ABSTRACT Backscatter communication is a promising technology in the hyper-connected era. Because of its ultra-low energy consumption, it can be used in various applications, but there are performance issues due to high uncertainty. We propose a signal-to-data translation model that can transform an entire backscatter signal into the original data. To train the translation model, we developed an automation framework that can efficiently collect datasets. We also proposed a data augmentation technique suitable for backscatter signals. In extensive experiments, our model significantly outperformed a simple rule-based decoding method and a commercial RFID reader. The proposed model showed consistent performance gains across different locations, obstacles, and mobility scenarios indicating a good generalization of learning.

INDEX TERMS Backscatter Communication, Signal-to-Data Translation, Deep Learning, Data Augmentation

I. INTRODUCTION

In the hyper-connected era, machine-to-machine communications and Internet of things technologies are essential. As communication consumes significant energy, low-energy communication technologies are receiving a lot of attention. Backscatter communication can transmit data by reflecting (or backscattering) surrounding RF signals without generating its own carrier signal. It can reduce energy consumption by tens to hundreds of times compared to regular wireless communication technologies such as WiFi and Bluetooth. It also has the advantage of reducing device cost and size as it does not require a battery to be installed.

RFID (Radio-Frequency Identification) is a typical technology based on backscatter communication that has achieved commercial success, and it is used in various fields such as logistics, and transportation [1]. In addition, RFID has been used for indoor positioning [2], movement awareness [3], and estimating the number of people in a crowd [4]. More recently, beyond RFID, backscatter communication has been proposed for video transmission [5], voice calls [6], and implant sensors [7], and more application fields are expanding [8].

Recently, Ambient Backscatter (AmBC) has been studied [9]. AmBC enables that tags can reflect ambient RF signals (e.g. TV broadcasting signals) without a dedicated signal of a reader. In early studies [9], [10], [11] on AmBC, tags directly communicate (backscatter and receive) each other using ambient RF signals. Therefore a backscattered response is decoded by a tag which has limited power and computing resources. More recently, AmBC studies extend to consider that a dedicated reader exists to improve the backscatter communication performance [12], [13], [14] or security [15]. [16]. More advanced works have been studied such as WiFi-backscatter [17], [18], [19], [20], Bluetooth backscatter [21], [22], FM radio-backscatter [23], and LoRa backscatter [24], [25] to improve the backscatter communication performance using the existing communication systems, but they require to increase the complexity of a tag.

Although research-level proposals involving backscatter communication have emerged, it isn’t easy to commercialize these ideas beyond the prototype level. The backscatter signal strength is inversely proportional to the fourth order of the distance between the tag and the reader. As the distance increases, the signal strength exponentially weakens, limiting communication coverage. An even bigger problem is that the uncertainty of backscatter communication is very high. According to recent research results, even a tiny change (e.g., a change in the distance, the angle, or the surrounding environment) can affect the signal amplitude by up to -10 dB and the phase by up to π radian [26]. This uncertainty makes
it difficult to accurately model (or estimate) communication performance, which hinders its use in real-world applications.

This paper proposes a signal-to-data translation model that can transform an entire backscatter signal into the original data. In a typical reader and tag configurations, a tag is cheap, small, and computation-constrained, while a reader is relatively rich in terms of both energy and computing resources. We design the tags to keep the current, simple encoding method (e.g., FM0), and the reader to use a learning-based translation model to improve the decoding success ratio significantly. To learn the uncertainty of backscatter communication, we need a wide variety of training data. We design an efficient data collection framework using programmable tags and reader systems for the first time. We also propose a data augmentation method to support the frequency error tolerance of tags.

Through extensive experiments, our proposed model improved the decoding success ratio by an average of 4.16x compared to a simple rule-based decoding method where tags are located in multiple positions with several angles. Our model also outperformed the Hidden Markov Model (HMM) approach and the commercial RFID reader, Impinj Speedway R420. To evaluate the generalization of our trained model, we performed additional experiments with different tag devices in another room. Our model consistently improved the decoding success ratio by an average of 4.43x compared to the rule-based decoding method without additional training. Our model also showed better performance when an (static or moving) obstacle was placed in front of the tag, and when the tag itself was moving.

Our contributions are as follows:

1) A signal-to-data translation model has been proposed for the first time without the need for tag modification.

2) We have developed a framework to efficiently collect both backscatter signals and their labels, and designed a data augmentation technique to increase the amount of backscatter signal data.

3) Our model greatly outperformed the simple rule-based decoding method, the HMM method, and the commercial RFID reader. For a certain position, the decoding success ratio increased from 0% up to 100%.

4) We have shown that our learning method can be generalized well with different tag devices, as well as in different locations, and even in the obstacle and moving scenarios.

Our paper is composed as follows. We explain backscatter communication and its characteristics, the traditional rule-based decoding method, and its limitations in Section II. We present the design considerations in Section III and we show our system and model design in Section IV. Our data collection procedure is explained in Section V and then the implementation is described in Section VI. We demonstrate our experimental method and the performance results in Section VII and provide a discussion in Section VIII. Finally, we show related works in Section IX and provide a conclusion in Section X.

II. BACKGROUND
A. BACKSCATTER COMMUNICATION

Backscatter communication is a communication technology in which a tag transmits data by backscattering (or reflecting) surrounding RF signals. Because it does not need to generate its own carrier wave, it is able to deliver information with W-level power. As a result, it is attracting attention as an ultra-low power communication technology.

The typical commercial technology based on backscatter communication is RFID. It is used in various fields, such as logistics, transportation, agriculture, healthcare, and sports [1]. The RFID market size is expected to reach $31.06 billion by 2026 [27]. In the RFID communication standard protocol EPC Gen2 [28], a very simple method is used as an encoding technique for tags (e.g., FM0 or Miller encoding). The FM0 symbol and the sequence of symbols are shown in Figure 1. One FM0 symbol represents one bit and consists of two amplitudes. The symbol representing zero has an amplitude inversion in the middle of the signal, and the symbol representing one has the same amplitude during the signal. Amplitude inversion must occur between the previous symbol and the next symbol. We use the terminology low (L) and high (H) for low amplitude and high amplitude, respectively.

B. TRADITIONAL RULE-BASED DECODING METHOD

A widely known traditional rule-based decoding method finds a preamble pattern in the received backscatter signal, and then trying decoding the remaining signal bit by bit. For each bit, it is selected as one of the most similar FM0 symbol candidates.

If the amplitude change (H→L or L→H) can be clearly detected, we can determine the FM0 symbol for each bit, as shown in the high-SNR case in Figure 2(a). However, in the low-SNR case in Figure 2(b), we need to calculate the correlation value of signal sample amplitudes and four FM0 symbols. The FM0 symbol that has the highest correlation value could be the chosen. This is a little bit more robust, but such a rule-based method still has a poor performance in real world applications because of the following reasons.
C. BACKSCATTER SIGNAL CHARACTERISTICS

The strength of the backscatter signal is inversely proportional to the fourth order of the distance between the reader and the tag, as described in (1) where $P_{tx}$ is the transmission power of the reader, $\lambda$ is the wavelength of the backscatter signal, $d$ is the distance between the reader and the tag, and $C$ is a constant value. Thus, the strength of the backscatter signal decreases exponentially as the communication distance increases.

$$R(dB) = 10 \log \left( \frac{C \cdot P_{tx} \cdot \lambda^4}{d^4} \right)$$ (1)

FIGURE 3. Decoding success ratio of the rule-based decoding method varies irregularly with respect to the distance between the reader and the tag. However, S2DT model has a consistent high decoding success ratio regardless of the distance.

Moreover, as shown in Figure 3, the decoding success ratio according to the distance in the rule-based decoding method shows an irregularly oscillating curve. With a commercial RFID reader, Impinj Speedway R420, also shows a similar irregular curve. According to the recent research, it has shown that the amplitude and phase of the backscatter signal can be sensitively affected by even small environmental changes [26]. This uncertainty makes it difficult to provide a stable performance on backscatter communication.

D. BLF ERROR TOLERANCE

To decode a backscatter signal bit by bit, determining the exact starting position and the length of each bit is important. When the number of bits is fixed (for example, an RFID response), the whole signal length should be the same for responses from all tags using the same BLF (backscatter link frequency).

If we guarantee the signal length, we may simply divide the received signal by the number of bits. However, the real tag’s BLF is slightly different due to the low-cost hardware precision limitation [29]. The EPC Gen2 protocol allows different BLF error tolerance with respect to the value of BLF. For example, for the a 40 KHz BLF, the BLF error tolerance should be $\pm 4\%$, so it should be acceptable between 38.4 kHz and 41.6 kHz.

FIGURE 4. Decoding success ratio varies with respect to the BLF. Each tag has a different optimal value of BLF which accomplishes the highest decoding success ratio.
In our preliminary experiment, Figure 4 shows that the actual BLFs for four tags are slightly different. All tags are designed to have a BLF of 40 kHz, but due to hardware limitations, the value of the BLF cannot be exactly 40 kHz. To decode backscatter signals from different tags, we need to know the optimal BLF value on each tag. However, it is difficult to know the optimal BLF before receiving a tag response.

E. ASYMMETRIC COMMUNICATION PROPERTIES

In general, a tag should be low-cost and simple design. These restrictions limit the tag’s operations and protocols. Therefore, it is difficult to implement a robust and advanced data encoding scheme in the tag. On the contrary, a reader can have more (several orders of magnitude) powerful computing resources than the tag. It means that the reader can afford to do something to improve the backscatter communication performance.

Considering the inherent asymmetry communications, we let the tag keeps the traditional and simple encoding scheme. However, we design more powerful and advanced decoding techniques for the reader system that can overcome the high dynamics and uncertainties of backscatter communication.

III. DESIGN CONSIDERATION

Backscatter signals are easily distorted according to the location and the angle of the reader and the tag. They are also vulnerable to multi-path effects caused by the surrounding environment. Because of these high uncertainties, it is difficult to design a (deterministic) rule-based approach. For example, the key parameters required to decode backscatter signals such as 1) the tags’ BLF, 2) the starting position of each bit, 3) the threshold value that distinguishes between low and high amplitudes, need to be configured in a heuristic manner. If the values of these parameters are not suitable, decoding errors are inevitable.

To address these challenges, we propose a signal-to-data translation model that uses deep neural networks to input the entire response signal and output a series of decoded bits (Figure 5), rather than a heuristic rule-based method.

A. BIT-LEVEL DECODING LIMITATIONS

Traditional rule-based decoding methods perform the decoding process on each bit one by one. For example, assume a tag transmits 128-bit data at 40 KHz and a reader samples the signal at 2 Msp (Mega sample per second). The number of samples of each bit would be 50, and dividing the entire signal by 50 results in 128 segments which map to 128 bits.

If the tag’s BLF is not precisely estimated, this simple division approach causes inaccurate bit separation, resulting in decoding failure. The reader may try to find the tag’s BLF within the acceptable error ranges of the EPC Gen2 standard. Such an exhaustive search requires lots of time and computation overhead. In addition, to decode the signal successfully, we need to consider the characteristics of the entire signal as much as individual bit information. The entire signal may include information about channel noise, self-interference, multi-path effects, and so on.

B. DEEP NEURAL NETWORK

A deep neural network is a kind of artificial neural network. It consists of multiple hidden layers which are connected between the input layer and the output layer [30]. The feature of the input information is extracted through the multiple hidden layers. Deep neural networks are used in speech recognition, computer vision, pattern recognition, etc. [31].

In a backscatter signal, original data, channel information, noise, and others are all mixed together. We have to extract our target information that a tag originally transmitted. Using a deep neural network, we can extract the features of the symbol from the received signal and translate them into the original data.

C. DATASET QUANTITY AND QUALITY

Deep neural networks need to be trained with a large amount of data to achieve good performance. For our model, data collection should bring together the response signals received and the corresponding label (original transmitted bits). However, collecting and labeling backscatter signals is very challenging for the following reasons.

1) Commercial RFID tags always answer the same data consist of 96-bits of EPC (electronic product code) and 32-bits of CRC (cyclic redundancy check). If tags backscatter their own EPC data, we must prepare lots of tags and collect the response individually to avoid model overfitting.

2) When the reader receives a certain amount of samples after it sends a query command, it must distinguish between situations where which the tag responds, but the signal is very weak, and the tag does not respond actually.

3) Backscatter communication has an uncertain nature that is highly affected by slight changes in the position and angle. It takes a lot of time and effort to collect data for all combinations, such as numerous positions and angles, to learn these uncertainties.

4) We should consider so many things for sufficient learning, such as covering the acceptable BLF error ranges of the EPC Gen2 standard.

Therefore, it is essential to collect large enough datasets and to label them precisely. Also, all of these processes should be efficient.

IV. SYTSTEM & MODEL DESIGN

In this paper, we design a deep learning model to improve the decoding performance of backscatter communication. We also propose efficient data collection and augmentation schemes for learning.

A. SIGNAL-TO-DATA TRANSLATION MODEL

Instead of matching the FM0 symbol on a bit level, we take the whole signal as an input for the deep neural network
model, and train the model to output the original data (i.e., the bit sequence) like a translation process. This allows the deep neural network to take all the important features into account comprehensively. Some of the features may be a bit-level aspects including the preamble detection, the starting position of each symbol, and the amplitude inversion between neighboring symbols. The other features could be the characteristics of the whole signal such as wireless channel information, self-interference, and multi-path effects.

We believe that having a comprehensive understanding from the whole signal can allow the features to be extracted more effectively, resulting in robust and superior decoding performance over the bit-level pattern matching method which only focuses on the local perspective.

B. DATA COLLECTION FRAMEWORK

To overcome the data collection issues mentioned in Section III-C, we designed an efficient framework, as follows. First, to overcome the limitation where commercial RFID tags always provide the same answer, we used a programmable tag, named WISP (i.e., the wireless identification and sensing platform). WISP is a tag capable of software-defined implementation while complying with the EPC Gen2 protocol. We programmed the tags to be able to backscatter random EPCs every round.

As shown in Figure 6, when the reader collects the signal after the query, it needs to distinguish whether the tag responds but the signal is weak or the tag does not respond. We modified the reader-to-tag communication protocol using a software defined radio (SDR) device, called USRP (universal software radio peripheral). We let the reader program generates random data (96 bits + CRC 32 bits), and piggyback the random data after a query command. When the tag receives the query, it extracts the random data from the query and stores it in non-volatile memory before backscattering. This allows us to check whether the tag responded or not, and what random data is backscattered (it can be used as the label).

C. DATA AUGMENTATION TECHNIQUE

Data augmentation is a technique used to increase the amount of data by modifying existing data. It helps reduce the overfitting of the trained model.

As mentioned above, we need to support a full range of BLF error tolerance recommended by an EPC Gen2 protocol. Because collecting the signal with all error range of BLFs is impossible, we apply the following augmentation technique. We first collect a dataset using a single tag (with a fixed BLF). Using the dataset, we can augment collected signals to be rescaled versions to cover the full error range.

While training, we also add Gaussian noise to our deep

---

**FIGURE 5.** System Overview

**FIGURE 6.** Flowchart for modified protocol between reader and tag for automated data collection.
learning model in order to reduce overfitting and improve model generalization.

V. DATASET COLLECTION

We built a testbed in an obstacle-free, 6×10 m office, as shown in Figure 7. The Rx antenna and Tx antenna were located at a distance of 1 m to reduce self-interference. The tags were arranged in a rectangular grid with 1 m spacing up to 4 m away, located to the left, center, and right of the Rx antenna, as shown in Figure 8. The installed height of the tag was the same as that of the Rx antenna. At each location, the tag had four angles rotated by 45 degrees. Therefore, the dataset was collected at 12 locations with four angles (a total of 48 cases).

To collect the dataset, we used an SDR device, USRP N210, and the programmable tag, WISP. We found an open-source code [33] that supports a basic EPC Gen2 protocol and then modified it to be our data collection framework.

The reader receives a signal at $2 \times 10^6$ samples per second, and records the sampled signal along with its label, which is the original data generated by the reader.

We collected 3,000 signals for each of the 12 locations with four angles. The total size of the dataset is 144,000 signals ($3,000 \times 12 \times 4$). We used 20% of the dataset as the test set. The remaining 80% was divided into the training set and the validation set at a ratio of 4:1. As shown in Figure 9, the training set, the validation set, and the test set contains 92,160, 23,040, and 28,800 signals for each location, respectively.

We have four programmable tags with slightly different BLFs, as shown in Table 1. During the dataset collection, we used only one tag, i.e., tag #2, and the rest of the tags were used in the learning generalization evaluation, as discussed in Section VII.

VI. IMPLEMENTATION

A. MODEL SPECIFICATION

Our S2DT model structure is presented in Figure 10. It is composed of encoder blocks with convolution layers, and decoder blocks with dense layers.

The input size is 7,300 x 2, where 7,300 is the number of samples for 38.4 KHz (i.e., the slowest frequency within the allowable error range). Each signal data is composed of in-phase (I) and quadrature (Q) components. The input signal passes the Gaussian noise layer first. This layer adds arbitrary Gaussian noise to each input, which generalizes the learning. The Gaussian noise layer is only applied during learning and is automatically disabled during testing.

In each encoder block, the input goes through a convolution layer, a pooling layer, a batch normalization layer, and a ReLU activation function. The extracted features from the encoder part are unfolded in one dimension through a flatten layer. Next, it feeds into the decoder blocks which are fully connected, multi-layered perceptron. Each block is composed of a dense layer, a batch normalization layer, a ReLU activation function, and a dropout layer to prevent overfitting.

The final output size is 128 with a softmax function. An output of 128 represents the decoded original data of 128 bits. The binary cross entropy is applied as a loss function to classify data as 0 or 1.

We use various model sizes for both the encoder and the decoder parts as shown in Table 2. According to the model size, the performance is evaluated in Section VII-C. The structure shown in Figure 10 indicates the largest version.

We use various model sizes for both the encoder and the decoder parts as shown in Table 2. According to the model size, the performance is evaluated in Section VII-C. The structure shown in Figure 10 indicates the largest version.

The number of parameters for each block is shown in Table 3. Case (a) is the model with the deep encoder and the deep decoder, case (b) is the model with the deep encoder and...
the shallow decoder, case (c) is the model with the shallow encoder and the deep decoder, and case (d) is the model with the shallow encoder and the shallow decoder. Notice that only half of the parameters in the batch normalization layer are trainable.

In the largest version, case (a), the number of total parameter is 482,520,224. Even the smallest version has hundreds of millions of parameters.

**B. MODEL IMPLEMENTATION**

Model learning and inference were conducted on a GPU-equipped machine. An open-source Python library, i.e., Tensorflow-2, was used to write the program code. The values of the parameters used in the learning are as shown in Table 4.

The learning epoch was not fixed. After saving the model weight when the validation success ratio reached the peak, if the peak was not updated for five epochs, the model weight stored at the peak was restored and learning was terminated.

**C. DATASET AUGMENTATION**

As noted in Section II-D, the EPC Gen2 standard should allow 4% tolerance for a tag’s BLF. For example, the error tolerance ranges from 38.4 kHz to 41.6 kHz if the BLF is 40 kHz. We collect the dataset using tag #2 in Section V, which has an average BLF of 40.72 kHz. If we train our model using only this dataset, the model must overfit to the BLF at 40.72 kHz.

To prevent over-fitting for a fixed BLF, we proposed the following augmentation scheme. 1) Pick a signal sequentially from the dataset, which is collected by using a single BLF at 40.72 kHz. 2) Randomly select a new BLF randomly with uniform distribution from 38.4 kHz to 41.6 kHz. 3) Resize the signal to the new BLF version, as shown in Figure 11. Resizing can be done by sub-sampling (to decrease the original signal) or linear interpolation (to increase the original signal).
FIGURE 11. The range of BLF allowed by the EPC Gen2 standard is 38.4kHz to 41.6kHz. We randomly select eight BLF values, and the signal is converted to each BLF value. The change in the value of BLF changes the number of samples representing one bit. (a) The number of samples increases when BLF decreases. (b) The number of samples decreases when BLF increases.

VIII. EVALUATION

We define the decoding success ratio as the ratio of the number of successfully decoded packets and the number of packets backscattered by the tag. Specifically, only when the entire 128-bit of the packet is successfully decoded, the packet is considered to be successfully decoded. This metric is valid for evaluating the performance, because we used fixed data rate.

The data rate is defined by the ratio of the value of BLF and M (Number of subcarrier cycles per symbol). Our work only focus on FM0 modulation, so M is 1. In other words, the data rate is same as the value of BLF in our work. In the EPC Gen2 protocol, the reader specifies the BLF in the query, and all tags are required to respond accordingly. Since we let the reader use the fixed value of BLF, all tags respond with the same fixed data rate.

A. DECODING METHODS

We compared the decoding performance of our S2DT model with the following three decoding methods: a simple rule-based, HMM, and a commercial RFID.

The rule-based decoding method is a traditional correlation approach as described in Section II-B. We used the optimal BLF length per tag (as shown in the Table 1), and divide the backscatter signal by the BLF length. For each divided part, we calculated a correlation value with the four FM symbols shown in Figure 1 and we selected one that maximizes the correlation value.

In the HMM method [34], we used the Viterbi algorithm [35]. While calculating the correlation value in the same way as the rule-based method, the Viterbi algorithm used this value as the emission probability. The transition probability was set equal to 0.5 for both 0 and 1 bits, and we constructed a trellis diagram to find out the Viterbi path and decoded the signal with that path.

Third, we used Impinj® Speedway® R420 [36] as a commercial RFID reader. The R420 is one of the most popular high-end RFID readers on the market, but it still uses the rule-based decoding method.

B. OVERALL PERFORMANCE

First, when the reader and tag were in a straight line without changing the angle, signals were collected at 0.1m intervals from 1.0m to 4.6m (the maximum distance a tag can hear the reader’s commands). Then, we compared the decoding success ratio of collected signals with respect to the rule-based method, Speedway R420, and the S2DT model in the Figure 12. As mentioned before, the result of rule-based method and Speedway R420 shows irregularly oscillating curve after 2.5m. However, the S2DT model achieved almost the maximum decoding success ratio regardless of the distance.

Next, we evaluate the performance according to the position and angle of the tag introduced in Section V. Figure 12 shows the results. For the average results on four angles at 12 positions, the rule-based method achieved the lowest decoding success ratio of 22.77%. HMM got 23.99% slightly higher than the rule-based method. The performance of Speedway R420 (34.19%) showed quite better performance than the rule-based method and HMM. Our S2DT model achieved 94.83%, significantly outperforming other methods.

Figure 13 presents the decoding success ratio for each pair of locations and angles. It can be seen that the rule-based method works well only when the tag is close to and facing the reader (this trend was almost the same for HMM and R420), while our model had a consistently high decoding success ratio in all cases. Even the S2DT model achieved 100% decoding success ratio at the positions and angles where the rule-based, HMM, and R420 were zero percent.

C. PERFORMANCE EFFECT ACCORDING TO MODEL SIZE

Figure 14 compares the decoding success ratio with different model sizes in S2DT. We designed the encoder and decoder to have Shallow and Deep types. Each type size is as shown in Table 2.

As the model size goes deeper (or larger), the decoding success ratio becomes higher. When the encoder was shallow,
the model performance was relatively low, and the decoder size had a large impact as shown in Figure 14 (a) and (b). When the encoder was deep, the model performance was high regardless of the decoder size in Figure 14 (c) and (d). Even the smallest model version still showed better performance than other methods.

D. MODEL GENERALIZATION

Regularization is a method to reduce overfitting as well as to increase generalization in the learning process. For regularization techniques, we used batch normalization, dropout, and the addition of Gaussian noises. We also applied data augmentation as described in Section VI-C to support the full range of BLF error tolerance recommended by an EPC Gen2 protocol.

To evaluate the effect of our regularization techniques and data augmentation, we collected the new test-set from another room. In an obstacle-free, 6x8 m office, which was narrower than the previous location. Almost all of the conditions were the same as those in Section V, but we placed tags nine positions in a grid topology, and we used three tags (1, 3, and 4) as shown in Table V. All the three tags were not used in the train-set. We did not conduct any additional training on the new location and different tags.

Table 5 shows the ablation test results. The rule-based method showed low decoding success ratio 21.41% as expected. Without both regulation techniques and data augmentation, our S2DT model only achieved 24.68%. The poor performance is due to that the model was overfitted to the tag #2 in the previous location.

Using regularization techniques (without data augmentation), the performance was increased to 42.65%. The model with only data augmentation achieved decoding success ratio of 94.64%. When both regularization techniques and data augmentation were applied, S2DT showed 94.93%. It indicates that data augmentation had a significant impact on...
The rule-based method showed 91.40% in case (a), but without additional training or fine-tuning. We used the S2DT model repeatedly in front of the fixed tag. We used the S2DT model signals. Finally, in case (d), we move one hand up and down repeatedly while collecting the signals between the reader antenna and the fixed tag. In case (c), we move the tag up and down dynamically. This is why the rule based method failed to decode. Although these distorted signals were not seen during training, the S2DT model performed robustly well.

### F. NUMBER OF BIT ERRORS

In Figure 17, we plot the number of bit errors occurred within a single response when decoding failed by the rule-based method and our S2DT model. The rule-based method failed to decode frequently even though there are only a few bit errors in response, but the S2DT model decoded successfully in most cases, except where the number of bit errors is very severe (around 50 to 70). It means that our S2DT model can recover from a small number of bit errors pretty well.

### VIII. DISCUSSION

This section presents topics of discussion for future study.

Our proposed model performed an average of 4.16 times better than the traditional rule-based method where tags are located at grid places with several angles.

However, in certain locations, we found that the tag could not receive the reader's command, and the tag had no chance to respond. Addressing this issue is beyond the scope of this paper. However, in our other work, we are doing research on a beam-forming technology with multiple antennas for backscatter communication. It can help the tag better hear the reader’s query command.

Since FM0 modulation has a distinct feature in which there exists a reversal of an amplitude between the two signals, most errors appear continuously unit of even bits. 99.86% of the bits that failed in the rule-based decoding method and 92.06% of the bits that failed in the S2DT model were consecutive errors.

In this work, we focus on FM0/40 KHz modulation. To support multiple modulation schemes and data rates, we need to collect additional datasets for modulated signals with various data rates. After that, we need to perform the same training process from scratch. Alternatively, we can try to do transfer learning [37] or domain adaptation [38]. We also plan...
to decode various data lengths in the future, rather than 128-bit fixed.

We have yet to deeply consider the reader’s decoding, and the decoding time depends on our model inference time. As described in Section VI-A our model has hundreds of millions of parameters which make the inference time longer and require hardware resources such as NPU and GPU. Therefore, it is difficult to apply to actual applications in the current form.

However, our work focus on proposing the possibility and the effectiveness of improving the decoding performance of backscatter communication which is vulnerable with noise by using deep learning. We will apply model compression techniques such as quantization [39], pruning [40] or knowledge distillation [41] to practically reduce the size of the model (memory) and the inference time.

IX. RELATED WORKS

- **Backscatter Communication Efficiency**: BLINK [42] is a bit-rate and channel adaptation protocol used to improve communication throughput. CARA [43] also tries to improve throughput by taking advantage of channel-aware rate selection. Both of these techniques focus on rate selection, while we concentrate on improving the decoding performance on weak and distorted signals.

- **Parallel Decoding**: Laissez-Faire [44], BiGroup [45], Fliptrancer [46], Hubble [47], FreeScatter [48], and Fireworks [49] propose to decode parallel backscatter signals by analyzing signals in the time and/or IQ domains. Because these works require moderate SNR levels, they can be used to complement our proposed model.

- **New Backscatter Paradigms**: Recently, novel backscatter communication systems have been proposed that operate with different wireless systems, such as WiFi-backscatter [17], [18], [19], [20], Bluetooth backscatter [21], [22], FM radio-backscatter [23], and LoRa backscatter [24], [25]. These systems can extend the communication coverage and improve throughput, but require tag modification and increase tag cost. In contrast, our work can operate with commercial tags or low-cost, highly constrained tags.

- **Deep Learning for the Physical Layer**: These works seek to apply deep learning to the physical layer and use it for wireless communication. These works require deep learning components on both transmitters and receivers [50], [51], [52], [53]. Our work differs from these in that only the reader side uses deep learning techniques, which keeps the tags simple, low-cost, and unmodified.

X. CONCLUSION

This paper proposes a signal-to-data translation model for backscatter communication which is vulnerable to slight changes in ambient environments. Our proposal allows the tags to use an existing simple modulation scheme without any modification. By training the S2DT model, the reader learns how to translate the entire backscatter signal into the original data. To realize the S2DT model, we designed a flexible data collection framework using an SDR-based reader and programmable tags. We also proposed a data augmentation method to support EPC Gen2 standard frequency error tolerance.

In our evaluation, our S2DT model significantly outperformed the simple rule-based method, HMM, and the commercial RFID reader in multiple positions and angles. We also evaluated our model in another room with different tags without additional fine-tuning. S2DT showed consistent performance gains over other methods, while supporting the different tags’ BLF error tolerance. Even in the obstacle and moving scenarios that were not exposed during training, the S2DT model demonstrated much better performance, indicating that our learning was well generalized.

REFERENCES

[1] E. Ilie-Zudor, Z. Kemény, F. Van Blommestein, L. Monostori, and A. Van Der Meulen. “A survey of applications and requirements of unique identification systems and RFID techniques,” Computers in Industry, vol. 62, no. 3, pp. 227–252, 2011.

[2] J. Zhou and J. Shi, “RFID localization algorithms and applications—a review,” Journal of intelligent manufacturing, vol. 20, no. 6, p. 695, 2009.

[3] H. Ding, L. Shangguan, Z. Yang, J. Han, Z. Zhou, P. Yang, W. Xi, and J. Zhao, “Femo: A platform for free-weight exercise monitoring with RFID,” in Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, 2015, pp. 141–154.

[4] H. Ding, J. Han, A. X. Liu, J. Zhao, P. Yang, W. Xi, and Z. Jiang, “Human object estimation via backscattered radio frequency signal,” in 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015, pp. 1652–1660.

[5] A. Saffari, M. Hessian, S. Naderiparizi, and J. R. Smith, “Battery-free wireless video streaming camera system,” in 2019 IEEE International Conference on RFID (RFID). IEEE, 2019, pp. 1–8.

[6] Q. Zhang, L. Zhang, Y.-C. Liang, and P.-Y. Kam, “Backscatter-NOMA: A symbiotic system of cellular and Internet-of-Things networks,” IEEE Access, vol. 7, pp. 20000–20013, 2019.

[7] A. Khaleghi, A. Hasanvand, and I. Balasingham, “Radio frequency backscatter communication for high data rate deep implants,” IEEE Transactions on Microwave Theory and Techniques, vol. 67, no. 3, pp. 1093–1106, 2018.

[8] C. Xu, L. Yang, and P. Zhang, “Practical backscatter communication systems for battery-free Internet of Things: A tutorial and survey of recent
D. Zanetti, B. Danev, and S. ˇCapkun, “Physical-layer identification of G. Research, “Global radio frequency identification (RFID) market 2020–J. Jiang, Z. Xu, F. Dang, and J. Wang, “Long-Range ambient LoRa V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, “FM backscatter: M. Zhang, S. Chen, J. Zhao, and W. Gong, “Commodity-level BLE backscatter,” in Proceedings of the 19th Annual International Conference on Mobile Computing and Networking, 2019, pp. 1–15.

Q. Wang, S. Chen, J. Zhao, and W. Gong, “RapidiRider: Efficient WiFi backscatter with uncontrolled ambient signals,” in IEEE INFOCOM 2021-IEEE Conference on Computer Communications. IEEE, 2021, pp. 1–10.

J. F. Ensworth and M. S. Reynolds, “BLE-backscatter: Ultralow-power IoT nodes compatible with Bluetooth 4.0 low energy (BLE) smartphones and tablets,” IEEE Transactions on Microwave Theory and Techniques, vol. 65, no. 9, pp. 3360–3368, 2017.

M. Ziang, S. Chen, J. Zhao, and W. Gong, “Commodity-level BLE backscatter,” in Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, 2021, pp. 402–414.

A. Y. Joo, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota, “FM backscatter: Enabling connected cities and smart fabrics,” in 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 2017, pp. 243–258.

V. Talla, M. HESSAR, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, “LoRa backscatter: Enabling the vision of ubiquitous connectivity,” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 3, pp. 1–24, 2017.

J. Jiang, Z. Xu, F. Dang, and J. Wang, “Long-Range ambient LoRa backscatter with parallel decoding,” in Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, 2021, pp. 684–696.

J. Wang, L. Chang, O. Abari, and S. Keshav, “Are RFID sensing systems ready for the real world?” in Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, 2019, pp. 366–377.

G. Research. “Global radio frequency identification (RFID) market 2020–2026 by component, product type, frequency band, wafer size, application, industry vertical, and region: Trend forecast and growth opportunity,” Tech. Rep. 5019726, Apr. 2020.

GS1, EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Protocol Version 2.1, Jul. 2018.

D. Zanetti, B. Danev, and S. Čapkun, “Physical-layer identification of UHF RFID tags,” in Proceedings of the sixteenth annual international conference on Mobile computing and networking, 2010, pp. 353–364.

Y. Bengio, Learning deep architectures for AI. Now Publishers Inc, 2009.

W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, vol. 234, pp. 1–26, 2017.

J. Smith, A. P. Sample, P. S. Powlode, S. Roy, and A. Namishiv, “A wirelessly-powered platform for sensing and computation,” in International Conference on Ubiquitous Computing. Springer, 2006, pp. 495–506.

N. Kargas, F. Mavromatis, and A. Bletsas, “Fully-coherent reader with commodity SDR for Gen2 FM0 and computational RFID,” IEEE Wireless Communications Letters, vol. 4, no. 6, pp. 617–620, 2015.

L. Rabine, and B. Jiang, “An introduction to hidden markov models,” IEEE ASSP Magazine, vol. 3, no. 1, pp. 4–16, 1986.

G. Forney, “The viterbi algorithm,” Proceedings of the IEEE, vol. 61, no. 3, pp. 268–278, 1973.

“Impinj Speedway RAIN RFID Reader for Solution Development,” [Online]. Available: https://www.impinj.com/products/readers/impinj-speedway-

S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A theory of learning from different domains,” Machine learning, vol. 79, no. 1, pp. 151–175, 2010.

S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, quantization and huffman coding,” arXiv preprint arXiv:1510.00149, 2015.

Y. He, X. Zhang, and J. Sun, “Channel pruning for accelerating very deep neural networks,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1389–1397.

J. Yin, D. Joo, J. Bae, and J. Kim, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4133–4141.

P. Zhang, J. Gummesson, and D. Ganesan, “Blind: A high throughput link layer for backscatter communication,” in Proceedings of the 10th international conference on Mobile systems, applications, and services, 2012, pp. 99–112.

W. Gong, H. Liu, K. Liu, Q. Ma, and Y. Liu, “Exploiting channel diversity for rate adaptation in backscatter communication networks,” in IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, 2016, pp. 1–9.

P. Hu, P. Zhang, and D. Ganesan, “Laissez-faire: Fully asymmetric backscatter communication,” ACM SIGCOMM computer communication review, vol. 45, no. 4, pp. 255–267, 2015.

J. Ou, M. LI, and Y. Zhang, “Come and be served: Parallel decoding for cots RFID tags,” in Proceedings of the 21st annual international conference on mobile computing and networking, 2015, pp. 500–511.

M. Jin, Y. He, X. Meng, Y. Zheng, D. Fang, and X. Chen, “FlipTracer: Practical parallel decoding for backscatter communication,” IEEE/ACM Transactions on Networking, vol. 27, no. 1, pp. 330–343, 2019.

M. Jin, X. He, X. Meng, D. Fang, and X. Chen, “Parallel backscatter in the wild: When burstiness and randomness play with you,” in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 471–485.

Q. Huang, G. Song, W. Wang, H. Dong, J. Zhang, and Q. Zhang, “FreeScatter: enabling concurrent backscatter communication using antenna arrays,” IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7310–7318, 2020.

M. Jin, Y. He, C. Jiang, and Y. Liu, “Parallel backscatter: Channel estimation and beyond,” IEEE/ACM Transactions on Networking, 2021.

T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” IEEE Transactions on Cognitive Communications and Networking, vol. 7, no. 4, pp. 563–595, 2020.

S. Donner, S. Cammerer, J. Hoydis, and S. Ten Brink, “Deep learning based communication over the air,” IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 132–143, 2017.

A. Felix, S. Cammerer, S. Donner, J. Hoydis, and S. Ten Brink, “OFDM-autoencoder for end-to-end learning of communications systems,” in 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, 2018, pp. 1–5.

H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, “Model-driven deep learning for physical layer communications,” IEEE Wireless Communications, vol. 26, no. 5, pp. 77–83, 2019.
SINGI JEONG received the B.S. degree in Computer Engineering from Sungkyunkwan University, Suwon, Republic of Korea, in 2018, where he is currently pursuing the integrated Ph.D. degree with the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea.

His current research interests include ultra low power wireless communication and UWB signal processing.

JAEMIN SHIN received the B.S. degree in Computer Engineering from Sungkyunkwan University, Suwon, Republic of Korea, in 2018, where he is currently pursuing the integrated Ph.D. degree with the Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea.

His current research interests include ultra low battery MIMO communication and 5G RAN communication.

YUSUNG KIM received the B.S. degree in the Department of Computer Science from Kyungpook National University, Daegu, Republic of Korea, in 2002, and the M.S. and Ph.D. degrees in the Department of Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, in 2004 and 2008, respectively. He had been a Senior Engineer at Samsung Electronics from 2008 to 2011. He is currently an Associate Professor with the Department of Software, Sungkyunkwan University, Suwon, Republic of Korea.

His current research interests include intelligent systems and learning-based system optimization.