RAScatter: Achieving Energy-Efficient Backscatter Readers via AI-Assisted Power Adaptation

Kai Huang, Rui
University

***Abstract*—Backscatter communication reduces the battery device's power consumption at the cost of extra RF ene transmitted from backscatter readers. Such extra cost res in extremely low energy efficiency at readers, but is ignored existing systems that always use the highest transmit RF po for maximum goodput. Instead, we envision that the maxim goodput is unnecessary in many practical scenarios, allow adaptation of transmit RF power to the required goodput. In paper, we present *RAScatter*, a new backscatter system of prec adaptive and lightweight power adaptation towards ener efficient backscatter readers. RAScatter learns the entang correlation between backscatter channel conditions, transmit power and goodput by designing a modular neural netwo which decomposes the complex learning task into multiple rela but simplified subtasks. This decomposition avoids redunda in neural networks and eliminates any confusion in training to insufficient training data in low-speed backscatter syste Experiment results over commodity batteryless tags show that RAScatter improves the energy efficiency at backscatter readers by 3.5× and reduces the readers' power consumption in backscatter communication by up to 80%.**

I. Introduction

Backscatter enables wireless communication on batteryless devices, such as RFID tags and energy-harvesting powered sensors [21], [17] that are essential components of Internet of Things (IoT). In backscatter communication, batteryless devices are powered by RF signals transmitted from the reader device. While reducing the batteryless devices' power consumption, such RF power delivery consumes high amounts of extra energy at readers.

Such energy efficiency of backscatter readers is ignored by most of existing backscatter systems [18], [8], [14]. Instead, these systems usually assume unlimited power supply at readers and use the highest RF power at readers for best backscatter goodput [35], [12]. In these cases, through a measurement study that compares different backscatter techniques with commodity wireless radios, our results in Section II show that the backscatter readers' energy efficiency is at most 20% of that on commodity radios, and could be up to 12,000 times lower than high-speed radios such as WiFi.

With the recent technical advances of backscatter systems, we have witnessed the migration of backscatter readers from stationary and wall-powered to portable and battery-powered (e.g., handheld RFID readers, laptops [34] and smartphones [17]). Low energy efficiency of these power-constrained readers, hence, seriously reduces the readers' lifetime and becomes the key barrier of using backscatter technologies in many applications, such as batteryless video streaming [24] and real-time biomedical monitoring [14]. For example, handheld RFID readers with continuous readings have a battery life of <2 hours [5], which could be further shortened on recent long-range [27] and high-throughput [8] backscatter systems.

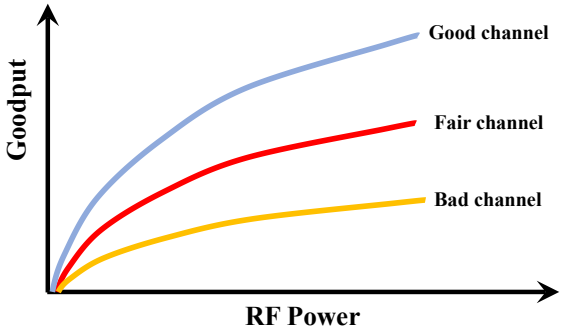


Fig. 1. Correlation between goodput and transmit RF power in different backscatter channel conditions.

Figure 1: Illustration of Correlation between Power, Optimal Rate and Goodput

To tackle with this barrier, we envision that neither the highest backscatter goodput nor the maximum transmit power at the reader is necessary in many practical scenarios. For example, although the maximum data rate for a batteryless RFID tag could be hundreds of kbps [2], 5-10 kbps is sufficient for RFID applications such as activity recognition [11]. Similarly, data rates of most body sensors that monitor humans' glucose level, breath rate and body temperature do not exceed 1 kbps [6]. On the other hand, our experimental studies in Section II show that the batteryless device can transmit data via backscatter communication at reduced goodputs, even with much lower transmit power from the reader. As a result, according to the required goodput, we can adaptively reduce the reader's transmit RF power, but still keep backscatter communication to be operable at all times.

Such power adaptation at backscatter readers, however, is challenging because it is hard for a reader to precisely determine how backscatter goodput is affected by the transmit RF power. As shown in Figure 1, such correlation mainly depends on the condition of backscatter channel[1], as the same transmit RF power could lead to higher goodput in a channel with good condition. As a result, a viable solution is to measure the channel condition at the reader from its received signal. However, this received signal may be highly fluctuating and unreliable, because it is a reflection of the

[1]We refer to the communication channel from batteryless device to reader.

978-1-6654-9641-4/22/$31.00 ©2022 IEEE
DOI 10.1109/IoTDI54339.2022.00016

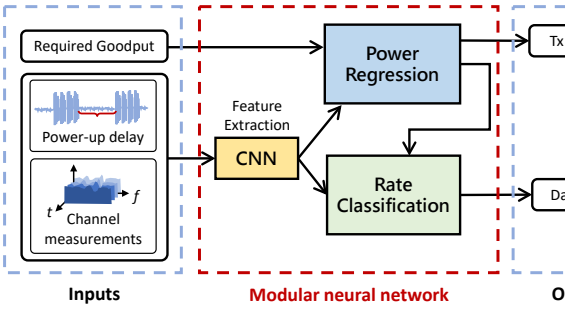


Fig. 2. RAScatter uses a modular neural network to depict the between backscatter channel conditions, transmit RF power and g

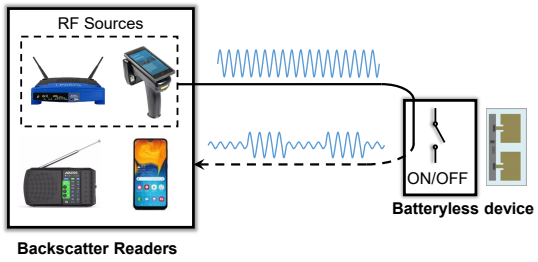


Fig. 3. Overview of backscatter communication

reader's transmitted signal and could have been distorted on the air before it reaches the batteryless device. Using numerical channel metrics such as RSSI [35] or packet loss [12] is highly inaccurate, especially when the backscatter channel changes due to device mobility or environmental dynamics.

In this paper, we present Reader Adaptation in Backscatter (*RAScatter*), a new backscatter system that precisely adapts the transmit RF power in backscatter systems to the required goodput. As shown in Figure 2, RAScatter uses a *modular neural network* at the reader to explicitly represents the channel condition, RF power and goodput in backscatter communication as separate but smaller modules. These modules are then interconnected with respect to their entangled correlations: a convolutional neural network (CNN) is first used to extract features from backscatter channel measurements, which are then used as inputs to two neural networks that individually decide transmit RF power and backscatter data rate. In this way, RAScatter avoids any redundant component from a single monolithic neural network, and hence eliminates confusions in training that are caused by complicated channel dynamics and cause errors in power adaptation.

The major challenge of training the modular neural network is backscatter's low data rate, which limits the available amount of channel measurements as training data[2] and could easily result in overfitting. To address this challenge, we design the modular neural network with respect to the unique characteristics in backscatter communication, so that the neural network could always be trained within the appropriate scope. More specifically, neural network modules in RAScatter are interconnected based on the causal relationship between the backscatter communication parameters that they represent. We use these characteristics to explicitly identify the most prominent features from backscatter channel measurements for training, and also incorporate these characteristics into the neural network's loss function.

In practice, the backscatter channel may continuously vary and the current backscatter goodput may hence deviate from the required goodput. In these cases, RAScatter adaptively probes the channel and uses current channel measurements to refine the neural network models online. Since our design of modular neural network minimizes the neural network's complexity, it ensures fast inference that allows timely adaptation to any instantaneous channel variation.

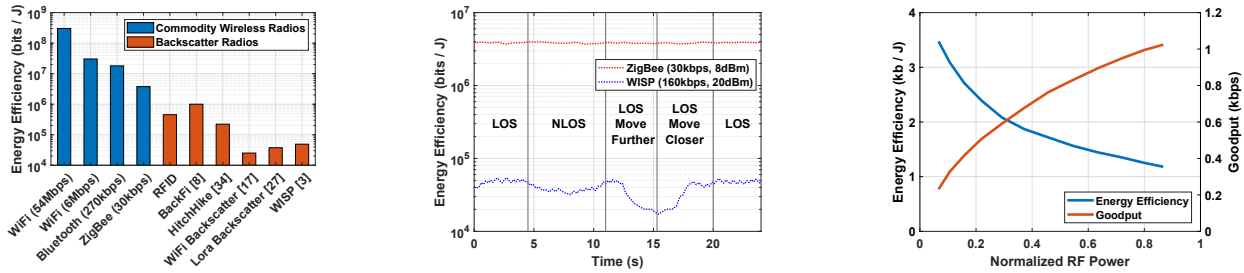
Our detailed contributions are as follows:

- We depict the entangled correlation between backscatter channel condition, RF power and goodput, by designing a modular neural network with the minimum complexity and redundancy. This design ensures unbiased training with limited data.
- We can timely reflect changes of the backscatter channel by adaptively refining the neural network models online. This online refinement also allows flexible run-time changes on the required goodput to adapt to different application requirements.
- We restrain all RAScatter's operations at backscatter readers, and require neither extra computation nor hardware change on batteryless devices. It can hence be adopted to most commodity batteryless devices.

We implemented RAScatter by using a USRP N210 SDR as the backscatter reader, which communicates with WISP 5.1 batteryless tags [3]. We have evaluated the performance of RAScatter under different backscatter channel conditions in both indoor and outdoor scenarios. From our experiment results, we have the following conclusions:

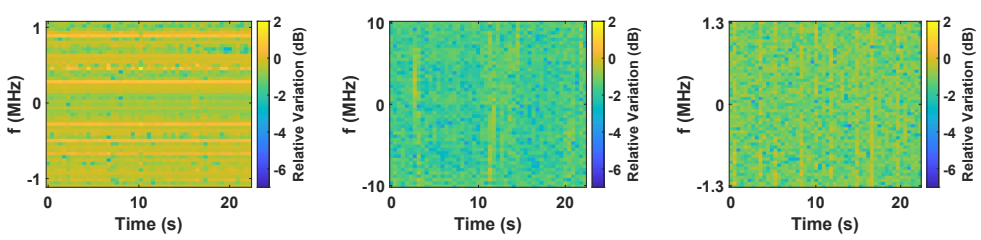
- RAScatter is *accurate*. Compared to traditional monolithic neural networks, the accuracy of power adaptation in RAScatter is more than 95% under different channel conditions and goodput objectives. Such accurate power adaptation improves the backscatter reader's energy efficiency by up to 3.5×, and reduces the readers' power consumption in backscatter communication by up to 80%.
- RAScatter is *adaptive*. RAScatter well adapts to different application scenarios with heterogeneous goodput objectives, communication distances, device mobility and surrounding objects. RAScatter retains its accuracy of power adaptation and improvement of energy efficiency in these scenarios, and exhibits 2× better resilience to channel variations and disturbances.
- RAScatter is *lightweight*. The energy consumed by neural network inference is at most <1.4% of RF energy consumption, and the inference latency is within 1ms. Hence, it is widely applicable to any portable backscatter reader with severe power constraints.

[2]Wireless channel measurements are obtained by probing the channel with special packets.



(a) Comparison with commodity radios (b) Energy efficiency with device mobility (c) Correlation with backscatter goodput

Fig. 4. Energy efficiency of backscatter readers



(a) Backscatter (b) WiFi (c) ZigBee

Fig. 5. Variations of channel frequency response over time

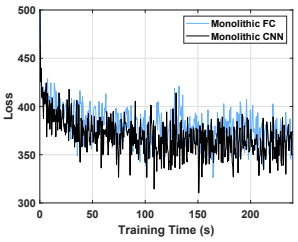


Fig. 6. Training loss over fully-connected (FC) and convolutional neural networks

II. Background & Motivation

To better understand the design of RAScatter, we first introduce the basics of backscatter communication, and demonstrate the low energy efficiency of backscatter readers with a measurement study. We then motivate our design by showing the ineffectiveness of using monolithic neural networks for backscatter power adaptation.

A. Backscatter Communication

Power-up delay measurement

The wisp tag is placed 0.8m away from the reader. The measurements are obtained by directly attaching the oscilloscope's probe to the wisp's buffer.

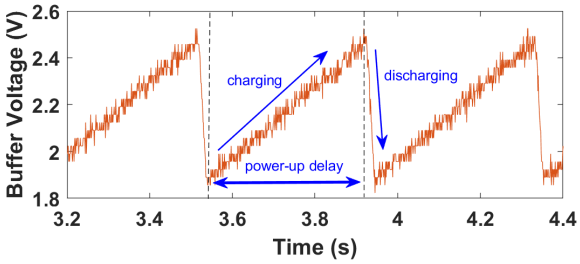


Fig. 7. Power-up delay on a batteryless tag

A batteryless device is usually equipped with an MCU and sensors. Then, the amount of RF power may be insufficient for the device's continuous operation, and duty cycling must be adopted. In each cycle, the device accumulates enough RF energy before being powered up, and such power-up delay depends on the condition of backscatter channel. For example in Figure 7, a WISP tag has a power-up delay of up to 0.2s to support its MSP430 MCU and ADXL362 IMU sensor to operate for 20ms, limiting its average data rate to 2 kbps. Such low data rate prevents collecting a sufficient amount of channel measurements for neural network training, and motivates us to adopt a modular neural network instead.

B. The Readers' Energy Efficiency

We compare the readers' energy efficiency in different backscatter systems with that of commodity wireless radios, which are all configured to operate at their maximum transmit power. As shown in Figure 4(a), the backscatter reader's energy efficiency is at least 80% lower. When the backscatter data rate is low (e.g., $<$ 20 kbps), such energy efficiency will further drop by 50 times and is more than 12,000 times lower when compared with high-speed radios such as WiFi.

Further, Figure 4(b) shows that the backscatter reader's energy efficiency is largely affected by the reader's mobility, and drops by 35% when the distance between the reader and the batteryless tag increases to 30cm. Hence, in practical applications when backscatter readers are battery-powered mobile devices, their energy efficiency will be very limited.

At the same time, as shown in Figure 7, the maximum transmit RF power at the reader helps reduce the power-up delay at the batteryless device, but is not a necessary condition for backscatter communication to be operable. For example, Figure 4(c) shows that a WISP tag can transmit data even when the reader's transmit power drops to 20% of the maximum, and the reader's energy efficiency actually increases with lower goodputs. These experiment results verify that when the required backscatter goodput is lower than the maximum, we can adaptively decide the reader's transmit RF power to

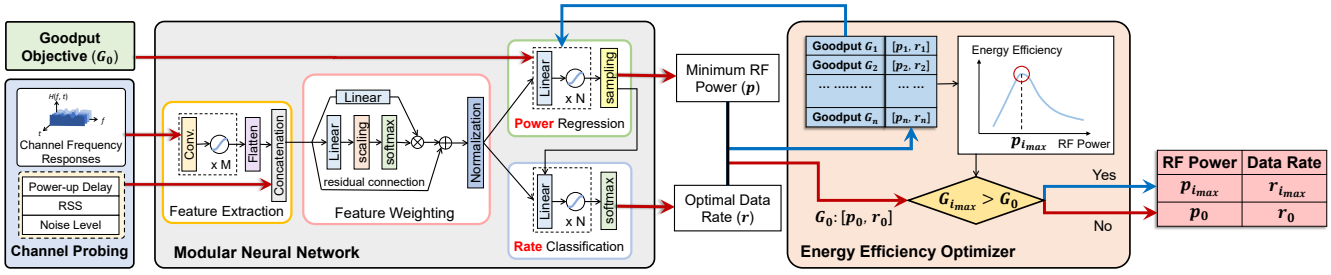


Fig. 8. RAScatter system overview

achieve the required goodput with better energy efficiency, without interrupting the backscatter communication.

C. Power Adaption with Neural Networks

The backscattered signal is a reflection of transmitted RF signal that also subjects to channel path loss and distortion. Backscatter communication, hence, is much more susceptible to wireless channel fluctuations. To demonstrate this, we experimentally compared the variation of backscatter channel with that of WiFi and ZigBee channels[3]. As shown in Figure 5, variation of backscatter channel's frequency response over time is >8 dB higher than that of WiFi and ZigBee radios.

Due to such high channel variation, it is difficult to precisely determine how the backscatter goodput is affected by transmit RF power. A solution is to use a *monolithic* neural network, which takes the backscatter channel condition as input and the corresponding goodput and transmit power as output labels in training. This solution, however, is ineffective due to the following reasons. First, when the channel quickly varies, backscatter channel measurements may lag behind and not correspond to the output labels being used. Second, backscatter's low data rate results in limited training data. For example, at 1 kbps, it takes >80 hours to collect 10k channel measurements, which are the minimum training data size required in monolithic neural networks. Hence, the neural network training, as shown in Figure 6, usually retains high loss and even does not converge. Increasing the network complexity helps, but incurs high computing cost at portable backscatter readers with power constraints.

III. System Overview

As shown in Figure 8, RAScatter addresses these limitations by using a *modular neural network* [10], which decomposes a complex learning task into multiple related but simplified subtasks. Existing designs are based on Mixture of Experts [15] that divide the learning task into multiple *identical* subtasks, but cannot remove the ambiguity in correlation between backscatter channel conditions, RF power and goodput.

Instead, RAScatter constructs the modular neural network based on the unique characteristics of backscatter communication, to avoid training bias and ambiguity due to insufficient amount of training data. After having known how the backscatter goodput is determined by the channel condition and transmit RF power, an Energy Efficiency Optimizer will further improve the energy efficiency of achieving the required goodput, by opportunistically increasing the transmit RF power and reducing the power-up delay.

A. Channel Probing

RAScatter obtains backscatter channel measurements through channel probing, which pings the batteryless device and calculates channel's frequency response from the received reply. To ensure accuracy, the highest transmit RF power and data rate are used. In addition, numeric channel metrics, including the received signal strength, noise level and power-up delay, are also included.

We collect data under different channel conditions for training. Under each channel condition, we increase transmit RF power from 0 to the maximum in small levels, and maximize the goodput in each level by finding the optimal data rate. Each pair of transmit RF power and optimal data rate are then used as the output labels, and the correspondingly achieved goodput is used as the input objective.

B. Modular Neural Network

RAScatter first extracts features from backscatter channel measurements, and then applies these features to two sub-neural networks (NNs) that represent transmit RF power and backscatter data rate, respectively. The first sub-NN (*Power Sub-NN*) correlates channel conditions and goodput objective with transmit RF power via regression, and the second sub-NN (*Rate Sub-NN*) uses the output of Power Sub-NN to decide the optimal data rate under the current channel condition. Such interconnection between sub-NNs builds on the fact that, the transmit RF power required to achieve the goodput can only be minimized with the optimal data rate. More details about sub-NN construction are in Section IV-A.

Both the accuracy and overhead of neural network operations depend on its complexity. A complicated neural network achieves higher accuracy but also increases the computation overhead. In RAScatter, we extract channel features with two $2 \times 3 \times 2$ convolutional layers, and each sub-NN then contains three 10-neuron fully-connected layers. We will further show how the neural network's accuracy and overhead are affected by its complexity in Section VIII-A.

To improve the neural network's training efficiency and inference accuracy, RAScatter uses a learning-based weighting procedure to adaptively enhance the most prominent features,

[3]We use USRP and XBee S2C [4] as WiFi and ZigBee transceivers.

before applying them as inputs to the sub-NNs. The features' weights are expected to reflect their impacts on the sub-NNs, which feedback such impacts through their residual connections. Details about such weighting procedure will be described in Section IV-B.

To retain the accuracy of power adaptation over time, RAScatter also probes the channel online and uses knowledge about real-time channel condition changes to refine the neural network model, when the current backscatter goodput deviates from the required goodput. Details of such online model refinement are in Section V.

C. Energy Efficiency Optimizer

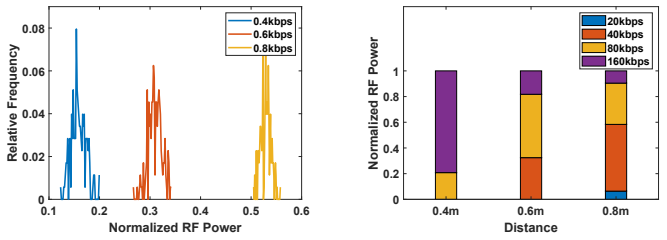
Using the neural network's output of the minimum RF power (p_0) and optimal data rate (r_0) may not always maximize the energy efficiency when achieving the required goodput (G_0), because it incurs long power-up delay and reduces the time for data transmission. RAScatter, instead, uses an *Energy Efficiency Optimizer* to seek for opportunities that further improve energy efficiency. This optimizer applies a series of pre-defined goodput values (G_1, G_2,..., G_n) as input objectives to the neural network, and identifies $i_{max} = \arg\max_i G_i/p_i$ as the highest energy efficiency. If $G_{i_{max}} > G_0$, the corresponding $[p_{i_{max}}, r_{i_{max}}]$ are used for better energy efficiency. Details are in Section VI.

IV. Modular Neural Network

In this section, we provide details of our design of the modular neural network.

A. Sub-NN Construction

In the modular neural network, the Power sub-NN uses regression to decide the minimum RF power. It is subjective to various practical factors such as channel fluctuations and hardware imperfection, which may make the collected training data deviate from the ground truth. For example in Figure 9(a) where we measure the minimum RF power to achieve the required goodput for 180 times, large variance is observed at all goodput levels. To eliminate such ambiguity in training data, the Power sub-NN probabilistically learns the transmit RF power as a Gaussian distribution $\mathcal{N}(\mu_P, \sigma_P^2)$, and its output in inference is then sampled from this distribution.



(a) Minimum RF power to achieve the required goodput

(b) Optimal data rate with different transmit RF power

Fig. 9. Correlation between goodput, transmit RF power and data rate on a WISP tag

Backscatter systems usually shift their data rates between discrete levels, and the Rate sub-NN learns to identify the optimal data rate through classification. However, our experiment results in Figure 9(b) show that the optimal data rate to achieve the required goodput largely varies with the transmit RF power, and such variation could be up to 8× when the tag is far away from the reader. Hence, it is difficult for the Rate sub-NN to decide such optimal data rate by itself.

In RAScatter, instead of separately training the two sub-NNs, we use the Power sub-NN's output as the Rate sub-NN's input as shown in Figure 8, and the two sub-NNs are then jointly trained by minimizing both the relative error loss for power regression and cross-entropy loss for rate classification. The joint loss function is expressed as

$$\mathcal{L}_{joint} = \mathcal{L}_{power} + \lambda \mathcal{L}_{rate} = \left| \frac{\overline{p} - p}{p} \right| - \lambda \cdot \sum\nolimits_i r_i \log(\overline{r}_i),$$

where p is Power sub-NN's output in training, r_i is Rate sub-NN's output in training as the probability for the i-th data rate to be optimal, and $\overline{p}, \overline{r}_i$ are the corresponding labels. λ is a hyperparameter that controls the contribution of each individual loss. In this way, the power sub-NN's training will also receive the feedback from the Rate sub-NN, which can enhance its convergence and power adaptation accuracy.

B. Feature Weighting before Sub-NNs

The features being extracted from the backscatter channel measurements are highly variant, due to the dynamic conditions of backscatter channel. Some features, as shown in Figure 10(a), will be more prominent and have higher impact on sub-NNs than others. The sub-NNs, however, are unable to identify the prominence of these features by themselves with an insufficient amount of data, and their accuracy will be affected by those features with low prominence.

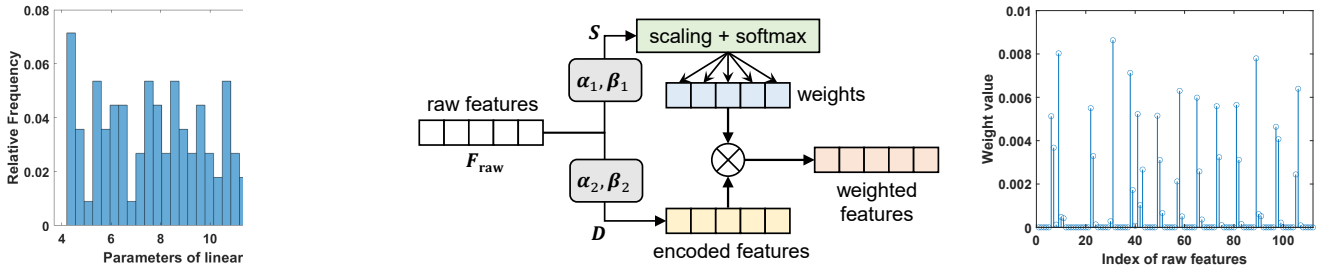
Instead, RAScatter explicitly identifies the features' prominence and applies such prominence as weights to raw features, before using these features as sub-NN inputs. Being different from current weighting schemes [28] that focus on information equalization, we aim to further enhance the impacts of those prominent features on the sub-NNs, so as to facilitate accurate and timely inference. As shown in Figure 10(b), given the raw features $\mathbf{F}_{\text{raw}}$, two trainable linear projectors $\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, i = 1, 2\}$ are used to encode the raw features into the disentangled representation $\mathbf{D}$ and the corresponding score vector $\mathbf{S}$ as

$$\mathbf{D} = {\boldsymbol{\alpha}_1}^T \mathbf{F}_{\text{raw}} + \boldsymbol{\beta}_1, \quad \mathbf{S} = {\boldsymbol{\alpha}_2}^T \mathbf{F}_{\text{raw}} + \boldsymbol{\beta}_2,$$

and these projectors are trained by the feedback from sub-NN loss through a residual connection[4]. Then, $\mathbf{S}$ is used to calculate the feature weights via scaling[5] and softmax operations, and these weights are applied to the encoded features $\mathbf{D}$. As shown in Figure 10(c), the weights generated from features' impacts in Figure 10(a) are highly skewed, indicating that only few features are enhanced.

[4]Training is conducted by recursively propagating gradient-based feedback backwards through the neural network.

[5]Scaling is needed in advance to prevent exponential saturation in training.



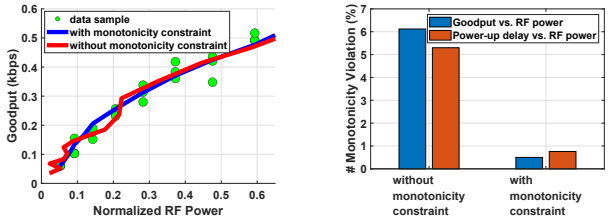
(a) Impacts of different featu (b) Workflow of feature weighting (c) Generated feature weights

Fig. 10. Feature weighting before sub-NNs in RAScatter

By using 1,000 backscatter channel measurements (train:test = 4:1), we have verified that the feature weights could efficiently prevent the sub-NN training from being interfered by less prominent features, so as to reduce the inference error of power regression (in Power sub-NN) and rate classification (in Rate sub-NN) by 6% and 5%, respectively.

C. Adding Monotonicity Constraints

Neural network inference should be always consistent with the naive monotonicity characteristics in backscatter systems: first, higher RF power is needed to achieve higher goodput; second, to achieve the same goodput, higher RF power is needed when the power-up delay is longer. However, such monotonicity may be violated due to overfitting in training, leading to inaccurate power adaptation. For example in Figure 11(a), the Power sub-NN sometimes outputs lower RF power even if higher goodput is required, and Figure 11(b) further shows that such violations frequently happen in neural network inference: 6.1% and 5.3% violations on two monotonicity characteristics are observed, respectively.



(a) Violation of monotonicity (b) Reduction of violations

Fig. 11. Adding monotonicity constraints

We explicitly makes the neural network to learn the monotonicity through training, by adding monotonicity constraints to sub-NNs' loss. We will first check if the monotonicity characteristics are being violated in training, by comparing sub-NNs' inputs and outputs in each training epoch via differentiation. If violation is found, a penalty will be added to sub-NNs' joint loss. Such penalty is separately constructed for the two monotonicity characteristics as

$$\mathcal{P}_1 = -\lambda_1 \mathscr{D}_{\text{power, goodput}}, \quad \mathcal{P}_2 = -\lambda_2 \mathscr{D}_{\text{power, power-up delay}},$$

where $\mathscr{D}_{u,v}$ denotes the differentiation of output u over input v, and λ_1, λ_2 are hyperparameters for scaling. Since the penalty will always be positive if the monotonicity is violated, the sub-NN training will be instructed to avoid violation of monotonicity. As shown in Figure 11(b), adding these constraints could significantly suppress the occurrences of these violations to the minimum (<1%).

V. Online Model Refinement

As shown in Section 2.3, the conditions of a backscatter channel is highly dynamic and fluctuating over time. To ensure the accuracy of power adaptation, RAScatter uses online learning [25] to refine the neural network models with real-time backscatter channel measurements, when the current backscatter goodput deviates from the required goodput. This online refinement, on the other hand, also allows flexible changes on the required goodput at run-time.

The major challenge of online learning, however, is the small amount of new training data that can be collected at real-time. For example, online learning for each refinement needs at least 32 backscatter channel measurements [7], but collecting these measurements would take at least several seconds in a low-speed backscatter system such as WiFi Backscatter [17]. To address this challenge, in RAScatter we design a new loss function for online learning, to make sure that online learning aims to achieve the required goodput with the minimum transmit RF power.

For current time τ, this loss function is computed over a sliding window with size T as

$$\mathcal{L}_{online} = \sum_{t=\tau}^{\tau+T} (E_t + \lambda \cdot P_t),$$

where λ is a scaling factor, P_t is the current transmit RF power and E_t measures the difference between the current goodput (G_t) and the required goodput (G_0). Intuitively, E_t can be defined as $|G_t - G_0|/G_0$, which is however, not related to any sub-NN's output. Instead, we incorporate the logarithmic likelihood of the neural network's prediction to define E_t. Given the neural network's predicted probability (R_t) of the optimal rate and the predicted transmit RF power P_t,

$$E_t = \left[\log(R_t) + \log\left(\frac{1}{\sqrt{2\pi}\sigma_P} e^{-\frac{(P_t-\mu_P)^2}{2\sigma_P^2}}\right)\right] \cdot \frac{|G_t - G_0|}{G_0}$$

where (μ_P, σ_P) are the parameters of Gaussian distribution learned by the Power sub-NN as described in Section IV-A.

To demonstrate the effectiveness of this online learning, we test the neural network model trained offline in a new

environment, where metal objects are placed nearby to change the backscatter channel. As shown in Figure 12, online learning can progressively approach to the required goodput, and improve the energy efficiency accordingly. More detailed evaluations on online refinement are in Section VIII-B.

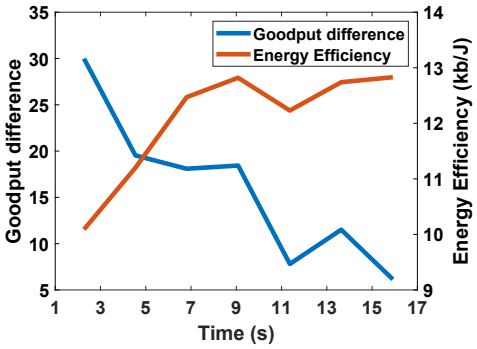


Fig. 12. Effectiveness of online model refinement

VI. Energy Efficiency Optimizer

The Backscatter goodput G increases with RF power P as

$$G = k_1(\sqrt{P} - V_{max})\log(1 + k_2 P), \quad (1)$$

where $V_{\max}$ is batteryless device's maximum electrical voltage from energy harvesting, and k_1 and k_2 are system-specific constants. Proof of Eq. (1) is in the Appendix.

Eq. (1) is illustrated by Figure 13, and shows that the energy efficiency of backscatter communication (indicated by G/P) generally drops when the goodput improves with higher RF power, because the majority of the harvested RF energy will be used to increase the duty cycle. On the other hand, when the transmit RF power is too low, the energy efficiency will also be impaired because of the long power-up delay and the short time for data transmission. Hence, if the required goodput (G_0) in RAScatter is too low, we could use the transmit RF power higher than the output of modular neural network (p_0), to further improve the energy efficiency[6].

As shown in Figure 13, G/P has and only has one turning point over P when $P = p_{\max}$ to reach the maximum energy efficiency. However, it is hard to calculate $p_{\max}$ from Eq. (1) due to the difficulty of estimating the constants k_1 and k_2. Instead, we first decide if $p_0 < p_{\max}$ by comparing the energy efficiency between G_0 and $G_0 + \Delta_G$, and then find $p_{\max}$ via numerical iteration from p_0 if $p_0 < p_{\max}$.

Details of such numerical iteration are in Algorithm 1. In each iteration, we first decide the most appropriate goodput objective $\overline{g}_{\max}$ that maximizes the energy efficiency, and then obtain the transmit RF power to achieve $\overline{g}_{\max}$ from the neural network output. Since the logarithm component in Eq. (1) increases much slower than the other polynomial component, Eq. (1) can be approximated by $\overline{G} = a_0\sqrt{P} + a_1$, and its coefficients a_0 and a_1 can be computed for every given pair of (S_l, S_u). Afterwards, $\overline{g}_{\max}$ is identified when G/P is

[6]The achieved backscatter goodput, in this case, will also be increased to be higher than G_0.

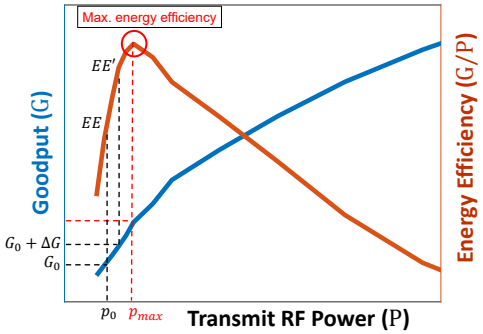


Fig. 13. Energy efficiency optimizer

maximized, and the estimation of $p_{\max}$ (denoted as $\overline{p}_{\max}$ in the current iteration) will be produced as $N_{RAS}(\overline{g}_{\max})$.

Algorithm 1 Energy Efficiency Optimizer

Input: G_0: the required goodput; $N_{RAS}(\cdot)$: RAScatter's neural network

Output: $(p_{\max}, r_{\max})$: the transmit RF power and data rate that maximize the energy efficiency.

$G_1 \leftarrow G_0 + \Delta G$
$(p_0, r_0) \leftarrow N_{RAS}(G_0), \quad (p_1, r_1) \leftarrow N_{RAS}(G_1)$
$EE \leftarrow G_0/p_0, EE' \leftarrow G_1/p_1$ //$p_0 < p_{\max}$ if $EE' > EE$
if $EE' > EE$ **then**
  $S_l \leftarrow (p_0, G_0), \quad S_u \leftarrow (p_1, G_1),$
  $\overline{p}_{\text{oldmax}} \leftarrow 0, \overline{p}_{\max} \leftarrow p_1, \Delta_p \leftarrow |\overline{p}_{\max} - \overline{p}_{\text{oldmax}}|/\overline{p}_{\max}$
  while $\Delta_p > \theta$ **do**
    Decide (a_0, a_1) in $G = a_0\sqrt{P} + a_1$ with respect to (S_l, S_u)
    $\overline{g}_{\max} = \arg\max_G G/P$
    $S_l \leftarrow S_u, \quad S_u \leftarrow (\overline{p}_{\max}, \overline{g}_{\max}) = N_{RAS}(\overline{g}_{\max})$
    $\Delta_p \leftarrow |\overline{p}_{\max} - \overline{p}_{\text{oldmax}}|/\overline{p}_{\max}$
  end while
  $(p_{\max}, r_{\max}) \leftarrow (\overline{p}_{\max}, \overline{r}_{\max})$
end if

In practice, Algorithm 1 can converge within a few iterations (see Section VIII-B). Since only one neural network inference is needed in each iteration, it incurs very low computation cost. Its performance mainly depends on 1) the choice of Δ_G to decide where the iteration is initiated; and 2) the choice of θ to decide when the iteration ends. We will experimentally investigate the impact of these parameters in Section VIII-B.

VII. Implementation

We implemented RAScatter in the 915MHz UHF band with WISP 5.1 batteryless tags. A USRP N210 with two SBX-40 RF daughterboards is used as the reader. The two daughterboards are connected to two RFMAX S9028PCRJ circularly polarized antennas for Tx and Rx. The analog Tx and Rx gains are 25dB and -10dB, respectively.

A. Backscatter Device Implementation

A WISP batteryless tag follows the EPC Gen 2 RFID protocol to backscatter data obtained from its on-board sensor

[32], but it lacks retransmission and rate selection mechan that are well-implemented in commodity wireless rad To tackle with possible data transmission failures, we b the tag's 32-bit sensory data in its non-volatile memory,

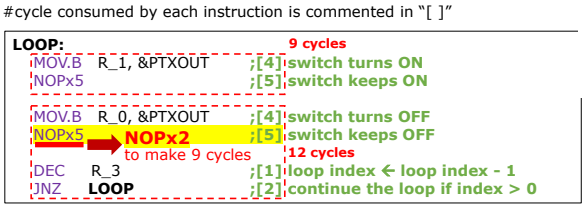


Fig. 14. Generating clock signals on a WISP tag

The major challenge of implementing rate selection on a WISP tag is that its RF switch is not connected to its MCU's clock output, and the tag can only backscatter with a fixed data rate of 160 kbps. Instead, we modify the tag's firmware to generate clock signals for different data rates. For example, generating a 320kHz clock signal needs to toggle the RF switch every 9 cycles. However, as shown in Figure 14, loop maintenance also consumes 3 cycles, and we avoid these

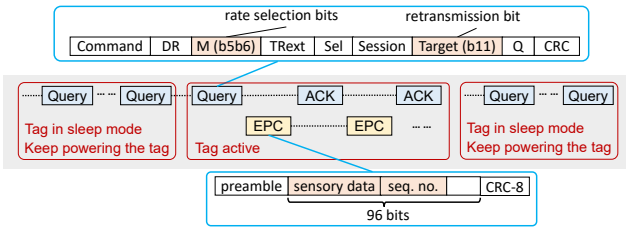


Fig. 15. Transmission protocol and packet formats

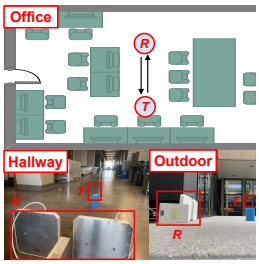
B. Reader Implementation

As shown in Figure 15, the USRP reader follows the EPC Gen 2 RFID protocol to power up and communicate with the WISP tag. The reader continuously sends queries to the tag every 62.5us, and controls the transmit RF power by adjusting USRP's Tx baseband gain. ASK modulation is used in all packets, and the reader appoints the tag's data rate among 160/80/40/20 kbps. A timeout of 62.5us is set to detect transmission failures. With 3 consecutive timeouts, the reader requests retransmission through the query's retransmission bit.

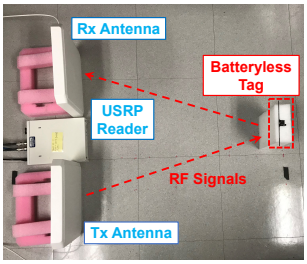
C. Neural Network Implementation

We construct and train the modular neural network using TensorFlow, and deploy the trained model with TensorFlow C API on a Raspberry Pi 3 microcontroller that is connected to the USRP reader. Having received the outputs from RAScatter, the reader applies the transmit RF power through its baseband gain controller, and apply the optimal data rate through the query being sent to the tag.

[7]On other backscatter tags with these mechanisms, RAScatter does not require any hardware or firmware modification.



(a) Deployment (R: reader, T: tag)



(b) Device layout

Fig. 16. Performance evaluation settings

VIII. Performance Evaluation

As shown in Figure 16, we use our implementation described in Section VII to evaluate RAScatter in both indoor (office and hallway) and outdoor scenarios, where most backscatter applications such as smart locks [29], indoor localization [22], occupation monitoring [20] and gesture detection [16] are being operated. 2,114 backscatter channel measurements are collected for offline neural network training, by placing the tag in different locations and orientations.

We use bits per Joule (BpJ) as the metric to evaluate the energy efficiency of backscatter communication. Such energy efficiency is measured as the ratio between the amount of backscattered sensory data and the amount of RF energy (including probing) consumed by the reader, and is being averaged over the tag's all duty cycles. We also measure the transmission delay of each piece of sensory data. Note that, due to possible data transmission failures and retransmissions, the goodput in each duty cycle and the delay of transmitting different data pieces could be different.

In each experiment, to evaluate the accurcacy of RAScatter's power adaptation, we first decide the optimal energy efficiency with respect to the required goodput objective, by exhaustively examining all the possible transmit RF power levels and data rates. Such optimal energy efficiency is then compared with the energy efficiency achieved by RAScatter, to see how small the difference between the two is.

In existing backscatter systems using the highest RF power, different rate adaptation schemes are adopted to maximize the goodput by choosing the best data rate. Hence, we compare RAScatter with the following rate adaptation schemes:

- **BLINK [35]**: Rate adaptation is instructed by an offline trained map that correlates the optimal data rate with RSSI and packet loss.
- **MobiRate [12]**: The optimal data rate is decided based on past history of data transmissions and the reader's empirical estimation of tag's mobility.

A. Impact of Neural Network Complexity

We first investigate how RAScatter's accuracy and overhead of power adaptation are affected by the complexity of neural network, with a fixed goodput objective of 63kbps. We then vary the numbers of $2 \times 3 \times 2$ convolutional layers (# CONV) for feature extraction[8] and 10-neuron fully connected layers

[8]For simplicity, we fix the number of neurons in each layer.

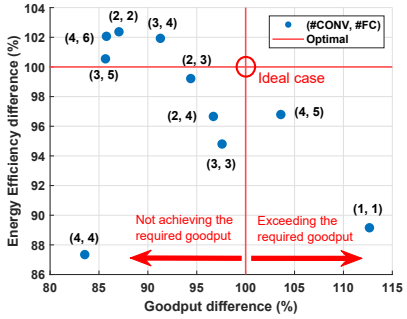
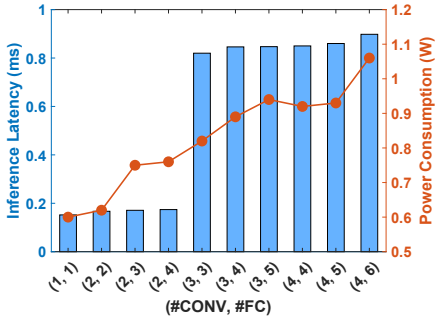
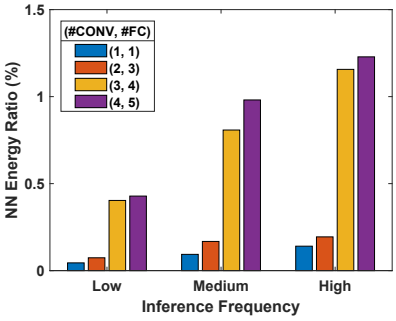


Fig. 17. Inference accuracy with different neural network complexities

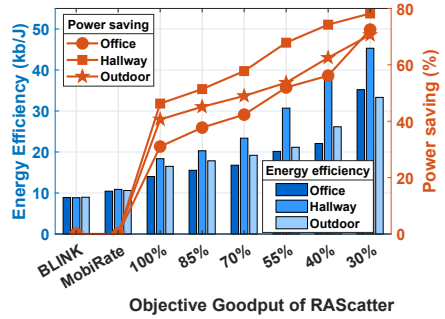


(a) Latency and power consumption

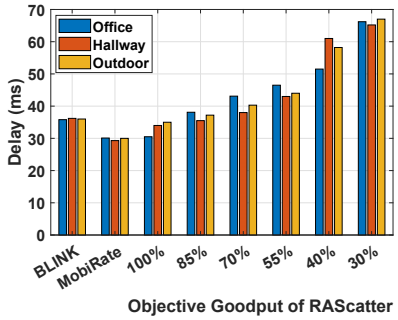


(b) Energy consumption

Fig. 18. Overhead of neural network inference



(a) Energy efficiency



(b) Data transmission delay

Fig. 19. Energy efficiency under different goodput objectives

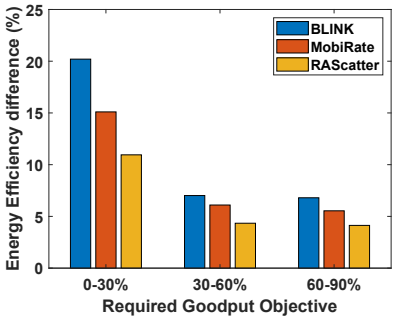


Fig. 20. Accuracy of power adaptation under different goodput objectives

(# FC) for sub-NN construction.

Ideally, goodput achieved by RAScatter should exactly match the goodput objective, achieving the optimal energy efficiency at the same time. However in practice, while the accuracy of power adaptation generally increases when the neural network becomes more complicated, Figure 17 shows that high complexity sometimes also results in overfitting and hence low accuracy. In particular, since the optimal energy efficiency relates to the goodput objective, RAScatter's achieved energy efficiency could possibly exceed 100% when compared with the optimal energy efficiency, if the goodput objective is not achieved. Based on these results, we use (#CONV, #FC)=(2,3) in all the following evaluations. On the other hand, Figure 18 shows that RAScatter's latency of neural network inference on Raspberry Pi 3 is generally lower than 1ms, and the energy consumption of such inference is $<$1.4% of RF energy consumption, even with the highest inference frequency (10 times per second). Hence, RAScatter is lightweight and affordable on most portable backscatter readers.

B. Performance of Power Adaptation

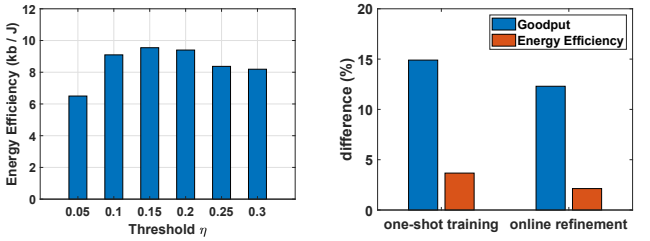
To evaluate RAScatter's performance of power adaptation, we use the goodput achieved by MobiRate with the highest transmit RF power as the baseline, and adopt different percentages of this baseline as the goodput objectives for RAScatter. In all experiments, we vary the communication distance between 20cm and 120cm to produce different channel conditions, but the tag remains stationary in each experiment.

Figure 19(a) shows that the reader's energy efficiency increases when the goodput objective drops. When the goodput objective drops to 30% of the baseline, RAScatter can achieve 3.5$\times$ improvement on the energy efficiency in the indoor office scenario, which is equivalent to reducing the reader's power consumption by 71%. In hallway and outdoor scenarios, RAScatter performs even better due to less multipath and irrelevant wireless signal interference, and the amount of power saving can reach 80% in the hallway scenario. Meanwhile, Figure 19(b) shows that, since using lower transmit RF power incurs more data transmission failures, RAScatter also incurs moderate increase of data transmission delay. However, it is shown that such amount of delay increase can be effectively restrained to $<$10ms if the goodput objective is $>$50%. In practice, these results allow users to flexibly balance between energy efficiency and timeliness of backscatter communication, based on the application requirements.

Based on these results, we conduct the rest of experiments in this section in the indoor office scenario, which is the most difficult scenario for power adaptation and its results can hence be considered as the baseline of RAScatter's performance.

Further, we evaluate RAScatter's accuracy of power adaptation with respect to the optimal energy efficiency. For cases of BLINK and MobiRate in these experiments, we manually reduce the reader's RF power from the maximum to exactly achieve the goodput objective. Results in Figure 20 show that, the energy efficiency achieved by RAScatter is very close to the optimum. Even when the goodput objective is

very low (<20%) and the channel becomes more dynamic due to the low RF power being used, its deviation from optimum can be restrained within 10%. This means that, for most goodput objectives, RAScatter can achieve near-optimal energy efficiency at the backscatter reader.

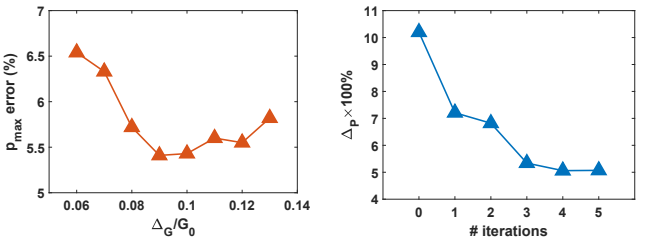


(a) Different thresholds (b) Accuracy of online refinement

Fig. 21. Performance of online model refinement

Performance of online model refinement. As described in Section V, online model refinement is only conducted when the current goodput sufficiently deviates from the required goodput objective. Figure 21(a) shows that, when such threshold (η) of deviation that triggers online model refinement varies from 5% to 20%, the neural network model is refined more frequently, leading to another 40% improvement of the energy efficiency. On the other hand, higher thresholds also reduce the sensitivity to channel variations and result in slight drop of energy efficiency. Based on these results, we use η=15% in the following evaluations.

Further, after the neural network has been trained offline, we test the system in a new untrained environment with densely placed metal objects nearby, and then evaluate the performance of online refinement. Figure 21(b) shows that in this case, online refinement well adapts to the new environment and achieves lower error in power adaptation.



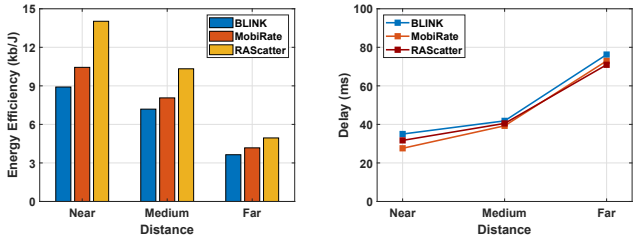
(a) Choice of Δ_G (b) Number of iterations

Fig. 22. Performance of energy efficiency optimizer

Performance of energy efficiency optimizer. As described in Section VI, its performance depends on the parameters in numerical iteration. As shown in Figure 22(a), if Δ_G/G_0 is too high or too low, the iteration's initiation will be more susceptible to the random error from neural network prediction, leading to lower accuracy. On the other hand, as shown in Figure 22(b), the iteration usually converges within 3 rounds, and more iterations produce little improvement.

C. Impact of Communication Distance

To evaluate RAScatter's performance over different communication distances, we set the goodput objective to be 90% of baseline, and place the tag at near (20∼50cm), medium (50∼80cm), far (80∼120cm) distances from the reader. As shown in Figure 23, when the tag moves far away from the reader, the backscatter signal becomes weaker, leading to lower energy efficiency and longer transmission delay. However, while achieving the similar delay with BLINK and MobiRate, RAScatter achieves >50% improvement of energy efficiency with the near distance, and 30% improvement with the medium distance. Such benefit reduces to 30% and 20% at far distance, respecitvely, since higher transmit RF power is required to maintain the same goodput.



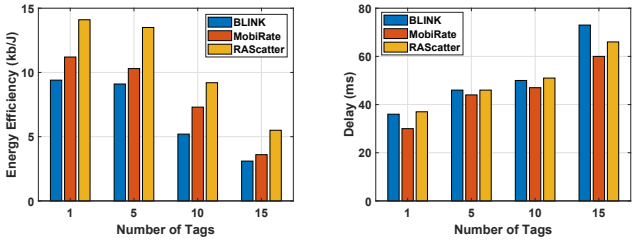
(a) Energy efficiency (b) Data transmission delay

Fig. 23. Impact of communication distance

D. Performance with Multiple Tags

We further evaluate RAScatter's performance of power adaptation when a reader communicates with multiple batteryless tags. We follow the EPC Gen 2 RFID standard [2], and let the reader sequentially communicate with each tag. In this setting, nearby inactive tags could affect the backscatter communication on active tags, since they could be considered as metal objects that create extra interference.

In our experiments, we place different number of tags covering near and medium range from the reader. The goodput objective is set to be 90%. As shown in Figure 24, the reader's energy efficiency in RAScatter is at least 50% higher than that in BLINK and MobiRate. In partciular, when the reader communicates with 15 tags, the advantage of RAScatter enlarges to >100%, because RAScatter can promptly adapt to the complicated channel condition due to mutual interference among co-located tags. At the same time, when the number of tags increases, the increase of data transmission delay is always within 10%.



(a) Energy efficiency (b) Data transmission delay

Fig. 24. Impact of multiple tags

E. Impact of Device Mobility

In practical scenarios, the batteryless device is usually attached to other bigger objects (e.g. human bodies, boxes, tiny robots, etc) and continuously move. Due to its limited throughput, most of its application scenarios only involve moderate mobility such as sleep monitoring [13] and livestock

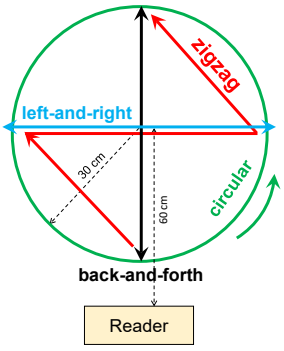
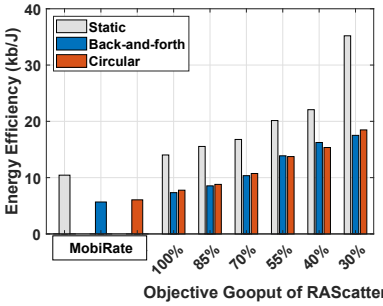
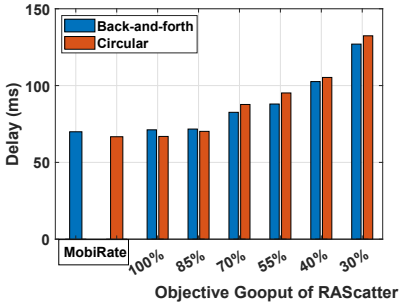


Fig. 25. Mobility patterns



(a) Energy efficiency

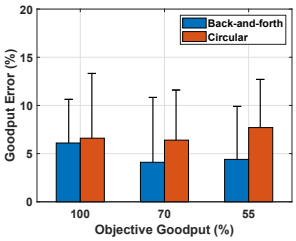


(b) Data transmission delay

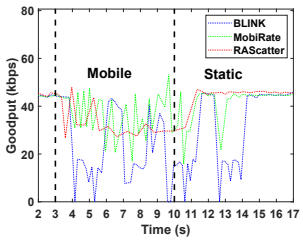
Fig. 26. Impact of device mobility

tracking [9]. To evaluate such impact of device mobility, as shown in Figure 25, we move the tag following different mobility patterns at 0.2m/s. As shown in Figure 26, RAScatter achieves 3.5× higher energy efficiency over MobiRate, and retains similar performance over different mobility patterns. Figure 27(a) further shows that the error of RAScatter's power adaptation can be reliably restrained within 7% in different mobility patterns. These results demonstrate that RAScatter can do timely and precise power adaptation under highly dynamic channel conditions.

Furthermore, we investigate the goodput variation in such highly dynamic backscatter channel conditions. As shown in Figure 27(b), when the tag continuously moves, RAScatter exhibits much smaller goodput variation compared to BLINK and MobiRate. When the tag stops moving, power adaptation in RAScatter also ensures that goodput objective could be quickly met. Compared to BLINK and MobiRate, RAScatter reduces such stabilization delay by more than 50%.



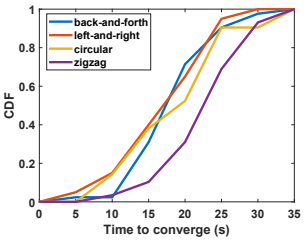
(a) Accuracy of power adaptation



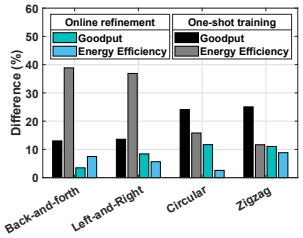
(b) Achieved goodput over time

Fig. 27. Goodput variations in highly dynamic channel conditions

Different mobility patterns may also impact the performance of online model refinement. Being similar to Section VIII-B, we switch RAScatter to an untrained environment and evaluate the accuracy of power adaptation after online refinement with different mobility patterns. As shown in Figure 28(a), overall 70% of such refinements can converge within 25 seconds, and such latency could be further significantly reduced in high-throughput backscatter systems such as Passive WiFi [18]. Further, Figure 28(b) shows that our online learning scheme is robust to different mobility patterns. Slightly higher errors are observed in circular and zig-zag tag movements, because of frequent changes of the tag's orientation to the reader[9].



(a) Latency of online refinement

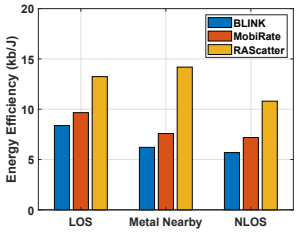


(b) Accuracy of power adaptation

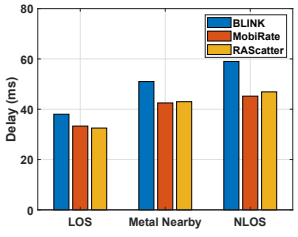
Fig. 28. Online model refinement on a mobile tag

F. Impact of Surrounding Objects

The surrounding objects can absorb or reflect the RF signal, hence affecting backscatter communication. To investigate such impact, we consider two different scenarios: 1) placing a metal object very close to the tag (Metal Nearby), and 2) blocking the line of sight between the tag and the reader with a thin metal panel (NLOS). The goodput objective is also set to 90%. As shown in Figure 29, RAScatter achieves over 2× higher energy efficiency than BLINK and 1.5× higher than MobiRate, while resulting in <1% delay increase. Compared to the line-of-sight case, such difference is enlarged by the interference from surroundings, which demonstrates the RAScatter's robustness of power adaptation.



(a) Energy efficiency



(b) Data transmission delay

Fig. 29. Impact of different surrounding objects

IX. Related Work

Backscatter Communication. RAScatter builds on prior work of backscatter communication. Traditional systems (e.g.

[9]The tag's orientation affects the signal strength and power-up delay.

RFIDs) use dedicated devices to communicate with batteryless tags, but are limited to very short communication range and low goodput. Recent research efforts, instead, use commodity wireless devices or infrastructure such as smartphones [17], TV stations [21], FM stations [30], and significantly improve their goodput and communication range [8], [27]. These efforts, however, aim to improve the communication performance and ignore reader's energy efficiency.

RAScatter is related to early research efforts on backscatter power scaling [23], [31] , which are however, limited to traditional NFC and RFID systems but cannot be applied to new backscatter devices and readers (e.g., smartphones). In addition, these schemes either calibrate the reader's power for fixed channel conditions offline, or exhaustively search the optimal power at runtime with high costs. They, hence, have limited applicability in practical scenarios.

RAScatter also relates to prior work on rate adaptation in backscatter systems [35], [12], which improves goodput with the given transmit RF power by choosing the optimal data rate. These existing techniques, however, are incapable of reducing the reader's power consumption. Their accuracy could be easily affected by backscatter channel dynamics due to the numerical channel metrics they use (e.g., SNR [33]).

AI-assisted Communication Systems. Our design is inspired by prior work on using neural networks to improve the performance of communication systems. For example, neural networks have been widely adopted for uplink throughput prediction [19] and congestion control [26]. However, these schemes are all based on general neural network models whose designs and structures are independent from any domain knowledge. Instead, RAScatter uses a modular neural network that allows designing the neural network structure based on special characteristics of backscatter communication. This domain-specific design hence, ensures accurate power adaptation even with highly dynamic backscatter channel and limited amounts of training data.

X. Discussions

Throughput vs. goodput. Most existing backscatter systems aim to enhance the communication throughput, without considering individual data losses and possible retransmissions. Such data losses and retransmissions, however, are crucial in many practical application scenarios that are time-sensitive. For example, losses of timely temperature readings may result in device overheat and severe damage. Instead, RAScatter defines the objective of power adaptation as goodput, so as to minimize possible data losses.

Low-speed backscatter systems. In this paper, we use the WISP batteryless tag as experimentation platform. However, since RAScatter requires no extra computation nor hardware change on the batteryless device, it can be applied to most types of batteryless devices, and our design of modular neural network can then be used to address the challenge of limited training data on low-speed devices such as LoRa Backscatter (18bps-37.5kbps) [27], WiFi Backscatter (<1kbps) [17] and Ambient Backscatter (<1kbps) [21].

Impact of device & environment heterogeneity. RAScatter's accuracy of power adaptation relies on precise measurements of backscatter channel, which may be affected by the batteryless device's hardware properties (e.g. antenna sensitivity, buffer capacity, etc) and the environment's multipath patterns. These factors can be generally resolved with additional training for different device models and environments. Another solution is to adopt transfer learning or semi-supervised learning to ensure that the trained model can well adapt to such heterogeneity. This will be our future work.

Practical deployment. As shown in Section VII, RAScatter involves the minimum hardware changes on backscatter readers, and it can hence be easily deployed to commodity reader devices. When being deployed to reader devices with severe power constraints (e.g., future mini-scale robots), RAScatter can leverage the emerging hardware, such as low-power AI chips [1] with power consumption of <1 mW, for more efficient neural network computations with the minimum latency and power consumption.

XI. Conclusion

In this paper, we present RAScatter, a new backscatter system that enhances the energy efficiency of backscatter readers by adapting the reader's transmit RF power to the required goodput. We ensure accurate power adaptation by learning the correlation between backscatter channel condition, transmit RF power and goodput with a modular neural network, and enhance the reader's energy efficiency by >3.5×.

Acknowledgments

We thank the anonymous reviewers for their comments and feedback. This work was supported in part by the National Science Foundation (NSF) under grant number CNS-1812407, CNS-2029520 and IIS-1956002.

Appendix

With certain channel condition, the maximum backscatter goodput is a discounted Shannon's ceiling by duty cycle D:

$$G = D \cdot W \log(1 + S/N), \tag{2}$$

where W, S, N are channel bandwidth, backscatter signal power, and noise level respectively. Since the batteryless device's live time is usually much smaller than power-up delay, we have $D \approx T_d$, and Eq. (2) can be rewritten as

$$G = k \cdot \log(1 + S/N)/T_d.$$

T_d, on the other hand, can be elaborated based on the capacitor charging formula as

$$T_d = \tau \cdot \ln(1 + (V_{\max} - V_{\min})/(V_c - V_{\max})),$$

where V_c is the batteryless device's harvested voltage in the current cycle, and $V_{\max}$ and $V_{\min}$ are the maximum and minimum harvested voltage, respectively. Since $V_{\max} \approx V_{\min}$ in practice for good cycle rates, T_d can be approximated[10] by $(V_{\max} - V_{\min})/(V_c - V_{\max})$. Eq. (1) can then be proved by considering that the transmit power $P \propto V_c^2$ and $P \propto S$.

[10] $\ln(1 + x) \approx x$ if $|x| \ll 1$

REFERENCES

[1] Ecm3532. https://en.wikichip.org/wiki/eta_compute/ecm353x/ecm3532.

[2] Epc gen 2 standard. https://www.gs1.org/sites/default/files/docs/epc/gs1-epc-gen2v2-uhf-airinterface_i21_r_2018-09-04.pdf.

[3] Wisp 5 tag firmware. https://github.com/wisp/wisp5.

[4] Xbee s2c data sheet. https://www.digi.com/resources/library/data-sheets/ds_xbee-s2c-802-15-4.

[5] Zebra rfd8500 uhf rfid handheld reader. https://www.zebra.com/us/en/products/rfid/rfid-handhelds/rfd8500.html.

[6] M. M. Alam and E. B. Hamida. Strategies for optimal mac parameters tuning in ieee 802.15. 6 wearable wireless sensor networks. *Journal of medical systems*, 39(9):1–16, 2015.

[7] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[8] D. Bharadia, K. R. Joshi, M. Kotaru, and S. Katti. Backfi: High throughput wifi backscatter. *ACM SIGCOMM*, 2015.

[9] M. Buettner, R. Prasad, A. Sample, D. Yeager, B. Greenstein, J. R. Smith, and D. Wetherall. Rfid sensor networks with the intel WISP. In *Proceedings ACM SenSys*, 2008.

[10] K. Chen. Deep and modular neural networks. In *Springer Handbook of Computational Intelligence*, pages 473–494. Springer, 2015.

[11] 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 rfids. In *Proc. ACM SenSys*, 2015.

[12] W. Gong, S. Chen, J. Liu, and Z. Wang. Mobirate: Mobility-aware rate adaptation using phy information for backscatter networks. In *Proc. INFOCOM*, 2018.

[13] E. Hoque, R. F. Dickerson, and J. A. Stankovic. Monitoring body positions and movements during sleep using wisps. In *Wireless Health 2010*, pages 44–53. 2010.

[14] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proc. ACM SIGCOMM*, 2016.

[15] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[16] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti. Wideo: Fine-grained device-free motion tracing using RF backscatter. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 189–204, 2015.

[17] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall. Wi-fi backscatter: Internet connectivity for rf-powered devices. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 607–618, 2014.

[18] B. Kellogg, V. Talla, S. Gollakota, and J. R. Smith. Passive wi-fi: Bringing low power to wi-fi transmissions. In *Proc. NSDI*, 2016.

[19] J. Lee, S. Lee, J. Lee, S. D. Sathyanarayana, H. Lim, J. Lee, X. Zhu, S. Ramakrishnan, D. Grunwald, K. Lee, et al. PERCEIVE: Deep learning-based cellular uplink prediction using real-time scheduling patterns. In *Proc. ACM MobiSys*, 2020.

[20] N. Li, G. Calis, and B. Becerik-Gerber. Measuring and monitoring occupancy with an rfid based system for demand-driven hvac operations. *Automation in construction*, 24:89–99, 2012.

[21] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: Wireless communication out of thin air. *ACM SIGCOMM Computer Communication Review*, 43(4):39–50, 2013.

[22] Y. Ma, X. Hui, and E. C. Kan. 3d real-time indoor localization via broadband nonlinear backscatter in passive devices with centimeter precision. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 216–229, 2016.

[23] M. Menghin, N. Druml, C. Steger, R. Weiss, R. Bock, and J. Haid. Nfc-dynfs: A way to realize dynamic field strength scaling during communication. In *Intl' Workshop on Near Field Communication*, 2013.

[24] S. Naderiparizi, M. Hessar, V. Talla, S. Gollakota, and J. R. Smith. Towards battery-free HD video streaming. In *Proc. NSDI*, 2018.

[25] D. Saad. Online algorithms and stochastic approximations. *Online Learning*, 5:6–3, 1998.

[26] C. Sander, J. Rüth, O. Hohlfeld, and K. Wehrle. Deepcci: Deep learning-based passive congestion control identification. In *Proceedings of the Workshop on Network Meets AI & ML*, 2019.

[27] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota. Lora backscatter: Enabling the vision of ubiquitous connectivity. *Proc. ACM IMWUT*, 1(3):1–24, 2017.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[29] G. K. Verma and P. Tripathi. A digital security system with door lock system using rfid technology. *International Journal of Computer Applications*, 5(11):6–8, 2010.

[30] A. Wang, V. Iyer, V. Talla, J. R. Smith, and S. Gollakota. FM backscatter: Enabling connected cities and smart fabrics. In *Proc. NSDI*, 2017.

[31] L. Xie, Q. Li, C. Wang, X. Chen, and S. Lu. Exploring the gap between ideal and reality: An experimental study on continuous scanning with mobile reader in rfid systems. *IEEE Transactions on Mobile Computing*, 14(11):2272–2285, 2015.

[32] D. J. Yeager, P. S. Powledge, R. Prasad, D. Wetherall, and J. R. Smith. Wirelessly-charged uhf tags for sensor data collection. In *IEEE International Conference on RFID*, 2008.

[33] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A practical snr-guided rate adaptation. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 2083–2091. IEEE, 2008.

[34] P. Zhang, D. Bharadia, K. Joshi, and S. Katti. Hitchhike: Practical backscatter using commodity wifi. In *Proc. ACM SenSys*, 2016.

[35] P. Zhang, J. Gummeson, and D. Ganesan. Blink: A high throughput link layer for backscatter communication. In *Proc. ACM MobiSys*, 2012.