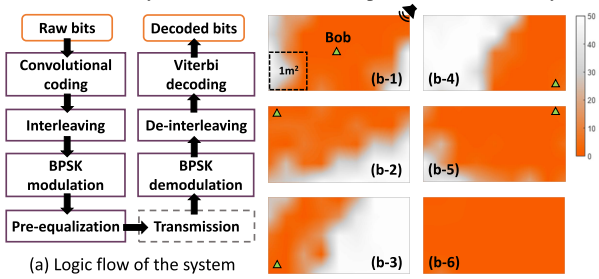


Fig. 18. Performance of pre-equalization method. (b-1)∼(b-5) show EVE's BER tested in scene 1; (a-6) shows Bob's BER tested in scene 1.

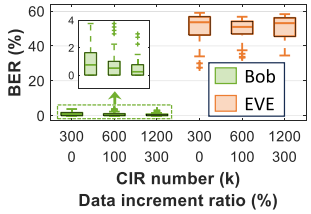


Fig. 19. Performance with models trained under different levels of data augmentation.

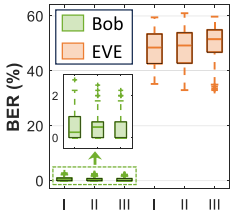


Fig. 20. Performance with different DL models.

experiments in Sec. VI-B with models II and III. Fig. 20 shows the performance of all three models. In the figure, we do not observe an obvious performance gap between Model I and the other two models, although its size is reduced by 74.7%, compared with Model III. The average BER on Bob achieved by Model I is only 0.17% higher than that achieved by Model III. So, we use Model I as the default model in SpotSound.

E. Ability in Generating Spots With Customized Size

We in this experiment evaluate SpotSound's ability to generate spots with different sizes. Fig. 21 (a) shows the setting of the experiment. We place Alice and Bob at the center and the corner of the room, respectively. We let Alice focus its signal to Bob with different spot sizes. The size of the signal spot is adjusted by fine-tuning the network with different security coefficients (SC), which are set at 0.4, 0.5, 0.7, and 0.9, separately. Under each setting of SC, Bob tries to decode Alice's signal on different test points (represented

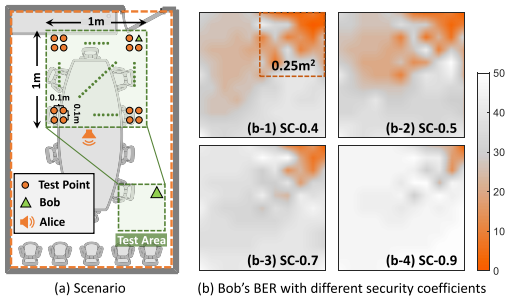


Fig. 21. Evaluation of Bob models under different security levels in training.

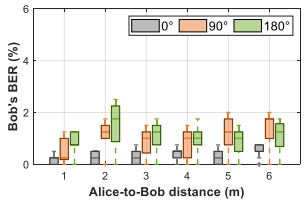


Fig. 22. Impact of Alice-to-Bob distances.

as the orange circles in Fig. 21 (a)), which are distributed on a $1m^2$ area with the spacing of 10cm.

The results are shown in Fig. 21 (b). As can be seen, the generated signal spot shrinks as the security coefficient increases in general, which verifies SpotSound's ability to generate spots with customized sizes. Another observation is that, for each generated spot, its shape cannot be precisely controlled due to the complicated multipath effects indoors. Moreover, we can see that when the SC is smaller than 0.5, the size of the spot changes only marginally with the decreased SC. On the other hand, when the SC reaches 0.9, the BER will exceed 30% even when the receiver is only 20cm away from the target position. The results in Figs. 21(b-1)∼(b-4) show that the smallest and largest spot that SpotSound can generate are $0.04m^2$ and $0.5m^2$, respectively.

F. Impact of Practical Factors

1) Distance: Then we evaluate the impact of signal propagation distance. We first evaluate Bob's performance under different Alice-to-Bob distances. Specifically, we vary the distance from 1m to 6m in the living room. On each distance, we test Bob's BER under three different angles, i.e., 0°, 90°, and 180°. The results are shown in Fig. 22. We can see that the impact of distance is not as obvious as that of orientation. On a certain orientation, we do not observe an apparent change in Bob's BER under different distances.

We further evaluate EVE's performance under different Alice-to-EVE distances and different Bob-to-EVE distances. Fig. 23 shows the results obtained under different orientations

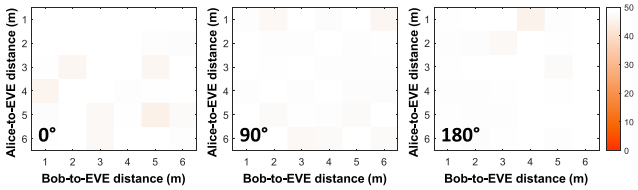


Fig. 23. EVE's BER under different Alice-to-EVE distances and Bob-to-EVE distances.

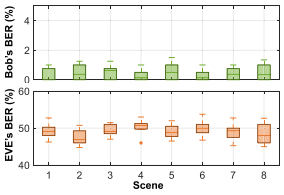
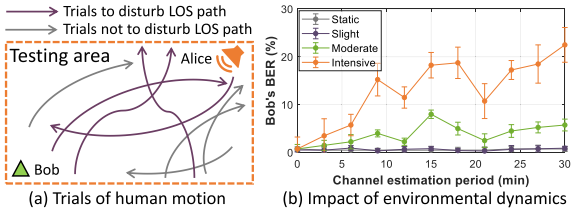


Fig. 24. EVE's BER under different environmental clutter levels.



(a) Trials of human motion (b) Impact of environmental dynamics

Fig. 25. Performance under different environmental dynamic levels.

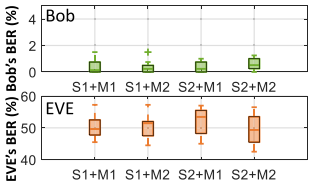


Fig. 26. Performance with different devices.

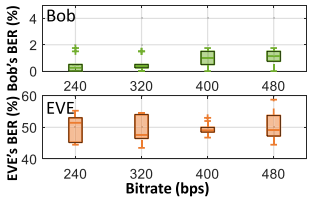


Fig. 27. Performance under different bitrates.

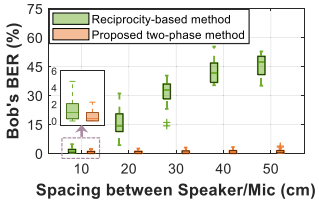


Fig. 28. Impact of different intra-device distances.

of Alice. We can see that EVE's BER keeps higher than 40% in all cases.

2) Environments: To evaluate SpotSound's performance in different environments, we perform experiments in 8 testing sites shown in Fig. 15. In each experiment, we fix the location of Alice and put Bob at 3 different locations. When Alice transmits signals to Bob, EVE tries to decode the signal at 3 randomly selected locations. Fig. 24 shows the BER obtained by Bob and EVE in different environments. As expected, SpotSound achieves higher secrecy while lower reliability in multipath-denser environments. For example, on average, the EVE's BER obtained in a cluttered kitchen is 2.5% higher than that obtained in the study room. Even in scene 8, where EVE is always placed in a position symmetrical to Bob with respect to Alice, EVE's BER remains higher than 40%. This is because the relative position between the transmitter and receiver only determines the LOS path, whereas the multipath channel is highly affected by the complex reverberations generated from NLOS paths in real-world environments. It is extremely rare for both the reflectors in the environment and the positions of Bob and EVE to be symmetric with respect to Alice.

We further test SpotSound's performance in dynamic scenarios, where users are allowed to move around the testing areas, and the locations of the furniture can also change. We tested at four different dynamic levels in scene 1: (a) Static, where there is no human motion and the environment is entirely static; (b) Slight dynamics, where there is simple human motion that does not disturb the LOS path between transceivers. The trials of human motion at this level are shown in Fig. 25 (a); (c) Moderate dynamics, where small reflectors (e.g., chairs and clutter on the table) between the transceivers are moved a noticeable distance (>10 cm) every minute, while the human motion mentioned in (b) exists simultaneously; (d) Intensive dynamics, where large reflectors (e.g., large tables and cabinets) between the transceivers are moved in the same way as in (c), and the human motion tends to interfere with the LOS path. The trials of human motion at this level are also shown in Fig. 25 (a). At each level, we measure the BER under different channel estimation periods, and perform 10 measurements for each period. The results are shown in Fig. 25 (b). As can be seen, SpotSound can still achieve a lower than 5% BER in moderate dynamics even when the channel estimation period increases to 25 min. This is because, compared to the large reflectors in the environment (e.g., the walls and the large furniture), the moving people and the chairs have only a marginal effect on the channel. Such small changes in the CIR measurement are considered in the training process, as we have discussed in Sec. IV-E. However, when working in an intensively dynamic environment, SpotSound has to re-estimate the channel every 5 minutes or more frequently to keep its BER lower than 5%. The above results show that we should provide an adaptive channel estimation method which can increase the channel estimation frequency automatically in dynamic environments. We leave this to our future work.

3) Channel Asymmetry: In this section, we explore how the channel asymmetry affects the performance of channel estimation methods and, consequently, the overall system performance. Specifically, we repeat the experiment in Sec. VI-B five times. For each time, we vary the spacing between the microphone and speaker on one device, which brings various degrees of channel asymmetry. Under each setting, we evaluate Bob's BER with two different channel estimation methods, including i) the reciprocity-based pre-transmission method; and ii) the proposed two-phase pre-transmission method. Fig. 28 shows the experimental results. As can be seen, with the reciprocity-based method, Bob's BER keeps rising and becomes unstable as the speaker-microphone spacing increases. Bob's average BER is 45.6% when the spacing reaches 50cm. In contrast, the proposed two-phase channel estimation method can largely compensate for the impact of channel asymmetry. With the proposed method, Bob's average BER is 0.97% even when the spacing is 50cm.

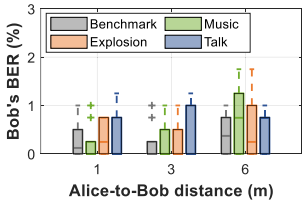


Fig. 29. Performance under different noise.

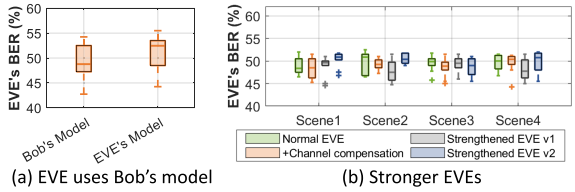


Fig. 30. Performance in resisting different EVEs.

4) Device Diversity: In this section, we evaluate SpotSound's transmission reliability when Alice and Bob use different types of microphones and speakers. We tested two types of microphones (TakStar TCM-400 (M1) and LATZZ A3 (M2)) and two types of speakers (EDIFIER R1200TII (S1) and Swan M200MKIII (S2)). The performance achieved with different pairs of speakers and microphones is shown in Fig. 26. As can be seen, the type of the device does not have an apparent impact on SpotSound's performance.

5) Noise: We further evaluate SpotSound's performance under different types of environmental noises. We consider three types of noises: rock music, explosions, and human talk. For each type of noise, we play it 1m away from Bob with a 50% volume, during which we measure Bob's BER under different Alice-to-Bob distances. We also test the performance in a quiet environment as the benchmark. As shown in Fig. 29, Bob's average BER is 0.38%, which shows SpotSound's resistance to environmental noises.

G. Throughput

We in this section evaluate whether SpotSound can achieve a bitrate higher than 320 bps by embedding more bits in one signal symbol. To do so, we implement another three versions of SpotSound, whose bitrates are 240 bps, 400 bps, and 480 bps, respectively. Note that we just change the size of the model input (i.e., X). The parameter size stays the same as the original version.

Figure 27 shows Bob's BER with different bitrates. As can be seen, Bob can still achieve a lower than 2% BER even at the bitrate of 480bps. Fig. 27 also shows EVE's BER, which does not show apparent variation under different bitrates.

H. Resistant to Stronger EVEs

We in this section evaluate SpotSound's ability to resist different EVEs. We first consider a special case where the eavesdropper directly uses Bob's model to decode Alice's signal at a non-target location. Fig. 30 compares the eavesdropper's performance when using EVE's and Bob's models. As can be seen, the two models achieve very similar BER.

Then we evaluate SpotSound's ability to resist stronger EVEs. We consider two cases. In the first case, EVE can communicate with Alice as a malicious legitimate receiver, so it can measure and compensate its channel h_E. This is equivalent to directly obtaining the source signal from Alice. In the second case, we consider the strengthened EVEs as mentioned in Sec. VI-A.

We compare the BER obtained by the normal EVE and the three stronger EVEs in different environments as shown in Fig. 30 (b). As can be seen, the four types of EVEs achieve very similar decoding performance. That is to say, one can hardly improve EVE's decoding performance by either increasing its channel quality or increasing its decoding capability. This is because Alice can modulate the signal in a channel-selective manner. The information embedded in the signal can pass through only the target channel while is largely destroyed on the non-target channels (e.g., EVE's channel). So, the EVE cannot detect the signal as long as its channel is different from the target channel. We will show in Sec. VII that the signal received on a non-target channel resembles a random noise, which can even pass the randomness test.

TABLE V
PROCESSING TIME FOR A 60-BIT PACKET

RPi (Alice)	RPi (Bob)	SD870 (Alice)	SD870 (Bob)
20.49 ms	7.5 ms	3.05 ms	1.04 ms

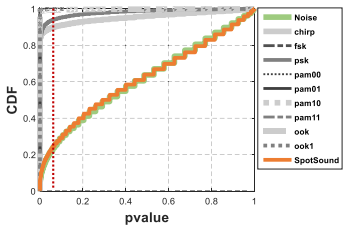


Fig. 31. Evaluate the randomness with KS test.

I. Time Overhead

We have also measured the time overhead of SpotSound on Raspberry Pi (RPi) and Snapdragon 870 (SD870). Table V shows the latency required in generating/decoding a 60-bit packet. As can be seen, the latency for packet generation and decoding is much shorter than the packet length.

VII. Security Analysis

A. Randomness of EVE's Signal

We first use the Kolmogorov-Smirnov test to evaluate the generated signal's goodness-of-fit with respect to random noise. Fig. 31 shows that the signal generated by SpotSound passes the test with P-value higher than 0.05 and achieves a similar level of fitness as the real ambient noise. In comparison, conventional modulation signals show a very low level of fitness, where only 1%∼10% cases can pass the test.

B. Resistant to Brute Force Attack

In a brute-force attack, EVE tries to guess Bob's channel's CIR and reconstruct Bob's signal. Experimental results show that EVE can successfully decode Alice's signal if its channel CIR shows a higher than 0.7 similarity with Bob's CIR. However, obtaining such a channel is computationally intractable. Specifically, the measured CIR used in SpotSound contains 3000 samples, so there will be 2^{3000} possible CIRs even when the depth of each sample is 2. Fig. 32 shows the possibility that EVE can obtain a satisfying CIR with one guess under

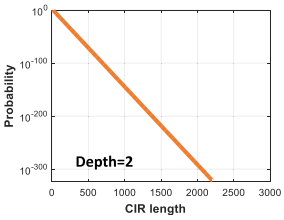


Fig. 32. Computation cost for brute force attack.

different CIR lengths L. As shown, the possibility decreases to 10^{-292} when $L > 2000$. So, it is almost impossible for EVE to obtain Bob's CIR and reconstruct its signal.

VIII. Related Work

We in this section discuss related work that we have not touched in the previous sections.

Static signal reflectors. Besides leveraging antenna arrays or meta-surfaces, another way to control the signal's propagation behavior is to use static reflectors [40], [41]. By adding 3D-printed static reflectors around the sender or receiver, one can shape the outgoing or incoming signals. However, these methods lack reconfigurability and cannot adapt to changing environmental conditions.

Communication with side channel. Another way to achieve covert communication is to build a side channel leveraging physical characteristics like vibration [42], heat [43], screen light [44], [45], electromagnetic [46], [47], etc. These methods, however, require either physical contact or specialized equipment, which makes them unsuitable for the IoT scenario. Besides, they assume that the attackers do not know the channel that the transceivers use. While none of the above is required in SpotSound.

Acoustic communication. Acoustic communication has been studied in [48], [49]. With a variety of modulation methods, they build a reliable acoustic channel for long-range, short-range, and underwater communication. Different from those methods, SpotSound aims to build a spatial selective channel with the acoustic signal.

IX. Discussions and Limitations

In this section, we discuss important design choices of SpotSound and limitations worth further exploration:

Selection of acoustic communication We utilize acoustic rather than RF modalities (e.g., Wi-Fi) because: i) SpotSound requires sample-level manipulation of the signal, while existing commercial RF modules support only symbol-level manipulation. ii) SpotSound relies on fine-grained estimation of the CIR of the legitimate channel, which means fine-grained estimation of the arrival time of the multipath signals. Due to the slow propagation speed, acoustic signals provide better CIR estimation granularity than RF signals under the same bandwidth. Note that SpotSound can be theoretically extended to RF communication with SDR devices (e.g., USRP). The encoding/decoding architecture can be directly extended to RF communication. We can adjust the band-pass filter's frequency band during training to adapt to RF signals.

Selection of frequency band We modulate the signal on the 17-20 kHz band because: i) acoustic signals below 17 kHz are easily audible to the human ear; and ii) the low-cost audio components in IoT devices often exhibit significant attenuation in frequency bands approaching the Nyquist frequency (e.g., 24 kHz for a 48 kHz sampling rate) [50], so we empirically set the upper bound as 20 kHz to avoid the attenuation on high-frequency band. Adaptively modifying the working band based on current channel conditions might further improve SpotSound's performance, which we leave to our future work.

Limited data rate. The data rate of our current prototype is still very low, *i.e.*, below 480bps, hence it is only applicable to those low data-rate applications that communicate with gateways intermittently. Accordingly, improving the data rate of SpotSound becomes an immediate next step.

Reliance on multipath effect. One assumption underlying SpotSound is that the signal transmitted to Bob and EVE will propagate along different multipath channels. Accordingly, SpotSound achieves inferior performance in low or moderate multi-path environments. To address this issue, one possible direction is to explore environment-independent channel effects (e.g., device distortion) for information hidden, and we leave it for our future work.

X. Conclusion

Eavesdropping attacks have compromised the security of billions of IoT devices. SpotSound tries to build spatial-selective channels among those devices. By controlling the signal's fine-grained shape, SpotSound can embed information into a random signal in a spatial-selective manner, making the modulated signal decodable only when received at a target location. The evaluation of our current implementation of SpotSound demonstrates that it can concentrate the signal on spots with customized sizes, ranging from $0.04m^2$ to $0.5m^2$. Meanwhile, it achieves a data rate of up to 480 bps.

References

[1] Q. Wang et al., "Walls have ears! Opportunistically communicating secret messages over the wiretap channel: From theory to practice," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 376–387.

[2] A. Chaman, J. Wang, J. Sun, H. Hassanieh, and R. Roy Choudhury, "Ghostbuster: Detecting the presence of hidden eavesdroppers," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 337–351.

[3] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J.-M. Merolla, "Applications of LDPC codes to the wiretap channel," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2933–2945, Aug. 2007.

[4] X. Na, X. Guo, Y. He, and R. Xi, "Wi-attack: Cross-technology impersonation attack against iBeacon services," in *Proc. 18th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jul. 2021, pp. 1–9.

[5] L. Xu, X. Zheng, X. Li, Y. Zhang, L. Liu, and H. Ma, "WiCAM: Imperceptible adversarial attack on deep learning based WiFi sensing," in *Proc. 19th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Sep. 2022, pp. 10–18.

[6] X. Lei, G.-H. Tu, C.-Y. Li, T. Xie, and M. Zhang, "SecWIR: Securing smart home IoT communications via Wi-Fi routers with embedded intelligence," in *Proc. 18th Int. Conf. Mobile Syst., Appl., Services*, Jun. 2020, pp. 260–272.

[7] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, "Information exposure from consumer IoT devices," in *Proc. Internet Meas. Conf.*, Oct. 2019, pp. 267–279.

[8] Y. Qiao, O. Zhang, W. Zhou, K. Srinivasan, and A. Arora, "PhyCloak: Obfuscating sensing from communication signals," in *Proc. NSDI*, 2016, pp. 685–699.

[9] Y. Zhu et al., "Et tu alexa? When commodity WiFi devices turn into adversarial motion sensors," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–15.

[10] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: Enabling robust 60 GHz links under blockage," in *Proc. NSDI*, 2016, pp. 193–206.

[11] S. Madani, S. Jog, J. O. Lacruz, J. Widmer, and H. Hassanieh, "Practical null steering in millimeter wave networks," in *Proc. NSDI*, 2021, pp. 903–921.

[12] V. Arun and H. Balakrishnan, "RFocus: Beamforming using thousands of passive antennas," in *Proc. NSDI*, 2020, pp. 1047–1062.

[13] Z. Li et al., "Towards programming the radio environment with large arrays of inexpensive antennas," in *Proc. NSDI*, 2019, pp. 285–300.
[14] C. Feng et al., "RFlens: Metasurface-enabled beamforming for IoT communication and sensing," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2021, pp. 587–600.
[15] M. Dunna, C. Zhang, D. Sievenpiper, and D. Bharadia, "ScatterMIMO: Enabling virtual MIMO with smart surfaces," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, Apr. 2020, pp. 1–14.
[16] J. Nolan, K. Qian, and X. Zhang, "RoS: Passive smart surface for roadside-to-vehicle communication," in *Proc. ACM SIGCOMM Conf.*, Aug. 2021, pp. 165–178.
[17] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. ICML*, 2016, pp. 1060–1069.
[18] D. D. Carlo, P. Tandeitnik, C. Foy, N. Bertin, A. Deleforge, and S. Gannot, "DEchorate: A calibrated room impulse response dataset for echo-aware signal processing," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, pp. 1–15, Dec. 2021.
[19] K. Qian et al., "AIRCODE: Hidden screen-camera communication on an invisible and inaudible dual channel," in *Proc. NSDI*, 2021, pp. 457–470.
[20] Q. Zhang, D. Wang, R. Zhao, Y. Yu, and J. Shen, "Sensing to hear: Speech enhancement for mobile devices using acoustic signals," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 5, no. 3, pp. 1–30, Sep. 2021.
[21] K. Sun and X. Zhang, "UltraSE: Single-channel speech enhancement using ultrasound," in *Proc. 27th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2021, pp. 160–173.
[22] F. Scholkmann, "Exposure to high-frequency sound and ultrasound in public places: Examples from Zurich, Switzerland," *Acoustics*, vol. 1, no. 4, pp. 816–824, Oct. 2019.
[23] T. Chen, L. Shangguan, Z. Li, and K. Jamieson, "The design and implementation of a steganographic communication system over in-band acoustical channels," *ACM Trans. Sensor Netw.*, vol. 19, no. 4, pp. 1–25, Nov. 2023.
[24] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proc. NIPS*, vol. 28, 2015, pp. 1135–1143.
[25] S. Gur, S. Benaim, and L. Wolf, "Hierarchical patch VAE-GAN: Generating diverse videos from a single sample," in *Proc. NIPS*, vol. 33, 2020, pp. 16761–16772.
[26] M. Akbari and J. Liang, "Semi-recurrent CNN-based vae-gan for sequential data generation," in *Proc. ICASSP*, Apr. 2018, pp. 2321–2325.
[27] X. Di and V. M. Patel, "Face synthesis from visual attributes via sketch using conditional VAEs and GANs," 2017, *arXiv:1801.00077*.
[28] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[29] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[30] T. Chen, L. Shangguan, Z. Li, and K. Jamieson, "Metamorph: Injecting inaudible commands into over-the-air voice controlled systems," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–17.
[31] Y. Wang, X. Zheng, L. Liu, and H. Ma, "PolarTracker: Attitude-aware channel access for floating low power wide area networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1807–1821, Aug. 2022.
[32] F. Yu, X. Zheng, L. Liu, and H. Ma, "LoRadar: An efficient LoRa channel occupancy acquirer based on cross-channel scanning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2022, pp. 540–549.
[33] T. Fan et al., "Towards spatial selection transmission for low-end IoT devices with SpotSound," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, Jul. 2023, pp. 1–14.
[34] A. Mahmood and S. Khan, "Exploiting transitivity of correlation for fast template matching," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2190–2200, Aug. 2010.
[35] P. Xie, J. Feng, Z. Cao, and J. Wang, "GeneWave: Fast authentication and key agreement on commodity mobile devices," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2017, pp. 1–10.
[36] Y. Ren et al., "Proximity-echo: Secure two factor authentication using active sound sensing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2021, pp. 1–10.
[37] J. Zhang, X. Guo, H. Jiang, X. Zheng, and Y. He, "Link quality estimation of cross-technology communication: The case with physical-level emulation," *ACM Trans. Sensor Netw.*, vol. 18, no. 1, pp. 1–20, Feb. 2022.
[38] Y. He, X. Guo, J. Zhang, and H. Jiang, "WIDE: Physical-level CTC via digital emulation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1567–1579, Aug. 2021.
[39] Y. Li et al., "MuDiS: An audio-independent, wide-angle, and leak-free multi-directional speaker," in *Proc. 30th Annu. Int. Conf. Mobile Comput. Netw.*, May 2024, pp. 263–278.
[40] J. Chan, C. Zheng, and X. Zhou, "3D printing your wireless coverage," in *Proc. 2nd Int. Workshop Hot Topics Wireless*, Sep. 2015, pp. 1–5.
[41] X. Xiong et al., "Customizing indoor wireless coverage via 3D-fabricated reflectors," in *Proc. 4th ACM Int. Conf. Syst. Energy-Efficient Built Environ.*, Nov. 2017, pp. 1–10.
[42] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: Communicating through physical vibration," in *Proc. Usenix NSDI*, 2015, pp. 265–278.
[43] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici, "BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations," in *Proc. IEEE 28th Comput. Secur. Found. Symp.*, Jul. 2015, pp. 276–289.
[44] K. Zhang et al., "ChromaCode: A fully imperceptible screen-camera communication system," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2018, pp. 575–590.
[45] A. Wang, C. Peng, O. Zhang, G. Shen, and B. Zeng, "InFrame: Multiflexing full-frame visible communication channel for humans and devices," in *Proc. 13th ACM Workshop Hot Topics Netw.*, Oct. 2014, pp. 1–7.
[46] Z. Yang, Q. Huang, and Q. Zhang, "NICScatter: Backscatter as a covert channel in mobile devices," in *Proc. ACM MobiCom*, 2017, pp. 356–367.
[47] M. Gao et al., "Deaf-aid: Mobile IoT communication exploiting stealthy speaker-to-gyroscope channel," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2020, pp. 1–13.
[48] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su, "Messages behind the sound: Real-time hidden acoustic signal capture with smartphones," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 29–41.
[49] Y. Bai, J. Liu, L. Lu, Y. Yang, Y. Chen, and J. Yu, "BatComm: Enabling inaudible acoustic communication with high-throughput for mobile devices," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, Nov. 2020, pp. 205–217.
[50] *TLV320ADC3100 Low-Power Stereo Adc for Voice-Activated Systems and Portable Audio*, Texas Instruments, Dallas, TX, USA, 2018.