Inventing Codes for Channels With Active Feedback via Deep Learning

Karl Chahine, *Graduate Student Member, IEEE*, Rajesh Mishra, and Hyeji Kim

***Abstract*—Designing reliable codes for channels with feedback, which has significant theoretical and practical importance, is one of the long-standing open problems in coding theory. While there are numerous prior works on analytical codes for channels with feedback, the majority of them focus on channels with noiseless output feedback, where the optimal coding scheme is still unknown. For channels with *noisy* feedback, deriving analytical codes becomes even more challenging, and much less is known. Recently, it has been shown that deep learning can, in part, address these challenges and lead to the discovery of new codes for channels with noisy output feedback. Despite the success, there are three important open problems: (*a*) deep learning-based codes mainly focus on the passive feedback setup, which is shown to be worse than the active feedback setup; (*b*) deep learning-based codes are hard to interpret or analyze; and (*c*) they have not been successfully demonstrated in the over-the-air channels with feedback. We address these three challenges. First, we present a learning-based framework for designing codes for channels with active feedback. Second, we analyze the latent features of the learned codes to devise an analytical coding scheme. We show that the approximated analytical code is a non-trivial variation of the state-of-the-art codes, demonstrating that deep learning is a powerful tool for deriving a new analytical communication scheme for challenging communication scenarios. Finally, we demonstrate the over-the-air performance of our neural codes by building a wireless testbed that consists of two separate N200 USRPs operating as the transmitter and the receiver. To the best of our knowledge, this is the first over-the-air hardware implementation of neural codes for interactive channels.**

***Index Terms*—Deep learning for channel coding, channels with feedback, active feedback, software-defined radios, interpretable machine learning.**

I. Introduction

A CRITICAL aspect of reliable communication involves designing codes that allow transmissions to be robustly and computationally efficiently decoded under noisy conditions. Advances in the design of reliable and efficient codes have been driven by information, communication, and coding theories, featuring several codes such as Turbo, Low Density Parity Codes (LDPC), and Polar codes [1], [2] that are near optimal for the canonical point-to-point Additive White Gaussian Noise (AWGN) channels.

However, designing reliable codes becomes much more challenging when the channel deviates from AWGN channels. An important such scenario is the channels with *feedback*, where the transmitter and the receiver can *cooperate* via feedback to communicate a message reliably. On the practical front, modern communication relies on the Automatic Repeat reQuest (ARQ) and the hybrid ARQ, where the transmitter retransmits the entire message block or transmits a pre-assigned message block upon receiving a repeat request. On the theoretical front, in 1956 [3], Shannon initiated the study of channels with noiseless output feedback, where the received value is fed back to the transmitter with a unit-step delay, and showed that the output feedback significantly improves reliability [4]. Shalkwijk and Kailath provided a linear encoding scheme, referred to as the SK scheme, which achieves such improved reliability, leading to a doubly exponential error exponent, for finite-block length settings [4].

Despite several theoretical studies on channels with feedback [4], [5], [6], [7], [8], [9], [10], there are many important open problems. First, the optimal coding scheme for channels with noiseless output feedback is still unknown. In [10], Polyanskiy et al. introduced an encoding scheme, referred to as the PPV scheme, for communicating a single bit over channels with noiseless output feedback. which is shown to be asymptotically optimal, achieves zero error probability with minimal energy requirement, and outperforms the SK scheme. Whether the PPV scheme is optimal for communicating a single bit over finite rounds of communications, however, is an open problem.

Furthermore, the elegant theory and results from the SK and PPV schemes break if there is noise in the output feedback channel. As the noise in the feedback channel prohibits the encoder and the decoder from perfectly synchronizing, an estimation error accumulates over time, which leads to poor performance. Kim et al. showed that linear codes that utilize the noisy output feedback could not achieve a positive rate of communication [8]. For non-asymptotic settings, in [7], Chance and Love proposed a linear coding scheme that improves upon the SK scheme for channels with noisy output feedback, which is later further improved by Mishra et al. [11]. Nevertheless, the optimal coding scheme for channels with noisy feedback is still unknown.

More recently, with the advances in deep learning, it has been shown that coding schemes can be learned in a data-driven manner. Neural network-based nonlinear codes, called

Manuscript received 2 June 2022; revised 2 September 2022 and 11 October 2022; accepted 16 October 2022. Date of publication 25 October 2022; date of current version 16 March 2023. This work was supported in part by the Office of Naval Research (ONR) under Award N00014-21-1-2379; in part by the National Science Foundation (NSF) under Award CNS-2008824; and in part by the 6G@UT Center within the Wireless Networking and Communications Group (WNCG) at the University of Texas at Austin. *(Karl Chahine and Rajesh Mishra contributed equally to this work.) (Corresponding author: Karl Chahine.)*

The authors are with the Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: karlchahine@utexas.edu).

Digital Object Identifier 10.1109/JSAIT.2022.3216515

2641-8770 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

Deepcode, were introduced in [12]; they are shown to outperform both the Chance-Love and Mishra-Vasal-Kim schemes for the AWGN channel with noisy output feedback. This success is attributed to the deep learning tools which enable efficient search over nonlinear codes.

Several works have generalized Deepcode by using different neural architectures, combined with novel training methodologies. Deep Extended Feedback (DEF) codes and Deep SNR-Robust Feedback (DRF) codes were introduced in [13] and [14] respectively, both of which build upon the Deepcode architecture and achieve improved reliability via different training procedures. In [15], the authors introduce AttentionCode, a new class of deep learning based feedback codes designed by a variation of the transformer architecture. As a follow-up, [16] presents the Generalized Block Attention Feedback (GBAF) code, a generalization of AttentionCode that addresses some limitations of existing neural designs, such as communication overhead and a limited set of feasible rates.

Despite the success of neural codes, there are three important open problems of significant theoretical and practical importance. First, the underlying channel model considered in [12], [13], [14], [15], [16] is the channel with noisy *output* feedback, where the receiver *passively* sends the received output to the transmitter. In other words, the receiver is not allowed to *encode* its received values into a feedback signal. In [17], Ben-Yishai and Shayevitz showed that the reliability can be noticeably improved by letting the receiver encode its feedback signal. Their proposed feedback encoding scheme, referred to as the Modulo-SK scheme, relies on the modulo-lattice operation, which is piecewise linear. However, learning a *nonlinear* feedback encoder along with the channel encoder and decoder is more challenging than learning only the channel encoder and decoder. In [12], this challenge, along with some negative results, is explicitly noted; encoding the feedback signal in the Deepcode architecture of [12] led to a minimal gain in reliability.

Second, the interpretation of neural codes remains a challenging open problem. Although a first-order analysis of learned codes is provided in [12], deriving an analytical approximation of learned codes is a widely open problem. Recently, there has been an analytical analysis of deep-learned codes for the point-to-point channels without feedback [18]. However, there is very little understanding and interpretation of deep-learned codes for channels with and without feedback besides this work.

Last, the channel model considered in the literature focuses on the Additive White Gaussian Noise (AWGN) channel. However, the practical channels are not always AWGN, and verifying the performance of neural codes in the *over-the-air* wireless environment is a necessary precursor for deploying neural codes in practical systems. However, most work on deep-learned codes focuses on the synthetic channels [12], [13], [14], [15], [16], [19], [20], [21].

In this paper, to address all three challenges mentioned above, we (*a*) introduce a framework, which we call ActiveFB, to learn nonlinear coding schemes for channels with *active* feedback, (*b*) derive an analytical approximation for the neural codes and provide the interpretation analysis, and (*c*) demonstrate them in the fading channels and over-the-air system. We specifically focus on communicating a *short message sequence*, which is a problem setup introduced and studied by Polyanskiy et al. [10], instead of a long message sequence as in [12], [13], [14]. Surprisingly, we show that our neural codes for a 2-bit message sequence outperform the neural codes learned for a 50-bit message sequence in [12], [13], [14]. Furthermore, we run systematic interpretation analysis and derive a precise analytical approximation of the learned codes, which could not be done for longer-blocklength codes in [12], [13], [14]. We show that the analytical approximation is a variation of the PPV scheme in [10] and provide an interpretation of how the two schemes are different. Our main contributions are as follows.

- *Learning framework for channels with active feedback:* We introduce ActiveFB,[1] a two-layer Recurrent Neural Network (RNN) encoder coupled with a two-layer RNN feedback encoder and a decoder for communicating a short message bit sequence over channels with active feedback. We empirically show that ActiveFB outperforms all the state-of-the-art codes such as the PPV scheme, SK scheme, Modulo-SK scheme, and neural codes for channels with noisy feedback as well as channels with noiseless feedback (Sections III, V-B, IV-C). In addition, a comparison against non-feedback schemes can be found in the Deepcode work [12].
- *Analytical approximation of learned codes for noiseless feedback:* For channels with noiseless feedback, we show that our ActiveFB code can be closely approximated as a variation of the PPV scheme, i.e., we establish a new state-of-the-art *analytical* scheme for channels with noiseless feedback. In addition, we show that the PPV scheme, which has been mainly considered for asymptotic settings in [10], can be further optimized by controlling the power of each transmission. We provide a dynamic programming approach for this optimization (Section IV-D).
- *Analytical approximation of learned codes for noisy feedback:* For channels with noisy feedback, we show that the ActiveFB scheme can still be approximated as a PPV scheme. Compared to the Modulo-SK scheme, which encodes the feedback in a piecewise linear manner, we observe that ActiveFB encodes the feedback in a *nonlinear* manner which can be modeled using a PPV-like scheme (Section V-C).
- *Over-the-air demonstration of ActiveFB:* We also demonstrate the superior performance of our neural network-based schemes by implementing them on an end-to-end communication system on an over-the-air channel in real-time. This is the first demonstration of active feedback coding schemes on a dynamic fading channel, to the best of our knowledge (Section VI).

II. Problem Formulation

We consider AWGN channels with active feedback depicted in Fig. 1. The forward channel is modeled as an AWGN channel, i.e., $y_i = x_i + z_i$, where $z_i \sim \mathcal{N}(0, \sigma^2)$ denotes the Gaussian noise. Following each i^{th} forward transmission, the

[1]Source codes are available at https://github.com/karlchahine/ActiveFB.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

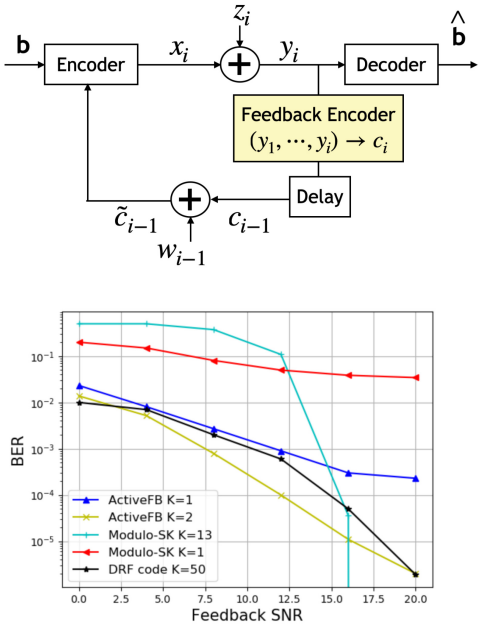

Fig. 1. (Top) Channels with active feedback. We jointly learn the encoder, the decoder, and the feedback encoder. (Bottom) We design a neural code, labeled as ActiveFB, which outperforms the state-of-the-art coding scheme for active feedback channels, labeled as Modulo-SK, and the state-of-the-art neural coding scheme for passive feedback channels, labeled as DRF code for channels with noisy feedback. K denotes the number of message bits to communicate.

feedback encoder maps its received values $y^i = (y_1, \ldots, y_i)$ to a feedback symbol c_i which is fed back to the transmitter with one step delay. The feedback channel is also modeled as an AWGN channel, i.e., $\tilde{c}_i = c_i + w_i$, where $w_i \sim \mathcal{N}(0, \sigma_f^2)$ denotes the Gaussian noise.

We assume that the encoder wishes to communicate K binary messages $\mathbf{b} \in \{0,1\}^K$ over N rounds of transmissions. We mainly focus on communicating one bit, i.e., $K = 1$. The *encoder* function is inherently causal and given as $x_{i+1} = f_{\text{enc}}(\mathbf{b}, \tilde{c}^i)$ for $i = 1, 2, \ldots, N-1$. The *feedback encoder* function is also causal, where $c_i = f_{\text{fbenc}}(y^i)$. The *decoder* function is non-causal; after N rounds of communications, the decoder estimates the message based on its received sequence as $\hat{\mathbf{b}} = f_{\text{dec}}(y^n)$. The power constraints are given as $\mathbb{E}[\mathbf{x}^2] \leq 1$ and $\mathbb{E}[\mathbf{c}^2] \leq 1$, where the expectations are over both the uniform distribution of the bits generated and the randomness in the Gaussian noise.

Our goal throughout this paper is to design $(f_{\text{enc}}, f_{\text{fbenc}}, f_{\text{dec}})$ which jointly minimizes the probability of error $P_e := P\{\hat{\mathbf{b}} \neq \mathbf{b}\}$. Designing codes for channels with active feedback is a challenging task as it involves the *joint* design of the feedback encoder as well as the encoder and the decoder. In the following section, we introduce the ActiveFB framework, where we model the encoder, the feedback encoder, and the decoder as RNNs, and learn them jointly, the results of which are analyzed, interpreted, and compared against existing codes in Sections IV and V.

III. ActiveFB: Joint Coding and Feedback Coding via RNN

For concreteness, we consider the transmission of a single bit $b \in \{0,1\}$ over $N = 3$ rounds of interactive communications. As feedback systems are sequential by nature, we model the channel encoder, the feedback encoder, and the channel decoder as RNNs, as illustrated in Fig. 2.

A. Architecture

Our ActiveFB framework consists of the channel encoder, the feedback encoder, and the channel decoder parametrized by RNNs, reflecting the sequential nature of the feedback. While the RNN parametrization is a natural fit, we empirically observe that the architectural choices have a negligibe impact on the reliability. For example, RNNs with 50 hidden units and RNNs with 10 hidden units lead to very similar reliability. Our experiments are simulated using 50 hidden units. Moreover, we observe that the appropriate power allocation is crucial, as we explain in the following.

Encoding: We model the channel encoder as an RNN, which takes the newly available feedback as input in a causal manner. As illustrated in Fig. 2, at every time-step i, $i \in \{1, 2, 3\}$, the input to the RNN encoder consists of the message bit b and all the feedbacks from the previous steps $1, \ldots, i-1$. At times $i = 1$ and $i = 2$, we pad the missing entries with 0.5. The encoder's output $\tilde{x}_i$ is passed through a power constraint block $x_i = h(\tilde{x}_i)$.

Power allocation: The power constraint block $h(.)$ operates in two successive stages.

(1) Power normalization: In the first stage, we estimate the first and second moments of each batch of size J. More specifically, at time i and for training example j, $x_i^j = (\tilde{x}_i^j - \mu_i)/\sigma_i$, where $\mu_i = \frac{1}{J}\sum_{j=1}^{J} \tilde{x}_i^j$ and $\sigma_i = \sqrt{\frac{1}{J}\sum_{j=1}^{J}(\tilde{x}_i^j - \mu_i)^2}$ are the batch mean and standard deviation, respectively. This results in having $\mathbb{E}[x_i] = 0$ and $\mathbb{E}[x_i^2] = 1$, satisfying power constraints in wireless systems. During the training phase, μ_i and σ_i are estimated from each batch. On the other hand, during the testing phase, μ_i and σ_i are pre-computed with multiple batches.

(2) Power scaling: In the second stage, we introduce trainable weights p_1, p_2, and p_3, which control the power allocated to x_1, x_2, and x_3, respectively. The weights are updated throughout the training such that the average power should satisfy $\sum_{i=1}^{3} p_i^2/3 \leq 1$. As shown in Fig. 2 (Bottom), the trainable power scaling leads to reliability improvement.

Feedback encoding and channel decoding: The receiver performs both the feedback encoding and channel decoding via an RNN, which takes y_i at time i for $i = 1, 2, 3$, as shown in Fig. 2. For time-steps $i = 1$ and $i = 2$, the decoder RNN generates a coded symbol which then goes through the power constraint block $h(.)$. The power-constrained coded bit c_i is then transmitted to the encoder through a feedback channel; the transmitter receives $\tilde{c}_i = c_i + w_i$. At time $i = 3$, the decoder generates $\hat{b}$, the final estimate of the message b. In our experiments, we tried having two separate RNNs for the feedback encoding and channel decoding, and having a single RNN for both the feedback encoding and channel decoding (as depicted in Fig. 2 (Top)). We observed that having a single RNN led to a better performance. We therefore show the results of adopting a single RNN.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

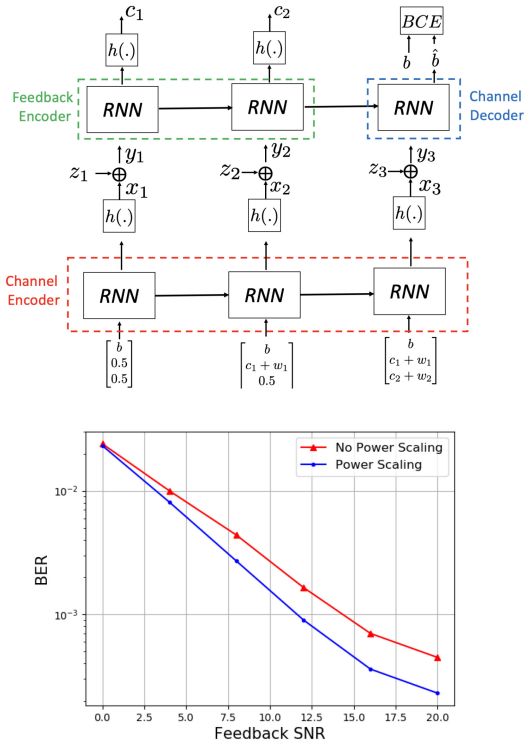

Fig. 2. (Top) Neural architecture of ActiveFB. We model the encoder, the feedback encoder, and the decoder as RNNs. We also entangle the feedback encoder and the decoder into one RNN. (Bottom) Power allocation is critical in achieving the high reliability of ActiveFB.

B. Training

We train the channel encoder, the feedback encoder, and the channel decoder jointly under AWGN channels with noisy feedback so that the probability of error in recovering the information bit b is minimized. Both the encoder and the decoder RNNs are jointly trained using binary cross-entropy as the loss function with a batch size 500 via an Adam optimizer. We also use a decaying learning rate and gradient clipping; we reduce the learning rate by 10 times after training with 10^6 examples, starting from 0.01. We train and test at the same Signal to Noise Ratio (SNR) defined as $SNR = -10\log_{10}(\sigma^2)$, where σ^2 is the noise variance. As illustrated in the following sections, we will evaluate ActiveFB over a range of SNR points. The forward and feedback channels do not necessarily share the same SNR value.

We also note that the architecture of ActiveFB depicted in Fig. 2 (Top) and our training methodology can be easily specialized to channels with *noiseless* feedback. The only change we make for channels with *noiseless* feedback is that the feedback noise samples w_1 and w_2 are set to 0. In the following sections, we run a set of experiments to describe the performance of ActiveFB under various settings, including both channels with noisy feedback and channels with noiseless feedback.

IV. Channels With Noiseless Feedback

As a precursor to learning codes for channels with noisy feedback, we begin our experiments and analysis for channels with noiseless feedback. Our ActiveFB framework is directly applicable to channels with noiseless feedback, for which the optimal coding scheme is not known to date.

In Section IV-A, we provide an overview of two existing coding schemes for channels with noiseless feedback, namely, the SK scheme and the PPV scheme. The SK scheme is a celebrated linear coding scheme which minimizes the mean square error in communicating a discrete message, and the PPV scheme is the state-of-the-art coding scheme for communicating a single bit. In Section IV-B, we show that we can improve the PPV scheme by introducing a dynamic program-based power optimization. We then show in Section IV-C that ActiveFB learns a code that outperforms both the power-optimized PPV scheme and the SK scheme. We provide an analytical approximation of the ActiveFB codes and interpretation in Section IV-D, where we show that ActiveFB codes are a non-trivial variation of the PPV scheme.

A. Existing Coding Schemes

Shalkwijk-Kailath (SK) scheme: Shalkwijk and Kailath introduced a celebrated *linear* communication scheme for channels with noiseless feedback [4]. The transmitter (Tx) first maps the message $\mathbf{b} \in \{0,1\}^K$ to the real-valued variable θ using a Pulse Amplitude Modulation (PAM). In the first round, it sends a scaled version of θ satisfying the power constraint P. In subsequent rounds, for example, at round i, the receiver (Rx) maintains an estimate $\hat{\theta}_i$ of θ given all the observations it has and feeds its observation back to Tx. Tx then computes the estimation error $\epsilon_i = \hat{\theta}_i - \theta$ and sends a power-scaled version of ϵ_i to Rx so that the Rx can correct the error. After a fixed number of rounds, Rx decodes the message using a minimum distance rule. Theoretically, the SK scheme relies on exactly noiseless feedback and does not extend to channels with an even arbitrarily small amount of noise in the feedback [4], [6]. An extension of SK to noisy settings is described in Section V-A.

Polyanskiy, Poor and Verdú (PPV) scheme: Polyanskiy, Poor and Verdú introduced nonlinear coding schemes for communicating a single bit over channels with noiseless feedback [10]. The encoder uses Log Likelihood Ratio (LLR) it receives as feedback at the end of each round to compute the error in the estimate of the transmitted bit at the receiver. It then scales the error value appropriately to satisfy the power constraint and sends it in the subsequent transmission. This proposed scheme was shown to achieve the minimum energy per bit, i.e., the probability of error approaches zero as the number of transmission rounds goes to infinity with the minimum energy spent per bit.

More specifically, to communicate a binary message $W \in \{-1, 1\}$, where $W = 2b - 1$ for $b \in \{0, 1\}$, at the i^{th} round, the encoder uses the function

$$x_i\left(W, y^{i-1}\right) = \frac{Wd_i}{1 + e^{WS_{i-1}}}, \tag{1}$$

where $S_{i-1} = \log \frac{P[W=+1|y^{i-1}]}{P[W=-1|y^{i-1}]}$ denotes the LLR for $i = 1, 2, \ldots, N$ and y^{i-1} are the noisy received symbols at the receiver up till time $i-1$, i.e., $y^{i-1} = (y_1, \ldots, y_{i-1})$. The

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

scaling factor d_i denotes the power scaling constant for the i^{th} round of transmission. After N rounds, the receiver recovers the message $\hat{W}$ based on the final LLR S_N.

B. Power Optimization for the PPV Scheme

The PPV scheme is shown to achieve the zero error probability with a minimal power as the number of rounds $N \to \infty$ with a constant power scaling factor $d_i = d$. However, for a non-asymptotic regime where $N < \infty$, as observed in ActiveFB codes, choosing the right set of power scaling factors d_i's is important. To this end, we build on the PPV scheme to address this shortcoming. Specifically, we formulate the power optimization problem and provide a dynamic program algorithm that numerically computes the optimal set of d_i's according to a sum power constraint. We show that this modification leads to improved reliability up to 1dB as shown in Fig. 3.

Numerical optimization and dynamic program: We define the objective for solving for the optimal parameters d_i's as

$$\tilde{d}_1, \ldots, \tilde{d}_N = \arg\min_{d_{1:N}} P_e^N,$$

where P_e^N denotes the error after N rounds of transmissions obtained when the encoder follows (1).

To solve the optimization formulated above, we propose a dynamic program that expresses P_e^N in terms of d_i and the statistics of LLR S_{i-1}, $i = N \ldots 1$ backward recursively, and solves for the optimal d_i at each step for any given statistics of S_{i-1} with sum power constraint. Specifically, we show the following two theorems to solve the optimization.

Theorem 1: The probability of error at the end of N transmissions, P_e^N, can be expressed as a recursion function in terms of the statistics of LLR, μ_i and σ_i, given by

$$P_e^N = 1 - Q\left(-\frac{\mu_N}{\sigma_N}\right), \tag{2}$$

where μ_N and σ_N are described recursively in terms of d_i as

$$\mu_i = \mu_{i-1} + \frac{d_i^2}{2\sigma^2}, \tag{3}$$

$$\sigma_i^2 = \sigma_{i-1}^2 + \left(\frac{d_i}{\sigma^2}\right)^2, \tag{4}$$

and $Q(\cdot)$ is the complementary error function.

Proof: Without loss of generality, we can express the probability of error at the end of N rounds as $P_e^N = Pr\{S_N < 0 | W = 1\}$ (due to symmetry).

We also note that given $W = 1$, the update to the value of LLR S_i can be expressed as

$$S_i = S_{i-1} + \frac{1}{2\sigma^2} d_i^2 + \frac{1}{\sigma^2} d_i z_i, \tag{5}$$

where $z_i \sim \mathcal{N}[0, \sigma^2]$ is the forward noise in the i^{th} transmission. It is evident from the recursive equation in (5) that the random variable S_N is just a sum of N Gaussian random variables scaled and shifted depending on noise variances and parameters $d_{1:N}$. Therefore, the value of P_e^N at the end of N transmissions can be expressed as in (2).

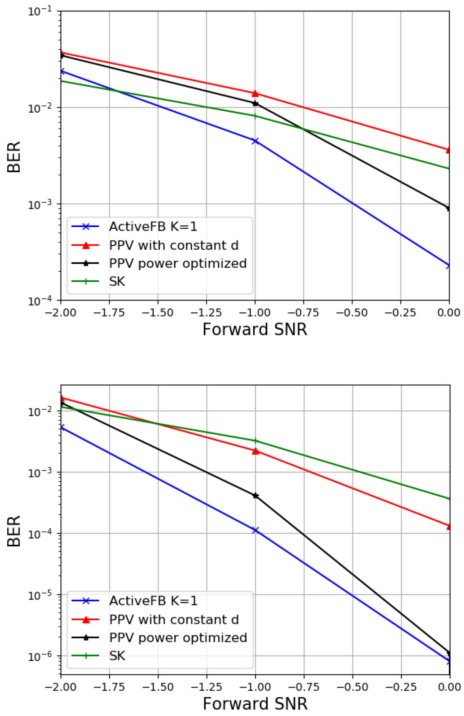

Fig. 3. (Top) ActiveFB outperforms all the baselines for channels with noiseless feedback for $N = 3$ rounds of communications. (Bottom) ActiveFB outperforms the baselines for channels with noiseless feedback for $N = 6$ rounds of communications.

We then obtain an updated equation for the statistics $\{\mu_i, \sigma_i^2\}$ as shown in (3) and (4). Using (2), (3), and (4), we can express P_e^N as a backward recursion. ■

Next, we need to constraint d_i to satisfy the sum power constraint. Let E_i^N denote the unallocated power for remaining transmissions i till N. Then, d_i is bounded such that

$$0 \le d_i^2 \left(\mu_i'^2 + \sigma_i'^2\right) \le E_i^N, \tag{6}$$

where μ_i' and σ_i' are the mean and variance of $\frac{1}{1+e^{S_{i-1}}}$ given $S_{i-1} \sim \mathcal{N}(\mu_{i-1}, \sigma_{i-1}^2)$.

Finally, we have the following optimization problem at any round i given as

$$\tilde{d}_i(\mu_{i-1}, \sigma_{i-1}, E_i^N) = \arg\min_{d_i} 1 - Q\left(-\frac{\mu_{i-1} + \frac{d_i}{2\sigma^2}}{\sqrt{\sigma_{i-1} + \left(\frac{d_i}{\sigma^2}\right)^2}}\right),$$

along with the constraint in (6). We solve this optimization numerically to obtain the set of optimal parameters $\tilde{d}_{1:N}$. We have summarized the steps in the dynamic program in Algorithm 1.

The algorithm is used to compute the set of optimal parameters d_i's, which can be used in the encoding function in (1). In Fig. 3, we plot the Bit Error Rate (BER) across different forward SNRs for noiseless feedback channel and compare it with other algorithms. We see that the PPV scheme with the optimized d parameters outperforms the original PPV scheme which assumed a constant d.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: Proposed DP to Optimize d_i in the PPV Scheme

Input: Blocklength: N, Sum power: E_1^N
Output: Parameters: $\tilde{d}_i$, $i = 1 \cdots N$
for $i = N, \ldots, 2$ **do**
  for *any* μ_{i-1}, σ_{i-1} *and* E_i^N **do**
    $\mu_i = \mu_{i-1} + \frac{d_i^2}{2\sigma^2}$ from (3)
    $\sigma_i^2 = \sigma_{i-1}^2 + \left(\frac{d_i}{\sigma^2}\right)^2$ from (4)
    if $i == N$ **then**
      $\tilde{d}_i = \underset{d_i}{\text{argmin}}\ 1 - Q\left(-\frac{\mu_i}{\sigma_i}\right)$
      $P_e^{N+1-i}\left(\mu_{i-1}, \sigma_{i-1}, E_i^N\right) = 1 - Q\left(-\frac{\mu_i}{\sigma_i}\right)$ at $d_i = \tilde{d}_i$
    end
    else
      $\tilde{d}_i = \underset{d_i}{\text{argmin}}\ P_e^{N-i}\left(\mu_i, \sigma_i, E_i^N\right)$
      $P_e^{N+1-i}\left(\mu_{i-1}, \sigma_{i-1}, E_i^N\right) = P_e^{N-i}\left(\mu_i, \sigma_i, E_i^N\right)$ at $d_i = \tilde{d}_i$
    end
  end
end
$\mu_1 = \frac{d_1^2}{2\sigma^2}$, $\sigma_1^2 = \left(\frac{d_1}{\sigma^2}\right)^2$
$\tilde{d}_1 = \underset{d_1}{\text{argmin}}\ P_e^{N-1}\left(\mu_1, \sigma_1, E_1^N\right)$
for $i = 2, \ldots, N$ **do**
  $\mu_i = \mu_{i-1} + \frac{d_i^2}{2\sigma^2}$, $\sigma_i^2 = \sigma_{i-1}^2 + \left(\frac{d_i}{\sigma^2}\right)^2$
  $\mu_i', \sigma_i' = \psi(\mu_{i-1}, \sigma_{i-1})$
  $E_i^N = E_{i-1}^N - d_i^2\left(\mu'^2 + \sigma'^2\right)$
  $\tilde{d}_i = \bar{d}\left(\mu_i, \sigma_i, E_i^N\right)$
end

Now we are ready to provide numerical results for ActiveFB. In the next section, we present the reliability of ActiveFB and compare it against the power-optimized PPV and SK schemes.

C. ActiveFB: Numerical Results

We analyze the performance of ActiveFB under noiseless feedback. In Fig. 3 (Top and bottom), we plot the BER vs. the forward channel Signal to Noise Ratio (SNR) for high rate ($N = 3$) and low rate ($N = 6$) for the following:

- *ActiveFB K=1:* This is our neural model described in Section III, where K is the number of transmitted information bits. Here, we transmit a single bit b ($K = 1$).
- *SK scheme:* We plot the BER of the celebrated SK scheme introduced in Section IV-A.
- *PPV scheme with constant d:* We plot the BER of the PPV scheme introduced in Section IV-A but with a constant power scaling factor $d_i = d$ for all the transmissions to meet the sum power constraint.
- *PPV scheme power optimized:* We plot the BER of the PPV scheme after power optimization using a dynamic programming algorithm. The details of the work are presented in Section IV-B.

We can see that the ActiveFB scheme outperforms both the power-optimized PPV and SK schemes for a wide range of forward SNR values for the noiseless feedback channel.

D. Interpretation of ActiveFB

In this section, we provide an interpretation analysis for ActiveFB. The *interpretability* can have various meanings. We focus on two aspects of interpretability: (*a*) providing a simple expression for the encoding and decoding functions and (*b*) being able to explain the difference between the neural codes and existing codes, and potentially provide an insight on how one can *modify* existing codes to improve the reliability based on neural results.

To this end, we first derive analytical schemes that approximate ActiveFB by studying the input-output relationships of the codes produced by the RNN. We consider the encoded bits as a function of the intended message and the received feedback and obtain an analytical scheme by fitting it to the already existing PPV scheme and its variant. By doing so, we obtain an insight into how one could alter the PPV scheme to further improve its reliability.

1) ActiveFB+: Derived Scheme From ActiveFB With a PPV Variation as a Backbone: We observe that the ActiveFB code resembles the PPV scheme, which motivates us to approximate the trained codes using the PPV scheme. In particular, we consider the following functional templates based on the PPV scheme to approximate ActiveFB codes.

- *PPV-like scheme:*
$$x_i\left(W, c^{i-1}\right) = \frac{Wd_i}{1 + e^{\boldsymbol{a_1} c^{i-1}}},$$
where $\boldsymbol{a_1} \in \mathbb{R}^{i-1}$ represents the vector that linearly combines the received feedback c^{i-1} (For passive feedback, $c^{i-1} = Y^{i-1}$). Note that this parameterization resembles the PPV scheme in (1).
- *Modified PPV scheme:*
$$x_i\left(W, c^{i-1}\right) = \frac{Wd_i}{1 + e^{\boldsymbol{a_1} c^{i-1} + a_2}},$$
where $\boldsymbol{a_1}$ denotes the vector that linearly combines the received feedback c^{i-1}, and a_2 denotes the bias.

With the two functional parameterizations, we fit the parameters $\mathbf{a_1}$ (and a_2) for the PPV-like and modified PPV schemes for 0 dB forward SNR.

Fitting with the PPV-like scheme: The encoder output x_i for different rounds of transmission are expressed as $x_1 = -1.1W$, $x_2 = \frac{-3.4W}{1+e^{1.7Wc_1}}$, and $x_3 = \frac{5.4W}{1+e^{1.462W(2c_1-c_2)}}$, where $W = 2b - 1$ with b as the original bit and c_1 and c_2 as the feedback received after the first and second rounds of transmissions, respectively. The final approximated decoding function is approximated as $\hat{W} = -15 + \frac{30}{1+e^{6c_1-3c_2+c_3}}$, where the final estimated bit $\hat{b} = 1$ if $\hat{W} > 0$ and 0 otherwise.

Fitting with the Modified PPV scheme: The fitting to the PPV scheme optimizes over the power allocation, which is the only degree of freedom available. However, by introducing a bias in the exponent in the modified PPV scheme, there

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

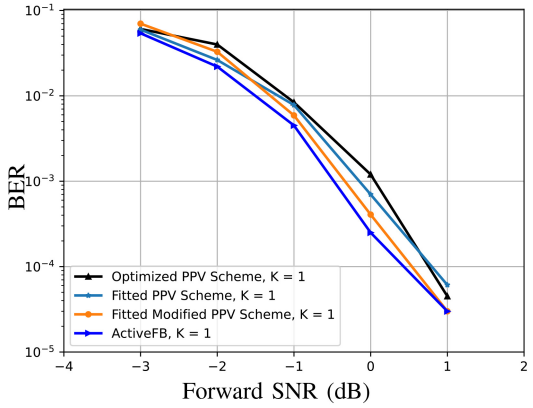

Fig. 4. We observe that ActiveFB outperforms the power-optimized PPV scheme. Moreover, the fitted model of the modified PPV scheme outperforms the fitted model of the PPV scheme, both of which are more reliable than the power-optimized PPV scheme. The blocklength was assumed to be $N = 3$, and the sum power was assumed as 0.5 dB. A forward SNR of 0 dB was considered for the fitting.

is a performance enhancement attributed to the extra flexibility in encoding the transmitted symbol as a function of the feedback. The transmissions assuming a modified PPV-like encoding function, can be expressed as $x_1 = -1.1W$, $x_2 = \frac{-3.9W}{1+e^{1.5Wc_1+0.3}}$, and $x_3 = \frac{9.2W}{1+e^{W(2c_1-c_2)+0.9}}$. The final decoding function is approximated as $\hat{W} = -15 + \frac{30}{1+e^{6c_1-3c_2+c_3}}$. The fitting model with $\mathbf{a_1}$ and a_2 fits the RNN data more closely than the corresponding PPV scheme with just $\mathbf{a_1}$.

In Fig. 4, we plot the BER of the different fitted models against the forward SNR and are compared against a baseline of the optimized PPV scheme and the RNN. We consider a noiseless feedback regime with finite blocklength $N = 3$ and restrict the sum power to 0.5 dB. As mentioned above, the fitting was done with data collected from RNN at 0 dB but was used to analyze the performance for the range of SNR. We see that the performance of the fitted models is an improvement over the baseline and is close to the results obtained for RNN, albeit with reduced complexities.

2) Comparison of ActiveFB vs. PPV Scheme: The PPV-like schemes that we consider for fitting are more general than the PPV scheme that was proposed in [10]. The basic assumption in the PPV scheme is that the difference in the transmitted symbols for intended bits $b = 0$ or 1, when the same feedback is received, is given by the parameter d_i, i.e.,

$$x_i\left(+1|c^i\right) - x_i\left(-1|c^i\right) = d_i. \tag{7}$$

This is assumed constant for the state-of-the-art PPV scheme. That is, the difference term does not depend on the feedback symbols c^i. However, we observe from the analytical approximation and the constellation plots that ActiveFB does not maintain a fixed distance between the encoded symbols as a function of the feedback. The distance between the symbols increases when the error is higher, i.e., more power is allocated for the encoded symbol if it is farther from the true value. It also means the power is suppressed for the symbol if it is close to the true value. Hence, we introduce variation in d_i as a function of feedback and modify the underlying function structure by introducing a bias in the exponent. This implies that the distance between the encoded symbols for given feedback for $b = 0, 1$ is no longer a constant. This provides further flexibility in modulating the power sent after each feedback.

V. Coding for Channels With Noisy Feedback

We now consider channels with *noisy* feedback, where the feedback encoding is crucial for combating the noise in the feedback channel. In Section V-A, we provide a brief overview of existing coding schemes, such as the Modulo-SK scheme and the DRF code, which serve as baselines for ActiveFB in Section V-B. We show that ActiveFB outperforms both baselines. In Section V-C, we provide an analytical approximation of the ActiveFB codes and interpretation, where we explain the nonlinearity of ActiveFB compared to the piecewise-linearity of the Modulo-SK scheme.

A. Existing Coding Schemes

Linear schemes for channels with passive feedback: Chance and Love in [7] and Mishra et al. [11] proposed linear schemes for channels with noisy output feedback. Chance and Love provided a concatenated code with the inner code as a linear encoding scheme that performs better than the SK scheme. Authors of [11] provided a dynamic program to solve for the linear sequential codes in closed-form.

Modulo-SK scheme for channels with active feedback: Ben-Yishai and Shayevitz in [17] generalized the SK scheme to *active* noisy feedback settings, where both Tx and Rx are allowed to employ coding and exchange signals on the fly. This should be distinguished from *passive* feedback, where no coding is allowed over the feedback channel. The previously described SK and PPV schemes belong to the passive feedback category.

A key observation in Modulo-SK is that the transmission of $\hat{\theta}_n$ over the feedback link (from Rx to Tx) can be regarded as a Joint Source Channel Coding (JSCC) problem with side information θ. More explicitly, at round n, Rx holds its estimate $\hat{\theta}_n$ and wants to communicate it with Tx, who knows θ and can use it as side information. To exploit this, the authors employ a lattice-based JSCC scheme with side information based on a more general scheme by Kochman and Zamir [22].

More precisely, Tx encodes its message into a scalar θ using PAM. In subsequent rounds, Rx computes a linear estimate of θ and feeds back an exponentially amplified version of this estimate, modulo a fixed interval. The modulo operation makes use of the fact that Tx knows θ and facilitates the essential "zoom-in" amplification without exceeding the power limit. In turn, Tx employs a suitable modulo computation and obtains the estimation error, corrupted by excess additive noise. This quantity is then properly scaled and sent over the feedforward channel to Rx. After a fixed number of rounds, Rx decodes the message using a minimum distance rule.

The Modulo-SK scheme is shown to outperform the SK scheme by a large margin, which demonstrates the advantage of utilizing the side information at the transmitter in compressing the receiver's outputs. In Section III, we show that by jointly designing (learning) the encoder, the decoder, and the feedback encoder, we can learn *nonlinear* feedback encoding

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

schemes that outperform the Modulo-SK scheme for communicating a single bit over channels with active feedback. The modulo operation in the Modulo-SK scheme, on the other hand, is piecewise linear.

Neural codes for channels with passive feedback: In [12], the authors present Deepcode, the first family of *nonlinear* codes obtained via deep learning for both noiseless and noisy *passive* feedback channels. They demonstrate a new family of RNN-driven nonlinear neural codes, which outperform the SK scheme. Deepcode progresses in two phases. In the first phase, K information bits $\mathbf{b}$ are sent raw (uncoded) over the AWGN channel ($K = 50$ is considered). In the second phase, $2K$ coded bits are generated based on the information bits $\mathbf{b}$ and (delayed) output feedback and sequentially transmitted. As for decoding, the authors propose a decoding scheme using two layers of bidirectional Gated Recurrent Units (GRU). Based on the received sequence of length $3K$, the decoder estimates K information bits $\hat{\mathbf{b}}$. While Deepcode outperforms the existing linear codes for channels with passive feedback, extending the Deepcode framework to channels with active feedback has been an open problem. In [12], the authors note that designing codes for this setting is challenging as it involves designing two encoders and one decoder jointly in a sequential manner. More recently, [13] and [14] extended the Deepcode framework, improving its performance. In [13], the authors introduce Deep Extended Feedback Code (DEF code), a generalization of Deepcode in two ways: (*a*) parity symbols are generated over longer time intervals in order to provide better error correction capability; and (*b*) high-order modulation formats are deployed to achieve increased spectral efficiency. In [14], the authors propose Deep SNR-Robust Feedback Code (DRF Code). The proposed code introduces two novelties over the previously proposed DNN-based codes: (*a*) An SNR-aware attention mechanism at the decoder which enables the reliable application of the same trained network over a wide range of SNR values; (*b*) A curriculum training with batch-size scheduling is used to speed up and stabilize training. DRF codes outperformed both Deepcode and DEF codes.

In the following, we show that by allowing the receiver to actively encode its feedback, ActiveFB (for $K = 2$ information bits) outperforms all the baselines, including DRF codes (for $K = 50$ information bits).

B. ActiveFB: Numerical Results

First, we evaluate the performance of ActiveFB under noisy feedback for different rates, rate 1/3 with $N = 3$ and rate 1/6 with $N = 6$ transmissions, by communicating a single bit $b \in \{0, 1\}$, the results of which are shown in Fig. 5 (Top and Bottom respectively). We observe that our model outperforms the baselines in both cases.

The following schemes are included as baselines.

- *ActiveFB K=1:* This is our neural model described in Section III, where K is the number of transmitted information bits. In this case, we transmit a single bit b ($K = 1$).
- *ActiveFB K=2:* Similar to the above, this is our neural model described in Section III. The difference is that we now transmit a vector $\mathbf{b} \in \{0, 1\}^2$, since $K = 2$. This time, we transmit a real symbol x_i every transmission i for a total of 6 transmissions to keep the rate as 1/3.
- *Modulo-SK K=1 [17]:* We consider Modulo-SK (described in Section V-A) when $K = 1$.
- *Modulo-SK K=13 [17]:* We also consider Modulo-SK for $K = 13$.
- *DRF Code [14]:* We include DRF Code for $K = 50$ information bits described in Section V-A.

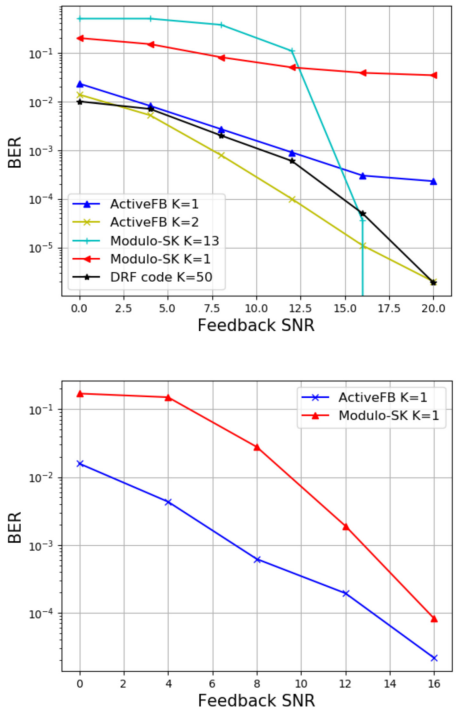

Fig. 5. ActiveFB outperforms the baselines for channels with noisy feedback for $N = 3$ rounds of communications (Top) and $N = 6$ rounds of communications (Bottom). The forward SNR is fixed to 0 dB.

From Fig. 5, we conclude that (*a*) we outperform all the baselines when the feedback SNR is moderate and (*b*) transmitting 2 bits instead of 1 results in a noticeable gain. It is interesting to note that the ActiveFB scheme for $K = 2$ outperforms DRF Code (for $K = 50$) and the Modulo-SK scheme for $K = 1$. Compared to the Modulo-SK scheme for $K = 13$, ActiveFB for $K = 2$ is comparable but is more reliable at low feedback SNR regimes. On the other hand, when feedback SNR is high, Modulo-SK with a sufficiently large K is highly reliable. We also note that generalizing ActiveFB to $K > 2$ is not straightforward. Neural coding for active feedback channels for a large K is an interesting future direction.

C. Interpretation of ActiveFB

In the previous section, we showed that ActiveFB outperforms the Modulo-SK scheme by a huge margin, as shown in Fig. 5. The interpretation of the generated codes can provide insights into the encoding process and help us formulate analytical encoding schemes that are less complex and easy to interpret.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

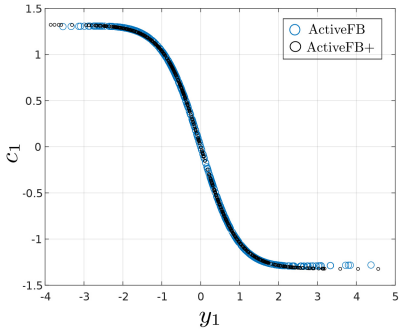

Fig. 6. Constellation plot showing the feedback encoder output at the end of round 1.

1) ActiveFB+: Derived Scheme From ActiveFB: We consider the encoder data from communication scheme with $N = 3$, forward SNR of 0 dB and several feedback SNRs to interpret the correlation between the input and the output data. We curve-fit the input data to the output data for each transmission round to get an approximate analytical expression. We then substitute it for RNN and make performance comparison.

Round 1 (Raw Bit Transmission and Likelihood Feedback): At Round 1, we verify that the encoder's output $x_1 = K_0(-2b+1)$ is a BPSK mapping of bit $b \in \{0, 1\}$ to $\{-1, 1\}$ where K_0 is chosen to match the data. The decoder receives $y_1 = x_1 + z_1$ where z_1 is the noise in the forward transmission. The feedback symbol $c_1 = \text{Normalize}(\hat{b}_1)$ is obtained from likelihood $\hat{b}_1 = P_{b|Y}(1|y_1)$. This can be interpreted as involving (a) the posterior likelihood $\hat{b}_1 = P_{b|Y}(1|y_1) = \sigma(2y_1/\sigma^2)$ and (b) the power normalization which is done by subtracting the mean and dividing by the standard deviation resulting in the constants. We show that the feedback symbol from the decoder to the encoder can be approximated as

$$c_1 = K_1 + \frac{K_5}{1 + e^{K_6 y_1 + K_9}},$$

where K_1, K_5, K_6 and K_9 are chosen from the input-output relationship of the data from the RNN.

Round 2 (Transmission of the Likelihood Error and Updated Likelihood): Fig. 7 (Top) shows the encoder's output x_2 as a linear function of the received noisy feedback $\tilde{c}_1 = c_1 + w_1$ and color-coded by the originally transmitted bit b. The encoder uses $\tilde{c}_1$ to estimate the error $\hat{e}_1$ between the bit b and the estimated likelihood after the first transmission, $\hat{e}_1 = (b - (K_1\tilde{c}_1 + K_2))$ where K_1 and K_2 are constants. It transmits this error after power normalization which can be expressed as $x_2 = K_3 + K_2\hat{e}_1$. The resulting fitted analytical function is given as $x_2 = 1.5x_1 - \tilde{c}_1$.

For the feedback transmission, the decoder receives $y_2 = x_2 + z_2$, i.e., a noisy version of the error $\hat{e}_1$ in the likelihood and computes a linear Minimum Mean Square Estimate (MMSE) estimation as $K_1 + K_2 y_2$. Now, from Fig. 7, we infer that the transmitted feedback is computed as a linear combination of the old likelihood $\hat{b}_1$ computed at the receiver at round 1 and the new likelihood function computed given the new information y_2 available at the receiver.

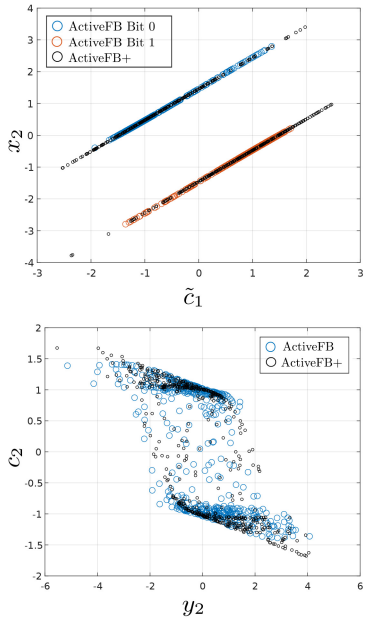

Fig. 7. The plot shows the Round 2 forward transmission (Top) and the feedback transmission (Bottom). The fitted models (black) are termed ActiveFB+ scheme for noisy feedback.

Let $\hat{b}_2$ represent the likelihood of the bit b when two symbols y_1 and y_2 are received, which can be expressed as

$$\hat{b}_2 = P_{b|Y_1,Y_2}(1|y_1, y_2) \tag{8}$$

$$= P_{b|\hat{b}_1,Y_2}\Big(1|\hat{b}_1, y_2\Big) \tag{9}$$

$$= \sigma\Big(K_1 + K_2\hat{b}_1 + K_3 y_2\Big), \tag{10}$$

where the expression inside the sigmoid function is basically estimating $b - \hat{b}_1$ from y_2 and then adding $\hat{b}_1$ (with scaling and shifting appropriately) in order to get

$$c_2 = K_1 + K_2\hat{b}_1 + K_3\sigma\Big(K_4 y_2 + K_5\hat{b}_2\Big), \tag{11}$$

which with constants can be expressed as

$$c_2 = 1.4 - 0.3c_1 - \frac{2.8}{1 + e^{3.2c_1 - 2y_2}}.$$

Round 3 (Final Transmission of the Error in Likelihood and Final Decoding): In the last round of transmission, the encoder chooses either to send the error between the actual bit b and the estimate of the likelihood $\hat{b}_2$ or send zero if the error has no effect on the final bit error performance as a means to save power, as is evident from the figure. The error is computed as a PPV-like function where the output becomes zero as it approaches 0 as the estimate at the receiver moves closer to the actual bit that was transmitted. The analytical function is modeled using both the previously received feedback symbols $\tilde{c}_1$ and $\tilde{c}_2$. The final transmitted symbol can be expressed as

$$x_3 = K_1 W + K_2\tilde{c}_1 + K_3\tilde{c}_2 + \frac{WK_5}{1 + e^{WK_6\tilde{c}_1 + WK_7\tilde{c}_2 + K_9}},$$

where K_i's are the constants obtained from fitting and W is the transmitted message from the first round.

Upon receiving the final transmission from the encoder, the receiver repeats the process that was done at the previous

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

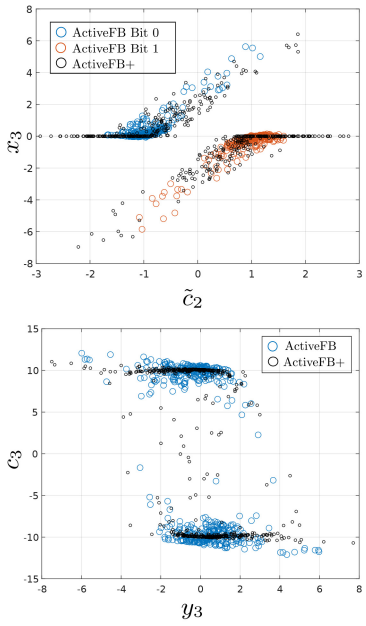

Fig. 8. Final encoded symbol is shown in the top figure while the bottom figure shows the final decoded output.

instant. It estimates the error in the likelihood from the transmitted symbol and adds it to the previous likelihood estimate. So we get one expression as

$$c_3 = K_1 + K_2 y_1 + K_3 y_2 + K_4 y_3 - \frac{K_5}{1 + e^{5.2c_2 - 1.6y_3}}. \quad (12)$$

The final decoding involves applying a sigmoid function on c_3 to obtain the final decoded bit. The fitting results for the third round of transmission are plotted in Fig. 8. The colored points denote the data obtained from the RNN plotted between the input on the x-axis and output on the y-axis. The fitting results are shown in black which closely resemble the relation shown by the RNN data (Fig. 8).

Results: In Fig. 9, we plot the BER of the proposed activeFB scheme, Modulo-SK, and the analytical scheme obtained by fitting the learned scheme to a predetermined function structure discussed above.

The fitting was conducted at forward SNR of 0 dB for various feedback SNRs. The comparison shows that the fitted curves, termed "Fitted", are better than the state-of-the-art schemes.

The significant gap between the ActiveFB and ActiveFB+ scheme was investigated by doing an ablation study where the encoders and decoders are sequentially replaced with the corresponding analytical model and the studying the effect on the BER performance.

We deduce that the intermediate stage, where the entire network except for the decoder at the final round is replaced by their analytical counterpart, performs much better than the case when the entire network is replaced. This implies that the PPV-like scheme is not sufficient to describe the relationship between the input and the output of the RNN for the last stage of the transmission. This is showcased in Fig. 9 thorugh the plot termed "Fitted*", which shows that the performance is much better than the case when the whole network is replaced.

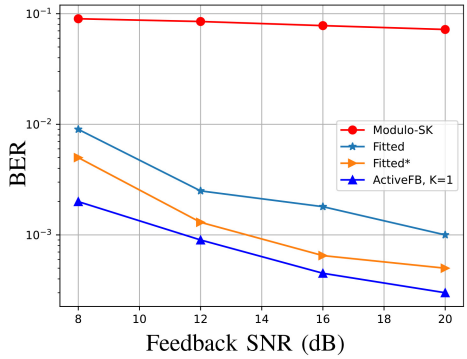

Fig. 9. The plot shows the BER comparison between the original RNN(ActiveFB), the Modulo-SK, and the fitted analytical models (ActiveFB+) for a noisy feedback regime across different feedback SNR. The blocklength was assumed to be $N = 3$, and the sum power was assumed as 0.5 dB. "Fitted*" plot shows the intermediate state in our ablation study where all parts except the decoder in the last round are replaced with analytical models.

2) Comparison of ActiveFB vs. Modulo-SK: An important interpretation to make is how ActiveFB is different from the baseline Modulo-SK scheme. A key difference is in the non-linear nature of the ActiveFB and the piecewise-linear nature of the Modulo-SK scheme.

As shown in Fig. 8, the learned neural codes are *nonlinear*, in contrast to the Modulo-SK codes, which are piece-wise linear codes. Based on the interpretation analysis of the learned *nonlinear* neural codes, we observe that the receiver is approximately sending its estimate of the log-likelihood-ratios, or more accurately, of the probability that $b = 1$, and then the transmitter creates a biased constellation based on the noisy feedback.

VI. Over-the-Air Demonstration

In the preceding sections, we developed codes for AWGN channels with feedback and focused on the theoretical analysis, deriving analytical coding schemes and building insights based on neural results. In this section, we focus on practical aspects, which are complementary to the theoretical analysis. We demonstrate our ActiveFB schemes for fading channels and in a hardware setup (software-defined radios) to study the performance in a practical communication scenario.

A. Rician Fading Channel

As a precursor to the over-the-air experiment, we first consider Rician fading channels parametrized by F and σ. A general description of a Rician fading channel is considered in which a channel is comprised of both a line-of-sight (LOS) and non-line-of-sight (NLOS) component. The amplitude of the LOS and NLOS components is dictated by F. The channel is defined as $y = hx + z$, where $z \sim \mathcal{N}(0, \sigma^2)$ and

$$h = \left| \sqrt{\frac{F}{(F+1)}}(1 + 1\mathrm{i}) + \sqrt{\frac{1}{(F+1)}} h^{\mathrm{NLOS}} \right|,$$

and h^{NLOS} is distributed as $\mathcal{CN}(0, I)$.

We apply this channel model to ActiveFB (described in Section III) and Modulo-SK (described in Section V-A). In

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

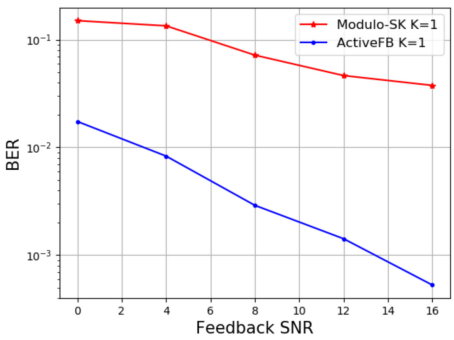

Fig. 10. ActiveFB outperforms Modulo-SK for the Rician fading channel. We considered $F = 10$ and set the forward SNR as 0 dB.

Fig. 10, we plot the BER vs. the feedback channel SNR, where we fix the forward channel SNR as 0 dB for both schemes, from which we observe that the ActiveFB outperforms Modulo-SK by a large margin.

B. Over-the-Air Channel

The channel encountered in practical communication systems are not AWGN due to physical world effects like scattering and reflections that cause unwanted fading. The noise affecting the transmitted message is neither additive nor Gaussian.

We can represent the received signal in an Over-the-Air (OTA) channel setup as

$$y = hx + z, \tag{13}$$

where y is the received value, x is the transmitted symbol, h is the Rician channel coefficient, and $z \sim N(0, \sigma_z^2)$ is the AWGN noise. The channel coefficient h represents both the Line of Sight (LOS) and the non-LOS component of the Rician fading channel and can be characterized as $h = h_{\text{LOS}} + h_{\text{NLOS}}$, where the line-of-sight component $h_{\text{LOS}} = \mu_h$ has a constant value μ_h while $h_{\text{NLOS}} \sim N(0, \sigma_h^2)$ is a zero-mean Gaussian random variable with variance σ_h^2.

For over-the-air channels, one can estimate the Rician channel parameter h and the noise parameter z based on the channel realizations by estimating σ_z^2, μ_h and σ_h^2.

Setup: As depicted in Fig. 11, our setup consists of two separate N200 USRPs as transmitter and receiver with antennas to communicate over the air. They are kept at a distance of around 3 meters from each other. The gain parameters are adjusted on the boards to simulate any desired channel SNR. The USRPs have two separate computers on the backend to perform data generation, encoding and the decoding processes.

We implement a communication system based on the 802.11a WiFi standard [23]. We use Orthogonal Frequency Domain Multiplexing (OFDM) modulation and demodulation for the baseband data processing as it is well-suited for wideband applications and all modern practical setups use OFDM signaling. The end-to-end OFDM setup is shown in Fig. 11. For synchronization, we use Short Training Sequences (STS) and the Long Training Sequences (LTS) which aid in the determination of symbol boundaries and are used for channel estimation and frequency offset compensation.

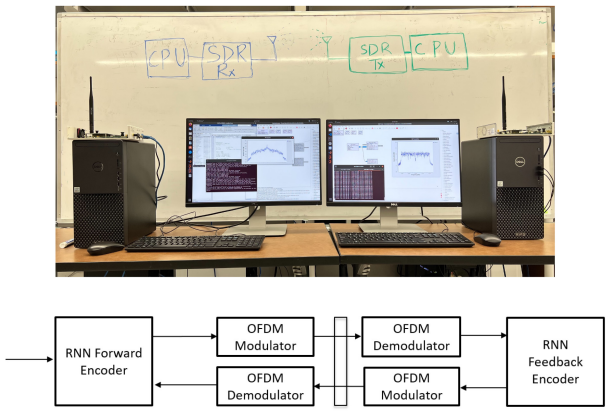

Fig. 11. (Top) Feedback setup showing the SDRs as Tx and Rx with the backend computers. (Bottom) Block diagram showing the implementation to illustrate the active feedback scheme with the over-the-air environment.

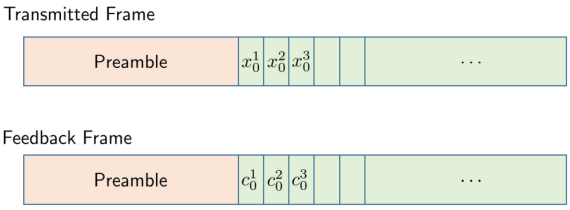

Fig. 12. Frame structures for the forward and feedback transmissions are shown. The beginning of each frame is a preamble for synchronization, offset correction, and channel estimation.

Frame arrangement: The frame structure for arranging the forward and feedback data is derived from the standard as shown in Fig. 12. We follow batch processing of the feedback wherein the feedback of all symbols in a frame is transferred after the entire frame's forward transmission is completed to ensure channels are uncorrelated. Each frame begins with a preamble to aid in synchronization, frequency, phase offset corrections, and channel estimation. The details of the preamble constituents can be found in the standard [23].

Channel Estimation: The first step in our training methodology is to determine the statistical parameters σ_z^2, μ_h, and σ_h^2 so that we can describe the channel precisely. We do a number of captures for the forward and the feedback channel and perform channel estimation to get a set of realizations of the channel h and noise z. Then, we plot the histogram of these realizations and fit it to a customized distribution with a known mean and variance to obtain the statistics of the captured channel data. The histograms of the raw channel data and the synthetic channel data are shown in Fig. 13. Finally, we use the learned statistics to define the channel model and use it in the training process.

Training: The next step is using the learned channel model in the training of the RNN. The training procedure of the RNN remains the same as was discussed in Section III. The trained encoders are then used in the setup for encoding and decoding data and the results are evaluated.

Remark 1: We approximate the statistics of the channel and then simulate the channel in the training process instead of including the actual over-the-air channel. This is because (*a*) the online training with the over-the-air channel requires a lot of captures which in turn would extend the time required

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

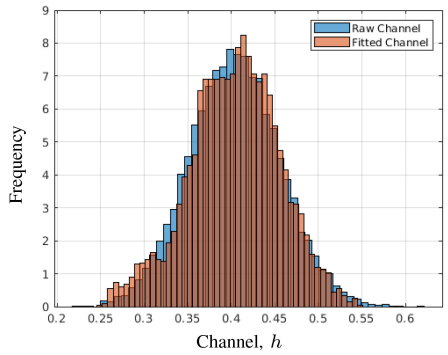

Fig. 13. Histogram plot of the raw channel data and the data generated using a custom Rician distribution captured at a particular subcarrier. It is worth noting that the channel statistics change with frequency; therefore, this process is done separately at each subcarrier.

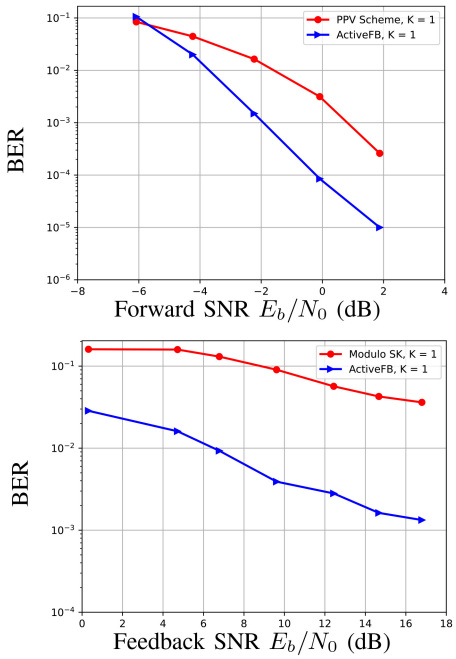

Fig. 14. The plot compares the BER of our proposed scheme (ActiveFB) and the existing schemes (PPV scheme, Modulo SK) in an over-the-air channel when implemented on a SDR hardware setup. (Top) The figure shows the performance in a noiseless regime. (Bottom) The figure shows the performance in a noisy feedback regime when the forward SNR $= E_b/N_0 =$ 0 dB.

for training and (b) we cannot backpropagate through the over-the-air channel measurement, which in turn requires the encoder to be learned via reinforcement learning, which is more time-consuming than the backpropagation.

C. Results

We consider both noiseless and noisy feedback regimes to demonstrate the performance improvement over the state-of-the-art encoding schemes. In Fig. 14 (Top), we plot BER as a function of the forward SNR, E_b/N_0, for channels with noiseless feedback while in Fig. 14 (Bottom), we plot BER for channels with noisy active feedback against the feedback SNR with forward SNR set at 0 dB.

Estimating the SNR of channels accurately is a daunting task. Therefore, we estimate the SNR per bit, i.e., E_b/N_0, by sending a random set of messages with unit power and determining the variance of noise with $N_0 = \boldsymbol{E}\left[\left(x - y_{eq}\right)^2\right]$, where y_{eq} are the received values after equalization. This is an approximate method to find the value for the SNR of the transmitted bit. This is done at all the channel settings before attempting communication to characterize the channel and to use it as a common metric for comparison of BER across different algorithms. We consider the PPV scheme as the baseline for the noiseless feedback; we consider the Modulo-SK scheme as the baseline for the noisy active feedback. In both cases, ActiveFB clearly outperforms these baselines.

VII. Conclusion and Discussion

We proposed ActiveFB, a family of RNN-based coding schemes for channels with active feedback. Here the decoder actively encodes its received values into the feedback signal, and the encoder uses the feedback to generate the subsequent transmission symbols. We demonstrated the effectiveness of the neural network technique in achieving superior reliability over state-of-the-art schemes. ActiveFB outperforms the state-of-the-art schemes, such as the PPV scheme for channels with noiseless feedback and the Modulo-SK scheme for channels with noisy feedback.

In order to compare the performance of the ActiveFB scheme against the best analytical baselines for channels with noiseless feedback, we developed a novel technique to optimize the power allocation for the PPV scheme based on dynamic programming which significantly improves the reliability. We showed that ActiveFB still outperforms the power-optimized PPV scheme, reflecting that ActiveFB is different from the PPV scheme.

We demonstrated that a new scheme termed ActiveFB+, approximated from ActiveFB as a *modified* version of the PPV scheme, closely mimics the RNN-based ActiveFB scheme and outperforms all the existing baselines both for noiseless and noisy feedback regimes. This approximation provides several benefits. First, we have an analytical scheme significantly less complex than the original RNN that gives similar performance. Besides being computationally efficient, the analytical expression makes more sense than the RNN parametrizations. Second, we use these analytical approximations to provide insights and interpretations of the ActiveFB codes, e.g., explain the difference between ActiveFB and the PPV scheme.

Finally, we also demonstrated the improved reliability of our ActiveFB schemes by implementing them on an end-to-end communication system with SDRs in the real-time over-the-air environment. This is the first over-the-air hardware demonstration of neural codes for interactive channels with feedback to the best of our knowledge.

There are several interesting open problems for future research. We mainly focused on short blocklength regimes. Surprisingly, ActiveFB for a very short blocklength regime is shown to outperform the state-of-the-art baselines designed for longer blocklengths for both channels with noisy feedback and channels with noiseless feedback. Whether one could

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.

extend ActiveFB to longer blocklength regimes is a challenging but very interesting open problem. On a related note, for longer blocklength regimes, we conjecture that architectural choices might be more crucial. Various architectures, such as the transformer architecture, can be considered and compared. For example, in [15], the authors introduce AttentionCode, a new class of deep learning based feedback codes designed by a variation of the transformer architecture. As a follow-up, [16] presents the Generalized Block Attention Feedback (GBAF) code, a generalization of AttentionCode that addresses some limitations of existing neural designs, such as communication overhead and a limited set of feasible rates. GBAF is shown to outperform AttentionCode and DRF codes by a large margin as it strongly leverages the blocklength gain at the cost of increased complexity for channels with noisy output feedback. Generalizing the GBAF code and its variants to the *active feedback* setting and developing an analyzable code that leverages the blocklength gain are left as very interesting open problems. Finally, another interesting open problem to study is to characterize how the approximated analytical codes vary depending on the system parameters like the forward and feedback noise variances.

ACKNOWLEDGMENT

The author Hyeji Kim would like to thank Assaf Ben-Yishai and Ofer Shayevitz for fruitful discussions.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.
[2] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 1173–1177.
[3] C. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 8–19, Sep. 1956.
[4] J. Schalkwijk and T. Kailath, "A coding scheme for additive noise channels with feedback–I: No bandwidth constraint," *IEEE Trans. Inf. Theory*, vol. 12, no. 2, pp. 172–182, Apr. 1966.
[5] J. Schalkwijk, "A coding scheme for additive noise channels with feedback–II: Band-limited signals," *IEEE Trans. Inf. Theory*, vol. 12, no. 2, pp. 183–189, Apr. 1966.
[6] R. G. Gallager and B. Nakiboğlu, "Variations on a theme by Schalkwijk and Kailath," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 6–17, Jan. 2010.
[7] Z. Chance and D. J. Love, "Concatenated coding for the AWGN channel with noisy feedback," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6633–6649, Oct. 2011.
[8] Y. H. Kim, A. Lapidoth, and T. Weissman, "The Gaussian channel with noisy feedback," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007, pp. 1416–1420.
[9] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, vol. 3, pp. 37–46, 1955.
[10] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Minimum energy to send k bits through the Gaussian channel with and without feedback," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4880–4902, Aug. 2011.
[11] R. Mishra, D. Vasal, and H. Kim, "Linear coding for AWGN channels with noisy output feedback via dynamic programming," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2021, pp. 13–18.
[12] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," in *Advances in Neural Information Processing Systems*, vol. 31. Red Hook, NY, USA: Curran, 2018.
[13] A. R. Safavi, A. G. Perotti, B. M. Popovic, M. B. Mashhadi, and D. Gunduz, "Deep extended feedback codes," 2021, *arXiv:2105.01365*.
[14] M. B. Mashhadi, D. Gunduz, A. Perotti, and B. Popovic, "DRF codes: Deep SNR-robust feedback codes," 2021, *arXiv:2112.11789*.
[15] Y. Shao, E. Ozfatura, A. Perotti, B. Popovic, and D. Gunduz, "AttentionCode: Ultra-reliable feedback codes for short-packet communications," 2022, *arXiv:2205.14955*.
[16] E. Ozfatura, Y. Shao, A. Perotti, B. Popovic, and D. Gunduz, "All you need is feedback: Communication with block attention feedback codes," 2022, *arXiv:2206.09457*.
[17] A. Ben-Yishai and O. Shayevitz, "Interactive schemes for the AWGN channel with noisy feedback," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2409–2427, Apr. 2017.
[18] N. Devroye et al., "Interpreting deep-learned error-correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022, pp. 2457–2462.
[19] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "LEARN codes: Inventing low-latency codes via recurrent neural networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.
[20] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: Symbol detection using a deep learning based Viterbi algorithm," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–5.
[21] Y. He, J. Zhang, S. Jin, C.-K. Wen, and G. Y. Li, "Model-driven DNN decoder for turbo codes: Design, simulation, and experimental results," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6127–6140, Oct. 2020.
[22] Y. Kochman and R. Zamir, "Joint Wyner–Ziv/dirty-paper coding by modulo-lattice modulation," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4878–4889, Nov. 2009.
[23] *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2016, 2016.

Karl Chahine (Graduate Student Member, IEEE) received the B.Eng. degree in computer and communications engineering from the American University of Beirut in 2020. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Texas at Austin, under the supervision of Dr. H. Kim. His research interests include machine learning, information theory, and wireless communications.

Rajesh Mishra received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology Warangal, Warangal, India, in 2011, and the master's degree in electrical engineering from the Indian Institute of Technology Bombay, India, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Texas at Austin. His research interests include information theory, mean field games, and machine learning in communications.

Hyeji Kim received the Ph.D. degree in electrical engineering from Stanford University in 2016. She is an Assistant Professor with the Department of Electrical and Computer Engineering, The University of Texas at Austin. She worked as a Postdoctoral Researcher with the University of Illinois at Urbana–Champaign from 2016 to 2018. She worked as a Researcher with Samsung AI Research Cambridge from 2018 to 2020. At UT Austin, she is a member of the Wireless Networking and Communications Group, 6G@UT, and the Machine Learning Laboratory. Her research interests include information theory, coding theory, machine learning, and wireless communications.

Authorized licensed use limited to: University of Texas at Austin. Downloaded on October 01,2023 at 22:04:02 UTC from IEEE Xplore. Restrictions apply.