On non-linearities of simple learned AWGN feedback codes

Y. Zhou, N. Devroye, Gy. Turán*, and M. Žefran
University of Illinois Chicago, Chicago, IL, USA, *HUN-REN-SZTE Research Group on AI
{yzhou238, devroye, gyt, mzefran}@uic.edu

***Abstract*—Several researchers have used deep learning to obtain novel feedback codes. Two such codes for AWGN channels with passive (possibly noisy) output feedback are Deepcode which employs a bit-by-bit rate 1/3 encoder, and Lightcode, which is a symbol-by-symbol code inspired by the Schalkwijk-Kailath (SK) scheme. Here, we build on prior work to interpret these codes by 1) providing the optimal maximum a posteriori (MAP) decoder for our simple non-linear interpretable encoder of a single-bit, two-round code that accurately approximates both single-bit Deepcode and Lightcode. This non-linear interpretable coding scheme, which mimics these codes, turns out to resemble both the functional form and performance of the Polyanskiy-Poor-Verdu (PPV) single bit feedback scheme that minimizes energy transmission asymptotically. 2) We extend our non-linear interpretable code to support more than one bit and two rounds, again providing an optimal MAP decoder. This remarkably simple and power-efficient nonlinear scheme provides insight into Lightcode.**

I. Introduction

Analytically constructed feedback codes [1]–[6] have been proposed over the past few decades. Among these, for passive feedback, Schalkwijk and Kailath introduce a linear coding scheme, known as the SK scheme, which achieves a doubly exponential error exponent [1]. Ankireddy et al. propose a variant of the SK scheme, the POWERBLAST scheme, in [6], which differs from the SK scheme in the final round of transmission. Additionally, Polyanskiy et al. introduce the PPV scheme, a nonlinear feedback code designed for single-bit communication, which asymptotically achieves zero error probability with minimum energy [5]. For large message bit lengths K ($K \geq 100$), a stop-feedback code – where feedback is used solely to signal the end of transmission – is sufficient [5]. For smaller K, refining the noise estimate with output feedback proves beneficial.

Deep learning has also been applied to develop learned error-correcting feedback codes (DL-ECFCs) [6]–[13]. These codes shine in low forward signal-to-noise ratio (SNR) and short block lengths over additive white Gaussian noise (AWGN) channels with (possibly noisy) output feedback. DL-ECFCs can be classified into two types: *bit-by-bit* [7]–[10], represented by Deepcode, where one bit of the message is transmitted at a time, and *symbol-by-symbol* [6], [12], [13], as in the current state-of-the-art Lightcode, where the entire message of length K is mapped to a symbol for refinement. Previous interpretations of these learned codes [6], [7], [14], [15], show that both schemes learn the same power-efficient non-linearity for the encoder. When forward noise pushes the received signal in the "correct" direction (correct decoding even without error correction), no power is wasted on a second transmission (expressed mathematically as a non-linear indicator function). If the noise pushes the signal in the "wrong" direction, the scaled noise is sent in the second round. This pattern continues in subsequent rounds, correcting only the noises that impact decoding.

Contribution: In this work, we present an analytically interpretable approximation of the encoder for the learned codes, Deepcode and Lightcode. It is power-efficient and, as yet unnoticed, resembles the functional form and performance of the optimized power PPV scheme in single-bit, two-round transmission. Although originally designed for noiseless feedback, it appears the PPV scheme is robust to noisy feedback, achieving performance comparable to learned codes. We also provide the corresponding MAP decoder for our nonlinear analytical encoder approximations. While the PPV scheme based on log-likelihood ratios (LLR) is difficult to generalize to larger message lengths $K \geq 2$, we extend our non-linear interpretable encoder to support larger K with a MAP decoder, achieving performance similar to Lightcode. Unlike learned codes with numerous parameters, our interpretable non-linear scheme is analytically constructed.

Notation: Random variables are denoted by capital letters, specific instances by lowercase letters, and vectors in boldface. The notation $\{x_i\}_{i=1}^N$ represents $\{x_1, \ldots, x_N\}$. Probability is represented by $\mathbb{P}(\cdot)$, and expectation by $\mathbb{E}(\cdot)$. The symbol $\mathbb{F}_2$ denotes the finite field with elements 0 and 1, while $\mathbb{R}^n$ represents n-dimensional real vectors. The function $\mathbb{I}(x)$ equals 1 if $x \geq 0$ and 0 otherwise. $Q(x)$ is the complementary distribution function of $\mathcal{N}(0,1)$, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du$. Given two integers a and b with $a < b$, $[a:b]$ represents a sequence of integers $[a, a+1, \ldots, b]$.

II. System Model

We consider the point-to-point AWGN channel with passive feedback, as shown in Fig. 1. The transmitter sends a message of length K, denoted as $\mathbf{M} = \{M_i\}_{i=1}^K \in \mathbb{F}_2^K$, to the receiver

This work was supported by NSF awards 2217023, 2240532 and by the AI National Laboratory Program (RRF-2.3.1-21-2022-00004). The contents of this article are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

using the channel N times. The code rate is defined as $R = K/N$. At channel use $i \in [1:N]$, the channel output is

$$Y_i = X_i + N_i \tag{1}$$

where $N_i \sim \mathcal{N}(0, \sigma_f^2)$ represents independent and identically distributed (i.i.d.) Gaussian noise, and $X_i \in \mathbb{R}$ is the transmitted symbol at time i, subject to the average power constraint $\frac{1}{N}\mathbb{E}\left(\sum_{i=1}^{N} X_i^2\right) \leq P$. The receiver sends the channel outputs to the transmitter through a feedback link in a causal manner. We term this "output feedback". For noiseless output feedback, the feedback is identical to the channel outputs, $\tilde{Y}_{i-1} = Y_{i-1}$. For noisy (passive) output feedback, the received feedback is corrupted by i.i.d. Gaussian noises, resulting in $\tilde{Y}_{i-1} = Y_{i-1} + \tilde{N}_{i-1}$, for $\tilde{N}_i \sim \mathcal{N}(0, \sigma_{fb}^2)$. We define $\text{SNR}_f = \frac{P}{\sigma_f^2}$ for the forward channel and $\text{SNR}_{fb} = \frac{P}{\sigma_{fb}^2}$[1] for the feedback channel, respectively. We assume that the channel statistics σ_f are perfectly known at both the transmitter and receiver.

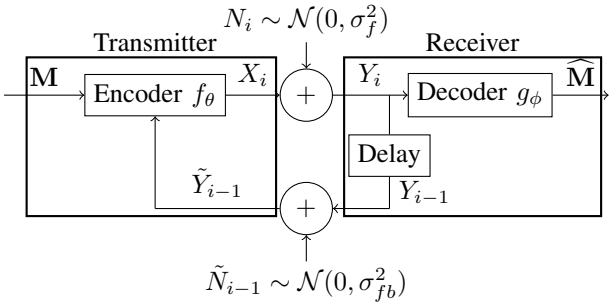

Fig. 1: AWGN channel with passive feedback

The encoder f_θ and decoder g_ϕ implicitly defined in Fig. 1 can be implemented either analytically or parameterized using neural networks (DL-ECFCs). Existing DL-ECFCs transmit messages in two ways: bit-by-bit or symbol-by-symbol.

Bit-by-bit: For bit-by-bit transmission [7]–[10], one bit of information is transmitted at a time[2],

$$\mathbf{X}_i = f_\theta\left(M_i, \{\tilde{Y}_j\}_{j=1}^{i-1}\right), \quad i \in [1:K] \tag{2}$$

For instance, Deepcode [7] is the first bit-by-bit DL-ECFC with a code rate $1/3$. Its encoding process involves two phases. In the first phase, each bit, modulated with BPSK and denoted by $X_{i,1} = 2M_i - 1$ is transmitted uncoded. In the second phase, the encoder sequentially generates *two* parity bits, $X_{i,2}$ and $X_{i,3}$, using the current message bit, the feedback from the first phase, and the delayed feedback from previous transmissions of parity bits. The transmission is described by $X_{i,k} = f_{\theta,k}\left(M_i, \tilde{Y}_{i,1}, \tilde{Y}_{i-1,2}, \tilde{Y}_{i-1,3}\right)$, where $k \in \{2,3\}$. Here, θ is parameterized by a recurrent neural network (RNN). The longer the dependency it builds between codewords, the better the error correction performance achieved [14], [15].

[1]A more precise definition would be $(P+\sigma_f^2)/\sigma_{fb}^2$. However, to maintain consistency with previous papers on DL-ECFCs, we use the current definition.

[2]$\{\tilde{Y}_j\}_{j=1}^{i-1}$ implicitly depend on the previous message bits.

Symbol-by-symbol: For symbol-by-symbol transmission [6], [12], [13], message $\mathbf{M} \in \mathbb{F}_2^K$ is mapped to a symbol which is iteratively refined as:

$$X_i = f_\theta(\mathbf{M}, \{\tilde{Y}_j\}_{j=1}^{i-1}), \quad i \in [1:N]. \tag{3}$$

This approach, inspired by the SK scheme [1], achieves rate K/N which is not restricted to $1/3$. Lightcode [6] is the current state-of-the-art in symbol-by-symbol DL-ECFCs. At channel use i, the encoder takes the message bits and feedback from previous $i-1$ rounds as the input, $X_i = f_\theta(\mathbf{M}, \tilde{Y}_1, \ldots, \tilde{Y}_{i-1}, 0, \ldots, 0)$, where zero-padding ensures a constant input dimension. Each transmitted symbol is normalized and power-optimized to satisfy the average power constraint and enhance performance. Lightcode employs feedforward feature extractors and multilayer perceptrons. The decoder maps the received noisy codewords to the estimated message bits $\widehat{\mathbf{M}} = g_\phi\left(\{Y_i\}_{i=1}^N\right)$. The performance metric is the block error rate $\text{BLER} := \mathbb{P}(\mathbf{M} \neq \widehat{\mathbf{M}})$.

III. Related Work

We now review analytical coding schemes for the AWGN channel with noiseless passive feedback. The SK scheme achieves doubly exponential error decay by minimizing the mean square error, while POWERBLAST differs from it only in the final round. The PPV scheme uses a nonlinear coding approach based on the log-likelihood ratio (LLR).

A. Schalkwijk-Kailath (SK) scheme

The SK scheme is a linear code designed for noiseless output feedback [1]. Initially, the message $\mathbf{M} \in \mathbb{F}_2^K$ is PAM modulated and transmitted over the forward channel. In subsequent rounds, the transmitter sends the estimation error made by the receiver, allowing the receiver to iteratively refine its estimate. For a fixed code rate, at low forward SNR, messages with smaller K perform better due to the larger distance between constellation points. Conversely, at high SNR, messages with larger K, and thus more channel uses N achieve better performance.

Initialization: The message $\mathbf{M}$ is mapped to a PAM symbol $\Theta \in \{\pm 1\eta, \pm 3\eta, \ldots, \pm(2^K-1)\eta\}$, where $\eta = \sqrt{\frac{3}{2^{2K}-1}}$ ensures the unit power constraint. The estimated message at receiver at time i is denoted by $\widehat{\Theta}_i$ with the error defined as $\epsilon_i = \widehat{\Theta}_i - \Theta$. The mean square error is given by $D_i = \mathbb{E}(\epsilon_i^2)$.

Encoding: In the first round, the transmitter sends with power P, $X_1 = \sqrt{P}\Theta$. In the subsequent rounds ($i \geq 2$), the encoder transmits the estimation error scaled appropriately, $X_i = \sqrt{\frac{P}{D_{i-1}}}\epsilon_{i-1} = \sqrt{\frac{P}{D_{i-1}}}\left(\widehat{\Theta}_{i-1} - \Theta\right)$.

Decoding: At the end of round 1, the estimate of the transmitted symbol at the receiver based on Y_1 can be either the minimum-variance unbiased estimator (MVUE), $\widehat{\Theta}_1 = \frac{Y_1}{\sqrt{P}}$ or the linear minimum mean square error estimator (LMMSE), $\widehat{\Theta}_1 = \frac{\sqrt{P}}{P+\sigma_f^2}Y_1$. In the subsequent rounds, the receiver estimates the error using the LMMSE estimator:

$$\hat{\epsilon}_{i-1} = \frac{\sqrt{PD_{i-1}}}{P+\sigma_f^2}Y_i, \quad i \in [2:N] \tag{4}$$

The receiver then updates its estimate as $\widehat{\Theta}_i = \widehat{\Theta}_{i-1} - \hat{\epsilon}_{i-1}$. Finally, the receiver applies a minimum distance decoder to $\widehat{\Theta}_N$, mapping it to the nearest PAM constellation point.

Performance: Using the biased LMMSE at the end of round 1 improves BLER performance for finite block lengths. The probability of error (BLER) based on the MVUE (more tractable than with the LMMSE) for a rate of K/N is

$$\text{BLER}_{sk} = 2(1-2^{-K})Q\left(\sqrt{\frac{3\text{SNR}_f(1+\text{SNR}_f)^{N-1}}{2^{2K}-1}}\right) \quad (5)$$

where SNR_f is expressed on a linear scale.

For a code rate of K/N, the first $N-1$ rounds of POWERBLAST follow the same procedure as the SK scheme. In the final round N, instead of transmitting the estimation error directly, POWERBLAST sends a discrete symbol representing the error in the PAM index estimate [6] $X_N = \frac{\sqrt{P}}{\sigma_u}(\hat{M} - M)$, where $\hat{M}, M \in [1 : 2^K]$ denote the PAM indices corresponding to $\widehat{\Theta}_{N-1}$ and the true message Θ, respectively, and $\sigma_u^2 = \mathbb{E}[(\hat{M} - M)^2]$. POWERBLAST demonstrates significantly better BLER performance than the SK scheme for rates $3/6$ and $3/9$, as shown in [6].

B. Polyanskiy, Poor, and Verdú (PPV) scheme

The PPV scheme is state-of-the-art for transmitting a single bit [5] over channels with noiseless output feedback. The encoder computes the estimation error using LLR and scales it to satisfy the power constraint.

Encoding: Specifically, for a single bit $M \in \mathbb{F}_2$, with BPSK modulation $W = 2M - 1 \in \{-1, 1\}$, the encoding function at the i-th round is defined as:

$$f_i(W, \{Y_j\}_{j=1}^{i-1}) = \frac{W d_i}{1 + e^{W S_{i-1}}} \quad (6)$$

where $S_{i-1} = \log \frac{\mathbb{P}\left(W=+1|\{Y_j\}_{j=1}^{i-1}\right)}{\mathbb{P}\left(W=-1|\{Y_j\}_{j=1}^{i-1}\right)}$ is the LLR, and d_i is a coefficient related to the power constraint. With a constant d_i, the PPV scheme has been shown to achieve zero error probability with the minimum energy per bit as the channel uses $N \to \infty$ [5]. For finite block lengths, optimizing d_i yields further BLER improvement using dynamic programming [11].

Decoding: After N rounds, the receiver decodes the message $\widehat{W}$ using the final LLR, S_N.

Performance: The BLER after N rounds can be represented as a recursion function [11]:

$$\text{BLER}_{ppv} = 1 - Q\left(-\frac{\mu_N}{\sigma_N}\right) \quad (7)$$

where $\mu_i = \mu_{i-1} + \frac{d_i^2}{2\sigma_f^2}$ and $\sigma_i^2 = \sigma_{i-1}^2 + \frac{d_i^2}{\sigma_f^2}$[3]. By symmetry, it is sufficient to analyze the case where $W = +1$. The BLER after N rounds can be expressed as $\mathbb{P}\left(S_N < 0 \mid W = +1\right)$. Given $W = +1$, the LLR at the i-th round can be written as $S_i = S_{i-1} + \frac{1}{2\sigma_f^2} d_i^2 + \frac{1}{\sigma_f^2} d_i N_i$, where $N_i \sim \mathcal{N}(0, \sigma_f^2)$. Since a linear combination of Gaussian random variables remains Gaussian, $S_N \sim \mathcal{N}\left(\frac{1}{2\sigma_f^2}\sum_{i=1}^N d_i^2, \sigma_f^2 \sum_{i=1}^N (\frac{d_i}{\sigma_f^2})^2\right)$. Thus, the BLER can be expressed as in (7).

[3]In [11], there is a minor typo in the expression $\sigma_i^2 = \sigma_{i-1}^2 + (\frac{d_i}{\sigma_f^2})^2$ as it omits the noise variance.

IV. Interpretable non-linear codes for $K = 1, N = 2$

In this section, we consider the simplest case: transmitting a single bit ($K = 1$) using the channel $N = 2$ times. Drawing insights from the interpretation of Deepcode and Lightcode, we design a very good approximation of the nonlinear encoder and derive the corresponding decoder. Interestingly, while this has not been explicitly observed before, this approximation closely resembles the PPV scheme.

For $K = 1$, the first round of transmission is the same for both bit-by-bit and symbol-by-symbol encoding, using BPSK modulation $Y_1 = \lambda_1 (2M - 1) + N_1$, with $\text{SNR}_1 = \frac{\lambda_1^2}{\sigma_f^2}$.

A. Linear encoder and linear decoder

For finite block lengths, the linear encoder follows the SK scheme with optimized power allocation for each round. We first analyze the MVUE at the end of round 1. For the i-th transmission, where $i \in [2 : N]$, the encoding is $X_i = \frac{\lambda_i}{\sqrt{D_{i-1}}}\epsilon_{i-1}$, with power λ_i^2, corresponding to $\text{SNR}_i = \frac{\lambda_i^2}{\sigma_f^2}$. Consequently, the BLER in (5) becomes $\text{BLER}_{linear} = 2(1-2^{-K})Q\left(\sqrt{\frac{3\text{SNR}_1 \prod_{i=2}^N (1+\text{SNR}_i)}{2^{2K}-1}}\right)$. After N rounds, the optimization problem involves minimizing BLER under the average power constraint $\sum_{i=1}^N \lambda_i^2 = NP$. This requires optimizing the power coefficients λ_i, which can be solved numerically using dynamic programming.

In this specific case, $X_2 = \frac{\lambda_2}{\sigma_f} N_1$, we can derive the maximum a posterior (MAP) decoder:

$$y_1 - \frac{\lambda_2 \sigma_f}{\lambda_2^2 + \sigma_f^2} y_2 \underset{\widehat{M}=1}{\overset{\widehat{M}=0}{\lessgtr}} 0 \quad (8)$$

For single bit, the BLER reaches its minimum $Q\left(\frac{P}{\sigma_f^2} + \frac{1}{2}\right)$ when $\lambda_1^2 = P + \frac{\sigma_f^2}{2}$. With optimized power allocation, the scheme is called "Linear MVUE" if the MVUE is used, and "Linear LMMSE" if the LMMSE is used (at round 1's end).

B. Nonlinear encoder and decoders: interpretable schemes and PPV similarities

The scatter plot of the encoder output in round 2 versus the noise in round 1 shows that, for first-order error correction of Deepcode, the second round of Lightcode, and the PPV scheme, the transmitter sends information in the second round only *when necessary*, as illustrated in Fig. 2. Notice the visual similarity between all three functions: this suggests that this functional form lies on the boundary of the energy / performance (as measured by BLER). This is the first time that the similarity of the learned schemes is compared with that of the analytical and known PPV scheme.

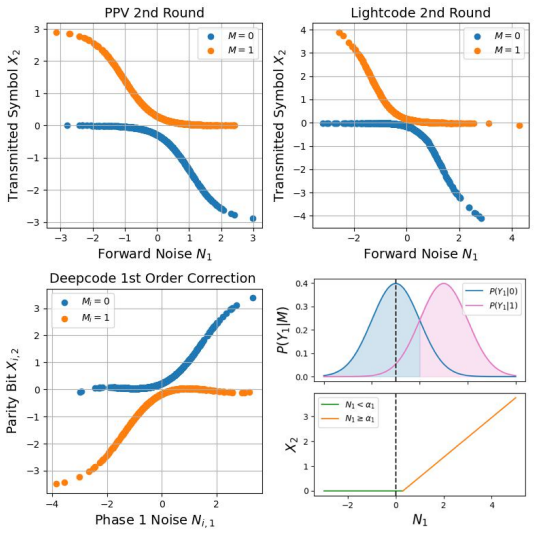

Fig. 2: Encoder output in round 2, X_2 (Deepcode: parity bit $X_{i,2}$) versus the forward noise in round 1, N_1 (Deepcode: phase 1 noise $N_{i,1}$) under forward SNR 0 dB and noiseless feedback, assuming $K = 1$. "Deepcode 1st order" means Deepcode interpretable model without outliers, as discussed in previous work [14], [15]. We approximate this shape using two linear segments, where the knee point α_1 depends on the forward SNR.

For all schemes, when $M = 0$ and the noise N_1 is negative (or $M = 1$ and N_1 is positive), binary detection successfully decodes the message, and no additional information is required. Otherwise, the transmitter sends the scaled version of the noise. This nonlinear structure is power-efficient, as it avoids wasting power on unnecessary information, thus improving performance.

As initially observed in [7] and made more explicit in [14] we can use a piecewise linear approximation of the transmitted symbol in the second round, modeled as follows

$$X_2 = \begin{cases} \frac{\lambda_2}{\sigma_z}(N_1 - \alpha_1)\mathbb{I}(N_1 - \alpha_1) & \text{if } M = 0 \\ \frac{\lambda_2}{\sigma_z}(N_1 + \alpha_1)\mathbb{I}(-N_1 - \alpha_1) & \text{if } M = 1 \end{cases} \tag{9}$$

by varying the knee point α_1, for $\sigma_z^2 = \int_0^\infty z^2 \frac{1}{\sqrt{2\pi\sigma_f^2}} e^{-\frac{(z+\alpha_1)^2}{2\sigma_f^2}} \, \mathrm{d}z$.

The MAP decoder may be easily derived and is expressed in (10), where $\lambda_1 \geq \alpha_1$ is required. α_1 can take either positive or negative values. We refer to this as "Nonlinear MVUE". We sample over λ_1 and α_1 to get the minimum BLER numerically.

Experiment: Fig. 3 illustrates the BLER performance of various feedback codes across different forward SNRs. The results show that POWERBLAST performs poorly in this region, where there is no advantage of high effective SNR over previous $N-1$ rounds. Moreover, the learned Lightcode, the analytical optimized PPV, and Nonlinear MVUE have nearly identical performance. Lightcode, trained for this block length, minimizes cross entropy at this specified rate. Both the PPV and the interpretable Nonlinear MVUE schemes provide accurate analytical approximations of this learned scheme. Interestingly, for $K = 1, N = 2$, Lightcode offers no improvement over PPV, indicating that while PPV scheme is asymptotically power-efficient, it appears to do so at finite block length N as well. Our experiments show that as the forward SNR increases, more power is allocated to round 1. This supports the intuition that at high SNR, corrections via feedback (in the second time slot) are needed only for large noise values, making it more efficient to allocate power to uncoded transmission in the first slot.

For noisy feedback, the noise estimation at the encoder is corrupted by feedback noises. For instance, the nonlinear encoder is expressed as $\frac{\lambda_2}{\sigma_z}(N_1 + \tilde{N}_1 \mp \alpha_1)\mathbb{I}\left(\pm(N_1 + \tilde{N}_1) - \alpha_1\right)$. In the PPV scheme, the LLR at the encoder is expressed as $\tilde{S}_{i-1} = \log \frac{\mathbb{P}\left(W=+1|\{\tilde{Y}_j\}_{j=1}^{i-1}\right)}{\mathbb{P}\left(W=-1|\{\tilde{Y}_j\}_{j=1}^{i-1}\right)}$. In contrast, the decoder retains its original form, as it has no information about the feedback noise. While there are claims that learned codes handle feedback noise better than analytical codes, our work here appears to refute this: Fig. 4 demonstrates that despite the mismatch induced by noisy feedback, both the PPV scheme and our Nonlinear MVUE interpretation remain robust, matching Lightcode performance. Lower feedback SNR shifts knee points α_1 to more negative values (we adjust the knee points to the forward and feedback SNR values), causing earlier noise transmission (for small forward noise values) – consistent with the pattern observed in our previous Deepcode interpretation [14]. In contrast, the linear scheme, which relies heavily on accurate noise estimation, is highly sensitive to feedback noise.

V. Interpretable Non-linear Codes for $K \geq 2, N = 2$

We now generalize the interpretable nonlinear encoding schemes for the message lengths $K \geq 2$ based on insights gained from Lightcode [6] and present the corresponding MAP decoders. While extending the PPV scheme based on LLR to $K \geq 2$ is not immediately obvious to us, our simple nonlinear scheme, which saves power on boundary messages, can be easily extended to mimic Lightcode performance.

The message $\mathbf{M} \in \mathbb{F}_2^K$ is mapped to PAM symbols $\{\Theta_1, \ldots, \Theta_T\}$, where $T = 2^K$. The boundary symbols are $\Theta_1 = -(T-1)\eta$ and $\Theta_T = (T-1)\eta$.

Encoding: In the first round, we send the PAM symbol $X_1 = \lambda_1 \Theta_j$, where $j \in [1 : T]$. In the second round, for Θ_1 (Θ_T), negative (positive) noise favors correct decoding so no power is allocated in these cases. However, for symbols in the middle, both positive and negative noise can affect the decoding results, requiring correction. The encoding function is defined as $X_2 = \frac{\lambda_2}{\sigma_z}\varphi$, where $\sigma_z^2 = \mathbb{E}\left(\varphi^2\right)$. φ is given by:

$$\varphi = \begin{cases} (N_1 - \alpha_1)\mathbb{I}(N_1 - \alpha_1) & \text{if } j = 1 \\ N_1 & \text{if } j \in [2 : T-1] \\ (N_1 + \alpha_1)\mathbb{I}(-N_1 - \alpha_1) & \text{if } j = T \end{cases} \tag{11}$$

where $\alpha_1 < \lambda_1(T-1)\eta$.

$$\frac{\lambda_2^2}{\sigma_z^2}(y_1-\lambda_1+\alpha_1)-4\lambda_1-\frac{4\lambda_1(\lambda_1-\alpha_1)}{y_1-\lambda_1+\alpha_1}-\frac{2\lambda_2}{\sigma_z}y_2 \underset{\widehat{M}=1}{\overset{\widehat{M}=0}{\lessgtr}} 0 \text{ if } y_1 \le \alpha_1-\lambda_1,$$

$$y_1-\frac{\sigma_z\lambda_2(\lambda_1-\alpha_1)}{\lambda_1\sigma_z^2+\lambda_2^2(\lambda_1-\alpha_1)}y_2 \underset{\widehat{M}=1}{\overset{\widehat{M}=0}{\lessgtr}} 0 \text{ if } \alpha_1-\lambda_1 < y_1 < \lambda_1-\alpha_1,$$

$$\frac{\lambda_2^2}{\sigma_z^2}(y_1+\lambda_1-\alpha_1)+4\lambda_1-\frac{4\lambda_1(\lambda_1-\alpha_1)}{y_1+\lambda_1-\alpha_1}-\frac{2\lambda_2}{\sigma_z}y_2 \underset{\widehat{M}=1}{\overset{\widehat{M}=0}{\lessgtr}} 0 \text{ if } y_1 \ge \lambda_1-\alpha_1 \tag{10}$$

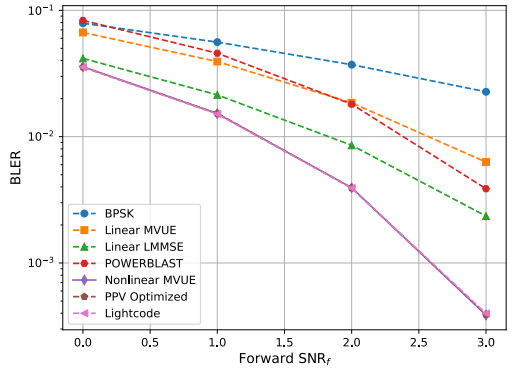

Fig. 3: BLER performance across different feedback schemes versus forward SNR with noiseless feedback, assuming $K=1$.

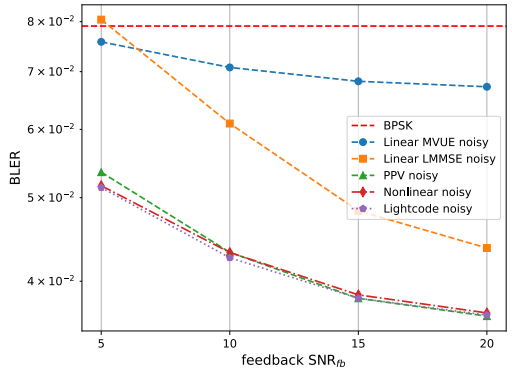

Fig. 4: BLER performance versus feedback SNR with fixed forward SNR$=0$ dB, assuming $K=1$.

Decoding: In the following let $X|Y$ denote the distribution of random variable X conditioned on random variable Y. The estimated symbol by the MAP decoder (with same prior) is given by:

$$\widehat{\Theta}=\arg\max_{\Theta_j}\mathbb{P}\left(\Theta_j \mid Y_1, Y_2\right) \tag{12}$$

where $Y_1 \mid \Theta_j \sim \mathcal{N}\left(\lambda_1\Theta_j, \sigma_f^2\right)$. For $j \in [2:T-1]$, the random variable $Y_2 \mid (Y_1,\Theta_j) \sim \mathcal{N}\left(\frac{\lambda_2}{\sigma_z}(Y_1-\lambda_1\Theta_j), \sigma_f^2\right)$.

At the boundaries $j=1$ or $j=T$, the classification depends on the value of Y_1. If $j \in \{1,T\}$,

$$Y_2 \mid (Y_1,\Theta_j) \sim \begin{cases} \mathcal{N}\left(0,\sigma_f^2\right), & \text{if } Y_1 \underset{j=T}{\overset{j=1}{\lessgtr}} \lambda_1\Theta_j \underset{j=T}{\overset{j=1}{\pm}} \alpha_1 \\ \mathcal{N}\left(\frac{\lambda_2}{\sigma_z}(Y_1-\lambda_1\Theta_j \underset{j=T}{\overset{j=1}{\mp}} \alpha_1)\right), \sigma_f^2), & \text{o.w.} \end{cases}$$

We conduct experiments with $K=2,3$, shown in Fig. 5. The interpretable nonlinear MVUE achieves BLER performance comparable to LightCode but uses only 2 analytically optimized parameters, compared to the 6736 trained parameters required by LightCode, which may increase for larger K. Deep-learned codes aim to conserve power for necessary transmissions; our interpretable schemes does the same but in a much simpler fashion. This shows the power of *interpreting* learned DL-ECFCs: they may be used to derive simple nonlinear analytical schemes which achieve similar performance.

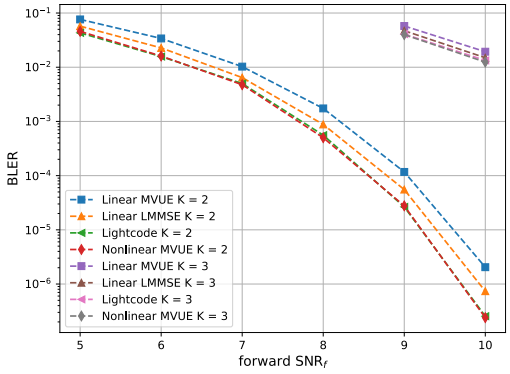

Fig. 5: BLER for $K \ge 2$ with noiseless feedback.

VI. Conclusion

We present a simple nonlinear, power-efficient, interpretable approximation of two learned feedback codes, Deepcode and Lightcode, for two-rounds of transmission and derive the corresponding MAP decoder. Our interpretable encoder, Deepcode, and Lightcode all resemble the optimized PPV scheme when $K=1$ bit is sent over 2 rounds. We extend our simple piecewise linear scheme to $K \ge 2$ bits over 2 rounds with an explicit MAP decoder whose performance again mimics that of the learned feedback code, Lightcode. Future work will explore the second order error correction of Deepcode and the third round of Lightcode, which also resemble the functional form and performance of the PPV scheme.

REFERENCES

[1] J. Schalkwijk and T. Kailath, "A coding scheme for additive noise channels with feedback–i: No bandwidth constraint," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 172–182, 1966.

[2] O. Shayevitz and M. Feder, "Optimal feedback communication via posterior matching," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1186–1222, 2011.

[3] Z. Chance and D. J. Love, "Concatenated coding for the awgn channel with noisy feedback," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6633–6649, 2011.

[4] A. Ben-Yishai and O. Shayevitz, "Interactive schemes for the awgn channel with noisy feedback," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2409–2427, 2017.

[5] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Minimum energy to send k bits through the gaussian channel with and without feedback," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4880–4902, 2011.

[6] S. K. Ankireddy, K. Narayanan, and H. Kim, "Lightcode: Light analytical and neural codes for channels with feedback," *IEEE Journal on Selected Areas in Communications*, 2025.

[7] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *NeurIPS*, vol. 31, 2018.

[8] A. R. Safavi, A. G. Perotti, B. M. Popovic, M. B. Mashhadi, and D. Gunduz, "Deep extended feedback codes," *arXiv preprint arXiv:2105.01365*, 2021.

[9] M. B. Mashhadi, D. Gunduz, A. Perotti, and B. Popovic, "Drf codes: Deep snr-robust feedback codes," *arXiv preprint arXiv:2112.11789*, 2021.

[10] Y. Shao, E. Ozfatura, A. Perotti, B. Popovic, and D. Gündüz, "Attentioncode: Ultra-reliable feedback codes for short-packet communications," *IEEE Transactions on Communications*, 2023.

[11] K. Chahine, R. Mishra, and H. Kim, "Inventing codes for channels with active feedback via deep learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 3, pp. 574–586, 2022.

[12] E. Ozfatura, Y. Shao, A. G. Perotti, B. M. Popović, and D. Gündüz, "All you need is feedback: Communication with block attention feedback codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 3, pp. 587–602, 2022.

[13] J. Kim, T. Kim, D. Love, and C. Brinton, "Robust non-linear feedback coding via power-constrained deep learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 16 599–16 618.

[14] Y. Zhou, N. Devroye, G. Turán, and M. Žefran, "Interpreting deepcode, a learned feedback code," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 1403–1408.

[15] Y. Zhou, N. Devroye, G. Turan, and M. Zefran, "Higher-order interpretations of deepcode, a learned feedback code," in *2024 60th Annual Allerton Conference on Communication, Control, and Computing*, 2024, pp. 1–8.