

Deep Learning-Based Joint Channel Coding and Frequency Modulation for Low Power Connectivity

Boxuan Chang, Chenyu Wang, Hun-Seok Kim, *Senior Member, IEEE*

Department of EECS, University of Michigan, Ann Arbor, MI

{bxchang, chyuwang, hunseok}@umich.edu

Abstract—Low-power, low-cost wireless communication is a fundamental requirement of Internet-of-Things (IoT) and massive machine-type communication (mMTC). Various low power connectivity standards such as Bluetooth and LoRa adopt non-coherent frequency modulation schemes as they exhibit significantly lower complexity and power consumption compared to coherent in-phase and quadrature (IQ) modulation schemes. In our paper, we propose a deep learning-based joint channel coding and modulation (JCM) scheme for digitally controlled oscillator (DCO)-based frequency modulation. The learned encoder takes an information bit sequence and produces DCO control samples that represent instantaneous frequency to modulate the radio frequency (RF) signal. The learned decoder recovers/decodes information bits from the received noisy samples without any preamble to assist time and frequency synchronization. We train and test the proposed scheme under significant phase noise and carrier frequency offset (CFO) of the DCO to successfully mitigate these practical impairments at the receiver.

Index Terms—Deep Learning, Digitally Controlled Oscillator, mMTC, GRU

I. INTRODUCTION

Massive machine-type communication (mMTC) in Internet-of-Things (IoT) networks is one of essential technologies to enable 5th and next generation communication systems [1]. The massive number of IoT devices calls for low-cost, low-power connectivity solutions. Popular low power wireless standards such as Bluetooth and LoRa adopt non-coherent frequency modulation schemes such as Gaussian frequency shift keying (GFSK) and frequency chirp that exhibit continuous phase constant envelope properties by modulating the RF signal using digitally-controlled oscillator (DCO) [2]. Unlike coherent IQ-based modulation schemes such as phase-shift keying and quadrature amplitude modulation (QAM), DCO-based frequency modulation does not necessarily require a phase lock-loop (PLL). Therefore, for low power low complexity wireless solutions, DCO-based transmitters have been widely adopted in Bluetooth Low Energy (BLE) [3], LoRa [4], and proprietary protocols [5]. The fact that DCO only controls the instantaneous frequency (not the phase of IQ samples) with a digital signal largely simplifies the RF transmitter implementation, and thus inspired us to explore a new DCO-based frequency modulation scheme via deep learning.

Deep learning has achieved tremendous success in various areas thanks to its strong representative capability and

increasing computation power. Recently, deep learning also appealed to the wireless communication research, especially to learn new source and channel coding schemes. Deep neural networks (DNN) first showed their advancement in the decoding of Polar [6] and LDPC [7] codes. Compared to the conventional decoding methods, the DNN-based algorithms have better block error rate (BLER) performance and robustness to variation in channel statistics. Further efforts are made to invent novel channel codes with different neural network structures. The authors in [8] first introduced a DNN-based autoencoder to learn a new (7, 4) code, which significantly outperforms (7, 4) Hamming code. However, the fact that the code uses one-hot encoded information bits as the model input, whose length grows exponentially with the block length, limits its application to very short input blocks. One approach to address this problem is demonstrated in [9], which segments the input block into shorter sequences before one-hot encoding. Each sequence is encoded with an independent DNN module, and the outputs are summed up to produce the superposition code. This design addresses the curse of dimension but it is not adaptive to various block lengths as each model is trained for a specific block length only. The models in [10] solve both of the aforementioned issues using a recurrent neural network (RNN) such as Gated Recurrent Unit (GRU) [11] and Long Short-Term Memory (LSTM) [12]. The popularity of these RNN-based models in sequential data processing tasks naturally invited attempts to create learned channel coding schemes. Prior works [13] and [10] adopted RNN-based autoencoder for channel coding and outperformed conventional Turbo and TBCC coding with short blocks.

Deep learning has been applied to learning new modulation schemes. The authors in [14] and [15] proposed DNN-based modulations for optical and molecular communications. Moreover, the unique capability of deep learning models to perform end-to-end optimization has inspired joint channel coding and modulation (JCM) designs. A prior work [16] applied a convolutional neural network (CNN) for channel coding and a time-distributed dense layer for constellation learning. Similarly, [17] used Turbo-Autoencoder as the channel coding model and multi-layer perceptron (MLP) module for modulation. These works have shown that under certain power constraints, the JCM scheme can outperform the combination of conventional channel coding and modulation schemes.

All aforementioned deep learning based communication

This work was funded in part by NSF CAREER Award #1942806. Source code available in https://github.com/boxuanCh/DCO_code_mod

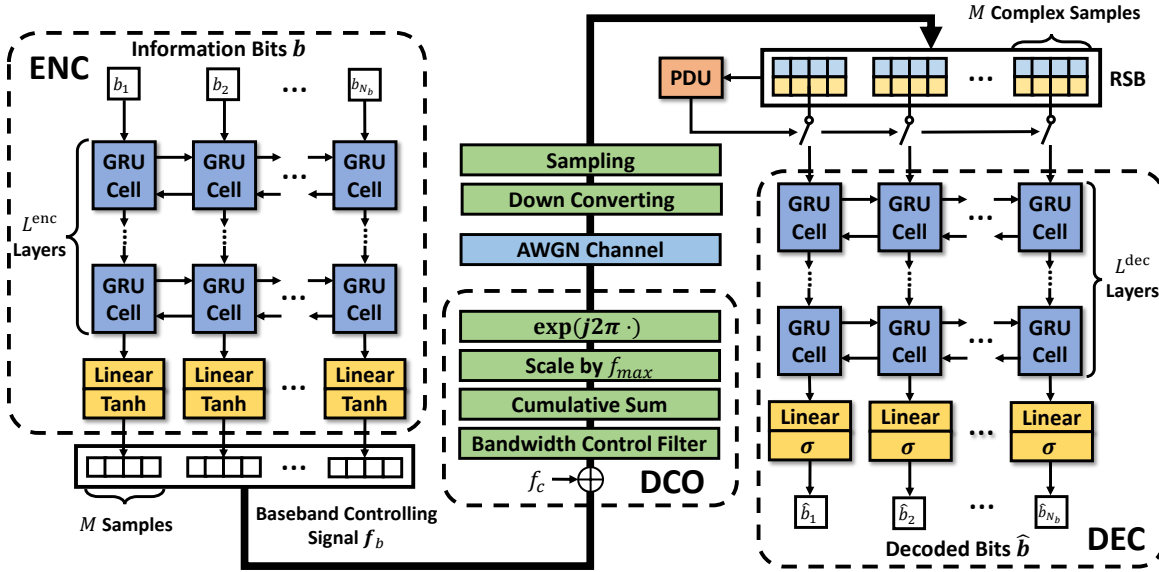


Fig. 1. Architecture of the proposed JCM model for DCO-based non-coherent frequency modulation communication

schemes are based on coherent baseband signaling with real-/complex-valued coherent in-phase and quadrature (IQ) samples. Such signaling requires a PLL to maintain phase-coherency without significant phase noise or carrier frequency offset (CFO). Therefore, those schemes are not directly applicable to DCO-based frequency modulation transmitters commonly employed in low power connectivity standards such as BLE and LoRa.

Inspired by the success of deep learning in communication systems, we design a new JCM scheme depicted in Fig. 1 for DCO-based non-coherent communication using a GRU-Autoencoder. In our design, the encoder maps the information bits to instantaneous frequency samples, which controls a free-running (i.e., without a PLL) DCO for RF signal transmission. At the receiver, another GRU module decodes the received noisy samples into bits. In the end-to-end training process, we take the phase noise and carrier frequency offset of the DCO into consideration to improve the robustness of the model. We proposed a preamble-less packet detection and time synchronization with a separate GRU module that is jointly trained with the encoder.

Contributions of this paper includes 1) introducing a novel deep learning-based JCM scheme for DCO-based non-coherent frequency modulation, 2) proposing a pilot-/preamble-less communication scheme for time/frequency synchronization and real-world impairments mitigation using deep learning, 3) providing extensive evaluations of the proposed scheme compared to the conventional channel coding and DCO-based GFSK modulation.

II. DCO-BASED FREQUENCY MODULATION

In this section, we construct the signal model of our proposed DCO-based non-coherent frequency modulation.

A continuous phase frequency modulated constant envelope signal is modeled as

$$y(t) = \exp \left[j \left(2\pi \int_0^t f(\tau) d\tau + \theta \right) \right], \quad (1)$$

where $f(t)$ is the instantaneous frequency at time t and θ is the unknown random initial phase of the DCO. The controlling signal of the DCO is equal to the instantaneous frequency. We define a discrete time sequence of the controlling signal $f[n]$ with sampling frequency f_s . We further decompose $f[n] = f_b[n] + f_c$, where $f_b[n]$ is the modulated baseband frequency and f_c is the constant carrier frequency. The discrete time frequency modulated baseband signal is modeled as

$$y[n] = \exp \left(j 2\pi \sum_{i=1}^n \frac{f_b[i]}{f_s} + \theta \right). \quad (2)$$

Practically, DCO experience phase noise caused by physical circuit impairments of the oscillator. Phase noise is modeled as a colored Gaussian random process $N_p(t)$. Then by interpolating the controlling signal, we obtain the continuous-time signal generated by the DCO as

$$y(t) = \text{BPF} \left\{ \exp \left[j \left(\frac{2\pi}{f_s} \left(\sum_{i=0}^{N-1} f_b[i] + \left(t - \frac{N-1}{f_s} \right) f_b[N] + f_c \right) + N_p(t) + \theta \right) \right] \right\}, \quad (3)$$

where $N = \lfloor t f_s \rfloor$ and BPF is bandpass filtering.

We suppose the transmitted signal undergoes an additive white Gaussian noise (AWGN) channel. The normalized received signal is modeled as

$$r(t) = y(t) + N_o(t) \quad (4)$$

where $N_o(t)$ is the channel noise. The receiver mixes the received signal with a local oscillation signal with frequency

$\hat{f}_c$ to down-convert it to the baseband signal, which is sampled with frequency f_s . Note that a practical (non-ideal) receiver experiences carrier frequency offset (CFO) defined by $\tilde{f}_c = f_c - \hat{f}_c$.

Suppose the starting time of the transmitted packet is 0 and the receiver detect the packet with time offset t_Δ . The discrete-time received baseband signal $r[n]$ is expressed by

$$r[n] = \exp \left[j \left(2\pi \left(\frac{1}{f_s} \left(\sum_{i=1}^{m+n} f_b[i] + \tilde{t}_\Delta f_b[m+n] + \tilde{f}_c n \right) \right) + N_p[n] + \tilde{\theta} \right) \right] + N_o[n] \quad (5)$$

where $\tilde{\theta}$ is the random phase offset, $m = \lceil f_s t_\Delta \rceil$, and $t_\Delta = m/f_s + \tilde{t}_\Delta$ holds. $N_p[n]$ and $N_o[n]$ are discrete-time phase noise and channel noise, respectively. We assume a non-coherent receiver with an unknown random phase $\tilde{\theta} \sim \mathcal{U}(0, 2\pi)$.

III. GRU-JCM DESIGN

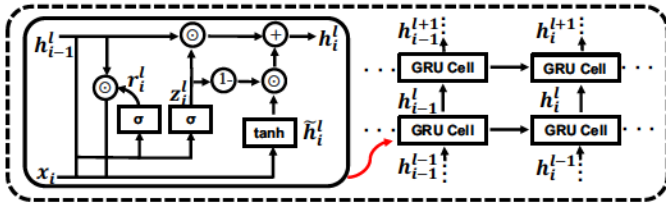


Fig. 2. Mechanism of a GRU cell

In this section, we introduce our GRU-based joint channel coding and modulation (JCM) model design.

A. GRU Model

GRU is a variant of RNN architecture that inherits the desired properties of RNN while alleviating the vanishing gradient problem encountered by traditional RNNs. Compared to other alternative architectures such as LSTM, GRU achieves comparable performance with faster computation speed.

In our JCM scheme, both the encoder and decoder adopt the GRU model. Fig. 2 illustrates the working mechanism of a GRU cell. At time step i and layer ℓ , a GRU cell takes the input $x_i \in \mathbb{R}^d$ and previous hidden state $h_{i-1}^\ell \in \mathbb{R}^h$ to compute the output controlled by the reset gate $r_i^\ell \in \mathbb{R}^h$ and the update gate $z_i^\ell \in \mathbb{R}^h$. These gates are calculated by:

$$\begin{aligned} r_i^\ell &= \sigma(x_i W_{xr}^\ell + b_{xr}^\ell + h_{i-1}^\ell W_{hr}^\ell + b_{hr}^\ell) \\ z_i^\ell &= \sigma(x_i W_{xz}^\ell + b_{xz}^\ell + h_{i-1}^\ell W_{hz}^\ell + b_{hz}^\ell) \end{aligned} \quad (6)$$

where $W_{xr}^\ell, W_{hr}^\ell \in \mathbb{R}^{d \times h}$, $W_{xz}^\ell, W_{hz}^\ell \in \mathbb{R}^{h \times h}$ are weight matrices, and $b_{xr}^\ell, b_{hr}^\ell, b_{xz}^\ell, b_{hz}^\ell \in \mathbb{R}^{1 \times h}$ are bias vectors. σ is the sigmoid activation function that forces each entry of gates to lie within the interval $(0, 1)$. With r_i^ℓ , we calculate the candidate hidden state $\tilde{h}_i^\ell \in \mathbb{R}^h$ with the following rule

$$\tilde{h}_i^\ell = \tanh(x_i W_{xh}^\ell + b_{xh}^\ell + (r_i^\ell \odot h_{i-1}^\ell) W_{hh}^\ell + b_{hh}^\ell) \quad (7)$$

where $W_{xh}^\ell \in \mathbb{R}^{d \times h}$ and $W_{hz}^\ell \in \mathbb{R}^{h \times h}$ are weight matrices, $b_{xh}^\ell, b_{hh}^\ell \in \mathbb{R}^{1 \times h}$ are bias vectors, and symbol $\odot$ is Hadamard

product operator. Finally, we obtain the final hidden state by taking element-wise convex combinations of h_{i-1}^ℓ and $\tilde{h}_i^\ell$ as

$$h_i^\ell = z_i^\ell \odot h_{i-1}^\ell + (1 - z_i^\ell) \odot \tilde{h}_i^\ell. \quad (8)$$

where the update gate z_i controls the update rate.

In the proposed JCM model, we leverage multi-layer bi-directional GRU for better performance. For a GRU of layer L , the GRU cell in the first layer takes the input x_i at time step i and generates hidden state h_i^1 . Layer ℓ with $2 \leq \ell \leq L$ takes the hidden state of the previous layer $h_i^{\ell-1}$ as input and produces hidden state h_i^ℓ . Furthermore, in order to produce more inter-bit correlation in the encoded signal, we adopt bi-directional GRU in the encoder and decoder. Bi-directional GRU is the combination of two independent GRUs, one of which takes the input with its original order and the other one with the inverse order.

B. Proposed JCM Structure

The architecture of our JCM scheme is shown in Fig. 1. In our JCM architecture, the encoder module consists of a multi-layer GRU model, a linear projection layer, and a hyperbolic tangent ($\tanh$) function as shown in Fig. 1. The transmitter encoder module generates the baseband controlling signal f_b to a DCO. The RF signal generated by the DCO is transmitted through the wireless AWGN channel to the receiver module. At the receiver, a packet detection unit constantly monitors received samples to detect a packet and to enable the decoder upon packet detection. The decoder module jointly demodulates and decodes the received packet.

For a packet with N_b information bits, the input bits $b \in \{0, 1\}^{N_b}$ are subject to independent Bernoulli distribution with $\Pr(b_i = 0) = \Pr(b_i = 1) = 0.5$. The encoder has a bi-directional GRU with L^{enc} layers and hidden state length of N_h^{enc} . We set input dimension $d = 1$ such that the GRU takes one bit at each time step. That is, for the first GRU layer, $x_i = b_i$ holds. The hidden state of the last GRU layer $h_i^{L^{\text{enc}}}$ is fed to the linear projection layer whose output size is determined to match the coding rate R . The signal duration per bit is given by $1/Rf_b$, which corresponds to $M = f_s/Rf_b$ samples per bit. We choose proper f_s and R so that M is an integer. Thus, weight W^{enc} and bias b^{enc} of the projection layer have the dimension of $W^{\text{enc}} \in \mathbb{R}^{2N_h^{\text{enc}} \times M}$ and $b^{\text{enc}} \in \mathbb{R}^M$. The output signal is restricted to $(-1, 1)$ by the $\tanh$ function and is scaled by the maximum instantaneous baseband frequency $f_{\max}$ to generate the output baseband DCO controlling signal $f_b \in (-1, 1)^M$. The baseband DCO control signal from the encoder model is expressed by

$$f_b = f_{\max} \tanh[W^{\text{enc}} h_N^{\text{enc}} + b^{\text{enc}}]. \quad (9)$$

It is worth noting that the proposed JCM signal *does not include any explicit pilot or preamble* to assist packet detection and carrier frequency offset (CFO) estimation/correction.

The receiver down-converts the RF signal to the baseband and stores the complex samples in a buffer (shift register). The packet detection unit (PDU) determines if the whole packet is received. If the PDU produces a positive output

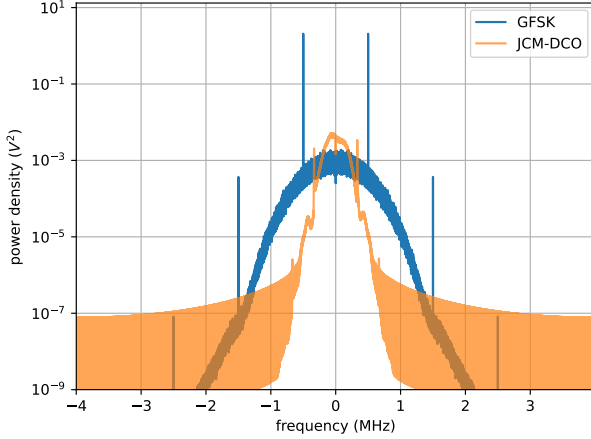


Fig. 3. Spectrum of the proposed scheme compared to the spectrum of BLE-GFSK modulation

(indicating packet detection), the samples are fed to the decoder module. Similar to the encoder, the decoder consists of a GRU with L^{dec} layers and a linear projection layer. Since most deep learning libraries (e.g., PyTorch) only implement neural networks for real inputs, we concatenate the real and imaginary parts of complex samples as the real-valued input to the decoder module. Hence, each GRU cell in the first layer of the decoder takes $2M$ samples at each time step. The linear projection layer with weight $\mathbf{W}^{\text{dec}} \in \mathbb{R}^{1 \times N_h^{\text{dec}}}$ and bias $b^{\text{dec}} \in \mathbb{R}$ maps the hidden state of the last layer of decoder GRU to a real value, which is then transformed into the probability $p[i] = \Pr(b_i = 1)$ by a sigmoid function. The i -th decoded bit is then determined by $\hat{b}_i = p[i] \stackrel{1}{\geq} 0.5$. The decoded bit probability has the form of

$$\mathbf{p} = \sigma [\mathbf{W}^{\text{dec}} \mathbf{h}_N^{\text{dec}} + \mathbf{b}^{\text{dec}}]. \quad (10)$$

C. Bandwidth Control

A practical communication system needs to obey certain frequency band restrictions. Although we have restricted the maximum instantaneous frequency $f_{\max}$ of the DCO controlling signal, abruptly changing instantaneous frequency creates spurious emissions that could potentially exceed the bandwidth constraints. Therefore, similar to the pulse-shaping filter in GFSK modulation, we add an additional low-pass filter to the instantaneous frequency samples $f_b[n]$ before feeding them to the DCO. Specifically, we applied a (discrete version of) root-raised cosine filter, which has the following impulse response

$$h(t) = \frac{2\beta}{\pi\sqrt{T}} \frac{\cos[(1+\beta)\pi t/T] + \frac{\sin[(1-\beta)\pi t/T]}{4\beta t/T}}{1 - (4\beta t/T)^2}. \quad (11)$$

The spectrum of DCO output before and after filtering is shown in Fig. 3. We tuned our filter such that the output bandwidth and spurious emissions follows the requirements of BLE physical layer standard. Table I shows the power spectrum evaluation result using 10dBm signal power (power class 1.5 in BLE).

TABLE I
SPECTRAL PROPERTIES OF JCM COMPARED TO BLE STANDARD

	Evaluated	BLE requirement
6dB bandwidth	670.60kHz	$\geq 500\text{kHz}$
$f_{\text{off}} = 2\text{MHz} \pm 500\text{kHz}$	-30.84dBm	$\leq -20\text{dBm}$
$f_{\text{off}} \geq 2500\text{kHz}$	-33.21dBm	$\leq -30\text{dBm}$

D. Packet Detection

In most wireless communication protocols, a preamble is attached at the beginning of each packet for packet detection, and frequency/phase synchronization. As a preamble takes extra time and energy for transmission, we propose a preamble-less packet detection (and implicit frequency synchronization) scheme to *completely eliminate such overhead*. The received samples are stored in a buffer with serial-in and parallel-out accesses, named as received-sample buffer (RSB) of length N_R samples. All samples in RSB are shifted by one position, discarding the oldest sample when a new sample arrives. A uni-directional GRU is used to detect a packet and determine the starting point of the packet. The hidden state of the GRU is mapped to the probability that the starting point of a packet is aligned with the head of the RSB. The samples are fed to the decoder only if the probability is larger than 0.5.

E. Training Methodology

JCM model can be regarded as an over-complete autoencoder pair aiming to communicate a message over a noisy random channel. Therefore, the training goal is to learn a reliable autoencoder that has robust hidden representation against the corruption of channel noise *as well as phase noise and CFO from a practical low power DCO circuit*. Towards that goal, we propose a deliberate training scheme. First, we use binary cross-entropy (BCE) as our loss function for both the JCM model and packet detection model. With batch size denoted as B , the loss function of the JCM model has the form

$$\ell(\mathbf{p}) = -\frac{1}{BN} \sum_{j=1}^B \sum_{i=1}^N b_{j,i} \log(p_j[i]). \quad (12)$$

We adopt Adam optimizer with learning rate 10^{-4} . Following the finding in [10], we train our encoder, decoder, and packet detector module separately in a round-robin fashion. That is, during encoder training, the parameters of the other networks (decoder and packet detector) are fixed, and vice versa. For better model generalization, we train the encoder with a fixed channel SNR ζ^{enc} and train the decoder with various SNR values in the range of $[\zeta_{\min}^{\text{dec}}, \zeta_{\max}^{\text{dec}}]$ (constant for a single packet). We also add phase noise and CFO in training to improve the robustness of the model against these impairments.

We train the packet detector module with various states of RSB placing the received packet at different time slots. We set its training label as 1 when the first sample of the packet is at the head of RSB and 0 otherwise.

This brings two challenges: 1) labels are heavily imbalanced, and 2) the difficulty of correctly detecting the packet depends on the packet start sample index k in RSB. Our

experiments showed that when $k(\neq 0)$ is smaller (closer to the head), the error probability of packet detection is higher as expected. To address these issues, instead of using uniformly distributed k , we sample k with the following PMF:

$$\Pr(k=i) = \begin{cases} 0.5 & , i = 0 \\ -\frac{1}{4\tilde{N}_R - 2} & , -\tilde{N}_R \leq i \leq \tilde{N}_R, i \neq 0 \end{cases} \quad (13)$$

where $\tilde{N}_R = N_R$ initially, and is halved every 50 epochs, until $\tilde{N}_R = 1$.

We leverage a hyperparameter optimization framework Optuna [18] to jointly optimize our model. Optuna implements a Bayesian optimization algorithm, Tree Parzen Estimators (TPE), with sampling and pruning strategies to dynamically construct search spaces for hyperparameters [19]. In our scheme, we use off-the-shelf interfaces of Optuna to determine important hyperparameters. Resulting optimized hyperparameters are listed in Table II

TABLE II
HYPERPARAMETERS DETERMINED BY OPTUNA BASED OPTIMIZATION

L^{enc}	L^{dec}	N_h^{enc}	N_h^{dec}	$\zeta_{\min}^{\text{dec}}$	$\zeta_{\max}^{\text{dec}}$	ζ^{enc}
2	5	30	90	3dB	7dB	7dB

IV. EVALUATIONS

In this section, we evaluate the performance of the proposed JCM scheme and compare it to conventional frequency modulation and error correction codes.

A. Complexity

The complexity of the proposed scheme consists of training complexity and complexity after model deployment. Although the proposed model needs a large training overhead, the training is done only once before deploying the trained model. Thus, we primarily care about the complexity after deployment. According to (6), (7), and (8), the number of operations (multiplications or additions) that each GRU cell requires is $6h(h+d) + 7h$. Therefore, we calculate the number of operations per bit needed for the encoder GRU as $12N_h^{\text{enc}2} + 26N_h^{\text{enc}} + 2(N_l^{\text{enc}} - 1)(12N_h^{\text{enc}2} + 7N_h^{\text{enc}})$. The projection layer in the encoder requires additional $4MN_h^{\text{enc}}$ operations. We estimate the baseband encoding power consumption with the power efficiency of 3 Tera-operations per second per Watt achieved with the low-power neural network processors such as [20], [21]. Thus, with $R = 0.5$, $N_l^{\text{enc}} = 3$, and $N_h^{\text{enc}} = 25$, the power consumption of the encoder is approximately 6.74 mW. For comparison, a typical low-power BLE transceiver consumes 24.42 mW [22]. Similarly, we estimate that our JCM decoder consumes ≈ 214.33 mW. Our main application scenario is an Internet-of-Things network where many low-power devices transmit messages to a powerful gateway (e.g., smartphone). Hence, achieving low power modulation at the transmitter is our primary goal, and the encoder power estimation (6.74 mW for encoding) justifies the practicality of the proposed JCM.

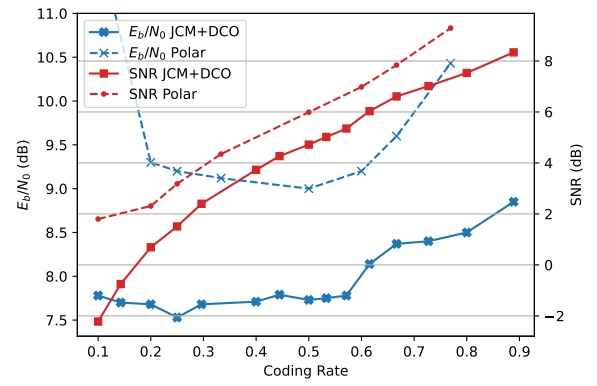


Fig. 4. Minimum E_b/N_0 and SNR required to achieve 0.01 block error rate with block length $N = 100$ bits

B. Performance

We use Monte-Carlo method to simulate the BER and BLER performance of the trained model. Our model is trained with CFO and phase noise for non-coherent communication under the BLE spectrum requirements. Therefore, we use GFSK in the BLE standard with non-coherent demodulation as our baseline. We compare our JCM scheme with GFSK protected by a Polar code with soft-decision successive cancellation list decoding (list size $L = 8$) and also a soft-decision based tail-biting convolutional code (TBCC) with constraint length $M = 8$ and the generator polynomial of 165,357 (in Octal). The data rate for coded bits is 1 Mbps for GFSK. The information rate is R Mbps, which is identical between our JCM and GFSK-Polar / GFSK-TBCC for a given rate R . The sampling rate is $f_s = 8\text{MHz}$.

It is difficult to obtain the theoretical capacity of the proposed DCO-based communication scheme. Thus we numerically evaluate the required E_b/N_0 for different coding rates R . E_b and N_0 denotes energy per information bit and noise power, respectively. Fig.4 shows the minimum E_b/N_0 and SNR ($\frac{E_b}{N_0 R}$) required for different coding rates to get BLER less than 0.01 when the message/block length is $N = 100$ bits. The numerical result shows that, in general, lower E_b/N_0 is required for lower coding rates. However, the decreasing slope is flattened when $R < 0.6$. It is observed $R \approx 0.6$ is an energy-efficient setup without extensive loss of performance in E_b/N_0 perspective. Setups with $R < 0.5$ are still useful when the main objective is longer communication distances at low SNRs. The proposed scheme requires significantly lower (1.5 – 2dB) E_b/N_0 and SNR to achieve a BLER of 0.01 compared to GFSK-Polar for the same rate.

Fig.5 shows the BER and BLER performance under different E_b/N_0 when $R = 0.5$ and message length N is 100 bits. The proposed JCM outperforms TBCC or Polar coded GFSK in the low E_b/N_0 region. The intersection between the B(L)ER curve of the proposed scheme and conventional baselines appears when E_b/N_0 is 9.5 – 10 dB with BER $\approx 10^{-6}$ and BLER $\approx 5 \times 10^{-4}$.

Another important property of the proposed scheme is the robustness against phase noise and CFO. The phase noise is modeled as colored Gaussian random process. The spectrum

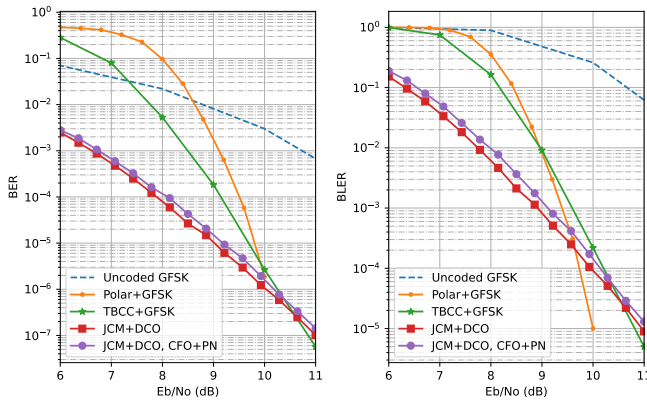


Fig. 5. BER and BLER performance of the proposed scheme compared to conventional soft-input Polar code and TBCC with GFSK modulation, when block length is $N = 100$ bits and coding rate is $R = 1/2$

of the phase noise is centered at the carrier frequency and approximated as a linear interpolation in the log-log domain of measured phase noise power $S(\Delta f)$ at frequency deviation Δf . In our experiments, we consider a phase noise spectrum model shown in Table III representing a conventional LC-tank-based low power free-running DCO (without a PLL) [23]. We assume that CFO is uniformly distributed between ± 150 kHz following the BLE requirement. The BER and BLER performance of our JCM with this realistic phase noise and CFO is shown in Fig. 5. For comparison, GFSK-Polar and GFSK-TBCC performance in Fig. 5 was evaluated *without* phase noise and CFO. Our JCM performance degradation brought by the impairments is small (≈ 0.1 dB) despite that our JCM waveform does not include any explicit preamble to estimate CFO. Note that BLE requires the overhead of 8-bit preamble for packet detection and CFO synchronization.

TABLE III

PHASE NOISE SPECTRA OF PLL-LESS FREE-RUNNING LOW POWER DCO

Δf (kHz)	0.1	1	10	100
$S(\Delta f)$ (dBc/Hz)	-30	-60	-80	-100

V. CONCLUSION

In this paper, we proposed a novel data-driven model for joint channel coding and continuous-phase frequency modulation. The proposed architecture complies with the BLE bandwidth specification and has an encoder that runs with lower encoder power than BLE device. Simulation shows that the proposed scheme outperforms Polar or TBCC code with GFSK modulation in the low E_b/N_0 region under severe phase noise and CFO without the need for preamble.

REFERENCES

- [1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5g: Physical and mac-layer solutions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, 2016.
- [2] R. B. Staszewski, C.-M. Hung, D. Leipold, and P. T. Balsara, "A first multigigahertz digitally controlled oscillator for wireless applications," *IEEE Transactions on Microwave theory and techniques*, vol. 51, no. 11, pp. 2154–2164, 2003.
- [3] X. Chen, A. Alghaihab, Y. Shi, D. S. Truesdell, B. H. Calhoun, and D. D. Wentzloff, "A crystal-less ble transmitter with clock recovery from gfsk-modulated ble packets," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 7, pp. 1963–1974, 2021.
- [4] D. Cherniak, L. Grimaldi, L. Bertulesi, C. Samori, R. Nonis, and S. Levantino, "A 23ghz low-phase-noise digital bang-bang pll for fast triangular and saw-tooth chirp modulation," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 248–250, 2018.
- [5] Texas Instruments, "CC1101: Low-Power Sub-1 GHz RF Transceiver," SWRS061I, 2013.
- [6] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6, IEEE, 2017.
- [7] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346, IEEE, 2016.
- [8] T. O'shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [9] C. Bian, M. Yang, C.-W. Hsu, and H.-S. Kim, "Deep learning based near-orthogonal superposition code for short message transmission," in *ICC 2022-IEEE International Conference on Communications*, pp. 3892–3897, IEEE, 2022.
- [10] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Learn codes: Inventing low-latency codes via recurrent neural networks," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 207–216, 2020.
- [11] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.
- [12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [13] R. Sattiraju, A. Weinand, and H. D. Schotten, "Performance analysis of deep learning based on recurrent neural networks for channel coding," in *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, IEEE, 2018.
- [14] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," *Journal of Lightwave Technology*, vol. 36, no. 20, pp. 4843–4855, 2018.
- [15] S. Mohamed, J. Dong, A. Junejo, *et al.*, "Model-based: End-to-end molecular communication system through deep reinforcement learning auto encoder," *IEEE Access*, vol. 7, pp. 70279–70286, 2019.
- [16] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication phy using neural network," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1364–1373, 2019.
- [17] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Joint channel coding and modulation via deep learning," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2020.
- [18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.
- [19] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.
- [20] Z. Fan, H. An, Q. Zhang, B. Xu, L. Xu, C.-W. Tseng, Y. Peng, A. Cao, B. Liu, C. Lee, Z. Wang, F. Liu, G. Wang, S. Jiang, H.-S. Kim, D. Blaauw, and D. Sylvester, "Audio and image cross-modal intelligence via a 10tops/w 22nm soc with back-propagation and dynamic power gating," in *2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, pp. 18–19, 2022.
- [21] Y. Toyama, K. Yoshioka, K. Ban, S. Maya, A. Sai, and K. Onizuka, "An 8 bit 12.4 tops/w phase-domain mac circuit for energy-constrained deep learning accelerators," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2730–2742, 2019.
- [22] Texas Instruments, "Measuring Bluetooth Low Energy Power Consumption," SWRA347a, 2012.
- [23] B. Razavi, *RF Microelectronics*. Prentice Hall communications engineering and emerging technologies series, Prentice Hall PTR, 1998.