

Learned codes for broadcast channels with feedback

Yingyao Zhou¹, Natasha Devroye¹

¹University of Illinois Chicago, Chicago, IL, USA
{yzhou238, devroye}@uic.edu

Abstract—We focus on designing error-correcting codes for the symmetric Gaussian broadcast channel with feedback. Feedback not only expands the capacity region of the broadcast channel but also enhances transmission reliability. In this work, we study the construction of learned finite blocklength codes for broadcast channels with feedback. Learned error-correcting codes, in which both the encoder and decoder are jointly trained, have shown impressive performance in point-to-point channels, particularly with noisy feedback. However, few learned schemes exist for multi-user channels. Here, we develop a lightweight code for the broadcast channel with feedback that performs well and operates effectively at short blocklengths.

Index Terms—Gaussian broadcast channel with feedback, deep-learned error correcting codes

I. INTRODUCTION

We study the transmission of independent messages over a two-user static Gaussian broadcast channel with either noiseless or noisy feedback, where both the forward and feedback links are corrupted by independent Gaussian noises. While feedback does not increase the capacity of a memoryless point-to-point (P2P) channel [1], it can expand the capacity region of a broadcast channel even when the receiver noises are independent [2], or with only one user’s feedback [3].

For the point-to-point additive white Gaussian noise (AWGN) channel with feedback, several well-known schemes exist: Schalkwijk-Kailath [4] introduced the “SK” scheme, a capacity-achieving linear coding scheme for noiseless feedback that achieves doubly exponential probability of error decay. For noisy feedback, Chance and Love proposed the CL scheme, a concatenated coding approach that uses passive feedback [5], while the Modulo-SK scheme introduced in [6] uses active feedback, achieving superior performance.

For the two-user Gaussian broadcast channel with feedback (GBCF), Ozarow extended the SK scheme to the “OL” scheme, which also achieves doubly exponential error decay [2]. Additionally, the work in [7] enhanced the OL scheme by adopting estimators with memory, i.e. that use the last two channel outputs y_{i-1}, y_i rather than simply the current channel output y_i , resulting in the “EOL” scheme. This enhancement not only improves the achievable rates of the OL scheme but also increases transmission reliability. In addition to these SK-type coding schemes, control-theory-based codes have been developed in for example [8], where LQG codes for the K-user GBCF, generalizing the schemes in [9] were proposed. For

The work of N. Devroye and Y. Zhou was partially supported by NSF under awards 2217023, 2240532, and 1900911. The contents of this article are solely the responsibility of the authors and do not necessarily represent the official views of the NSF.

the GBCF with noisy feedback, a linear concatenated coding scheme for the K-user GBCF is proposed in [10].

The focus of this paper is not on achievable rates or error exponents, but rather on constructing practical codes at *finite* blocklength, evaluating them based on their *probability of error* for a fixed rate. To achieve low bit or block error rates for finite block lengths, deep learning has successfully been used to design encoder/decoder for P2P error-correcting codes with feedback (DL-ECFCs). These DL-ECFCs [11]–[17] outperform traditional analytically constructed codes in most cases, especially in noisy feedback. DL-ECFCs can be categorized into two types: *bit-by-bit*, where one bit of information is transmitted at a time, and *symbol-by-symbol*, where message bits of length K are mapped to a symbol for transmission, similar to SK-type codes. Bit-by-bit schemes [11]–[14] use a two-phase encoding process with a fixed rate of $\frac{1}{3}$, as first introduced in Deepcode [11]. Symbol-by-symbol schemes [15]–[17] achieve variable rates by adjusting the number of channel uses. The lightweight symbol-by-symbol code in [17] achieves state-of-the-art performance for AWGN channels with feedback. However, few deep learning strategies have been applied to multi-user channels; [18] extends Deepcode to fading Gaussian broadcast channels with feedback, while [19] focuses on the multiple-access channel (MAC) with feedback.

Contribution: We generalize Lightcode [17] to the symmetric GBCF with independent feedback links¹, achieving superior performance in the short blocklength regime, even with noisy feedback. Parallel work by [20] also extends Lightcode to GBCF but follows a different structure. Our scheme uses time-division transmission in the first two rounds, while theirs transmits both messages in the first round. Additionally, we provide an initial interpretation of our learned codes, revealing an approximately linear relationship between the encoder output and the received feedback.

Notation: Random variables are denoted by capital letters and specific instances by lowercase letters. Vectors are represented in bold, with superscripts indicating their length. Subscripts u distinguish between users, and i denotes time indices. $\text{SNR}_{u,f}$ and $\text{SNR}_{u,fb}$ represent the Signal-to-Noise Ratios for the forward and feedback channels for user u , respectively. Probability is denoted by $\mathbb{P}(\cdot)$ and expectation by $\mathbb{E}(\cdot)$. Real values are represented by \mathbb{R} , and \mathbb{F}_2 denotes the finite field with elements 0 and 1. Additionally, $\text{sgn}(x) = 1$ if $x \geq 0$, and 0 otherwise. Moreover, $x \text{ dB} = 10 \log 10(x)$.

¹Our code is available at <https://github.com/zyy-cc/GBCF.git>

II. SYSTEM MODEL

We consider the real-valued Gaussian Broadcast Channel with feedback for two users (GBCF), as illustrated in Fig. 1. The transmitter sends independent messages $\mathbf{M}_u = (M_{u,1}, \dots, M_{u,K_u}) \in \mathbb{F}_2^{K_u}$ to each user $u \in \{1, 2\}$. A total of $K_1 + K_2$ message bits are sent over N channel uses, at the code rate for each user $R_u = \frac{K_u}{N}$. At time $i \in \{1, \dots, N\}$, the received symbols for user $u \in \{1, 2\}$ are given by:

$$Y_{u,i} = X_i + Z_{u,i}$$

where $X_i \in \mathbb{R}$ are the transmitted symbols of the codewords \mathbf{X} subject to the average power constraint $\frac{1}{N} \mathbb{E} [\|\mathbf{X}\|_2^2] \leq P$, and $Z_{u,i} \sim \mathcal{N}(0, \sigma_{u,f}^2)$ are independent and identically distributed (i.i.d.) Gaussian noises across different users and time. Each receiver sends the received symbols back to the transmitter through the feedback links with one unit delay. For noiseless feedback, the received feedback equals the channel outputs, $\tilde{Y}_{u,i-1} = Y_{u,i-1}$. However, for noisy feedback, the received feedback is expressed as $\tilde{Y}_{u,i-1} = Y_{u,i-1} + \tilde{Z}_{u,i-1}$ where $\tilde{Z}_{u,i-1} \sim \mathcal{N}(0, \sigma_{u,fb}^2)$ are also i.i.d. Gaussian noises.

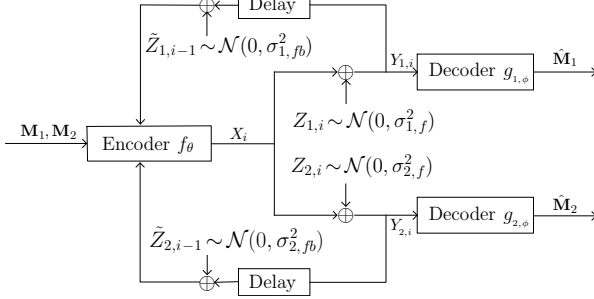


Fig. 1: Gaussian broadcast channel with noisy feedback

In this work, we focus on the symbol-by-symbol coding, similar to the OL scheme, which maps message bits of length K to a symbol and iteratively refines the transmitted symbols. At the i -th transmission, the encoding function f_θ generates the transmitted symbol X_i using the message bits \mathbf{M}_1 and \mathbf{M}_2 , along with the past feedback from both users $\tilde{\mathbf{Y}}_u^{i-1} = [\tilde{Y}_{u,1}, \dots, \tilde{Y}_{u,i-1}]$, defined as

$$X_i = c_i f_\theta(\mathbf{M}_1, \mathbf{M}_2, \tilde{\mathbf{Y}}_1^{i-1}, \tilde{\mathbf{Y}}_2^{i-1})$$

where c_i are scaling factors that ensure the power constraint is satisfied. After N transmissions, the decoder for each user estimates the message bits from their own received noisy codewords, expressed as $\widehat{\mathbf{M}}_u = g_{u,\phi}(\mathbf{Y}_u^N) \in \mathbb{F}_2^{K_u}$, where $\mathbf{Y}_u^N = [Y_{u,1}, \dots, Y_{u,N}]$. The performance metric is the average block error rate (BLER), defined as $\text{BLER}_u = \mathbb{P}(\mathbf{M}_u \neq \widehat{\mathbf{M}}_u)$ for user u . The joint BLER is taken as the average BLER between the two users, given by $\text{BLER} = \frac{1}{2}(\text{BLER}_1 + \text{BLER}_2)$. For simplicity, we set $K_1 = K_2$ and focus on the symmetric case where $\sigma_{1,f}^2 = \sigma_{2,f}^2$ and $\sigma_{1,fb}^2 = \sigma_{2,fb}^2$.

Most previous work has focused on analytical linear code design for the encoder f_θ and the decoders $g_{1,\phi}$ and $g_{2,\phi}$. In this work, we aim to replace them with neural networks parameterized by θ and ϕ to minimize the BLER.

III. PAST WORK

In this section, we provide a brief overview of deep-learned error-correcting feedback codes (DL-ECFCs) for point-to-point AWGN channels with feedback (P2P-AWGN-F) and the analytical linear coding schemes for the GBCF.

A. DL-ECFCs

Deepcode: Deepcode [11] is the first DL-ECFC for the P2P AWGN channel with passive feedback, achieving a code rate of $\frac{1}{3}$. The encoder and decoder are based on recurrent neural networks (RNNs) and are trained jointly. Our previous work [21], [22] explains how the Deepcode encoder utilizes past feedback for error correction and develops an analytical encoder and decoder with significantly fewer learned parameters that achieves comparable performance. The study in [18] extends Deepcode to the symmetric two-user fading GBCF; here we consider static channels. GBCF-Deepcode [18] encoding has two phases: in the first, the BPSK-modulated message bits for both users are transmitted uncoded; in the second, the encoder generates a **single** parity bit based on feedback from the two receivers.

Deepcode achieves a lower bit error rate (BER) than analytical codes with passive feedback but is limited to a code rate of $\frac{1}{3}$. Additionally, for message bits of length K , the total number of transmissions—including both forward and backward passes—grows linearly with K , potentially causing significant delays.

Lightcode: To address these limitations, [15] proposes a symbol-by-symbol coding method for P2P-AWGN-F using a transformer-based structure called GBAF, which achieves remarkably low BLER. Lightcode [17] eliminates the self-attention and positional encoding components of GBAF, reducing the number of parameters to just $\frac{1}{10}$ of those in GBAF while still outperforming it. We further expand on the structure of Lightcode in Section IV.

B. Analytical Linear Coding Schemes

In this subsection, we review the OL and EOL schemes developed for GBCF with **noiseless** feedback.

1) *OL scheme:* The OL scheme is an adaptation of the SK scheme [2]. Initially, the two message bits, \mathbf{M}_1 and \mathbf{M}_2 , are mapped to real numbers and transmitted separately over the forward channel. The receivers estimate the messages based on the received noisy symbols and send these estimates back to the transmitter via the noiseless feedback link. In subsequent rounds, the transmitter sends a linear combination of the estimation errors made by the two receivers. Each receiver updates its estimate, progressively reducing the errors.

Initialization: The message bits \mathbf{M}_u , where $u \in \{1, 2\}$, are mapped to a PAM modulation symbol Θ_u from the constellation $\{\pm 1\eta, \pm 3\eta, \dots, \pm(2^{K_u} - 1)\eta\}$, with $\eta = \sqrt{\frac{3}{2^{2K_u} - 1}}$ to satisfy the unity power constraint ². The estimated message

²In [2], message bits are mapped to Θ_u , which ranges from $-\frac{1}{2}$ to $\frac{1}{2}$, then scaled by $\sqrt{12P}$. When 2^{K_u} is large, $\mathbb{E}[\Theta^2] \approx \frac{1}{12}$. Since our K_u is small, we use the method described in our work.

at receiver u at time i is denoted by $\hat{\Theta}_{u,i}$, with the estimation error $\epsilon_{u,i} = \hat{\Theta}_{u,i} - \Theta_u$. The mean square error is represented by $\alpha_{u,i} = \mathbb{E}[\epsilon_{u,i}^2]$, and the correlation coefficient between the estimation errors of the two receivers is defined as $\rho_i = \frac{\mathbb{E}[\epsilon_{1,i}\epsilon_{2,i}]}{\sqrt{\alpha_{1,i}\alpha_{2,i}}}$. In the first two rounds, the transmitter sends with power P as

$$X_i = \sqrt{P}\Theta_i, \quad i = 1, 2$$

After the first transmission, receiver 1 estimates $\hat{\Theta}_{1,1} = \hat{\Theta}_{1,2} = \frac{Y_{1,1}}{\sqrt{P}}$ and ignores the second transmission. Similarly, receiver 2 ignores the first transmission and estimates $\hat{\Theta}_{2,1} = \hat{\Theta}_{2,2} = \frac{Y_{2,2}}{\sqrt{P}}$. In this case, $\alpha_{u,2} = \frac{\sigma_{u,f}^2}{P}$ and $\rho_2 = 0$. In this work, we focus on finite channel uses and apply LMMSE estimation in the first two rounds to improve BLER performance [6]. The estimate is given by

$$\hat{\Theta}_{u,i} = \frac{\sqrt{P}}{P + \sigma_{u,f}^2} Y_{u,i}, \quad i = 1, 2, \quad u = 1, 2$$

Encoding: For the i -th transmission, where $i \geq 3$, the transmitter sends:

$$X_i = \sqrt{\frac{P}{D_{i-1}}} \left[\frac{\epsilon_{1,i-1}}{\sqrt{\alpha_{1,i-1}}} + \frac{\epsilon_{2,i-1}}{\sqrt{\alpha_{2,i-1}}} g \operatorname{sgn}(\rho_{i-1}) \right]$$

where $D_{i-1} = 1 + g^2 + 2g|\rho_{i-1}|$, and g is a nonnegative constant that balances the trade-off between the two users. To achieve similar BLER for both users, we assume $g = 1$.

Decoding: At the receiver u , a memoryless MMSE estimator is used to estimate $\epsilon_{u,i-1}$ based **only** on $Y_{u,i}$, denoted by $\hat{\epsilon}_{u,i-1} = \mathbb{E}[\epsilon_{u,i-1}|Y_{u,i}]$. The receiver u then updates its estimate as follows:

$$\hat{\Theta}_{u,i} = \hat{\Theta}_{u,i-1} - \frac{\mathbb{E}[\epsilon_{u,i-1}Y_{u,i}]}{\mathbb{E}[Y_{u,i}^2]} Y_{u,i}, \quad u = 1, 2$$

where $\mathbb{E}[Y_{u,i}^2] = P + \sigma_{u,f}^2$, $\mathbb{E}[\epsilon_{1,i-1}Y_{1,i}] = \sqrt{\frac{P}{D_{i-1}}}\sqrt{\alpha_{1,i-1}}(1 + g|\rho_{i-1}|)$, and $\mathbb{E}[\epsilon_{2,i-1}Y_{2,i}] = \sqrt{\frac{P}{D_{i-1}}}\sqrt{\alpha_{2,i-1}}(g + |\rho_{i-1}|)\operatorname{sgn}(\rho_{i-1})$.

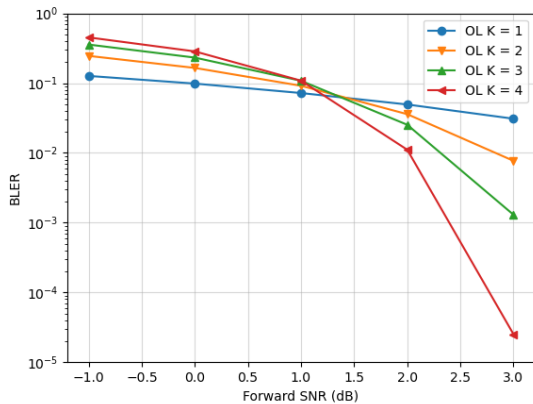


Fig. 2: BLER performance for code rate $\frac{1}{3}$ with varying message bit lengths K .

We experimentally observed that for the same code rate of $\frac{1}{3}$, different message bit lengths $K = K_1 = K_2$ result in varying BLER performance, as shown in Fig. 2. At low SNR $_f$, where noises significantly affect the signal, shorter message bit lengths perform better due to the increased distance between constellation points. Conversely, at high SNR $_f$, longer message bit lengths K —and thus more channel uses $N = 3K$ —lead to better performance, as the decoding error of the OL schemes decays doubly exponentially in N .

However, K cannot be too large due to precision issues and quantization errors associated with 2^K PAM modulation. The advanced analog-to-digital converters, such as the Texas Instruments ADS1263 and Analog Devices AD4134, support up to 32-bit quantization [14]. In our experiments, when $K \geq 25$, the BLER starts to increase rather than decrease. To accommodate longer lengths L , we recommend selecting an appropriate value for K to avoid precision issues—using a small K at low SNR and a large K at high SNR—and then dividing it into $l = \frac{L}{K}$ chunks of bits. Each chunk, with a message bit length of K , is processed using the desired coding scheme. The deep-learned codes also adhere to this rule; however, with more rounds, the input space increases, which raises training difficulty.

2) *EOL scheme:* The EOL scheme [7] extends the previous OL scheme by incorporating an MMSE estimator based on the current and previous channel outputs rather than just the current output. The transmission process remains nearly the same as before, except that the receiver now estimates $\hat{\epsilon}_{u,i-1} = \mathbb{E}[\epsilon_{u,i-1}|Y_{u,i-1}, Y_{u,i}]$. For further details, we refer readers to [7].

IV. GBCF-LIGHTCODE-SEP

We now extend Lightcode to GBCF-Lightcode-sep, which supports a wider range of code rates. While the concurrent [20] sends a symbol containing both messages in the first channel use and uses the remaining channel uses ($N - 1$) to help the decoder refine this, our scheme uses separate channel uses for each of the messages first, before simultaneously refining in the remaining channel uses ($N - 2$). For a fixed blocklength, our scheme thus has less refinement rounds compared to [20] but has perhaps “clearer” initial message transmission.

Learned codes for GBCF offer superior BLER performance compared to analytical schemes, particularly when the forward SNR is high. They also remain effective with noisy feedback, even when the noise variance in the feedback link is high.

A. Structure

The structure is inspired by the block coding approach in [15], [17], with the design consisting of a feature extractor (FE) and a multilayer perceptron (MLP) module.

Feature extractor: The structure is shown in Fig. 3. In [17], the original dimension d_H is set to 32. Here, we double the input information and experimentally find that the increased dimension improves performance. Therefore, we set $d_H = 64$. Layer normalization is added to stabilize training.

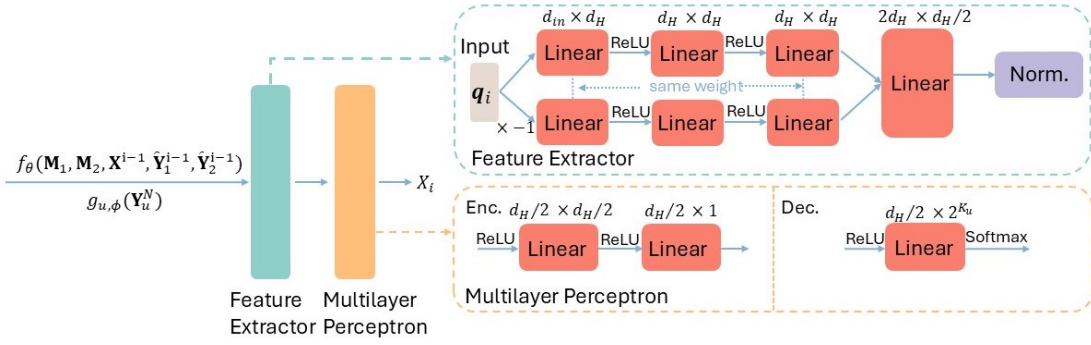


Fig. 3: Design of the GBCF-Lightcode-sep (left) and its detailed structure (right).

Multilayer perceptron: The output of the feature extractor is passed to the MLP layers. The encoder uses a two-layer MLP to generate a symbol for each channel use, while the decoder employs a one-layer MLP. For the decoder on the receiver side, $u \in \{1, 2\}$, the index for each message bit M_u ranges from 0 to $2^{K_u} - 1$. Thus, the output of the linear layer is projected to a vector of dimension $C_u = 2^{K_u}$, with a softmax function applied to denote the probability for each index. For a vector $\mathbf{v} \in \mathbb{R}^{C_u}$, the i -th element after applying softmax is given by $\text{softmax}(v_i) = \frac{e^{v_i}}{\sum_{j=1}^{C_u} e^{v_j}}$.

B. Transmission

The entire transmission design is inspired by the OL scheme [2]. The learned parameters are highlighted in red.

Initialization: Similar to the OL scheme, in the first two rounds, the message $M_u \in \mathbb{F}_2^{K_u}$ is mapped to the PAM modulation symbol Θ_u for $u \in \{1, 2\}$ and transmitted separately.

$$X_i = \beta_i \Theta_i, \quad i = 1, 2$$

Encoding: In the subsequent rounds, the learned encoder generates the transmitted symbols based on the previously generated symbols and the feedback from the two receivers from the last $i - 1$ rounds. The generated symbol is given by

$$X_i = \beta_i f_\theta(\mathbf{X}^{i-1}, \tilde{\mathbf{Y}}_1^{i-1}, \tilde{\mathbf{Y}}_2^{i-1}), \quad i = 3, \dots, N$$

where $\tilde{\mathbf{Y}}_u^{i-1} = [\tilde{Y}_{u,1}, \dots, \tilde{Y}_{u,i-1}]$ represents the feedback from receiver u , with each element given by $\tilde{Y}_{u,i-1} = X_{i-1} + Z_{u,i-1} + \tilde{Z}_{u,i-1}$. The whole transmission is subject to an average power constraint, meaning $\frac{1}{N} \sum_{i=1}^N \beta_i^2 = P$. The encoding process in [20] follows a different structure from ours. Instead of transmitting the messages separately, the encoder sends messages of length $2K_u$ for both receivers in the first round $X_1 = \beta_1 f_\theta(M_1, M_2)$, with subsequent rounds used for error correction $X_i = \beta_i f_\theta(M_1, M_2, \mathbf{X}^{i-1}, \tilde{\mathbf{Y}}_1^{i-1}, \tilde{\mathbf{Y}}_2^{i-1})$.

In the following discussion, we refer to our scheme as ‘‘GBCF-Lightcode-sep’’ and the scheme in [20] as ‘‘GBCF-Lightcode’’.

Decoding: For each receiver, the decoder estimates its message based on the received noisy codewords, as does [20]:

$$\hat{M}_u = g_{u,\phi}(\mathbf{Y}_u^N).$$

The encoder f_θ and the two decoders $g_{u,\phi}$ consist of FE and MLP modules, as shown in Fig. 3. The entire process is outlined in Algorithm 1. The input dimension to the neural networks is fixed, so the $*$ parts will be padded with zeros to maintain the required dimension. The function H_{ps} represents the power constraint block, while H_{d2b} converts the PAM index to its binary representation. In this work, we use the negative log-likelihood (NLL) loss as the objective function:

$$\ell_u(\mathbf{p}_u, \hat{\mathbf{p}}_u) = -\frac{1}{B} \sum_{i=1}^B \left(\sum_{j=0}^{C_u-1} p_{u,ij} \log(\hat{p}_{u,ij}) \right)$$

Here, B is the batch size, and $C_u = 2^{K_u}$ represents the number of classes (PAM indices) for receiver u , and $p_{u,ij}$ is 1 if class j is the correct class for the i -th sample, and 0 otherwise. $\hat{p}_{u,ij} \in \mathbb{R}$ denotes the predicted probability for class j of the i -th sample. Additionally, during training, to prevent one receiver from significantly outperforming the other, we add a regularization term $(\ell_1 - \ell_2)^2$ to ensure that the performance of both receivers remains comparable.

Algorithm 1 GBCF-Lightcode-sep (Encoder f_θ , Decoder 1 $g_{1,\phi}$, Decoder 2 $g_{2,\phi}$, Channel Uses N , Message Bits M_1, M_2)

- 1: **Initialization:** Map M_1, M_2 to PAM symbols Θ_1, Θ_2 ; set $X_1 = \beta_1 \Theta_1, X_2 = \beta_2 \Theta_2$
 - 2: **for** each $i \in [3, N]$ **do**
 - 3: Update encoder input vector:
 - 4: $\mathbf{q}_i = [X_1, X_2, \dots, X_{i-1}, *, \tilde{Y}_{1,1}, \dots, \tilde{Y}_{1,i-1}, *, \tilde{Y}_{2,1}, \dots, \tilde{Y}_{2,i-1}, *]$
 - 5: Generate transmitted symbol: $X_i = \beta_i H_{ps}[f_\theta(\mathbf{q}_i)]$
 - 6: Transmit X_i to both receivers
 - 7: **end for**
 - 8: Receiver 1: $\hat{\mathbf{p}}_1 = g_{1,\phi}(\mathbf{Y}_1^N) \in \mathbb{R}^{C_1}$
 - 9: Receiver 2: $\hat{\mathbf{p}}_2 = g_{2,\phi}(\mathbf{Y}_2^N) \in \mathbb{R}^{C_2}$
 - 10: Compute NLL loss: $\ell_1 + \ell_2 + (\ell_1 - \ell_2)^2$
 - 11: Classification: $I_u = \arg\max(\hat{\mathbf{p}}_u)$, where $u \in \{1, 2\}$
 - 12: Get bit representation: $\hat{M}_u = H_{d2b}(I_u)$, where $u \in \{1, 2\}$
-

The training parameters are listed in Table I, where training is performed at the corresponding forward and feedback SNRs.

TABLE I: Training parameters for GBCF-Lightcode-sep

Parameters	Values
Batch size B	100,000
Optimizer	AdamW
Weight decay	0.01
Epochs	120000
Initial learning rate	0.002
Clipping threshold	0.5
Power P	1

V. SIMULATIONS

In this section, we evaluate the performance of GBCF-Lightcode-sep and compare it with other existing codes.

A. Noiseless Feedback

In Fig. 4 we compare the BLER of codes with a code rate of $\frac{1}{3}$ under noiseless feedback. Although LQG codes outperform OL and EOL asymptotically in terms of achievable rates, LQG demonstrates poorer block error rate (BLER) performance in the finite-length regime. At low SNR³, shorter messages perform better at the same code rate. As forward SNR increases, the BLER of longer messages with more channel uses decreases significantly. GBCF-Lightcode-sep performs worse than GBCF-Lightcode when $K_u = 1$ but becomes nearly comparable at $K_u = 3$. This suggests that for small message lengths (K), transmitting both messages (M_1, M_2) in the first round for an additional error correction round is more effective, while for larger K , time-division in the first two transmissions is preferable.

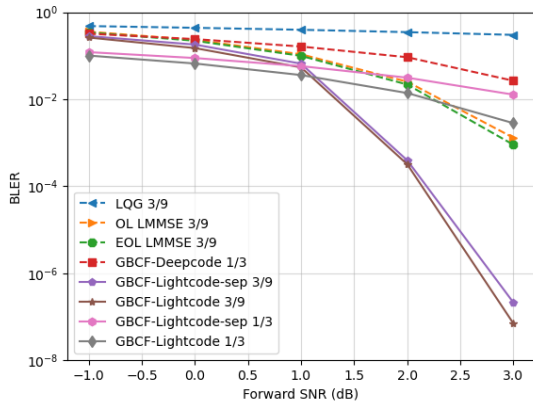


Fig. 4: BLER performance across different schemes with noiseless feedback

We also consider the Time-Division Lightcode (TD-Lightcode) [17] scheme as a benchmark, in which each user transmits their message during an assigned time slot. To ensure fairness between the users, we assume an even number of channel uses, N , with $\frac{N}{2}$ time slots allocated for transmitting

³For a fair comparison, we set the block length to 3 for GBCF-Deepcode.

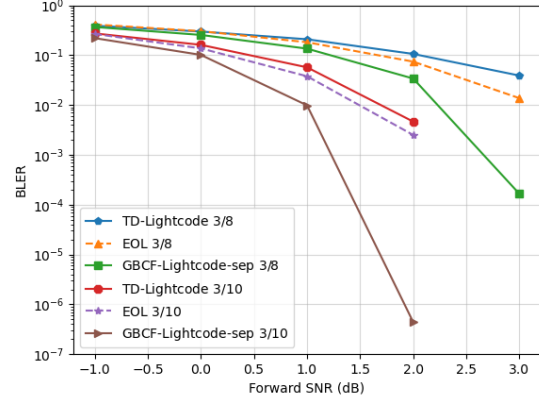


Fig. 5: BLER comparison for code rates $\frac{3}{8}$ and $\frac{3}{10}$ with noiseless feedback

M_1 and the other $\frac{N}{2}$ slots for transmitting M_2 . As shown in Fig. 5, the BLER performance is illustrated for code rates $\frac{3}{8}$ and $\frac{3}{10}$ with noiseless feedback. The results show that, at the same code rate, GBCF codes (including EOL schemes and GBCF-Lightcode-sep) outperform TD-Lightcode, demonstrating that cooperation between the encoder and decoders effectively leverages feedback from both users. Notably, GBCF-Lightcode-sep achieves the best performance among all tested codes. Additionally, the number of channel uses, N , is crucial for enhancing BLER performance; even a single additional channel use can lead to a significant improvement.

B. Noisy Feedback

We also investigate the performance of GBCF-Lightcode-sep under noisy feedback. Fig. 6 compares the performance of various schemes, assuming a fixed forward SNR_f = 3 dB. Results indicate that when feedback noise variance is low, symbol-by-symbol coding⁴ with a longer block length ($N = 9$) performs well, as additional channel uses allow for better message refinement when feedback is reliable. However, longer block lengths are more sensitive to feedback noise, while shorter block lengths ($N = 3$) are more robust. Notably, GBCF-Lightcode-sep outperforms other codes by adjusting the block length according to feedback noise levels.

C. Interpretation

In this subsection, we present an initial interpretation of the learned GBCF-Lightcode-sep encoder, examining the relationship between the encoder output at round 3, X_3 , and past noisy feedback from two receivers, as in [17]. Here, $I(\Theta_u)$, which ranges from 0 to $2^{K_u} - 1$, represents the PAM index for the message bits of user u . For feedback from receiver 1 ($\tilde{Y}_{1,1}$) and receiver 2 ($\tilde{Y}_{2,2}$), we analyze the relationship between X_3 and the noisy feedback by setting the message

⁴Although the OL and EOL schemes are designed for the noiseless feedback case, here we update the estimation error at the encoder with feedback noise, and apply normalization to ensure the transmitted symbol meets the power constraint.

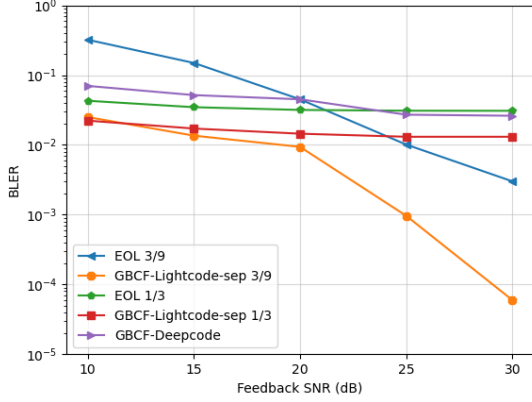


Fig. 6: BLER performance with noisy feedback and fixed forward SNR 3 dB

bits and forward noises of the opposite user to $I(\Theta_2) = 0$ and 0 for user 2 (left side of Fig. 7) and $I(\Theta_1) = 0$ and 0 for user 1 (right side of Fig. 7). From Fig. 7, we observe that X_3 is approximately a linear function of the feedback channel output, being negatively proportional to $\tilde{Y}_{1,1}$ and positively proportional to $\tilde{Y}_{2,2}$.

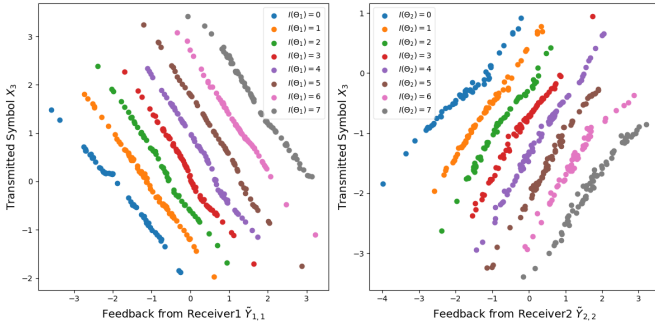


Fig. 7: Encoder output at round 3 X_3 with respect to the noisy feedback $\tilde{Y}_{1,1}$ (left) and $\tilde{Y}_{2,2}$ (right) at code rate $\frac{3}{9}$, forward SNR 3 dB and noiseless feedback.

VI. CONCLUSION

In this work, we propose a symbol-by-symbol, learned coding scheme for the static Gaussian broadcast channel with feedback. This scheme easily applies to different coding rates and performs especially well at high forward SNR, while remaining robust to noisy feedback. At the same code rate, shorter message lengths (e.g., sending 2^K PAM signals with smaller K) are preferred at low forward or feedback SNR. Conversely, at high SNR, longer messages (with larger K and correspondingly more channel uses N) are more suitable. We also provide an initial interpretation of our scheme, showing that the learned encoder output at round 3 is a linear function of the channel outputs. Compared to GBCF-Lightcode in [20], our GBCF-Lightcode-sep uses time-division in the first two rounds rather than simultaneously sending one symbol with two messages in the first round. We thus tradeoff one less refinement round (for 2 users) with a clearer initial guess.

Depending on the scenario, the learned GBCF codes achieve the best-known finite blocklength performance for $N \leq 10$.

REFERENCES

- [1] C. Shannon, "The zero error capacity of a noisy channel," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 8–19, 1956.
- [2] L. Ozarow and S. Leung-Yan-Cheong, "An achievable region and outer bound for the gaussian broadcast channel with feedback (corresp.)," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 667–671, 1984.
- [3] S. R. Bhaskaran, "Gaussian broadcast channel with feedback," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 5252–5257, 2008.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] Y. Murin, Y. Kaspi, and R. Dabora, "On the ozarow-leung scheme for the gaussian broadcast channel with feedback," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 948–952, 2014.
- [8] E. Ardestanizadeh, P. Minero, and M. Franceschetti, "Lqg control approach to gaussian broadcast channels with feedback," *IEEE transactions on information theory*, vol. 58, no. 8, pp. 5267–5278, 2012.
- [9] N. Elia, "When bode meets shannon: Control-oriented feedback communication schemes," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1477–1488, 2004.
- [10] Z. Ahmad, Z. Chance, D. J. Love, and C.-C. Wang, "Concatenated coding using linear schemes for gaussian broadcast channels with noisy channel output feedback," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4576–4590, 2015.
- [11] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *NeurIPS*, vol. 31, 2018.
- [12] 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.
- [13] M. B. Mashhadi, D. Gunduz, A. Perotti, and B. Popovic, "Drf codes: Deep snr-robust feedback codes," *arXiv preprint arXiv:2112.11789*, 2021.
- [14] Y. Shao, E. Ozfatura, A. Perotti, B. Popovic, and D. Gündüz, "Attention-code: Ultra-reliable feedback codes for short-packet communications," *IEEE Transactions on Communications*, 2023.
- [15] 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.
- [16] J. Kim, T. Kim, D. Love, and C. Brinton, "Robust non-linear feedback coding via power-constrained deep learning," *arXiv preprint arXiv:2304.13178*, 2023.
- [17] 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*, pp. 1–1, 2025.
- [18] S. Li, D. Tuninetti, and N. Devroye, "Deep learning-aided coding for the fading broadcast channel with feedback," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 3874–3879.
- [19] E. Ozfatura, C. Bian, and D. Gündüz, "Do not interfere but cooperate: A fully learnable code design for multi-access channels with feedback," in *2023 12th International Symposium on Topics in Coding (ISTC)*. IEEE, 2023, pp. 1–5.
- [20] J. Malayter, C. Brinton, and D. Love, "Deep learning aided broadcast codes with feedback," *arXiv preprint arXiv:2410.17404*, 2024.
- [21] 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.
- [22] Y. Zhou, N. Devroye, G. Turán, 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.