Towards Spatial Selection Transmission for Low-end IoT devices with SpotSound

Tingchao Fan*
Shanghai Jiao Tong University
kylin_f@sjtu.edu.cn

Huangwei Wu*
Shanghai Jiao Tong University
wuhuangwei@sjtu.edu.cn

Meng Jin†
Shanghai Jiao Tong University
jinm@sjtu.edu.cn

Tao Chen
University of Pittsburgh
tac194@pitt.edu

Longfei Shangguan
University of Pittsburgh
Longfei@pitt.edu

Xinbing Wang
Shanghai Jiao Tong University
xwang8@sjtu.edu.cn

Chenghu Zhou
Chinese Academy of Sciences
zhouch@lreis.ac.cn

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 this signal propagates along a certain multipath channel. Since the multipath channel decorrelates rapidly over the distance between different 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 achieves a $0.25m^2$ location isolation.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

IoT Security, Acoustic, DL-based Communication

1 INTRODUCTION

Today most wireless transceivers are omnidirectional sources of electromagnetic waves. Since wireless channel is a broadcast medium, wireless transmissions suffer more concerns on security and privacy [5, 16, 24, 27, 32]. 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 [14], their data packets are vulnerable to eavesdropping attack [20]. 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) [18, 37].

To address this security concern, a potential solution is leveraging beamforming [15, 23], 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 tradeoff 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

*Co-primary authors
†Corresponding author

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3592496

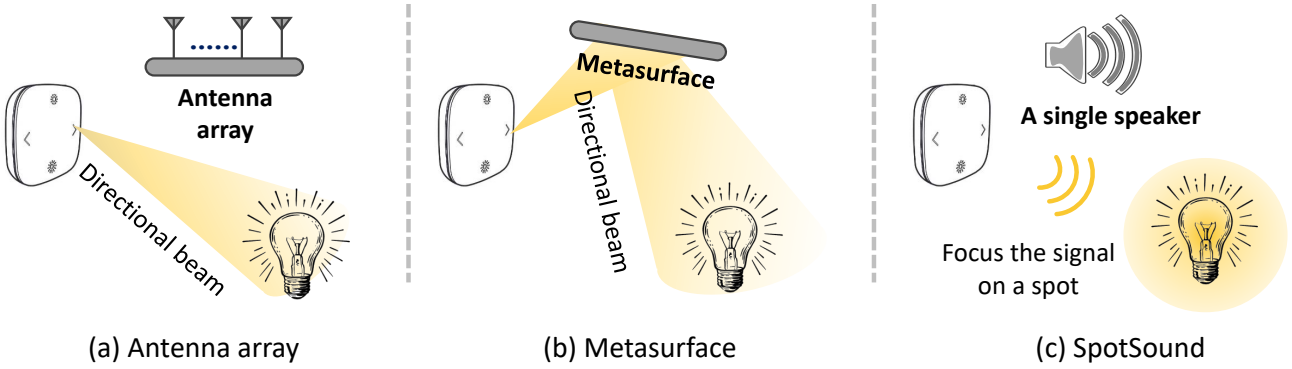


Figure 1: Spatial selective transmission with beamforming and SpotSound.

environment through meta-surface [1, 2, 8, 9, 17], 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 need to transmit and a measurement of the target channel, based on which it generates modulated signals. To 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 channels. 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 focus its signal in a less than $0.25m^2$ spot.

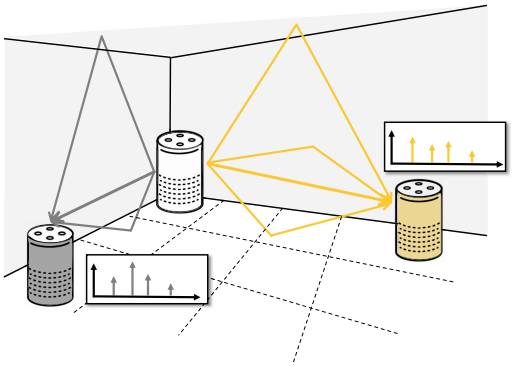


Figure 2: Multipath reflection.

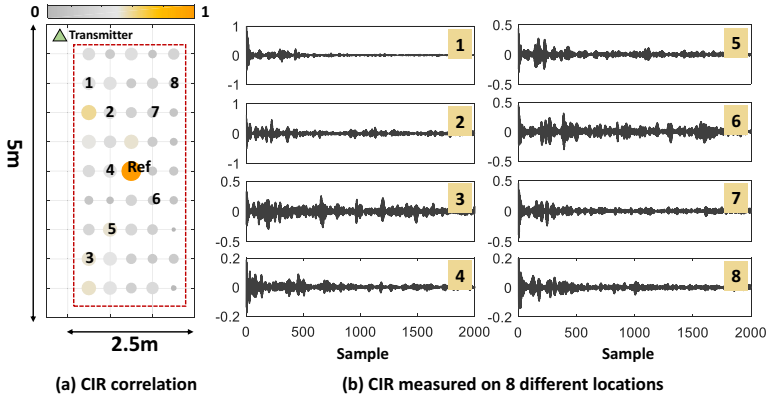


Figure 3: CIR measured in different environments.

2 THREAT MODEL

In our target scenario, the attacker is a malicious receiver (EVE) which 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's any advantages over EVE on 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).

3 BACKGROUND AND INTUITION

3.1 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:

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

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 be also written as the temporal convolution of transmitted signal $s(t)$ and the channel impulse response (CIR) $h(t)$:

$$r(t) = h(t) \otimes s(t) + w(t) \quad (2)$$

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.

3.2 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 propagates 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 uses 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

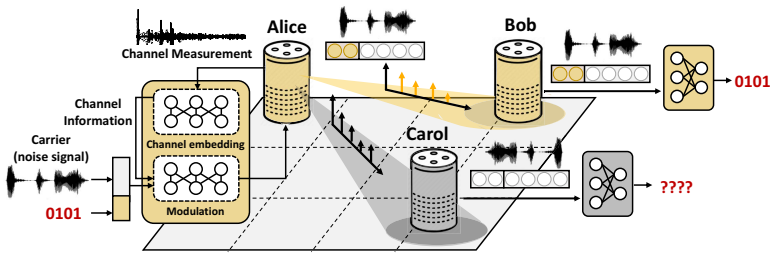


Figure 4: Intuition in using environment as beamformer.

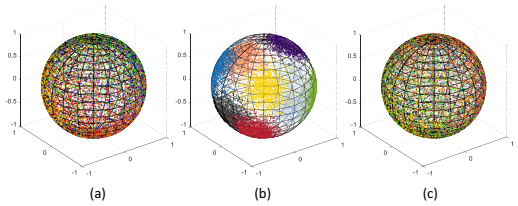


Figure 5: Feature map of modulation signals: (a) signal generated by Alice; (b) signal received by Bob; (c) signal received by Carol.

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 tradeoff 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 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.

3.3 Physical channel as a beamformer

We in this paper propose a new approach which 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, compare with its propagation behavior.
- Since the channel effect decorrelates rapidly to 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 matches only the target channel.

We deal with this complexity with deep learning (DL). Specifically, we borrow the idea from cGAN (conditional GAN) [19], 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 simultaneously, i.e., decodable on the target channel and undetectable on other channels, we use two discriminators, which act as the target receiver (denote 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 objects 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 backpropagation 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 shows 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 form 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 channel measurement, 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 CIR measurements (which may 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 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. 7 and 6 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.

4 SPOTSOUND'S DL MODEL

4.1 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 θ_{A}, θ_{B} and θ_{C}. In our design, Bob and Carol share the same structure but have different parameters.

Alice takes as inputs the data information X, CIR of the target channel $\hat{h}_B$, and a random 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 $X = \{x_1, ..., x_M\}$ and outputs the modulated signal $S = \{s_1, ..., s_M\}$. 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) [4].

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:

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

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

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

Similarly, we obtain the loss function for Bob as:

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

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

$$L(\theta_A, \theta_B) = L_B(\theta_A, \theta_B) - L_C(\theta_A, O_C(\theta_A)) \quad (6)$$

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

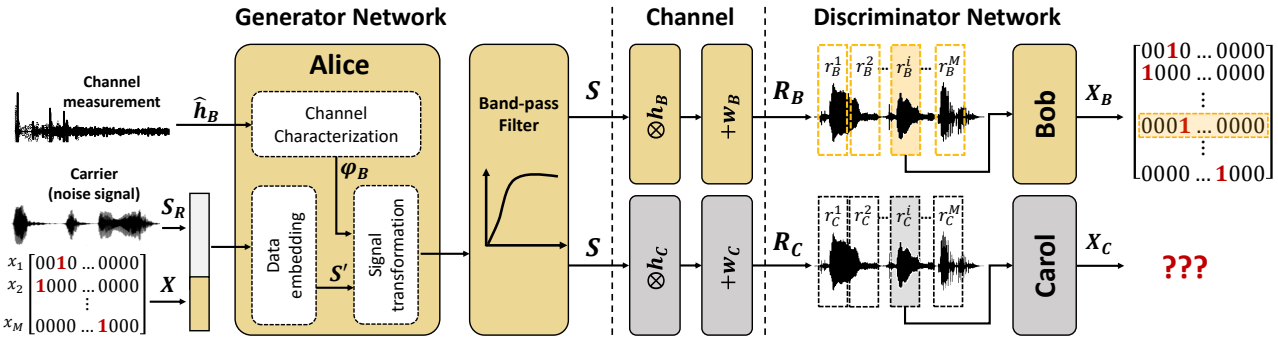


Figure 6: Architecture of SpotSound's DL model.

4.2 Design details

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

4.2.1 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 [25]. This makes the final output signal S perfectly and exclusively fits 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 module 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.

4.2.2 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).

4.2.3 Frequency selection. To make the modulated signal inaudible to human ears, the transmitter should transmit the signal only on the inaudible bandwidth (i.e., 17∼20KHz). 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 17∼ 20KHz band can finally arrive at the receiver side. This forces the transmitter to modulate the signal only on these bands. 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).

4.3 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 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 *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

Table 1: Memory and computation cost of the models.

Model	Description	Memory	Computation	Energy
I	LRA+DSC+ pruning	188,467	22.16 MFLOPS	$1.06\times10^{-4}J$
II	LRA+DSC	549,795	26.71 MFLOPS	$1.29\times10^{-4}J$
III	DSC	1,234,743	100.84 MFLOPS	$4.89\times10^{-4}J$

Table 2: 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

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 1 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 2 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. 6.3. The results show that Model I incurs only a slight performance degradation compared with the other two models.

We further analyse 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 need to be loaded from DRAM, which can be expressed as follows:

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

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 [12] 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 [12], we estimate the energy consumption of our three models in generating one 60-bit packet. The results are shown in Table 1. As can be seen, the energy consumption of SpotSound keeps lower than 1mJ, which is affordable on most IoT devices.

4.4 Training strategy

Dataset. To ensure that the model could learn the latent mapping between the channel information and the modulated signal, we train the model with CIR measurements collected in different environments.

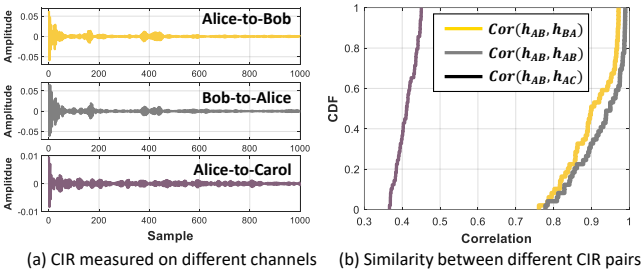
- *Diversity.* Our CIR measurements are collected from 12 different indoor environments on a university campus, including seminar rooms, cafeteria, offices, and dorms.
- *Scale.* In each environment, we put a pair of transceivers (denoted as A and B) on 40~90 pairs of different locations and for each pair of locations we collect more than 600 CIR measurements, where 300 measurements for the A-to-B channel, and the other 300 measurements for the B-to-A channel (we will explain in Sec. 5.1.1 that why we require the CIR on both directions). We in total collected more than **430,000 CIR measurements** for training.

Training. In practice, Alice cannot perfectly estimate the target channel. Thus, we should involve this error in the training process so that Alice and Bob can learn to tolerate this error. To achieve this, we find out CIRs measured on the same location while at different times and directions, and use them respectively as Alice's input and Bob's channel layer. Due to the errors in channel estimation, those CIRs have only 0.8~0.9 similarity, which forces Alice and Bob to handle the imperfect channel estimation. We train the model with different combinations of h_B and h_C. The training examples also include the cases where Carol can directly obtain the source signal (i.e., $h_C = 1$).

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.

5 SPOTSOUND IN PRACTICE

With only the core DL model introduced in Sec. 4, SpotSound cannot work in practice because of two unsolved issues: i) pre-transmission channel estimation; and ii) package detection. This section presents the design enhancement for SpotSound, in order to solve these two issues.



(a) CIR measured on different channels (b) Similarity between different CIR pairs

Figure 7: Channel reciprocity.

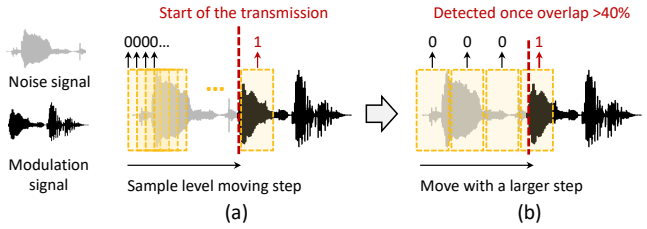


Figure 8: Packet detection. (a) moving the window sample by sample; (b) moving the window with a long step

5.1 Channel estimation

To estimate the channel (i.e., the CIR), a common practice [7, 29, 34] is to send a known training signal s_T. Then, CIR can be derived from the deconvolution of received signal s'_T and 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:

$$\hat{h} = \Im^{-1}\{S_T^* S'_T\} \tag{8}$$

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 of this section.

5.1.1 Achieving pre-transmission channel estimation. We perform pre-transmission channel estimation with 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. To validate channel reciprocity on the acoustic channel, we perform an experiment with three devices: Alice, Bob, and Carol. In the experiment, we keep the Alice-to-Bob distance at 2m and the Carol-to-Bob distance at 1m. We measure the Alice-to-Bob channel h_{AB}, the Bob-to-Alice channel h_{BA}, and Alice-to-Carol channel h_{AC}, and show the results in Fig. 7 (a). As can be seen, h_{AB} and h_{BA} are very similar while are significantly different from h_{AC}.

We further repeat the above experiments in 12 different environments, and in each environment we collect more than 100 CIRs for each channel. We calculate the pearson correlation coefficient between each pair of CIRs, including that between two reciprocal channels ($Cor(h_{AB}, h_{BA})$), the same channel measured at different times ($Cor(h_{AB}, h_{AB})$), and two different channels ($Cor(h_{AB}, h_{AC})$). The results in Fig. 7(b) show that the $Cor(h_{AB}, h_{BA})$ is only 3% lower than that of $Cor(h_{AB}, h_{AB})$, while is 2× higher than that of $Cor(h_{AB}, h_{AC})$.

Based on this phenomenon, Alice can estimate the Alice-to-Bob channel based on Bob's responses. Note that since such a reversed channel estimation brings additional errors in CIR measurement, we involve this error in the training process by using reciprocal CIR measures on Alice's input layer and Bob's channel layers, as mentioned in Sec. 4.2.

5.1.2 Making channel estimation imperceptible. To make the channel estimation imperceptible to an eavesdropper, we use random noise as pilot signal. In our design, each two transceivers Alice and Bob maintain two white noise signals, denoted as $s_T^{(req)}$ and $s_T^{(ack)}$. These two noise sequence is known only by Alice and Bob. When Alice wants to predict the Alice-to-Bob channel, it first broadcasts a request $s_T^{(req)}$. Bob detects this request by cross-correlation between the received signal and the template $s_T^{(req)}$. After detecting this request, Bob responses an ACK $s_T^{(ack)}$ to Alice, based on which Alice can measure the Alice-to-Bob channel. Since both $s_T^{(req)}$ and $s_T^{(ack)}$ are noise-like signal, the eavesdropper can hardly distinguish them from the noise and thus can hardly detect the communication between Alice and Bob.

The two noise sequences $s_T^{(req)}$ and $s_T^{(ack)}$ shared between two devices can be initialized either manually or leveraging the existing automatic pairing methods for IoT devices [21, 30]. To prevent the EVEs from learning the two sequences by sampling the channel over a long time, Alice and Bob also update the two sequences periodically. Specifically, they can use the old sequences as seeds to generate new sequences with a hash function. The hash function can be a public function which is known by EVE.

Note that we do not need to perform the above channel estimation process for every transmission. It can be triggered periodically. Our experiments in 6.4 tells that a low-frequent channel re-estimation (e.g., every 10 min) is sufficient for SpotSound to achieve a decent performance in highly dynamic environment.

5.2 Package detection

In a practical communication system, Alice has to transmit a known preamble signal at the beginning of each packet [13, 35], so that Bob can localize the start of a packet and then slices 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 the signal 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 NN module for packet detection, which can distinguish the modulated signals from noises. 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 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 noises 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. 8 (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. 8 (b). In this way, the receiver can locate the start of the packet in a coarse-to-fine manner. Specifically, we first move the window with a longer step, as shown in Fig. 8 (b). Once a symbol is detected, it further moves the window with a finer granularity (e.g., sample by sample) to finally locate the start of the packet with 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.

6 EVALUATION

6.1 Experimental Methodology

Implementation. We implement SpotSound on Raspberry pi 4 Model B with the help of TensorFlow lite. The sound is played with the EDIFIER R1200TII speaker and the TakStar TCM-400 microphone which are both connected to the UGREEN external sound card. The frequency range and the sampling rate of the sound are set at 17-20 kHz and 48 kHz respectively.

we select Model I in Table 1 as the default implementation and compare all the three versions of SpotSound model

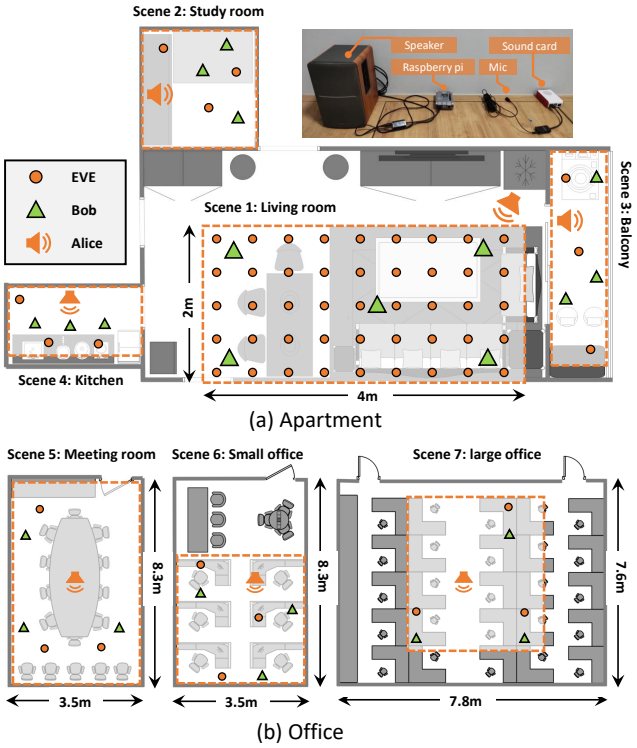


Figure 9: Experiment settings.

in Sec. 6.3. We have also trained a normal EVE and two strengthened EVEs. We use the normal one which shares the same structure with Bob as the default implementation of EVE. The strengthened version ① uses convolution layers with four times as many feature channels as Bob and the strengthened version ② replaces the first two convolution layers of version ① with FC layers. In Sec. 6.6 we evaluate SpotSound's performance in resisting all three EVEs. 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.

Experimental setup. The experiments are performed in two different 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) three different office rooms with different sizes. Fig. 9 shows the floor map for the experiment scenes. **Note that all of the testing environments are unseen by SpotSound in the training process.** 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,

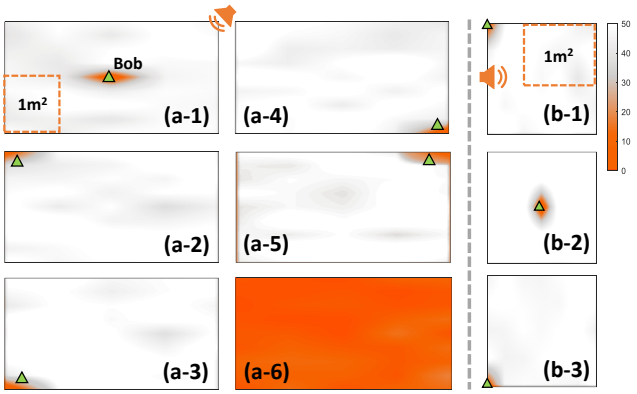


Figure 10: Overall performance. (a-1)~(a-5) shows EVE's BER tested in the living room when Bob is put at different locations; (a-6) shows Bob's BER tested in the living room. Fig. (b-1)~(b-3) shows EVE's BER tested in the study room.

maximize that on EVE. We also evaluate SpotSound's throughput in Sec. 6.5.

6.2 Overall Performance

We start by examining SpotSound's ability in concentrating the signal on a target area. In the experiment, we deploy Alice in the top right corner of a living room and deploy Bob on five different locations in the room, as marked in Fig. 9. At each location, we let Alice transmit signals to Bob in a spatial-selective manner, during which we put EVE on 45 different locations in the environment, and observe whether EVE can decode the transmission between Alice and Bob.

Fig. 10(a-1)~(a-5) shows 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×0.5m spot** –– EVE's BER keeps higher than 40% as long as it locates 0.5m far away from Bob. EVE cannot decode Alice's signal even in the case where it locates next to Alice.

However, we can also observe that SpotSound's signal control precision slightly decreases when Bob locates quite close to Alice, (see Fig. 10(a-5)). This is because SpotSound's spatial-selective transmission needs the support of the multipath environment. When both EVE and Bob locate close to Alice, the difference between their channels becomes marginal, which leads to the slight spread of the signal spot, as shown in Fig.10(a-5). As a comparison, we also put Bob in the 45 positions and observe Bob's BER. Fig. 10(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 1.6×2.8 area. Fig. 10(b-1)~(b-3) shows 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.

6.3 Impact of model complexity

We in this section compare the performance of the three models shown in Table 1. Specifically, we repeat the experiments in Sec. 6.2 with model II and III. Fig. 11 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 0.6% higher than that achieved by model III. So, we use Model I as the default model in SpotSound.

6.4 Impact of practical factors

6.4.1 Direction. We first evaluate the impact of the signal's propagation direction. We perform two groups of experiments. In the first group of experiments, we vary the angle between Alice and Bob from -180° to 180°, with a step of 10°, during which we fix the Alice-to-Bob distance at 1m. We measure Bob's BER on different angles and the results are shown in Fig. 12 (right). As can be seen, Bob's BER keeps lower than 1% when the angle changes from -60° to 60°. When the angle increases to ±90°, the BER begins to increase. However, although in the case where Bob locates behind the speaker (i.e., at ±180°), Bob's BER is still lower than 3.5%.

We further evaluate EVE's performance. During the experiment, we fix both the Alice-to-EVE distance and Bob-to-EVE distance at 1m, and change the angle between Alice and EVE from -180° to 180° by changing Alice's orientation. We let Alice concentrate its transmission to Bob, and observe EVE's BER from different angles. The results in Fig. 12 (left) show that BER keeps higher than 45% in all directions.

6.4.2 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. 13. We can see that the impact of distance is not so 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. Figs. 14 (a)-(c) show the results obtained under different orientations of Alice. We can see that EVE's BER keeps higher than 40% in all cases.

6.4.3 Environments. To evaluate SpotSound's performance in different environments, we perform experiments in 7 testing sites shown in Fig. 9. In each experiment, we fix the

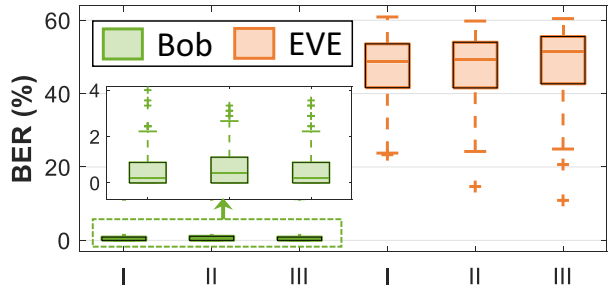


Figure 11: Performance with different DL models.

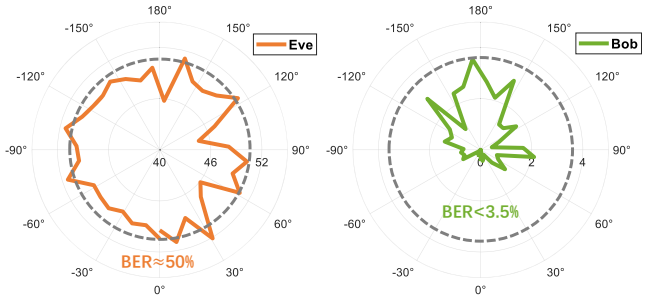


Figure 12: Impact of signal propagation direction

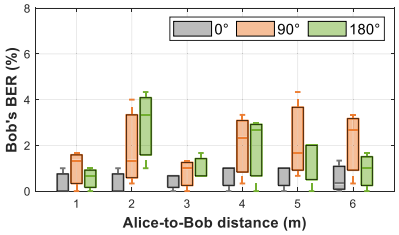


Figure 13: Bob's BER under different Alice-to-Bob distances.

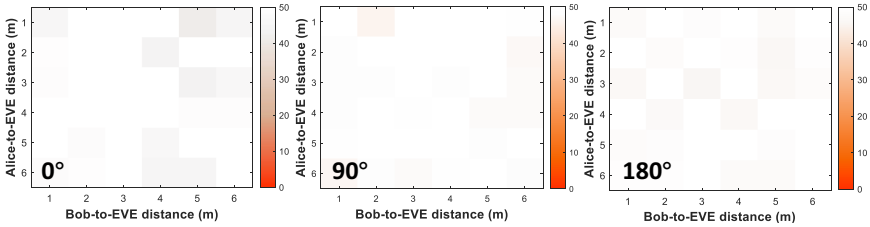


Figure 14: EVE's BER under different Alice-to-EVE distances and Bob-to-EVE distances

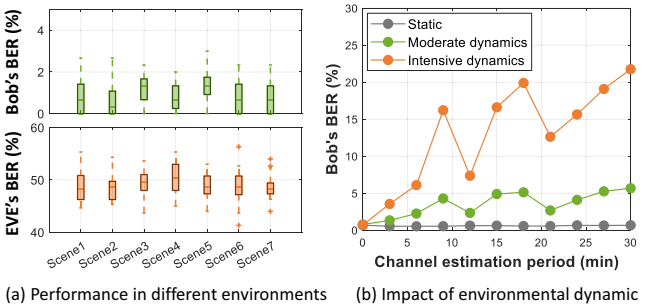


Figure 15: Performance under environments with different clutter levels and dynamic levels.

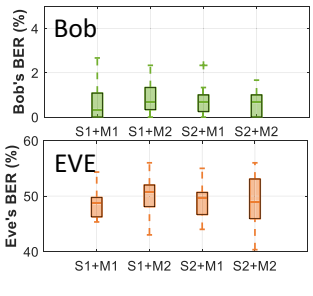


Figure 16: Performance with different devices.

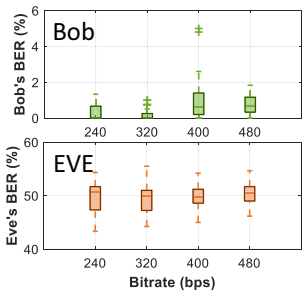
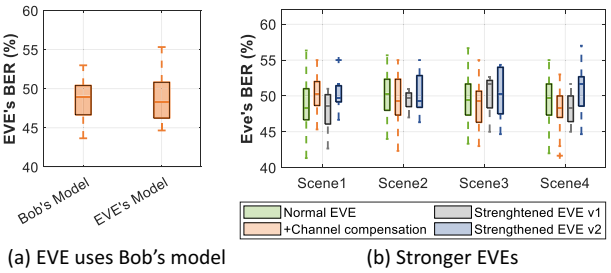


Figure 17: Performance under different bitrates.

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. 15 (a) 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, in average the EVE's BER obtained in a cluttered kitchen is 2.3% higher than that obtained in the living room.

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 (e.g., the chair) can also change. We tested two kinds of dynamic scenarios: i) moderate dynamics where the moving objects will not disturb the LoS channel between Alice and Bob; and ii) intensive dynamics where the moving objects will disturb the LoS channel. In each scenario, we observe SpotSound's performance obtained with different channel estimation frequencies. The results are shown in Fig. 15 (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 that, compared to the large reflectors in the environment (e.g., the walls and the large furniture), the moving people and the chairs have only marginal effect to the channel. Such small changes in the CIR measurement are considered in the training process, as we have discussed in Sec. 4.4. However, when working at a intensively dynamic environment, SpotSound has to re-estimate the channel every 5 min to keep its BER lower than 5%. The above results tell 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.



(a) EVE uses Bob's model (b) Stronger EVEs

Figure 18: Performance in resisting different EVEs.

6.4.4 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 speaker and microphone are shown in Fig. 16. As can be seen, the type of the device does not have apparent impact on SpotSound's performance.

6.5 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.

Fig. 17(a) shows Bob's BER with different bitrates. As can be seen, Bob can still achieve a lower than 3% BER even at the bitrate of 480bps. Fig. 17(b) shows EVE's BER, which does not show apparent variation under different bitrates.

6.6 Resistant to stronger EVEs

We in this section evaluate SpotSound's ability in resisting 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. 18 (a) compares the eavesdropper's performance when using EVE's and Bob's model. As can be seen, the two models achieve very similar BER.

Then we evaluate SpotSound's ability in resisting stronger EVEs. We consider two cases. In the first case, EVE can communicate with Alice (it may be another legitimate receiver of Alice who wants to eavesdrop on the communication between Alice and Bob), so it can measure and compensate its channel h_E. In the second case, we consider the strengthened EVEs as mentioned in Sec. 6.1.

We compare the BER obtained by the normal EVE and the three stronger EVEs in different environments as shown in Fig. 18 (b).

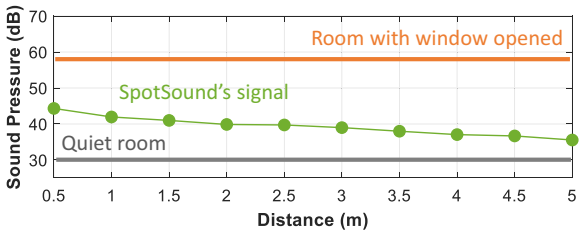


Figure 19: Sound pressure of SpotSound's signal measured from different distances.

Table 3: Processing time for a 60-bit packet.

	Alice	Bob
Raspberry pi	20.49 ms	7.5 ms
Snapdragon 870	3.05 ms	1.04 ms

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 that 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. 7 that the signal received on a non-target channel resembles a random noise, which can even pass the randomness test.

6.7 Sound pressure measurement

Although SpotSound transmits signal only on the inaudible bandwidth (i.e., 17-20KHz), it may still produce faint noises on the audible band due to the non-linearity in acoustic devices. We in this section conduct an experiment to examine whether SpotSound can produce noticeable sounds. In the experiment, we measure the sound pressure of the signal produced by Alice from different distances, the results are shown in Fig. 19. As a comparison, we also plot the sound pressure measured in a totally quiet room and an office with ambient noise from outside the window. As can be seen, the sound pressure produced by SpotSound is only slightly higher than that measured in a totally quiet room, and is lower than that measured in the office.

6.8 Time overhead

We have also measured the time overhead of SpotSound on Raspberry pi and Snapdragon 870. Table 3 shows the processing time required in generating/decoding a 60-bit packet. As can be seen, the processing time for packet generation and decoding is much shorter than the length of a packet.

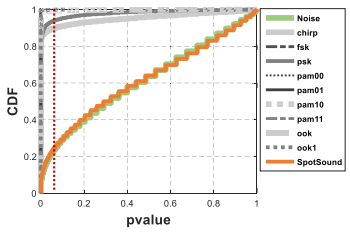


Figure 20: Evaluate the randomness with KS test.

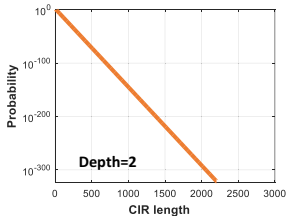


Figure 21: Computation cost for brute force attack.

7 SECURITY ANALYSIS

7.1 Randomness of EVE's Signal

We first use Kolmogorov-Smirnov test to evaluate the generated signal's goodness-of-fit with respect to random noise. Fig. 20 shows that the signal generated by SpotSound passes the test with Pvalue 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.

7.2 Resistant to brute force attack.

In the brute force attack, EVE tries to guess the CIR of Bob's channel and reconstruct Bob's signal. Experimental results tell 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 CIR measurement used in SpotSound contains 3000 samples, so there will be 2^{3000} possible CIRs even when the depth of each sample is 2. Fig. 21 shows the possibility that EVE can obtain a satisfying CIR with one guess under different CIR lengths L. As can be seen, 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.

8 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-surface, another way to control the signal's propagation behavior is to use static reflectors [6, 31]. 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 [22], heat [11], screen light [26, 36], electromagnetic [10, 33], and 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 [3, 28]. 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.

9 DISCUSSIONS AND LIMITATIONS

SpotSound still has limitations worth further exploration:

- **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 work achieve inferior performance in low or moderate multi-path environment. 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.

10 CONCLUSIONS

Eavesdropping attacks have compromised the security of billions of IoT devices. SpotSound tries to build spatial-selective channel among those devices. By controlling the signal's fine-grained shape, SpotSound can embed information into 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 a $0.25m^2$ target area, and meanwhile achieves a data rate of up to 480 bps.

11 ACKNOWLEDGMENT

We thank our shepherd and the anonymous reviewers for their constructive feedback. This work was supported by the National Key Research and Development Program of China under Grant 2020YFB1708700, and National Natural Science Fund of China (62272293,42050105).

REFERENCES

[1] V. Arun and H. Balakrishnan. 2019. Towards Programming the Radio Environment with Large Arrays of Inexpensive Antennas. In *NSDI*.

[2] V. Arun and H. Balakrishnan. 2020. RFocus: Practical Beamforming for Small Devices. In *NSDI*.

[3] Y. Bai, J. Liu, L. Lu, Y. Yang, Y. Chen, and J. Yu. 2020. BatComm: Enabling Inaudible Acoustic Communication with High-throughput for Mobile Devices. In *SenSys*.

[4] D. D. Carlo, P. Tandeitnik, C. Foy, N. Bertin, A. Deleforge, and S. Gannot. 2021. dEchorate: A Calibrated Room Impulse Response Dataset for Echo-Aware Signal Processing. *EURASIP Journal on Audio, Speech, and Music Processing* 39, 2021 (2021).

[5] A. Chaman, J. Wang, J. Sun, H. Hassanieh, and R. R. Choudhury. 2018. Ghostbuster: Detecting the Presence of Hidden Eavesdroppers. In *ACM MobiCom*.

[6] J. Chan, C. Zheng, and X. Zhou. 2015. 3D Printing Your Wireless Coverage. In *HotWireless*.

[7] T. Chen, L. Shangguan, Z. Li, and K. Jamieson. 2020. Metamorph: Injecting inaudible commands into over-the-air voice controlled systems. In *NDSS Symposium*.

[8] M. Dunna, C. Zhang, D. Sievenpiper, and D. Bharadia. 2020. ScatterMIMO: Enabling Virtual MIMO with Smart Surfaces. In *MobiCom*.

[9] C. Feng, X. Li, Y. Zhang, X. Wang, L. Chang, F. Wang, X. Zhang, and X. Chen. 2020. RFlens: Metasurface-Enabled Beamforming for IoT Communication and Sensing. In *MobiCom*.

[10] M. Gao, F. Lin, W. Xu, M. Nuermaimaiti, J. Han, W. Xu, and K. Ren. 2020. Deaf-Aid: Mobile IoT Communication Exploiting Stealthy Speaker-to-Gyroscope Channel. In *ACM MobiCom*.

[11] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici. 2015. BitWhisper: Covert Signaling Channel between Air-Gapped Computers using Thermal Manipulations. In *ACM CSF*.

[12] S. Han, J. Pool, J. Tran, and W. J. Dally. 2015. Learning both Weights and Connections for Efficient Neural Networks. In *NIPS*.

[13] Yuan He, Xiuzhen Guo, Jia Zhang, and Haotian Jiang. 2021. WIDE: Physical-Level CTC via Digital Emulation. *IEEE/ACM Transactions on Networking* 29, 4 (2021), 1567–1579. https://doi.org/10.1109/TNET.2021.3071782

[14] X. Lei, G-H Tu, C-Y Li, T. Xie, and M. Zhang. 2020. SecWIR: Securing Smart Home IoT Communications via Wi-Fi Routers with Embedded Intelligence. In *ACM MobiSys*.

[15] S. Madani, S. Jog, J. O. Lacruz, J. Widmer, and H. Hassanieh. 2021. Practical Null Steering in Millimeter Wave Networks. In *NSDI*.

[16] Xin Na, Xiuzhen Guo, Yuan He, and Rui Xi. 2021. Wi-attack: Cross-technology Impersonation Attack against iBeacon Services. In *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SECON52354.2021.9491605

[17] J. Nolan, K. Qian, and X. Zhang. 2021. RoS: Passive Smart Surface for Roadside-to-Vehicle Communication. In *SigComm*.

[18] Y. Qiao, O. Zhang, W. Zhou, K. Srinivasan, and A. Arora. 2016. PhyCloak: Obfuscating Sensing from Communication Signals. In *NSDI*.

[19] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. 2016. Generative Adversarial Text to Image Synthesis. In *ICML*.

[20] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi. 2019. Information Exposure From Consumer IoT Devices. In *ACM IMC*.

[21] Y. Ren, P. Wen, H. Liu, Z. Zheng, Y. Chen, P. Huang, and H. Li. 2021. Proximity-Echo: Secure Two Factor Authentication Using Active Sound Sensing. In *INFOCOM*.

[22] N. Roy, M. Gowda, and R. R. Choudhury. 2015. Ripple: Communicating through Physical Vibration. In *Usenix NSDI*.

[23] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra. 2016. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *NSDI*.

[24] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J. Merolla. 2007. Applications of LDPC Codes to the Wiretap Channel. *IEEE Transactions on Information Theory* 53, 8 (2007), 2933–2945.

[25] A. Thangaraj, S. Dihidar, A. R. Calderbank, S. W. McLaughlin, and J. Merolla. 2010. Exploiting Transitivity of Correlation for Fast Template Matching. *IEEE TIP* 19, 8 (2010), 2190–2200.

[26] A. Wang, C. Peng, O. Zhang, G. Shen, and B. Zeng. 2015. InFrame: Multiflexing Full-Frame Visible Communication Channel for Humans and Devices. In *ACM MobiSys*.

[27] Q. Wang, K. Ren, G. Li, C. Xia, X. Chen, Z. Wang, and Q. Zou. 2015. Walls Have Ears! Opportunistically Communicating Secret Messages Over the Wiretap Channel: from Theory to Practice. In *ACM CCS*.

[28] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, and L. Su. 2016. Messages Behind the Sound: Real-Time Hidden Acoustic Signal Capture with Smartphones. In *ACM MobiCom*.

[29] Yuting Wang, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2022. PolarTracker: Attitude-Aware Channel Access for Floating Low Power Wide Area Networks. *IEEE/ACM Transactions on Networking* 30, 4 (2022), 1807–1821. https://doi.org/10.1109/TNET.2022.3154937

[30] P. Xie, J. Feng, Z. Cao, and J. Wang. 2017. GeneWave: Fast Authentication and Key Agreement on Commodity Mobile Devices. In *ICNP*.

[31] X. Xiong, J. Chan, E. Yu, N. Kumari, A. A. Sani, C. Zheng, and X. Zhou. 2017. Customizing Indoor Wireless Coverage via 3D-Fabricated Reflectors. In *BuildSys*.

[32] Leiyang Xu, Xiaolong Zheng, Xiangyuan Li, Yucheng Zhang, Liang Liu, and Huadong Ma. 2022. WiCAM: Imperceptible Adversarial Attack on Deep Learning based WiFi Sensing. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 10–18. https://doi.org/10.1109/SECON55815.2022.9918564

[33] Z. Yang, Q. Huang, and Q. Zhang. 2018. NICScater: Backscater as a Covert Channel in Mobile Devices. In *ACM MobiCom*.

[34] Fu Yu, Xiaolong Zheng, Liang Liu, and Huadong Ma. 2022. LoRadar: An Efficient LoRa Channel Occupancy Acquirer based on Cross-channel Scanning. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. 540–549. https://doi.org/10.1109/INFOCOM48880.2022.9796845

[35] Jia Zhang, Xiuzhen Guo, Haotian Jiang, Xiaolong Zheng, and Yuan He. 2021. Link Quality Estimation of Cross-Technology Communication: The Case with Physical-Level Emulation. *ACM Trans. Sen. Netw.* 18, 1, Article 14 (oct 2021), 20 pages. https://doi.org/10.1145/3482527

[36] K. Zhang, C. Wu, C. Yang, Y. Zhao, K. Huang, C. Peng, Y. Liu, and Z. Yang. 2018. ChromaCode: A Fully Imperceptible Screen-Camera Communication System. In *ACM MobiCom*.

[37] Y. Zhu, Z. Xiao, Y. Chen, Z. Li, Max Liu, B. Y. Zhao, and H. Zheng. 2020. Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors. In *NDSS*.