creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

Master of Science

Enhanced Syndrome-based Reliability Decoding for Error Correction Code Transformers

The Graduate School

of the University of Ulsan

Department of Electrical, Electronics and

Computer Engineering

Nguyen Dang Trac

울산대학교
UNIVERSITY OF ULSAN

Enhanced Syndrome-based Reliability Decoding for Error Correction Code Transformers

Supervisor: Prof. Sunghwan Kim

A Master's Thesis

Submitted to

the Graduate School of the University of Ulsan

in Partial fulfillment of the Requirements

for the Degree of

Master of Science

by

Nguyen Dang Trac

Department of Electrical, Electronics and

Computer Engineering

University of Ulsan, Republic of Korea

June 2024

울산대학교
UNIVERSITY OF ULSAN

Enhanced Syndrome-based Reliability Decoding for Error Correction Code Transformers

This certifies that the Master's thesis of Nguyen Dang Trac is approved.

Committee Chair: Prof. Dr. Sungoh Kwon

Committee Member: Prof. Dr. Hee-Youl Kwak

Committee Member: Prof. Dr. Sunghwan Kim

Department of Electrical, Electronics and Computer Engineering

Ulsan, Republic of Korea

June 2024

울산대학교
UNIVERSITY OF ULSAN

ACKNOWLEDGEMENTS

First of all, I would like to give my utmost gratitude to my advisory professor, Prof. Dr. Sunghwan Kim for his guidance and support throughout the whole process of this research from the very beginning. Throughout the two years of this program, Prof. Kim had always been very appreciative towards the direction and effort that I brought into my works. Furthermore, Prof. Kim was also willing to support me during my difficult times. Over the course of two years studying under his supervision, I have gathered so much valuable lessons that would support a career ahead of me. Therefore, I am very much grateful for my time working with Prof. Kim.

In the context of this thesis, I would also like to express my gratitude toward Prof. Dr. Sungoh Kwon, thesis defense committee chair, and Prof. Dr. Hee-Youl Kwak, a committee member, for their insightful comments for the research contents that I present in this thesis. The contribution of this thesis is enhanced partially thanks to the committee members constructive comments.

This two year program and all of this learning opportunity would not been available to me if it would not for the permission for study and scholarship support our university, the University of Ulsan and the Department of Electrical and Electronics Computer Engineering. Therefore, I would like to give my thanks to the officials and Professors of our Department and our University working day

울산대학교
UNIVERSITY OF ULSAN

in day out and operating the academic process.

This journey has been one full of invaluable lessons, among which are those I received from my seniors, lab mates and friends. Along the road for my research before the finalization of this research work, there are moments that, if it was not for their precious advice, could have been regrettable missteps for me. Therefore, I would like to give my sincere thanks to my dear seniors and friends whom, despite them not knowing it, has been very supportive and walk with me on this journey. I trust they will have a wonderful and successful career that they deserve for the efforts that they put into their works.

Finally and absolutely most importantly, I want to sent a thanks from the bottom of my heart to my family, my dear mother, sister, and above all, my dear father who passed beyond my reach when it was only four months after I started this program. Up to this very moment, every member in my family has been so supportive to me, which means more to me than I believe they are aware of. And even at his last moments, my father did not even have a hint of intention to permit me giving up on my study. I am grateful for all of that and hoped that I can take care and support my family like they did me. If it was not for my family, there could never be any part of myself existed to this point in time.

울산대학교
UNIVERSITY OF ULSAN

ABSTRACT

In this work, dynamically adaptive refinement masking (DARM) and mirror-sharing variational U-shaped architecture are proposed in order to improve the performance of error correction code transformer (ECCT). Instead of fixed binary masking, DARM module is designed to create unique masking for each attention head that reinforces the self-attention mechanism by further enhancement of the contrast in the attention map with dynamic magnitudes taking the distribution of the attention weight as reference. Furthermore, under the use of sequential neural architecture, DARM modules are designed to be sequentially connected, creating an iterative refinement effect. For moderate coding lengths, the mirror-sharing variational U-shaped architecture is introduced to enhance the overall efficiency of the transformer-based decoder. The U-shaped architecture with variational-autoencoder-like skip-connection provides a segmentation like behavior that operates well with moderate length codes, especially at low coding rates. As the U-shaped model requires a certain level of depth to achieve desirable performance, an architectural-level parameter-sharing scheme called mirror-sharing is introduced to effectively scale the U-shaped model to achieve better efficiency and performance. Experimental results show considerable improvements in bit error rates compared to the baseline ECCT, while also significantly increasing the training convergence speed.

울산대학교
UNIVERSITY OF ULSAN

TABLE OF CONTENTS

울산대학교 UNIVERSITY OF ULSAN

울산대학교
UNIVERSITY OF ULSAN

울산대학교
UNIVERSITY OF ULSAN

LIST OF FIGURES

울산대학교
UNIVERSITY OF ULSAN

울산대학교
UNIVERSITY OF ULSAN

LIST OF TABLES

울산대학교
UNIVERSITY OF ULSAN

CHAPTER 1

INTRODUCTION

1.1 Motivation

Among the technologies utilized in wireless communication systems, error correction codes (ECCs) is one of the essential factors contributing to the reliability of data transmission. ECCs are known for their capability of correcting errors in the corrupted received signals by relying on complex likelihood decoding algorithms such as belief propagation (BP) decoding. Despite its their advantages, the NP-hard level of complexity for optimization of the decoding algorithms have also challenged the advancement of ECCs.

In recent years, the learning-based ECC decoding have gained favor due to their superior performance compared to traditional methods. The different approaches range widely from model-free designs [12–15] to model-based adapta-

울산대학교
UNIVERSITY OF ULSAN

tion of traditional decoding algorithm with neural networks [4,5,9–11]. While the model-based designs adopting the BP algorithm suffer from the complexity of the Tanner-graphs, the model-free approaches, though less restrictive, are hindered by the curse of dimensionality, especially when dealing with significantly large code length.

As an alternative to direct codeword estimation, a neural network with model-free syndrome-based decoding [16] focused on training the network to estimate the multiplicative noise from the channel output and removing the noise to obtain the approximated transmitted signal. By adopting this principle, the authors of [17] proposed the transformer-based method, namely error correction codes transformer (ECCT), that successfully adapted the components of the original transformer in [18] to decoding ECCs. The ECCT in [18] was able to achieve a significant reduction in error probability compared to the state-of-the-art BP-based neural decoding approaches as well as other previous learning-based models including that of [16]. Nevertheless, this approach was the first attempt at transformer-based decoders, which had not fully explored potential of transformer-based decoder through the reliability decoding.

The research on neural network architectural design has been highly successful, yielding renowned models such as ResNet [25, 26], DenseNet [27, 28], and MobileNet [29]. Among the popular architectural designs, U-Net [30] is known as the state-of-the-art image segmentation convolutional model. While it is well-

울산대학교
UNIVERSITY OF ULSAN

known for its application in the field of computer vision, many researchers have successfully adapted this design to time-series data and other fields of research, as well as combining it with models other than convolutional neural networks.

1.2 Contribution

As an attempt to explore the compatibility of transformer and reliability decoding as well as improving the performance of ECCT [17], this work proposes a dynamically adaptive refinement masking (DARM) mechanism and the variational U-shaped architecture. The proposal of DARM addresses problem of unexplored interaction between the transformer-based model and the syndrome-based reliability decoding. On the other hand, the design of variational U-ECCT aims to enhance performance of the ECCT at moderate code lengths while ensuring model efficiency through parameters sharing.

To further enhance the code-aware self-attention mechanism with the explored learning pattern, DARM is proposed as a learnable mask generating mechanism. DARM creates filtered masks that adapt to the attention dot-product matrices through a set of adaptive thresholds uniquely obtained for each attention head. Along each row of its input matrix, this filtering mechanism reduces the range of positive values below peak-magnitude. The filtered information is then dynamically learned through a parameterized transformation. The combination of these steps is considered a sub-max reduction transformation. The DARM is com-

울산대학교 UNIVERSITY OF ULSAN

pleted with two consecutive sub-max transformation. The input and output of the first sub-max transform operation are both transposed, forming a module akin to a mask-mixing mechanism.

Inspired by the U-Net model of [30], a U-shaped architecture for ECCT is proposed to improve the high-rate, moderate-length decoding performance. This architecture consists of two major network flows: the main flow and the shortcut connections. While the decoding process of the main network flow is similar to the baseline ECCT, the shortcut connections interact with the main flow through a multiplicative operator. With this mechanism, the proposed model learns to generate and combine two different segments of the embedding dimension, retaining common information between the two flows, which significantly enhances the decoding process, particularly targeting large code-length cases. Furthermore, the U-ECCT is designed with multi-level scaling with four different levels of hidden dimension, providing flexibility and reducing model size compared to the baseline ECCT [17].

In order to improve the general performance of the U-ECCT at all code rates, the U-shaped architecture is combined with the variational autoencoder (VAE) model to form a variational U-ECCT. In our improved model, the interaction between the main network flow and the shortcut connections is modified to resemble the VAE. This modified U-ECCT model provides a way to estimate the common distribution of the shortcut connections. By this effect, the large syndrome length

울산대학교
UNIVERSITY OF ULSAN

is fully utilized, enabling significant improvements in low-rate codes.

Finally, in an effort to enable an efficient U-ECCT design, a new weight-sharing strategy is proposed, called mirror-sharing. Our weight-sharing strategy is applied so that the second half of the U-shaped model reuses the parameters of its first half in a reflective manner. This method effectively compresses our proposed U-ECCT models, enabling a deep, light-weight model that performs better than the baseline ECCT.

In experimentation, the DARM-equipped models achieves considerable improvements in increasing the conversion speed during training as well as reducing bit error rates (BER) during inference stage, resulting in an performance gains of 0.3dB for Bose-Chaudhuri-Hocquenghem (BCH) codes and low-density parity-check (LDPC) codes, and up to 0.6dB for polar codes compared to the baseline model of [17]. The variational U-ECCT models achieve significantly better performance than the baseline ECCT for moderate code-length cases (from 256 to 1024), and still provide noticeable improvement for short code-length cases (less than 256). In terms of decoding performance, the the U-shaped model provides improvements that vary between 0.3dB and 0.5dB for moderate code-length cases, depending on the code structure and code rates. Especially for low code rate cases, the variational design is able to achieve improvements between 0.5dB and 0.65dB. Regarding efficiency, our models with a mirror-sharing strategy provide effective compression, achieving up to 73% reduction in the number of train-

울산대학교
UNIVERSITY OF ULSAN

able parameters compared to the baseline ECCT [17].

The rest of this paper is organized as follows. Chapter 2 introduces the related knowledge to this work. Chapter 3 consists of detailed explanations for proposed methodology. The experimental results and discussions are described in Chapter 4. Finally, we provide our conclusions to this work in Chapter 5.

울산대학교
UNIVERSITY OF ULSAN

CHAPTER 2

BACKGROUND AND RELATED WORKS

In this chapter, necessary background knowledge is provided for coding, syndrome-based reliability decoding for model-free neural network decoder [4], transformer [18], key features of the ECCT [17], Integrated Gradients [41] for model interpretation, U-Net and variational autoencoder (VAE).

2.1 Coding

In a standard wireless communication system, the transmission process employs a linear code C to encode an input message $\mathbf{s} \in \{0,1\}^k$ into a codeword $\mathbf{x} \in \{0,1\}^n$ by calculating the Galois field matrix product with a generator matrix

울산대학교
UNIVERSITY OF ULSAN

$\mathbf{G}$. The linear code is defined by a binary generator matrix $\mathbf{G}$ of size $k \times n$ and a binary parity-check matrix $\mathbf{H}$ of size $(n-k) \times n$, such that $\mathbf{G}\mathbf{H}^{\mathrm{T}} = 0$, where T denotes the matrix transposition. A codeword $\mathbf{x}$, satisfying $\mathbf{H}\mathbf{x} = 0$, is then transmitted via a binary-input-symmetric-output, additive white gaussian noise (AWGN) channel. The channel output can be modeled as $\mathbf{y} = \mathbf{x}_s + \mathbf{z}$, where $\mathbf{x}_s$ is the modulated codeword (commonly considered binary phase shift keying (BPSK)), and $\mathbf{z}$, defined by $\mathbf{z} \sim \mathcal{N}(0, \sigma^2)$, represents the AWGN which is independent of the codeword $\mathbf{x}$. Without loss of generality, the modulated codeword $\mathbf{x}_s$ can be obtained using $\mathbf{x}_s = bipolar(\mathbf{x})$, where $bipolar(\cdot)$ represents the conversion by 0 to 1 and 1 to -1, and is also obtainable as $bipolar(a) = 1 - 2a$ with input $a \in \{0, 1\}$.

At the decoder, the main objective is to provide a soft estimation $\hat{\mathbf{x}}$ of the original codeword. Thus, the decoding function can be viewed as a function of $\mathbf{y}$, which is $\hat{\mathbf{x}} = f(\mathbf{y})$. A basic decoding method for linear block codes is syndrome-decoding. This decoding scheme is performed by obtaining the syndrome $s(\mathbf{y})$ performing the Galois field matrix product between the hard-decision mapping $\mathbf{y}_b$ of $\mathbf{y}$ and the parity-check matrix $\mathbf{H}$. The syndrome $s(\mathbf{y})$ is defined by $s(\mathbf{y}) = \mathbf{H}\mathbf{y}_b \in \{0, 1\}^{n-k}$. In an ideal non-erroneous transmission, the received codeword is expected to yield a zero syndrome $\mathbf{H}\mathbf{y}_b = 0$. However, in practical transmission, a certain level of errors is anticipated. In such cases, the syndrome $s(\mathbf{y})$ is non-zero $\mathbf{H}\mathbf{y}_b \neq 0$ and is utilized to identify positions and values of the errors. For a single-bit error, basic syndrome decoding is adequate. However, when multiple error

울산대학교
UNIVERSITY OF ULSAN

bits occur in the same codeword, more advance likelihood decoding techniques are required, such as belief propagation (BP), min-sum, or maximum likelihood decoding (ML), all of which incorporates the principle of Tanner-graph involving the parity-check matrix $\mathbf{H}$.

2.2 Syndrome-based reliability decoding system model

The Syndrome-based reliability decoding model is presented in Fig. 2.1 which follows the model proposed in [17]. The general principle of communication system and coding considered for this model is explained in Section II.A. In the current section, the pre-processing and post-processing methods proposed by [16] for the receiver are discussed. These methods are regarded as an extended version of symdrome-based decoding that includes channel reliabilities, aiming to overcome overfitting by eliminating the need to simulate codewords during training.

The pre-processing step modifies channel output $\mathbf{y}$ with a vector of length $2n-k$, which can be defined as

$$\tilde{\mathbf{y}} = h(\mathbf{y}) = |\mathbf{y}| \sqcup s(\mathbf{y}), \tag{2.1}$$

where $\sqcup$ denotes the vector concatenation and $|\mathbf{y}|$ represents the element-wise magnitude of $\mathbf{y}$.

The binary hard-decision transformation of $\mathbf{y}$ can be obtained by $\mathbf{y}_b = \frac{1}{2}(1 - sign(\mathbf{y}))$. The sign function $sign(\cdot)$ is defined by

울산대학교
UNIVERSITY OF ULSAN

$$sign(c) = \begin{cases} 1, & c > 0, \\ 0, & c = 0, \\ -1, & c < 0. \end{cases} \tag{2.2}$$

The post-processing step multiplies the bipolar mapping of the channel output $\mathbf{y}$ with the decoder output to create the estimation $\hat{\mathbf{x}}$ of the original codeword $\mathbf{x}$. To be more specific, the prediction takes the below form

$$\hat{\mathbf{x}} = \mathbf{y} \otimes f_\theta(\tilde{\mathbf{y}}), \tag{2.3}$$

where $f_\theta(\cdot)$ denotes the neural decoder parameterized by θ and $\otimes$ is the component-wise multiplication.

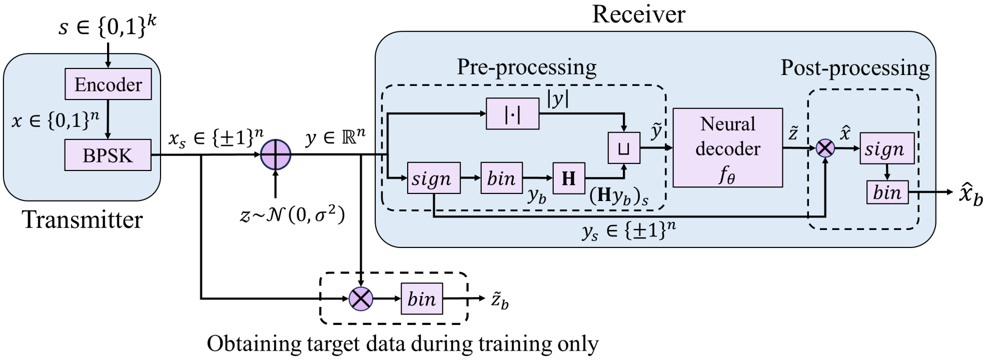


Figure 2.1: System model for syndrome-based reliability decoding framework.

The objective of the neural decoder is to generate an estimation which mitigates the effect of channel noise and recovers the original codeword when multiplied with the channel output $\mathbf{y}$. With such approach, syndrome-based decoding

울산대학교
UNIVERSITY OF ULSAN

proposed method in [16] focused on prediction of the multiplicative noise $\tilde{\mathbf{z}}$. Let $\tilde{\mathbf{z}}_s$ denote the soft multiplicative noise such that $\mathbf{y} = \mathbf{x}_s \otimes \tilde{\mathbf{z}}_s$ (proof of Lemma 1 in [24]), thereby obtaining the following expression $\tilde{\mathbf{z}}_s = \tilde{\mathbf{z}}_s \otimes \mathbf{x}_s^2 = \mathbf{y} \otimes \mathbf{x}_s$. However, since the target of prediction is the occurrence of bit-flipping, the bipolar problem of $\tilde{\mathbf{z}}_s$ can be simplified to a binary problem where the detected bit-flipped case is labeled as 1 and non-erroneous case as 0. Hence, the target data for the loss function is the binary the multiplicative noise, defined by $\tilde{\mathbf{z}}_b = bin(sign(\mathbf{y} \otimes \mathbf{x}_s))$. With such objective, the loss function for training the neural decoder is chosen as the binary cross-entropy (BCE) loss between the decoder output $f_\theta(\mathbf{y})$ and the target $\tilde{\mathbf{z}}_b$, which is formalized as

$$\mathscr{L}_{\text{BCE}} = -\sum_{i=1}^{n} (\tilde{\mathbf{z}}_{b,i} \log(f_\theta(\tilde{\mathbf{y}}_i)) + (1 - \tilde{\mathbf{z}}_{b,i} \log(1 - f_\theta(\tilde{\mathbf{y}}_i))). \tag{2.4}$$

Ultimately, the estimation of the original codeword $\mathbf{x}$ is obtained by $\hat{\mathbf{x}}_b = \frac{1}{2}(1 - sign(f_\theta(\tilde{\mathbf{y}}) \otimes sign(\mathbf{y})))$.

2.3 Transformer

The transformer was introduced in [18] as a successor to RNN in sequence-to-sequence learning tasks. The vanilla transformer in [18] consists of two major components: the positional encoding and multi-head attention mechanism. Currently, the transformer not only dominates the performance scale in the field of natural language processing, but also emerges as the leading model in multiple

울산대학교
UNIVERSITY OF ULSAN

mainstream topics including language-based classification [31,33], time series data analysis [34], and image classification [35]. Several researchers have also adopted the design of the transformer to enhance the communication system such as 6G intelligent network designs [37], and attention-inspired feedback codes [38, 39].

At the heart of the transformer, and also the key to its success, is the multi-head attention mechanism. By employing a learnable scaled dot-product mapping of a vector pair, specifically query and key, the model generates attention weights capturing the relation between each element to every other elements within the same sequence (self-attention) or between a reference query sequence and another sequence projected into key and value vectors (cross-attention). The scaled dot-product attention can be formulated as

$$A(Q,K,V) = Softmax(\frac{QK^{\mathrm{T}}}{\sqrt{d_h}})V, \tag{2.5}$$

where d_h is the dimension of the learned linear projections of query and key vectors, which is originally calculated by d_{model}/h. Herein, d_{model} denotes the embedding dimension after performing positional encoding on the input sequence and h represents the number of parallel heads during the linear projection of query, key, and value vectors.

The multi-head self-attention is formulated as $MHSA(Q,K,V) = Concat(head_1,...,head_h)W^O$, where $head_i = A(QW_i^Q, KW_i^K, VW_i^V)$. The linear projections of Q,K,V, and the attention output are parameterized by matrices $W_i^Q \in \mathbb{R}^{d_{\mathrm{model}} \times d_h}$, $W_i^K \in \mathbb{R}^{d_{\mathrm{model}} \times d_h}$, $W_i^V \in \mathbb{R}^{d_{\mathrm{model}} \times d_h}$, and $W_i^O \in \mathbb{R}^{hd_h \times d_{\mathrm{model}}}$. As explained

울산대학교 UNIVERSITY OF ULSAN

in [18], the masked self-attention module takes the summation of causal masks and the dot product of the query-key vector pair. This mechanism was proposed to prevent the language decoder from attending to subsequent positions in the sequence, thereby avoiding the network from learning to simply copy the input.

2.4 Error Correction Code Transformer

The authors of [17] introduced two key features, namely positional reliability encoding and code-aware self-attention, which will be discussed in this subsection.

2.4.1 Positional reliability encoding

In [17], the authors proposed a unique positional encoding method that provides more attention to the magnitude of the input sequence (i.e., the pair of channel corrupted codeword and its respective syndrome). Each dimension of $\{\tilde{\mathbf{y}}_i\}_{i=1}^{2n-k}$ in the positional encoding is regarded as the pre-processed version of the channel output $\mathbf{y}$ by (2.1). The encoding process projects the vector $\tilde{\mathbf{y}}$ to a d_{model} dimensional embedding $\{\phi_i\}_{i=1}^{2n-k}$ which is defined by

$$\phi_i = \begin{cases} |\mathbf{y}_i| W_i, & \text{if } i \leq n, \\ (1 - 2(s(\mathbf{y}))_{i-n+1}) W_i, & \text{otherwise,} \end{cases} \tag{2.6}$$

where $\{W_i \in \mathbb{R}^d\}_{i=1}^{2n-k}$ denotes the one-hot encoding vector corresponding to each bit position of $\tilde{\mathbf{y}}$. As shown in [17], this embedding method yields a desirable

울산대학교
UNIVERSITY OF ULSAN

effect during the self-attention dot product operation such that the unreliable information (i.e., low magnitude positions) would collapse to the origin, whereas the syndrome bits yield negative scaling. The proposed encoding can be understood as a positional encoding corresponding to the input reliability, hence the name positional reliability encoding.

2.4.2 Code-aware self-attention

Similar to the conventional decoding process that relies on the parity-check matrix, the transformer counterpart introduced in [17] is designed to incorporate unique sparse masks capturing the patterns of the parity-check matrix for each respective code type. Since each bit of the input vector is not necessarily related to all other bits, traditional causal masking as well as non-masking would yield sub-optimal performance.

Algorithm 2.1 Pseudo-code of the binary mask generating algorithm

```
Input: Parity-check matrix $\mathbf{H}$
Output: Mask based on $\mathbf{H}$
$\mathbf{M} \leftarrow I(2n-k)$
for $i$ = 1, 2, ..., $n-k$ do
    $idx \leftarrow$ where($\mathbf{H}[i] == 1$)
    for $j \in idx$ do
        $\mathbf{M}[n+i, j] \leftarrow 1$
        $\mathbf{M}[j, n+i] \leftarrow 1$
        for $h \in idx$ do
            $\mathbf{M}[j, h] \leftarrow 1$
            $\mathbf{M}[h, j] \leftarrow 1$
        end
    end
end
$\mathbf{M} \leftarrow (-\infty(\neg\mathbf{M}))$
```

울산대학교
UNIVERSITY OF ULSAN

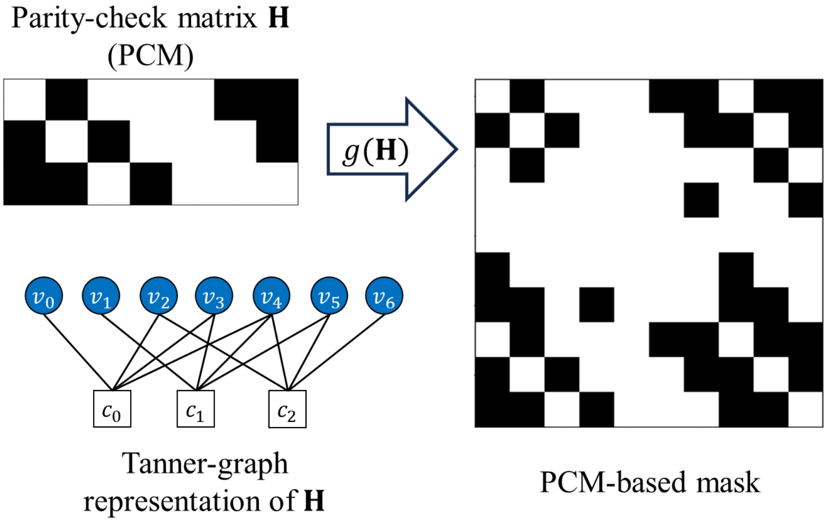


Figure 2.2: Demonstration of parity-check matrix for Hamming (7,4) with its equivalent representation in Tanner-graph and attention mask.

Thus, for a parity-check matrix **H**, the parity-check-based attention mask is denoted by $g(\mathbf{H}) : \{0,1\}^{(n-k)\times n} \rightarrow \{-\infty, 0\}^{(2n-k)\times(2n-k)}$. First, the mask is initialized with an identity matrix. Taking reference to each row of the parity-check matrix **H** at a time, for every bit *one* in **H**, a pair of positions (simultaneously considering row and column) in $g(\mathbf{H})$ is unmasked, i.e, assigned with *one*. Consequently, this process provides a symmetric mask containing information about every pairwise bit relations. The construction of the mask $g(\mathbf{H})$ is explained by Algorithm 2.1. An example of the generated mask is shown in Fig. 2.2.

In light of the given objective function (2.6), the ECCT design is utilized as the neural decoder to effectively achieve the estimation of the multiplicative noise. Considering the binary mask $g(\mathbf{H})$ generated by Algorithm 2.1, the masked

울산대학교
UNIVERSITY OF ULSAN

self-attention mechanism for ECCT can be defined as

$$A_{\text{mask}}(Q,K,V) = Softmax(\frac{QK^{\mathrm{T}} + g(\mathbf{H})}{\sqrt{d_h}})V. \tag{2.7}$$

By summation of $g(\mathbf{H})$ and the query-key dot product, the values of desirable positions in the product are retained while the prohibited positions theoretically yield $-\infty$. This value assignment is designed with consideration of *Softmax* activation. Positions assigned a value of $-\infty$ collapse to the nearest possible value to *zero* after the *Softmax* is applied. This prevents any information of the corresponding positions from influencing attention weights. Hence, the pattern of attention weight matrix also adheres to the symmetric pattern of $g(\mathbf{H})$, resulting in a *code-aware attention* mechanism [17]. While sustaining useful information learned by the dot product, the binary design only acts as a restriction against learning non-desirable positions. As shown by the works of [21–23], dynamic masking instead of binary masking has the potential to encourage the attention process to more effectively extract desirable information.

2.5 Integrated Gradients

Integrated gradients (IG) [41] is a gradient-based neural network interpreting method. IG explain the learning pattern by integrating the gradient $\nabla f_\theta(a)$ along an i^{th} dimension in input space. In the simplest form, the trajectory is chosen to be the segment $(\tilde{a}, a)$ connecting the baseline $\tilde{a}$ to an input a. Integrated gradients

울산대학교
UNIVERSITY OF ULSAN

define feature-wise scores as

$$\mathscr{R}_i(a) = (a_i - \tilde{a}_i) \times \int_0^1 (\nabla f_\theta(\tilde{a} + t \times (a - \tilde{a})))_i dt. \tag{2.8}$$

It is proven by the author of [40] that Integrated gradients satisfy an axiom called *Completeness* where the summation of attribution scores for all dimensions is equal to the difference between the output of $f_\theta(\cdot)$ at input a and baseline $\tilde{a}$ ($\sum_i \mathscr{R}_i(a) = f_\theta(a) - f_\theta(\tilde{a})$). For most of the tested neural network models, it is shown that the baseline can be chosen such that the prediction on the baseline point is near-zero. Hence, it can also be inferred that the accumulative scores is approximately equal to the neural network output at input x, which is $\sum_i \mathscr{R}_i(a) \approx f_\theta(a)$. In such cases, the attributions can interpreted as ignoring the baseline and distributing the output to each input features.

2.6 U-Net

In the field of computer vision, a popular convolutional network architecture called U-Net was proposed in [30] for semantic segmentation in images. This architecture is designed with one side for down-sampling and the other side for up-sampling operations. The features extracted by the down-sampling side are also forwarded and learned by the up-sampling side through skip connections, concatenation, and re-scaling modules. As a result of its success in the initial proposed problem set, U-Net was later combined with other neural network

울산대학교
UNIVERSITY OF ULSAN

families to provide efficient solutions for a variety of tasks, including U-shape swin transformers (SUNet [42]) and U-Net VAE [43] for image denoising, U-Net transformer (Ds-transunet [44]) for enhanced medical image segmentation, one-dimension convolutional U-Net [45] for times series learning, and Seq-U-Net [46] for sequence modelling. In the majority of its applications, U-Net was proposed due to its highly effective capability for segmenting different parts of the superimposed input data and extracting hidden patterns that enable restoration of distorted information. In this context, we note that U-Net architecture has great potential in noise segmentation problems such as channel decoding in communication systems.

2.7 Variational Autoencoder

Initially introduced in [47], VAE is one of the successful deep generative models. As suggested by its naming convention, this variant of the autoencoder belongs to the family of methods called variational Bayes methods. In general, this family of methods is used to solve the variational inference problem, where a posterior distribution of unobserved latent variables given some input data is approximated by a variational distribution. VAE operates as a parameterized model to optimize the process of discovering the variational distribution across all input data points.

This process is achieved by using an encoder neural network to map the

울산대학교
UNIVERSITY OF ULSAN

input data to a latent space. Then, a separate parameterized model learns the mean and variance of the encoded latent space to generate new samples from the prior distribution. Finally, an decoder network approximates the distribution of the input data (the variational distribution) by extracting the features from the newly generated samples. A combination of VAE and transformers had also been proposed for generative tasks [48]. In the communication point of view, VAE has also been proposed for channel coding [49, 50] and joint source-channel coding [51].

울산대학교
UNIVERSITY OF ULSAN

CHAPTER 3

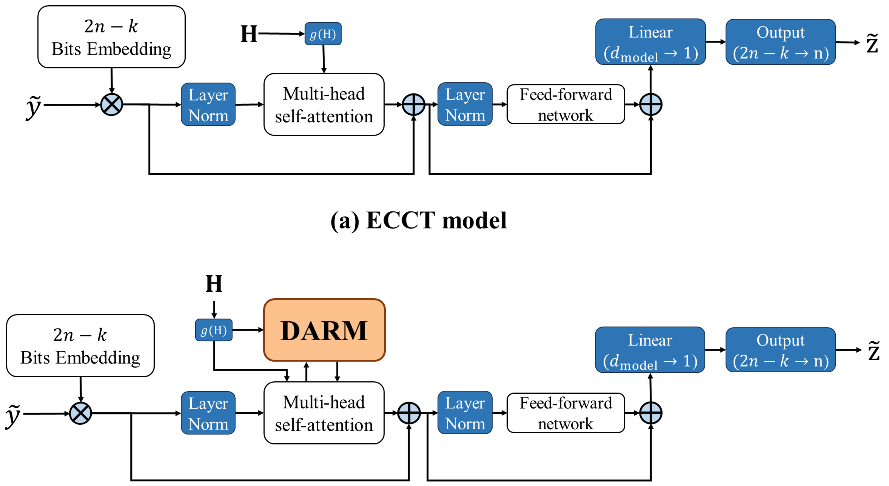
THE PROPOSED METHODS

In this chapter, the proposed ECCT is discussed in details. The first contribution, DARM, is a dynamic mask generating mechanism that learns to adapt and enhance reliable information detected by the query-key dot product of the self-attention mechanism. The second contribution, the variational U-ECCT, is designed to improve the ECCT performance for moderate code lengths. In Fig. 3.1, the overall transformer architecture including the proposed ECCT is presented.

3.1 Intuitive interpretation of ECCT learning pattern

Since our focus is effectively learning the reliable information from the channel output, the attention mechanism is regarded as the reliability learning problem (attention to high magnitude positions). In the original design of the self-attention mechanism, the objective is learning the relation between different po-

울산대학교
UNIVERSITY OF ULSAN



(a) ECCT model

(b) DARM ECCT model

Figure 3.1: Overview of system architecture for the proposed ECCT.

sitions within the given sequence. This is slightly different from the code-aware attention introduced in [17] for the syndrome-based reliability decoding. As explained with the syndrome-based decoder system model in **??** and its objective function (2.6), the model target of estimation is the binary multiplicative noise. Therefore, it can be inferred that the self-attention learning pattern is focusing on the relation between the erroneous positions and the rest of the input information. Following the usual behavior of the self-attention mechanism, the softmax attention matrix is anticipated to exhibit significantly higher values along the columns corresponding to the erroneous positions (reliable information) while assigning lower values to other positions (unreliable information). Henceforth, stronger contrast in softmax attention matrix between reliable and unreliable information can

울산대학교 UNIVERSITY OF ULSAN

lead to better decoding performance for the transformer-based model.

3.2 Dynamically Adaptive Refinement Masking

With this idea and inspiration from the works of [21–23], DARM is designed to produce adaptive mappings of the attention products that not only restrict unreliable information but also refine and accelerate the attention mechanism. Our design of DARM operates with a mechanism called sub-max reduction transform. This mechanism consists of two steps: low-magnitude reduction and multi-head projection. Let D_p indicates the dot-product $(Q_p(K_p)^{\mathrm{T}})$ for each attention head. The sub-max reduction step first generates a binary mask of positions with positive magnitudes less than the peak value along each row (or sub-max positive values) of D_p, which is defined as

$$M_{p,\text{submax}} = \begin{cases} 1, & 0 \leq D_{p,i,j} < D_{p,i,max}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

The $M_{p,\text{submax}}$ is used to generate two modifications of D_p: $D_{p,\text{submax}}$ which retains only the sub-max positive positions retained and $D_{p,\text{non}-\text{submax}}$ which masks out those same positions. Finally, these two are combined to generate a modified $D_p^{'}$ where the sub-max positive positions are reduced. The modified dot-product

울산대학교 UNIVERSITY OF ULSAN

$D_p^{'}$ can be obtained by

$$\begin{aligned} D_p^{'} &= \delta(D_p) \\ &= D_{p,\text{non-submax}} + (D_{p,\text{submax}} - \tau_p), \\ &= D_p \otimes (1 - M_{p,\text{submax}}) + (D_p \otimes M_{p,\text{submax}} - \tau_p), \end{aligned} \tag{3.2}$$

where $\delta(\cdot)$ represents the sub-max reduction operation, τ_p is the adaptive threshold obtained by calculating the arithmetic mean of the non-zero elements along the column dimension of $D_{p,\text{submax}}$. The threshold τ_p is defined by $\tau_p = \{\tau_{p,i}\}_{i=1}^{2n-k}$ $= \{\frac{1}{2n-k-o} \sum_{j=1}^{2n-k} (D_{p,\text{non-submax}})_{i,j}\}_{i=1}^{2n-k}$ with o represents the number of zero elements. The purpose of this adaptive threshold is to reduce the sub-max positive range so that the output dot-product has a better contrast with better highlighted the peak-value positions. This design aims to align the attention score to be more akin to the outputs shown for the deeper models and longer training epochs. From the analyzed behavior of the ECCT model, it can be expected that the more contrasted the attention score (Q-K dot-product), the better the network estimates the erroneous positions. Given the work of the code-aware attention in [17], which aims to highlight the erroneous positions, our design focuses on expediting this process by reducing, though not completely restricting, the range of values interfering the highlighted positions (typically the positions with peak magnitude).

However, one potential hurdle that can be anticipated from the using only the sub-max reduction step is that the code-aware attention may not be well-trained in early stages, which could lead to the model learning incorrect informa-

울산대학교 UNIVERSITY OF ULSAN

tion and providing undesirable performance. Therefore, to mitigate this problem, the multi-head transformation step is proposed to enable a fully-trainable mechanism for DARM. Each independent attention head $p \in \{1, ..., h\}$ in the sub-max multi-head learning process is defined by

$$\Omega(O_p) = F((O_p \cdot W_p) + g(\mathbf{H})), \tag{3.3}$$

where $\Omega(\cdot)$ represents the parameterized transformation, O_p is the input of $\Omega(\cdot)$ on each head and $F(\cdot)$ is the *Mish* activation introduced in [20]. The learning process is parameterized by $W_p \in \mathbb{R}^{(2n-k)\times(2n-k)}$. In order to reduce the model size and complexity, W_p is initialized only once and shared across all layers in a model with M number of layers.

The entirety of DARM is formed by consecutively applying the sub-max reduction transformation along the column and row dimension of the dot-product D_p, respectively. This formation allows all information in the attention scores to be learned by the sub-max reduction transform mechanism. The attention score is being treated similar to an image of size $(2n-k)\times(2n-k)$ with h number of channels. In this case, our proposed formation is similar to the placement of proposed by the MLP-mixer model for vision tasks in [40]. Thus, the DARM module is defined by

$$DARM_p = \Omega_2(\Omega_1(\delta(D)_p^{\mathrm{T}})^{\mathrm{T}}), \tag{3.4}$$

where Ω_1 and Ω_2 are the respective multi-head transformations for the column

울산대학교
UNIVERSITY OF ULSAN

and row dimensions, respectively. More specifically, the input of the Ω_1 is the transposition of the Q-K dot-product matrix, and the Ω_2 operation is performed on the transposition of the output of Ω_1. By including our dynamic mask, the attention in (2.5) can be rewritten as

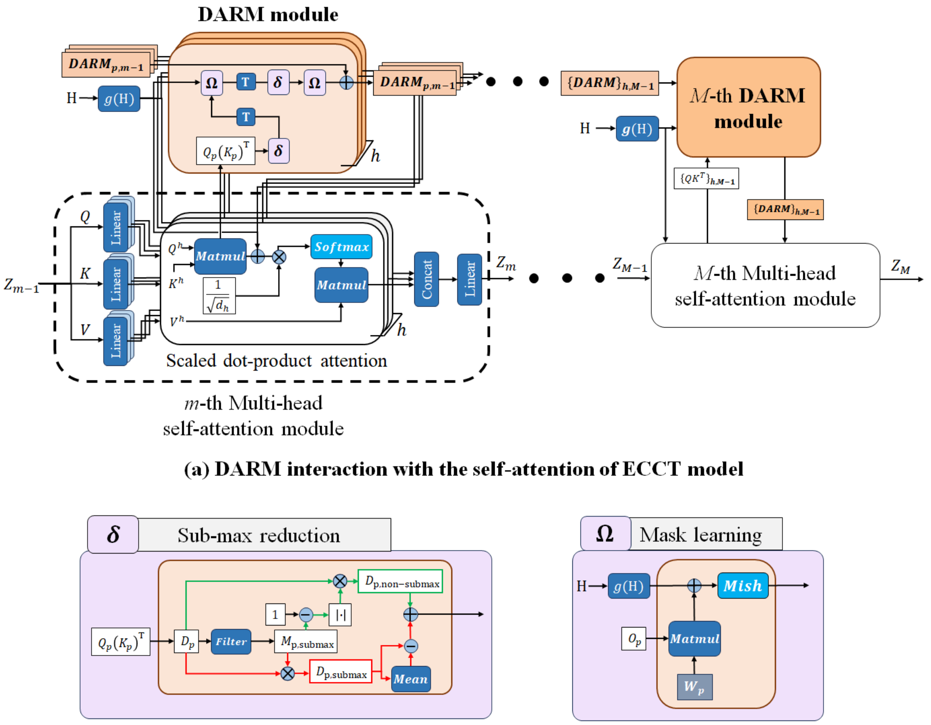
$$A_{\mathrm{DARM}}(Q,K,V) = Softmax(\frac{QK^{\mathrm{T}} + DARM + g(\mathbf{H})}{\sqrt{d_p}})V, \tag{3.5}$$

where *DARM* is the general output of the mask learning mechanism, defined by $DARM = \{DARM_p\}_{p=1}^{h}$.

The choice of *Mish* activation [20] in (3.13) is closely linked to the utilization of *Softmax* and the summation of the query-key dot product and the output of DARM during calculation of the attention weight matrix. By design, *Softmax* penalizes lower and negative input values while rewarding higher positive values. To prevent the attention to unreliable information (low and negative-value positions), the mask-generating function is designed so that the output mask contains negative values at undesirable positions, where the minimal allowed value is typically *zero*. Among many negative permitting activation functions, *Mish* allows for a wider range of negative outputs based on negative inputs, which is suitable for our design. Furthermore, as shown in [20], *Mish* is preferable in backward propagation compared to other ReLU-resemble activation functions because of its better gradients at zero-region.

In (3.16), the mask learning operation only considers the dot product matrices from its respective layer and the binary mask $g(\mathbf{H})$. However, when consider-

울산대학교 UNIVERSITY OF ULSAN



(a) DARM interaction with the self-attention of ECCT model

(b) Sub-max reduction operation

(c) Mask learning operation

Figure 3.2: DARM local relation with the multi-head self-attention module and sequential connection within the transformer architecture.

ing an architecture of M transformer layers, by independently considering DARM for individual attention layer, DARM modules may generate masks with varying patterns, which can be unstable during training. Therefore, at the m-th layer, the mask learning of each head $DARM_{p,m}$ takes the output of the $(m-1)$-th layer $DARM_{p,m-1}$ as one of its inputs. The updated form of (3.13) is expressed as

$$DARM_{p.m} = \Omega_2(\Omega_1(D^{\mathrm{T}}_{p.m})^{\mathrm{T}}) + DARM_{p,m-1}. \tag{3.6}$$

Through this mechanism, DARM layers are sequentially connected alongside the

울산대학교
UNIVERSITY OF ULSAN

existing relation with the stacked transformer layers, which is shown in Fig. 3.2. This structure ensures that every DARM layer in the transformer architecture can be computed with stability during backward propagation.

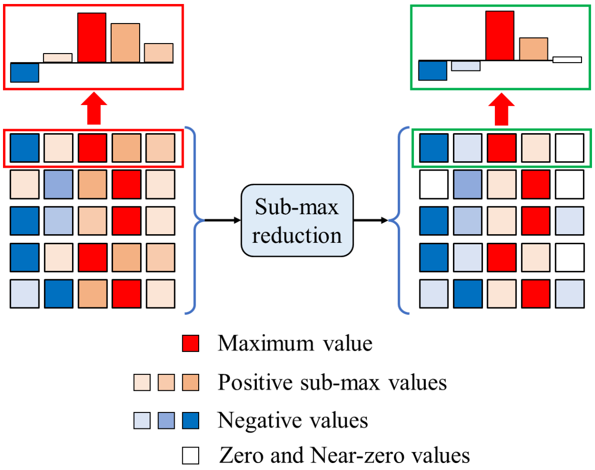


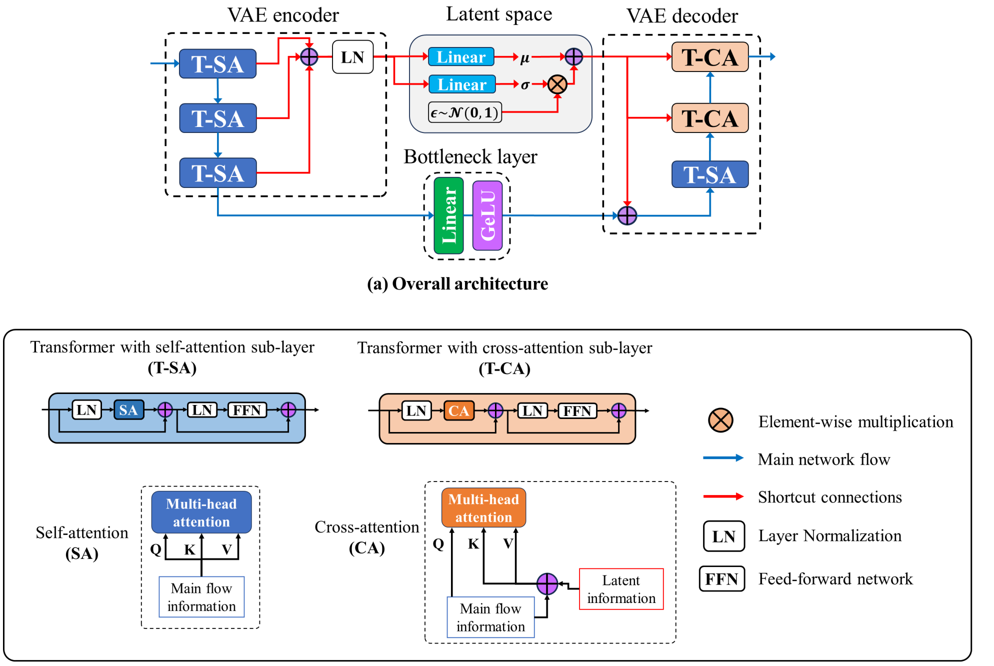
Figure 3.3: An example of sub-max reduction operation.

3.3 Variational U-ECCT

Inpsired by the U-Net from [30] and the variational autoencoder from [47], we propose a variational U-shaped architecture for the ECCT model with specific modifications to enable performance improvement at moderate code lengths. In Fig. 3.4, the architecture of variational U-ECCT is presented along with its sub-layer operations. This architecture is designed as a counterpart to the standard six-layer ECCT model in [17]. The proposed model consists of the main network flow of information and the shortcut connections. The main flow is a

울산대학교
UNIVERSITY OF ULSAN

network of sequentially connected components, including the bottleneck layer and transformer layers. The bottleneck layer is placed at the center of the network, dividing it into two halves.



(a) Overall architecture

(b) Sub-layer architecture

Figure 3.4: Diagrams for variational U-ECCT architecture.

Instead of the simple forward-passes and concatenations, the shortcut connections are combined to resemble a variational autoencoder. The variational Bayes inference method, inspired by the VAE in [47], is adopted to provide approximation for the outputs of transformer layers on the former half of the network. As shown in Fig. 3.4, the outputs of each layer on the first half of the network are combined through summation and normalized before feeding to the ap-

울산대학교
UNIVERSITY OF ULSAN

proximation module. The mechanism of the approximation module adopts the reparameterization trick proposed in [47], where the distribution of the latent space is approximated by a deterministic function, defined by

$$\tilde{\mathbf{s}} = \mu + \sigma \otimes \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1), \tag{3.7}$$

where $\tilde{\mathbf{s}}$ is the latent vector, μ is the learned mean vector of the latent distribution, σ is the learned variance vector of the latent distribution, and ε is the random vector drawn from the standard normal distribution. This formula is called the reparameterization trick, and in this particular case, it is designed to generate random vectors from the location-scale distribution family. The latent vector $\tilde{\mathbf{s}}$ is fed to the second half of the network. In Fig. 3.4, the second half is structured with two types of transformer layers, denoted by T-SA (transformer with self-attention sub-layer) and T-CA (transformer with cross-attention sub-layer). The two types of attention modules were first introduced by the authors of [18] in their transformer architecture and proven effective in learning the relation between different positions in a teach-forcing manner. The T-SA takes only one input, which is either the previous T-SA output (first half side) or the summation of the outputs from the bottleneck layer and the latent space approximation. For the T-CA, as shown in Fig. 3.4, the query vector Q is directly taken from the preceding transformer layer, whereas the key and value vectors, K and V, are obtained by taking in the summation between the main flow information and the latent approximation. By this mechanism, the second half of the U-shaped architecture

울산대학교 UNIVERSITY OF ULSAN

operates as a decoder, extracting high-reliability information from the combined distribution between the main flow and the latent approximation of the superimposed shortcut connections. In particular, the shortcut information obtained from approximating the distribution of encoder outputs enables the extraction of the common learning patterns at each T-SA layer on the encoder side, as opposed to the individual shortcut connections at each scaling level in our basic U-ECCT design. This mechanism mitigates the unwanted noise when the syndrome length of $n-k$ is larger than half the code length, contributing more than one-third of the total input length $(2n-k)$ in lower code rate cases. However, this mechanism is less effective when the syndrome length is small, rendering the performance of the variational U-ECCT closer to the basic U-ECCT design.

In the variational U-ECCT architecture, the dimensionality is uniform across all layers. In the case of the U-Net, the multi-level scaling provides flexibility to adjust the model size as well as effectively distribute the parameters to important parts of the network. On the contrary, the variational U-ECCT architecture becomes highly complex when applying multi-level scaling due to the additional use of rescale layers to provide an equal hidden dimension for the summation before and after the latent space approximation step. Thus, the model size increases significantly compared to the basic U-ECCT, while the performance slightly degrades conpared to the variant with a uniform hidden dimension, losing both the model reduction benefit of the basic U-ECCT and the intended performance su-

울산대학교
UNIVERSITY OF ULSAN

periority of the variational U-ECCT architecture.

As part of the VAE, the objective function of the model is modified to include the KL-divergence term in addition to the cross-entropy loss. This combination can be deduced from the evidence lower bound, which was detailed in [47]. The new loss function is defined as

$$\mathscr{L}_\theta = \mathscr{L}_{\text{BCE}}(\tilde{\mathbf{z}}_b, f_\theta(\tilde{\mathbf{y}})) + D_{\text{KL}}(\mathscr{N}(\mu, \sigma^2 I) \parallel \mathscr{N}(0, \sigma_{\mathbf{y}}^2 I), \tag{3.8}$$

where $\mathscr{L}_{\text{BCE}}$ is the binary cross-entropy (BCE) term which is obtained by (2.6), D_{KL} is the KL-divergence term, and $\sigma_{\mathbf{y}}$ is the standard deviation of the channel output $\mathbf{y}$. In our design, the target standard normal distribution in the KL-divergence is modified to the normal distribution with a zero mean and the standard deviation $\sigma_{\mathbf{y}}$ calculated from the observed channel output $\mathbf{y}$. The KL loss between two normal distribution is defined by

$$\begin{aligned} D_{\text{KL}}(\mathscr{N}(\mu_1, \Sigma_1) \parallel \mathscr{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\big[tr(\Sigma_2^{-1}\Sigma_1) \\ +(\mu_2 - \mu_1)^{\text{T}}\Sigma_2^{-1}(\mu_2 - \mu_1) + r + \log\frac{|\Sigma_2|}{|\Sigma_1|}, \end{aligned} \tag{3.9}$$

where r is the size of the random vector, which, in this case, is the hidden dimension d_{model} of the latent vector. Leveraging the formula derived by the authors of [49] (Appendix B, [49]) and the considered distributions in our case, the KL-

울산대학교
UNIVERSITY OF ULSAN

divergence loss term can be calculated as

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{N}(\mu, \sigma^2 I) \parallel \mathcal{N}(0, \sigma_{\mathbf{y}}^2 I)) = \frac{1}{2\sigma_{\mathbf{y}}^2} \sum_{j=1}^{r} \mu_j^2 \\ - \frac{r}{2}(1 - \frac{\sigma_{\mathbf{y}}^2}{\sigma^2} + \log \frac{\sigma_{\mathbf{y}}^2}{\sigma^2}). \end{aligned} \tag{3.10}$$

In this formulation, the latent distribution is designed to have a similar power factor to the channel output, further enhancing the estimation of the multiplicative noise. Furthermore, during training, the derived form in (3.10) is tractable during back-propagation.

The mechanism of the variational U-ECCT provides significant resistance to the unwanted noise provided by the extensively large syndrome length in low code rate cases. Furthermore, it is able to take advantage of the additional syndrome information to enhance the multiplicative noise estimation process. Therefore, the variational U-ECCT exhibits considerably better performance in low code rate cases compared to the basic U-ECCT design. However, as a trade-off, the multi-level scaling of the basic design is not applicable for the variational model, losing much of the scaling flexibility of the U-ECCT model.

The parameter utilization is determined by the total number of parameters in the entire model. This refers to the sum of the number of weights and biases from different types of layers trained for the given task. In the sub-layer architecture of both our proposed design and the baseline ECCT in [17], three main sub-layer types are used, namely layer normalization (LN), multi-head attention (MHA), and feed-forward network (FFN). For ECCT [17], the number of

울산대학교
UNIVERSITY OF ULSAN

parameters used by each type of sub-layer can be calculated as a function of the embedding dimension d_{model}. The building block of all sub-layers is referred to as the dense layer or fully connected layer, with d_{in} and d_{out} denoting its input and output dimensions, respectively. Its number of parameters, P_{dense}, is calculated by

$$P_{\text{dense}}(d_{\text{in}}, d_{\text{out}}) = d_{\text{in}} \times d_{\text{out}} + d_{\text{out}}. \tag{3.11}$$

The number of parameters for layer normalization, P_{LN}, is defined as

$$P_{\text{LN}} = 2d_{\text{model}}. \tag{3.12}$$

From (3.12) and (3.13), the number of parameters for MHA and FFN sub-layers can be defined as

$$\begin{aligned} P_{\text{MHA}} &= 4P_{\text{dense}}(d_{\text{model}}, d_{\text{model}}) \\ &= 4 \times (d_{\text{model}}^2 + d_{\text{model}}) \\ &= 4d_{\text{model}}^2 + 4d_{\text{model}}, \end{aligned} \tag{3.13}$$

$$\begin{aligned} P_{\text{FFN}} &= P_{\text{dense}}(d_{\text{model}}, d_{\text{FFN}}) + P_{\text{dense}}(d_{\text{FFN}}, d_{\text{model}}) \\ &= (d_{\text{model}} \times d_{\text{FFN}} + d_{\text{FFN}}) + (d_{\text{FFN}} \times d_{\text{model}} + d_{\text{model}}) \\ &= 4d_{\text{model}}^2 + 4d_{\text{model}} + 4d_{\text{model}}^2 + d_{\text{model}} + 2d_{\text{model}} \\ &= 8d_{\text{model}}^2 + 5d_{\text{model}}. \end{aligned} \tag{3.14}$$

울산대학교 UNIVERSITY OF ULSAN

where d_{FFN} is the dimensionality of the FFN module inner-layer, which was equal to four times the embedding dimension d_{model} in [17] and [18]. The FFN component, was originally proposed in [18] along with the MHA module, is composed of a neural layer with GELU activation [56] followed by a linear activated layer. In this pair of layers, the output dimension of the first layer and the input dimension of the second layer are equal to d_{FFN}.

By accumulating the sub-components parameters, the total number of parameters for a transformer layer is calculated by

$$\begin{aligned} P_{\text{Transformer}} &= P_{\text{MHA}} + P_{\text{FFN}} + 2P_{\text{LN}} \\ &= 4d_{\text{model}}^2 + 4d_{\text{model}} + 8d_{\text{model}}^2 + 5d_{\text{model}} + 4d_{\text{model}} \\ &= 12d_{\text{model}}^2 + 13d_{\text{model}}. \end{aligned} \tag{3.15}$$

Let M denote the number of transformer layers in the model. The size of an M-layers ECCT model is determined by multiplying $P_{\text{Transformer}}$ number of parameters of each transformer layer by M number of layers used in the model architecture, which results in the following form as

$$P_{\text{ECCT}} = M \times P_{\text{Transformer}} = 12Md_{\text{model}}^2 + 13Md_{\text{model}}. \tag{3.16}$$

In this context, the variational U-ECCT architecture is slightly expanded in the number of parameters with the addition of the bottleneck layer and the pair of

울산대학교 UNIVERSITY OF ULSAN

variational approximating layers. With the additional components, the parameter quantification for the variational U-ECCT model is obtained as

$$\begin{aligned} P_{\text{VariationalU}-\text{ECCT}} &= M \times P_{\text{Transformer}} + P_{\mu+P_{\sigma+P_{\text{Bottleneck}}}} \\ &= M \times P_{\text{Transformer}} + 3P_{\text{dense}}(d_{\text{model}}, d_{\text{model}}) \\ &= (12M+3)d_{\text{model}}^2 + (13M+3)d_{\text{model}}. \end{aligned} \tag{3.17}$$

3.4 Mirror-sharing for model compression

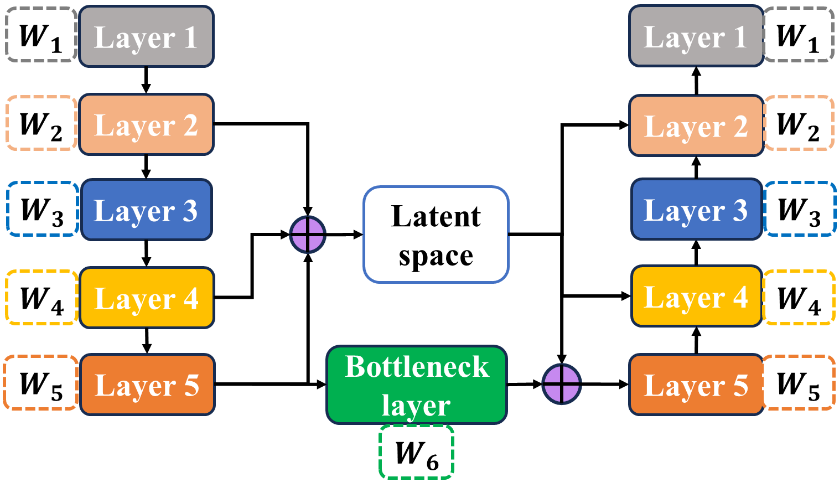


Figure 3.5: The proposed mirror-sharing strategy for the variational U-ECCT.

In this sub-section, a new model compression strategy, named mirror-sharing, is introduced. This strategy is based on the weight-sharing method, which increases learning efficiency. While there were numerous effective weight-sharing strategies proposed in previous works, we came to the conclusion that, in order to

울산대학교
UNIVERSITY OF ULSAN

effectively apply weight-sharing to the setting of U-ECCT, a new weight-sharing scheme is required.

In our weight-sharing strategy, each transformer layer in the first half of the U-shaped architecture is assigned an independent set of parameters. For the second half of the architecture, we reuse the parameter sets from the first half. In this structure, the proposed weight-sharing method is called mirror-sharing since one side of the network mimics the parameters of the other side in an U-shaped architecture that has reflective characteristics regarding dimensionality. During training, the first half and the second half of the U-shaped model share the same set of parameters, or, in other words, this set of parameters is trained twice during the forward and backward propagation. This means that when considering an M-layers model, only $\frac{M}{2}$ sets of parameters are used, which results in a new model with only half the number of parameters. Nevertheless, depending on the computational power required for specific cases, the sharing rate needs to be adjusted. Furthermore, the weight-sharing technique cannot be applied to rescale layers since they are not reflective in dimensionality. Each rescale layer has a unique dimensionality, which is highly complex when applying the weight-sharing technique.

Fig. 3.5 shows the mirror-sharing strategy for the variational U-ECCT architecture with $M = 10$. A fully mirror-sharing strategy is shown in Fig. 3.5, where both sides of the U-shaped model share the same five sets of parameters for transformer layer, denoted by W_1, W_2, W_3, W_4, and W_5. The parameters set W_6 of the

울산대학교
UNIVERSITY OF ULSAN

Table 3.1: Model size comparison between ECCT, proposed models with and without mirror-sharing.

Setting indicator	**Number of layers**	**Quantification of parameters**	
ECCT	$M = 6$	$72d^2_{\text{model}} + 78d_{\text{model}}$	
Variational U-ECCT	$M = 6$ (without mirror-sharing) $M = 10$ (with mirror-sharing)	**Without mirror-sharing**	**With mirror-sharing**
		$P_{\text{VU-ECCT}} = 75d^2_{\text{model}} + 81d_{\text{model}}$ $P_{\text{VU-ECCT}} \approx P_{\text{ECCT}}$ $\Delta P = 3d^2_{\text{model}} + 3d_{\text{model}}$	$P^{'}_{\text{VU-ECCT}} = 60d^2_{\text{model}} + 65d_{\text{model}}$ $P^{'}_{\text{VU-ECCT}} < P_{\text{ECCT}} < P_{\text{VU-ECCT}}$ $\Delta P^{'} = 12d^2_{\text{model}} + 13d_{\text{model}}$

bottleneck layer in Fig. 3.5 is not subject to sharing since this layer is used only once in the entire model. Five sets of parameters are specifically chosen for sharing to reduce model size and obtain optimal performance.

Table 3.1 presents the model size comparison for the baseline ECCT in [17] and variational U-ECCT models with and without mirror-sharing. Without mirror-sharing, the model size of the proposed architecture is approximately equal to the baseline ECCT. On the other hand, the mirror-sharing model enables a considerable model size reduction.

울산대학교
UNIVERSITY OF ULSAN

CHAPTER 4

EXPERIMENTATION

4.1 Setup and training

The experimentation incorporates the baseline of ECCT [17], Hyper-Graph-Net decoder [6], BP [43]. To demonstrate the improvement in our design, training and inference stages are executed on both our proposed models and the baseline model from [17] with identical hyperparameter settings. Similar to [17], our design is evaluated with three classes of codes, namely LDPC codes, Polar codes, and BCH codes. Specifically for Polar codes, simulation results of Successive Cancellation List decoding (SCL) [56] are included for comparison.

For LDPC (576,432) and LDPC (1248,936), which are respectively IEEE 802.11e and IEEE 802.16e standards, the results of the state-of-the-art neural normalized min-sum decoder (NNMS) reported in [7] are included for comparison.

울산대학교 UNIVERSITY OF ULSAN

Table 4.1: Hyperparameters setting for all methods.

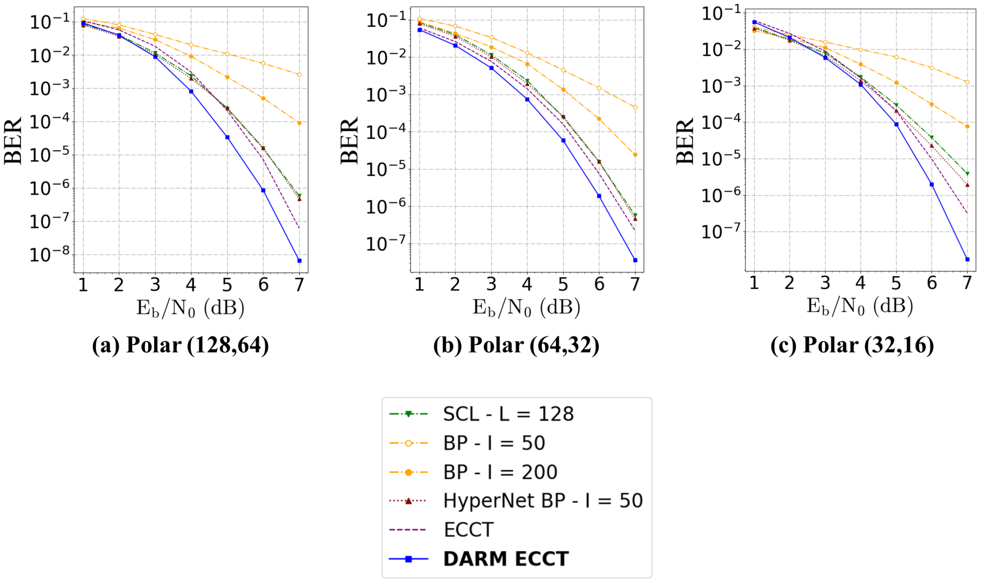
Method	**Hyperparameter**	**Code structures**
SCL [56]	List size $L = 128$	Polar
SNNMS-LR-Q [7]	Iterations $I = 10$ Number of layers $M = 22$	LDPC
BP [55]	Iterations: $I = 50, 200$	LDPC, BCH, Polar
Hyper-Graph-Net BP [6]	Iterations: $I = 50$ Network g hidden size $d_g = 16$neurons Hypernetwork f hidden size $d_f = 128$ neurons	
ECCT [17] and DARM-ECCT	Embedding size $d_{\text{model}} = 128$ FFN hidden size $d_{\text{ff}} = 4 \times d_{\text{model}}$ Number of attention heads $h = 8$ Number of layers $M = 6$ layers	
Variational U-ECCT	Embedding size $d_{\text{model}} = 128$ FFN hidden size $d_{\text{ff}} = 4 \times d_{\text{model}}$ Number of attention heads $h = 8$ Number of layers $M = 10$ layers	

The model from [7] used for comparison is the parameter-sharing NNMS with Leaky ReLU and 12-bit quantizer (SNNMS-LR-Q). The parity-check matrices for all codes are taken from [52].

Both the baseline ECCT [17] and the proposed models train and test results are evaluated for architectures with 6 layers, 8 attention heads, and embedding dimension $d_{\text{model}} = 128$. A summary of the hyperparameters settings for all methods is presented in Tab. 4.1. The notation shown in Tab. 4.1 are defined as follows: L: the list size of the SCL decoding. I: the number of iterations for BP-based methods. M: the number of layers in neural network models. d_g: the number of neurons for the decoder network g, and d_f: the number of neurons for the hyper-network f in Hyper-Graph Neural BP [6]. The results of Hyper-Graph-Net decoder [17] and SNNMS-LR-Q [7] are obtained from their respective hyperameters as well as training and testing setup. All simulation tasks are performed on

울산대학교
UNIVERSITY OF ULSAN

a workstation equipped with an AMD Ryzen Threadripper Pro 5995WX processor, 512GB of RAM, and two NVIDIA RTX 4090-24GB GPUs. All simulations are programmed using the PyTorch framework [53].



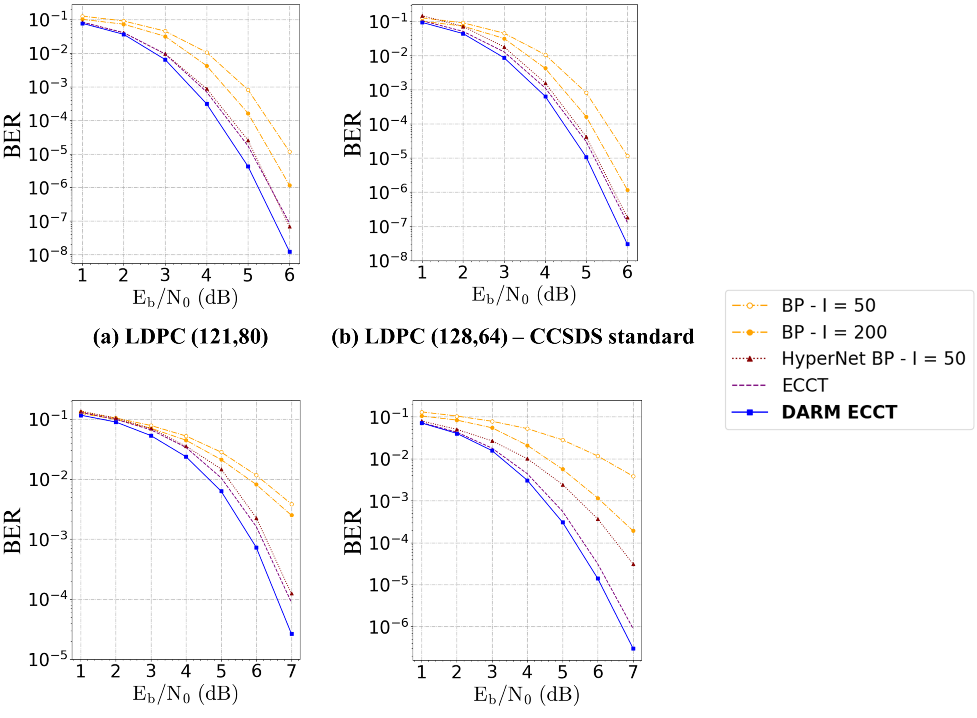
(a) Polar (128,64) **(b) Polar (64,32)** **(c) Polar (32,16)**

Figure 4.1: BER comparison for baseline models and our proposed models - Polar codes.

For training, data samples are generated for ECCT [17] and our models with 128 samples per batch for 1000 batches for cases with input length satisfy $2n-k \leq 200$ (all code cases with code length within range of $n \leq 128$ bits), which makes up 128000 samples. However, due to constraint on GPU memory, data are generated with 64 samples per batch for cases of input length within the range of $300 \leq 2n-k < 500$ (Polar (256,128), CCSDS (256,128)), 16 samples per batch for cases satisfy $500 \leq 2n-k < 1000$ (Polar (512,384), LDPC (576,432)), and 6 samples per batch for cases of input length $2n-k \geq 1000$ (Polar (1024,768),

울산대학교 UNIVERSITY OF ULSAN

LDPC (1248,936)). All training data are all-zero codewords as it is shown to be sufficient for training in [16,17] without overfitting. Similar to [17], all neural network models are trained on noise-corrupted codewords with the energy per bit to noise power spectral density ratio E_b/N_0 generated within the range of 2dB and 7dB. The Adam optimizer [54] is used with initial learning rate of 10^{-4}. Training is conducted over 1000 epochs using the cosine annealing weight decay method [57] without warm restart until learning rate reaches 5×10^{-7}.



(a) LDPC (121,80)

(b) LDPC (128,64) – CCSDS standard

(c) BCH (127,64)

(d) BCH (63,51)

Figure 4.2: BER comparison for baseline models and our proposed models - LDPC and BCH codes.

For inference on the trained models of the baseline ECCT and our de-

울산대학교
UNIVERSITY OF ULSAN

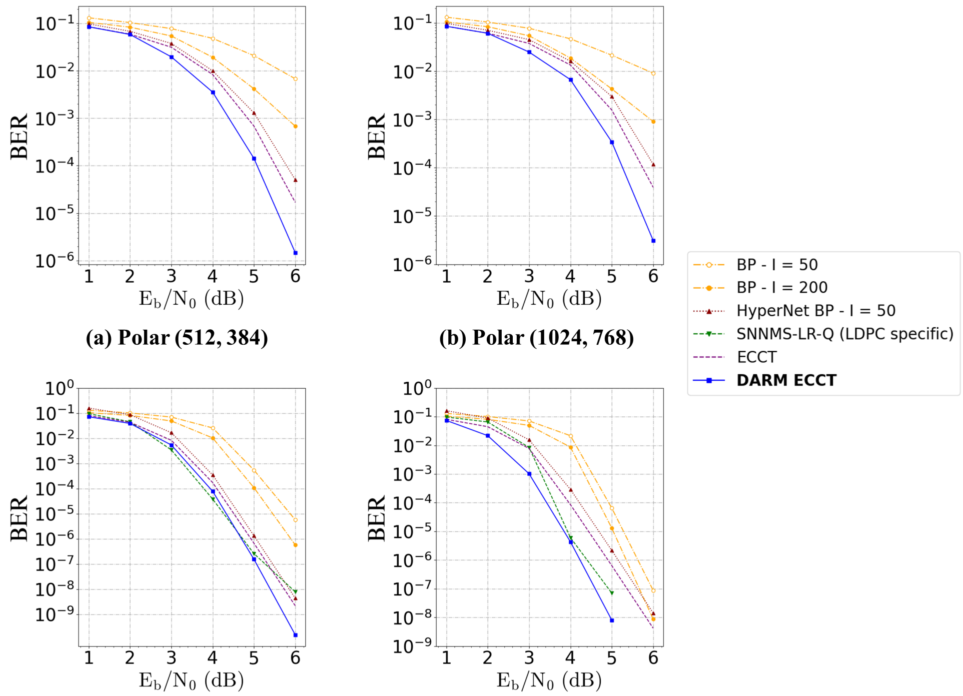
signs, the generator matrix $\mathbf{G}$ is obtained from the respective parity-check matrix and used to generate the codewords through Galois field matrix product, defined by $\mathbf{x} = \mathbf{s}\mathbf{G}$. With this operation, 128 samples are generated for each test batch for 1000 batches for E_b/N_0 below 7dB. During simulation, it is observed that the logarithmic BER is mostly below 10^{-6} for $E_b/N_0 \geq$ 7dB. As a result, for $E_b/N_0 \geq$ 7dB, test data is instead generated with total number of samples exceeding 10^7, ensuring a minimum of 100 error frames to obtain accurate BER results. Similarly, for BP with 50 iterations, Monte-Carlo simulations are conducted with a total of 10^7 samples to ensure accurate results across all E_b/N_0 regions. All methods are tested over the E_b/N_0 range between 1dB and 7dB.

4.2 Decoding performance of DARM-ECCT

Figs. 4.1 and 4.2 show the logarithmic BER over the test range of E_b/N_0 for codes with different code lengths and coding rates. As can be observed from the logarithmic BER results in Figs. 4.1 and 4.2, the performance superiority of CRPE, DARM, and CRPE-DARM ECCTs, are prominent for all code types and different coding rate. Especially, CRPE-DARM ECCT is the model with best performance in operational E_b/N_0 region (4dB - 6dB), able to achieve performance gains of 0.4dB to 0.6dB for Polar variants, 0.2dB to 0.5dB for LDPC variants, and 0.3dB for BCH variants. When applying DARM and CRPE separately, each method also shows noticeable improvements over ECCT, with DARM resulting

울산대학교 UNIVERSITY OF ULSAN

in better performance than CRPE. Evidently, this indicates the significance of DARM mechanism in guiding the self-attention operation to concentrate on the reliable information.



(a) Polar (512, 384)

(b) Polar (1024, 768)

(c) LDPC (576, 432) – IEEE 802.11e

(d) LDPC (1248, 936) – IEEE 802.16e

Figure 4.3: BER results for large code lengths.

Fig. 4.3 shows the logarithmic BER result for larger code-lengths. Across all results shown in Fig. 4.3, it is further proven that the proposed model achieves better performance than the baseline models for different code settings. Fig. 4.3 showcases the performance for codes with code length satisfy $n > 500$. When comparing with the SNNMS-LR-Q [**?**], it can be observed that the ECCT fall shorts for cases of $E_b/N_0 \geq 4dB$ while our models are capable of surpassing

울산대학교
UNIVERSITY OF ULSAN

the performance of SNNMS-LR-Q. Considering the overall results, the ECCT-based models are superior to all BP-based methods in terms of decoding performance. As the basic ECCT is less appealing for larger code lengths, the improvements provided by the proposed models are further emphasized. The performance proven the potential of well-designed model-free design can achieve significant gains over model-based decoders.

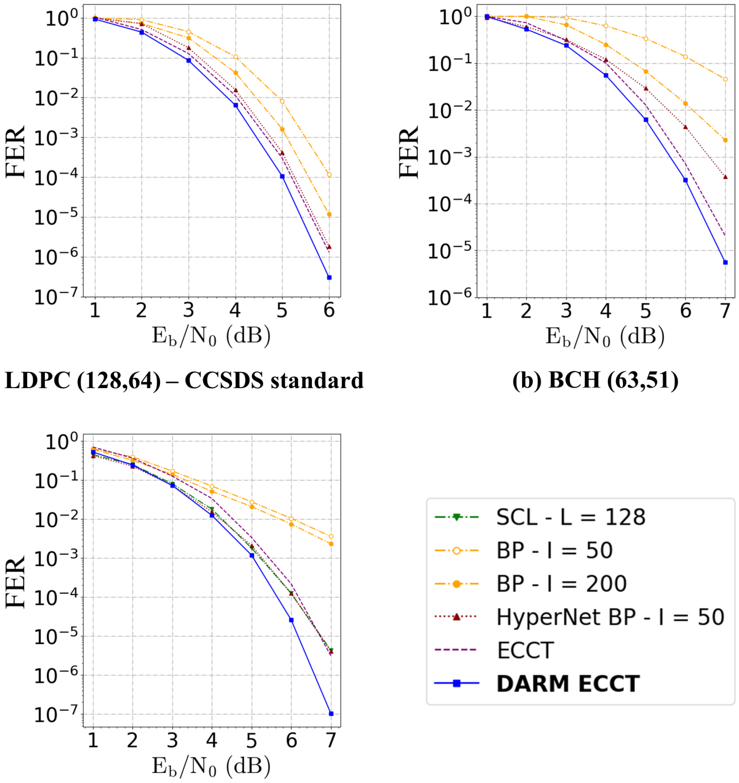
Additionally, frame error rate (FER) results are shown in Fig. 4.4. The FER results is utilized to confirm the decoder capability of recovering accurate entire frame (codeword) as appose to individual bits. From the FER results in Fig. 4.4, the proposed models also achieve superior whole codeword recovery performance compared to the other considered techniques.

4.3 Training convergence

In Fig. 4.12, the convergence of logarithmic BER over 1000 epochs is provided for Polar(64,32), BCH(63,51) and LDPC(121,80). Based on the BER validation curve over 1000 epochs, it is further confirmed that the combination of CRPE and DARM, named CRPE-DARM ECCT, provides the best result. From the results of both Figs. 4.1 and 4.5, CRPE-DARM ECCT shows improvements in both inference performance and convergence speed compared to either CRPE or DARM when they are considered separately.

The results in Fig. 4.5 also confirm that both DARM and CRPE provide

울산대학교
UNIVERSITY OF ULSAN



(a) LDPC (128,64) – CCSDS standard

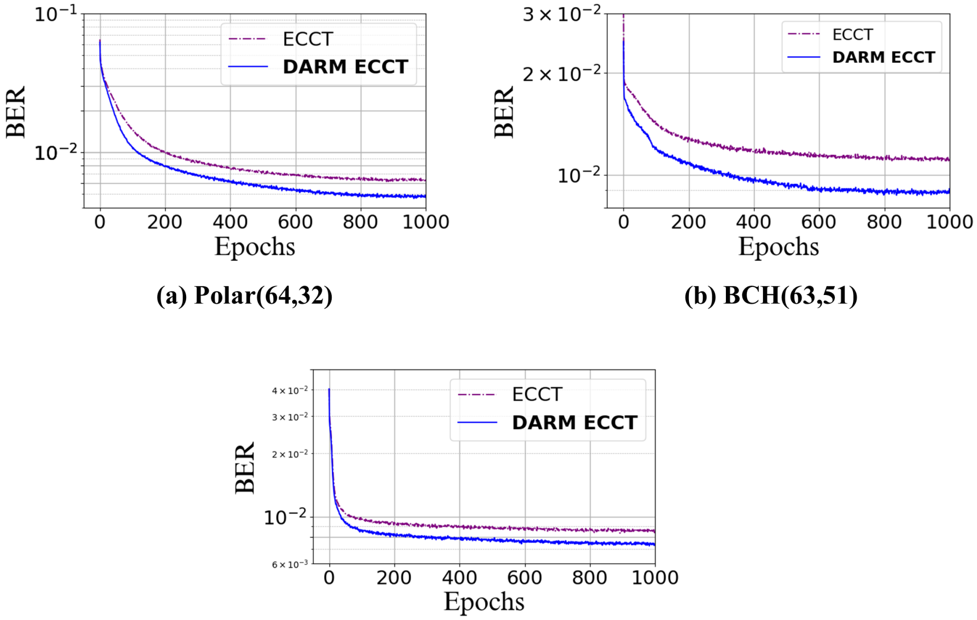
(b) BCH (63,51)

(c) Polar (64,32)

Figure 4.4: FER result across different code settings.

considerable improvements in convergence speed compared to the baseline ECCT when they are applied separately. Furthermore, DARM shows its significance in the BER convergence during training stage and BER performance during inference stage in comparison with CRPE results.

울산대학교
UNIVERSITY OF ULSAN



(a) Polar(64,32)

(b) BCH(63,51)

(c) LDPC(121,80)

Figure 4.5: BER convergence over 1000 epochs for Polar(64,32), BCH(63,51) and LDPC(121,80).

4.4 Ablation analysis for DARM

In this section, the proposed designs are further analyzed to better explain their learning pattern and potential. To verify the assumed learning patterns of self-attention in error detection, the softmax attention score is inspected in combination with analysis on the attribution score obtained through IG method. Furthermore, the sub-components in DARM are tested separately, proving the essentially of interactions between them.

울산대학교
UNIVERSITY OF ULSAN

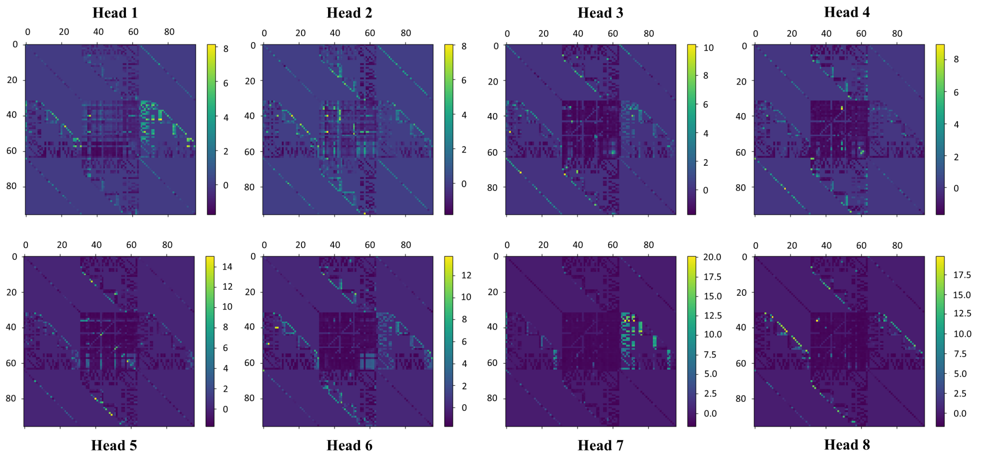
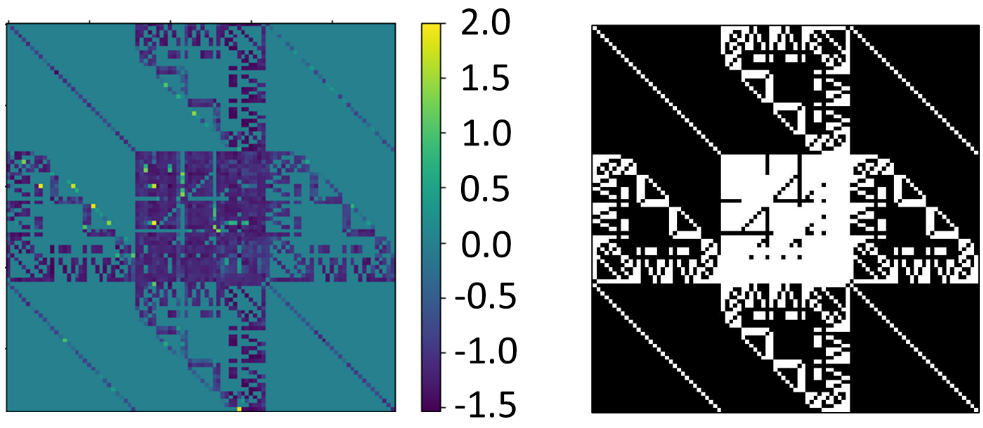


Figure 4.6: Dynamic masks generated by DARM for each attention head.



(a) Attention weight matrix of DARM ECCT

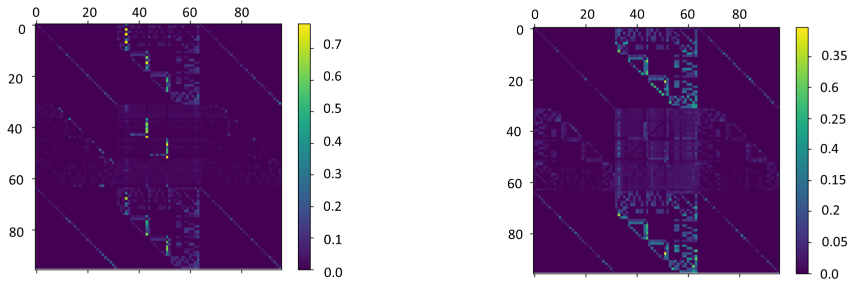
(b) Attention weight matrix of ECCT

Figure 4.7: Comparisons for masking between DARM and ECCT.

4.4.1 Softmax attention score analysis

In this subsection, the effect of the proposed DARM mechanism is analyzed for our tested Polar(64,32). In Fig. 4.6, the learned mask matrices are provided

울산대학교
UNIVERSITY OF ULSAN



(a) Attention Weight Matrix of DARM-ECCT **(b) Attention Weight Matrix of baseline ECCT**

Figure 4.8: Attention weight matrices during inference stage at 4dB for Polar (64,32) - baseline ECCT [17] vs DARM-ECCT.

for all 8 attention heads. Each head learns a slightly different distribution since each head provides independent version of the query-key dot-product. However, all output masks follow a similar pattern where the magnitudes of the masks strongly concentrate at only specific positions that have some spatial relation to one another. As intended, DARM not only adapts to the attention dot-product matrix but also learns to amplify specifically high magnitude positions. Fig. 4.7 shows a comparison between a dynamic mask generated by DARM ECCT and a binary mask of ECCT. The dynamic mask in Fig. 4.7a not only contains the positional information (the edges in the PCM) similar to the binary mask of ECCT (Fig. 4.7b) but also contains the magnitudes of each position through adaptively learning the information of the Q-K dot-product.

Fig. 4.8 demonstrates examples of attention weight matrices for DARM and baseline ECCTs during inference. From the attention weights of the trained DARM-ECCT (Fig. 4.8a) and baseline ECCT (Fig. 4.8b), it is certain that DARM pro-

울산대학교
UNIVERSITY OF ULSAN

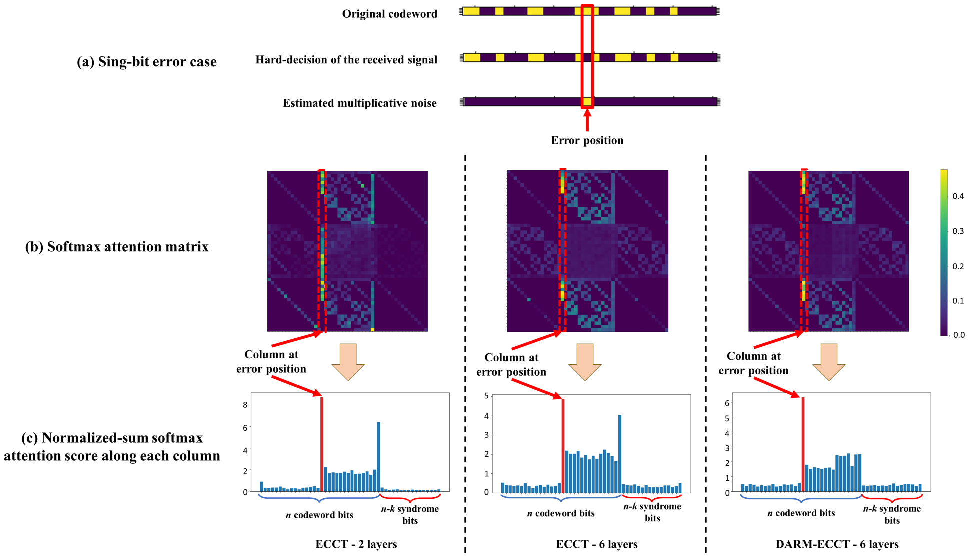


Figure 4.9: Attention score for single error bit at 5dB for Polar (32,16) - baseline ECCT [17] vs DARM-ECCT.

vides a better concentrated attention mapping than the masking of the baseline ECCT. Although the attention weights of the baseline ECCT shows a highlight of fewer positions, the spatial distribution in the attention matrix of ECCT is sparse and shows unrelated patterns, where medium to high magnitude positions are often located apart from each other.

In order to understand the quality of decoding, the ECCT-based models are tested on pre-defined error cases. In Fig. 4.9, the softmax attention scores of attention-based models tested on Polar (32,16) with single-bit error at $E_b/N_0 = 5dB$ are presented. The softmax score at each input position is obtained by summation of all values along the column dimension of the heads-average attention matrix of the last MHSA layer of each respective model. This result confirms

울산대학교
UNIVERSITY OF ULSAN

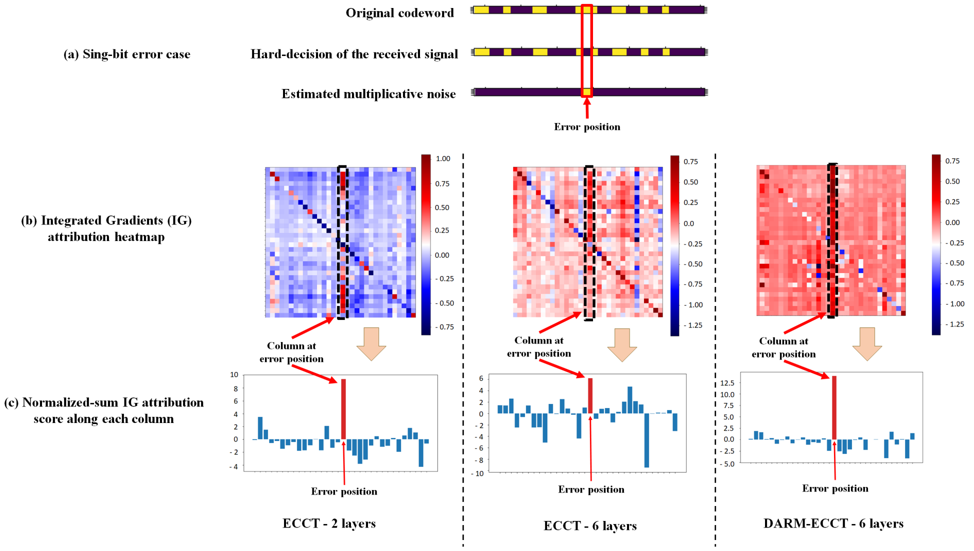


Figure 4.10: IG attribution score for single error bit at 5dB for Polar (32,16) - baseline ECCT [17] vs DARM-ECCT.

that code-aware self-attention highlights the reliable information that are beneficial to obtain the target which is the binary multiplicative noise. In this particular case, it assign very high attention score to the same position as the error bit in the received signal. The DARM-equipped models showcase a significantly better contrast between the softmax score at the error positions and the other positions. This result verifies the sub-max reduction functionality of DARM design and its reason for performance superiority. Nevertheless, inspection of the softmax attention score alone can be less appealing when it comes to interpreting the dynamic of the entire architecture.

울산대학교 UNIVERSITY OF ULSAN

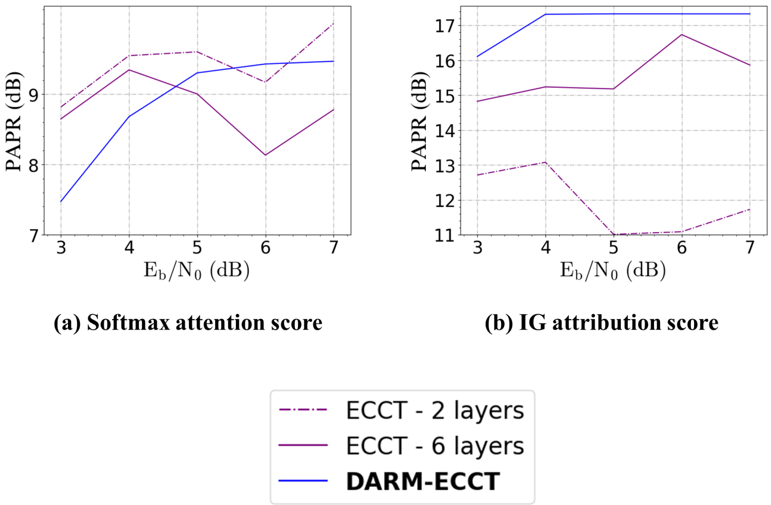
4.4.2 Integrated Gradient attribution score analysis

As there are certain general interpretability limitation through softmax attention matrices visualization, IG is adopted to analyze the general behavior of attention-based decoder. The IG attribution score is obtained for the similar single error bit case as with the softmax score analysis. Fig, 4.10a shows the IG attribution scores heat-map for each position against every other positions in the target data. Fig. 4.10b presents the mean-normalized IG scores where value at each position is the summation along the respective column of the IG heat-map in Fig. 4.10a. Since the IG performs the calculation over the entire gradient space of the model for each inspected positions of the output, the attribution score leads to a better representation of the model overall learning pattern. From the IG attribution scores obtained for attention-based models, it can be observed that the position with the highest attribution score is the same as the error position. This further provides confirmation for the pattern shown when inspecting the softmax attention scores. Furthermore, the contrast between the error position (peak-value) and other positions in the attribution scores for DARM-equipped models is much higher, confirming the model performance improvement.

4.4.3 Quantitative assessment

In order to confirm that DARM provide a consistent high contrast attention score and IG heat-map, peak-to-average-power-ratio (PAPR) is measured for

울산대학교
UNIVERSITY OF ULSAN



(a) Softmax attention score

(b) IG attribution score

Figure 4.11: PAPR for the column normalized summation for softmax attention scores and IG attribution scores.

the normalized summation score of bot the attention matrices and IG attribution score heat-map. The ECCT and DARM-ECCT models trained on Polar (32,16) are tested on random single-bit error cases for 1000 samples over the E_b/N_0 range between 3dB and 7dB. For each sample, number of cases where the peak-value position of the column summation of attention matrix and IG heat-map are monitored. For each matched cases, PAPR is obtained from the column summation score of attention matrix and IG heat-map. In Fig. 4.11, the described PAPR results of 2-layer ECCT, 6-layer ECCT and 6-layer DARM are shown, where Fig. 4.11a and Fig. 4.11b are the PAPR of the normalized column summation attention scores and the IG attribution scores, respectively. It can be observed that the DARM-ECCT model provide a consistent PAPR curve for both type of model information. Furthermore, for IG attribution score, the PAPR values is the high-

울산대학교
UNIVERSITY OF ULSAN

est among all models between 5dB and 7dB, remaining almost constant over this range. In contrast, the ECCT yield highly fluctuated PAPR characteristic, even at operating E_b/N_0 range.

4.5 Significance of mirror-sharing

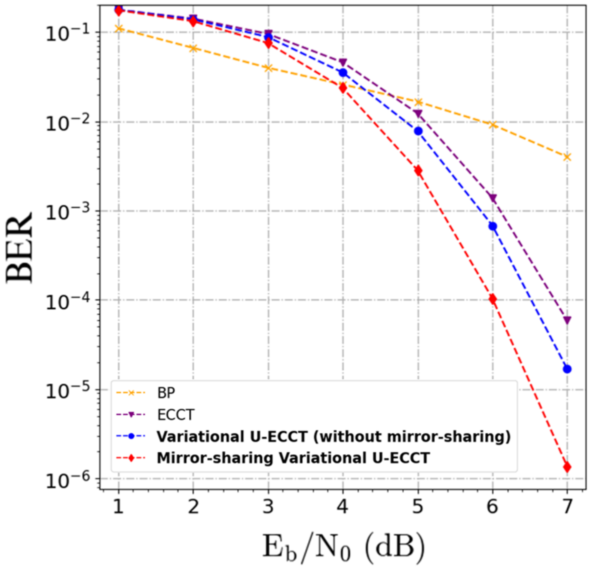


Figure 4.12: BER performance of variational U-ECCT with mirror-sharing and without mirror sharing for Polar (512,171).

The proposed mirror-sharing strategy described in Section III provides a solution to configure deeper U-ECCT networks without increasing the number of parameters. In Fig. 4.12, the BER performance is shown for the baseline methods, the variational U-ECCT with and without mirror-sharing. The effectiveness of mirror-sharing is confirmed by the result of variational U-ECCT models, where

울산대학교 UNIVERSITY OF ULSAN

Table 4.2: Calculated model size of ECCT and variational U-ECCT with chosen $d_{\text{model}} = 128$.

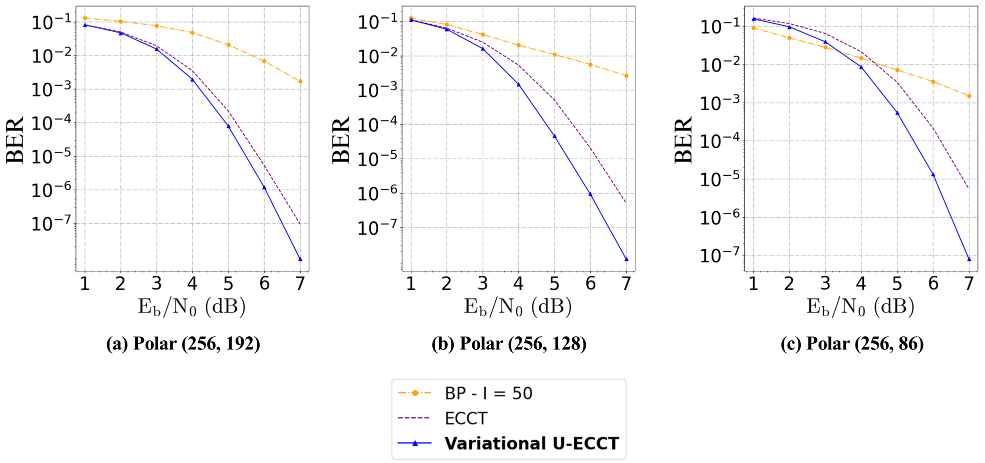
Setting indicator	**Without mirror-sharing**	**With mirror-sharing**
ECCT	1189632	N/A
Variational U-ECCT	1222656 (↑ 2.7%)	991360 (↓ **11**%)

the mirror-sharing U-ECCT significantly outperforms both its non-sharing variant and the baseline ECCT.

Nevertheless, the variational U-ECCT without mirror-sharing demonstrates its improvement over baseline ECCT with a noticeable gain in performance. Note that in this case, the BP continues to demonstrate its advantages in the low E_b/N_0 region up to the 4dB, while all neural models begin narrowing the gap and eventually outperform the traditional method. In this regard, our models, particularly the mirror-sharing variants, exhibit a significantly greater improvement over BP and the baseline ECCT.

Table 4.2 presents the calculated size of all tested neural network models based on the chosen embedding dimension of $d_{\text{model}} = 128$. ECCT models and proposed models without mirror sharing utilizes $M = 6$ layers. Mirror-sharing models utilizes $M = 10$ layers. The size difference columns consists of model size difference in percentage between ECCT and the models in the preceding column to the right. The model size reduction is enabled on all proposed models with mirror-sharing. Coupled with the advantage in performance, the variational U-ECCT variants are much more efficient in utilizing the parameters for mul-

울산대학교
UNIVERSITY OF ULSAN



(a) Polar (256, 192) **(b) Polar (256, 128)** **(c) Polar (256, 86)**

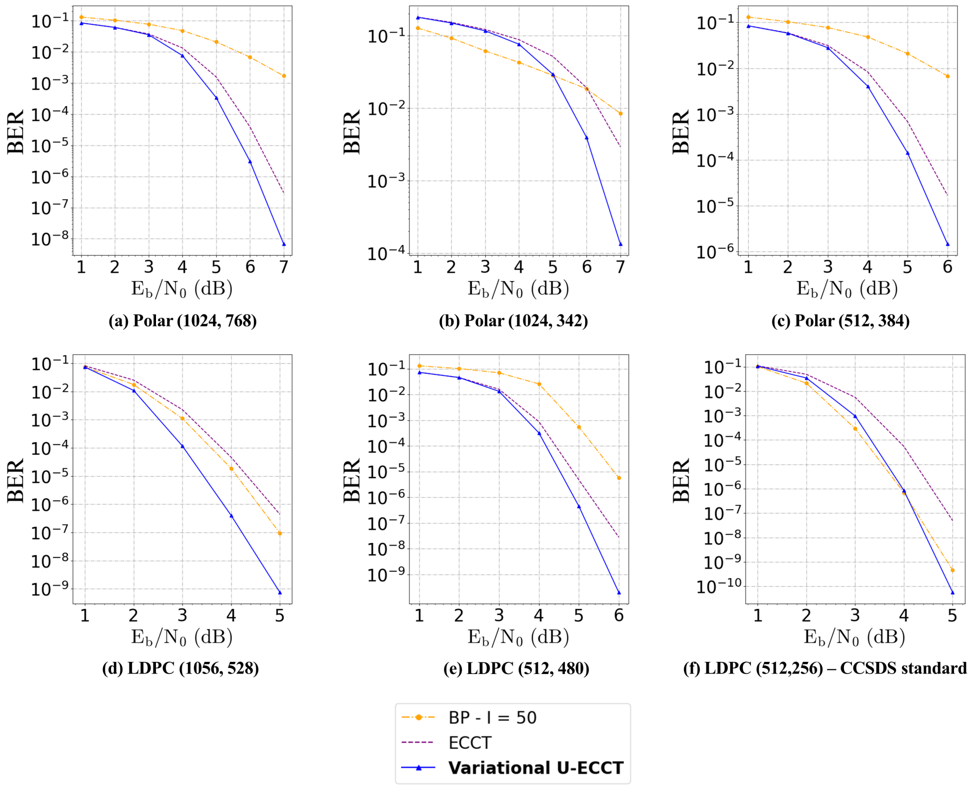
Figure 4.13: BER performance results for the baseline methods and U-ECCT models on polar codes with a code-length of 256 at different code rates.

tiplicative noise estimation. The results further prove the efficiency as well as superiority of the U-ECCT design compared to the baseline ECCT.

4.6 BER performance of U-ECCT variants

By testing the proposed designs on a variety of coding cases, it has been verified that the proposed variational U-ECCT provides considerable improvements compared to baseline ECCT. In Fig. 4.13, the BER performance results tested on different code rates for Polar codes with a code length of 256 are plotted for BP, ECCT, and our designs. It can be observed that variational U-ECCT outperform other methods, specifically within the range of E_b/N_0 between 4dB and 7dB. Furthermore, the variational U-ECCT model shows its advantage for low code rates, providing considerable performance gains compared to the baseline ECCT [17].

울산대학교 UNIVERSITY OF ULSAN



(a) Polar (1024, 768)

(b) Polar (1024, 342)

(c) Polar (512, 384)

(d) LDPC (1056, 528)

(e) LDPC (512, 480)

(f) LDPC (512,256) – CCSDS standard

Figure 4.14: BER performance results for the ECCT and U-ECCT variants on large code-length.

In Fig. 4.14, BER results are shown for ECCT and variational U-ECCT on larger code-length. As anticipated from the U-shaped design, the proposed model exhibits better performance at larger code-length cases. Comparing the performance on polar codes at the high code rate $R = \frac{3}{4}$, at a code-length of 1024 bits, the average performance gains of the variational U-ECCT to the baseline ECCT are 0.6dB in E_b/N_0 at $BER = 10^{-5}$ in Fig. 4.14(a), which improved by 0.2dB compared to the 0.4dB average performance gains at the smaller code-length of 256 bits in Fig. 4.13(a). In the three cases of LDPC codes in Figs.

울산대학교 UNIVERSITY OF ULSAN

4.14(d), 4.14(e), 4.14(f), the improvements provided by our models are obvious, with average performance gains of 0.6dB for LDPC (512, 256) and 0.4dB for LDPC (576, 480). Even at lower E_b/N_0 values of 3dB, where the noise level is higher, the variational U-ECCT model still provides a 0.4dB gain for LDPC (512, 256). For low rate codes of Polar (256, 86) in Fig. 4.13(c) and Polar (1024, 342) in Fig. 4.14(b), variational U-ECCT model demonstrates considerably superior performance to the basic U-ECCT, showcasing the combined advantages of the U-shaped model and the VAE-like structure.

Table 4.3 presents the performance results of baseline ECCT and our proposed mirror-sharing U-ECCT model for a variety of code structures at different code-length and coding rates. This table showcases results for BCH codes, LDPC codes, and polar codes. The results are presented in negative natural logarithm of BER for E_b/N_0 cases from 4dB to 6dB (larger value is better). As shown in Table 4.3, improvements can be observed over all code-length cases with larger gains for larger codes (with code-length from 256 to 1024). Evidently, based on the obtained results, the mirror-sharing variational U-ECCT outperforms the baseline ECCT in all cases, some less than others.

울산대학교 UNIVERSITY OF ULSAN

Table 4.3: Additional performance results of ECCT and our proposed variational U-ECCT with mirror-sharing.

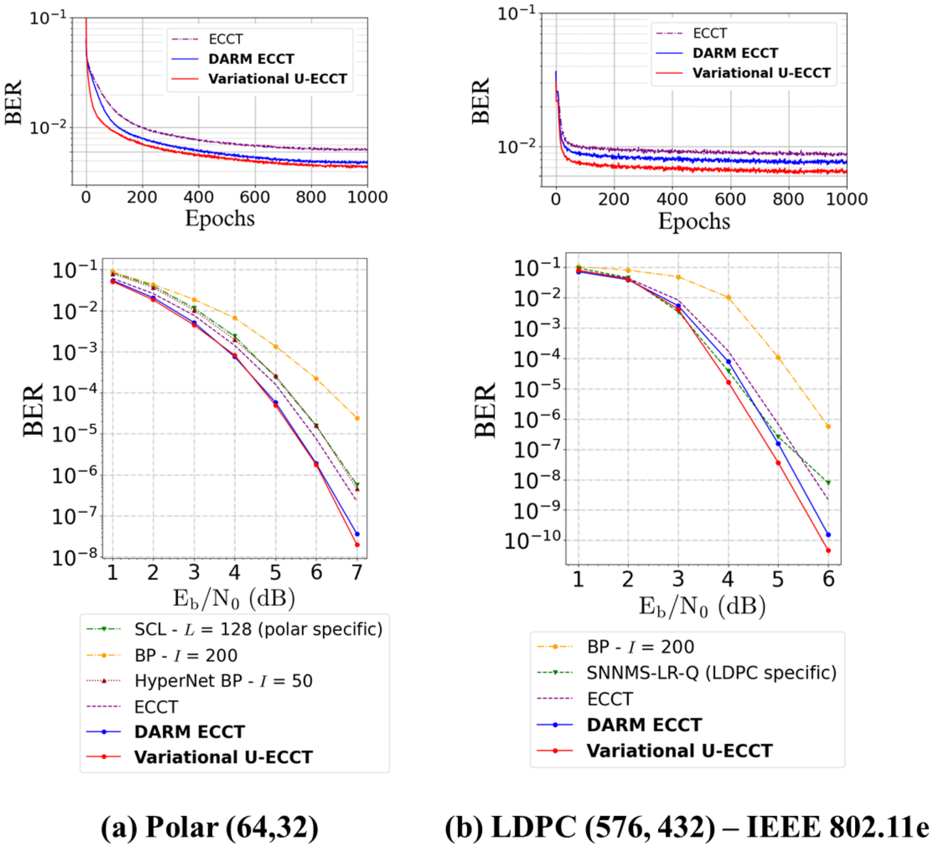
Code types	**ECCT**			**Variational U-ECCT**		
	4	5	6	4	5	6
Polar (512, 384)	4.79	7.29	10.99	**5.32**	**8.79**	**13.09**
Polar (128, 43)	5.75	7.86	10.56	**6.44**	**8.81**	**11.73**
Polar (64, 32)	6.99	9.36	12.55	**7.10**	**9.91**	**13.24**
LDPC (576, 480)	7.04	12.24	18.28	**7.89**	**14.18**	**21.93**
LDPC (121, 80)	7.00	10.83	15.38	**7.82**	**12.58**	**18.53**
BCH (255, 163)	4.29	6.11	9.26	**4.35**	**6.34**	**9.69**
BCH (127, 64)	3.39	4.56	6.43	**3.47**	**4.71**	**6.80**
BCH (63, 51)	5.52	7.72	10.67	**5.73**	**8.02**	**11.34**

4.7 Comparisons between DARM and mirror-sharing variational U-ECCT

In Fig. 4.15, comparisons between the DARM ECCT and mirror-sharing variational U-ECCT are presented. It can be observed that the DARM ECCT performs on similar level to mirror-sharing U-ECCT in both training and testing phases for Polar (64,32). However, when it comes to larger code lengths, DARM ECCT is inferior to the U-shaped model. Since the mirror-sharing variational U-ECCT enables more layers than DARM ECCT, it is anticipated that this model operates better when the input space is much larger.

Considering computational complexity, the mirror-sharing variational U-ECCT scale better with the size of the input due to having the parameter-sharing feature. On the other hand, the DARM ECCT faces an issue when the length of the input

울산대학교
UNIVERSITY OF ULSAN



(a) Polar (64,32)

(b) LDPC (576, 432) – IEEE 802.11e

Figure 4.15: Performance and training curve comparisons between DARM ECCT and mirror-sharing variational U-ECCT.

vector $(2n-k)$ is large. Since the mask learning mechanism utilizes the weight matrix of size $(2n-k \times 2n-k)$, the scale of the model is increased as the code lengths grew larger. Since our current focus for DARM is the performance enhancement of the self-attention in error estimation problem, the complexity cost at large code length was not well handled. Hence, the efficiency of DARM ECCT is not as beneficial as the mirror-sharing variational U-ECCT model.

울산대학교 UNIVERSITY OF ULSAN

CHAPTER 5

CONCLUSION

With the rise of model-free transformer-based design for ECC decoding, DARM and variational U-shape architecture are proposed in order to improve reliability syndrome-based decoding for ECCTs. The evaluation of DARM-ECCT is conducted based on various aspects, including comparison between the proposed methods and baseline methods using BER over operating E_b/N_0 range for different code types, analysis of the effect of DARM for attention weight matrix and learned mask matrix, evolution of DARM outputs over the sequential architecture. In decoding comparison, DARM- ECCT achieves best performance, resulting in improvements between 0.2dB and 0.6dB compared to baseline ECCT. Under examination of the attention weight matrix and the learned mask matrix of DARM, it is shown that DARM dynamically adapts to the spatial distribution of the query-key dot-product, guiding the self-attention mechanism to better

울산대학교
UNIVERSITY OF ULSAN

concentrate on reliable information.

The U-ECCT design forms a unique interaction between the shortcut connections and the main network flow, supporting an enhanced extraction of reliable information beyond just relying on the self-attention mechanism. This enables the model to achieve both performance enhancement and model size reduction. With the proposed mirror-sharing strategy, parameter utilization efficiency has significantly increased. This method also enables a deeper yet lighter model, thus achieving better performance and size reduction compared to the baseline ECCT. With the modification in the form of variational U-ECCT, the decoding performance at low code rates is significantly enhanced, providing up to 0.7dB in average performance gains on the scale of E_b/N_0.

However, we also observe few potential directions in which the design can be improve upon. In terms of CRPE, though providing improvement in performance, using the parity-check matrix for embedding requires the initial projection from the syndrome length to the chosen model embedding size, which cost additional parameters compared to the embedding in baseline ECCT [17]. When looking at DARM, since its mechanism relies on the existing query-key product, if the query-key product fails to capture the reliable information, DARM module can not detect this failure and will provide improper information back to the self-attention mechanism. However, such behavior is less likely to occur since the code-aware self-attention mechanism is already well-performed. While

울산대학교
UNIVERSITY OF ULSAN

effective for moderate code cases, the variational U-ECCT is yet to achieve desirable efficiency for code lengths larger than 1024 bits. Such large dimension cases requires thorough study and additional model scaling methods such as domain specific pruning and model quantization to achieve a desirable level of complexity for practical decoding. Finally, similar to the baseline ECCT [17], our proposed models requires learning each code structure, coding rate, and code length separately, in which the required training time can be considerably large under practical circumstances. In this case, a design allowing heterogeneous learning of different code structures within a certain range of code length can be a more effective and practical model.

울산대학교
UNIVERSITY OF ULSAN

LIST OF PUBLICATIONS

Article Submitted

[1] N. D. Trac and K. Sunghwan, “U-shaped error correction codes transformers,” Submitted to *IEEE Trans. Cogn. Commun. Netw.*. (Under Review)

[1] N. D. Trac and K. Sunghwan, “Dynamically adaptive refinement masking and code-domain reliability-enhanced positional encoding for error correction code transformer,” Submitted to *IEEE Trans. Neural Netw. Learn. Syst.* (Under Review)

International Conference

[1] N. D. Trac and K. Sunghwan, “DRF-ECCT: Dynamic reliability filter for error correction code transformer,” in *Procs Int. Conf. Green Human Inf. Tech. (ICGHIT)*, 2023, pp. 18-23.

울산대학교
UNIVERSITY OF ULSAN

BIBLIOGRAPHY

[1] Y. Jiang *et al.*, "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 2758-2768.

[2] Y. Jiang *et al.*., "Feedback turbo autoencoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 8559-8563.

[3] J. Clausius *et al.*, "Serial vs. parallel turbo-autoencoders and accelerated training for learned channel codes," in *Proc. 11th Inter. Symp. Topics Coding (ISTC)*, 2021, pp. 1-5.

[4] A. Buchberger, C. Häger, H. D. Pfister, L. Schmalen and A. Graell i Amat, "Pruning and quantizing neural belief propagation decoders," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1957-1966, July 2021.

[5] G. Larue, L. -A. Dufrene, Q. Lampin, H. Ghauch, and G.R.B. Othman, "Neural belief propagation auto-encoder for linear block code design," *IEEE Trans.*

울산대학교
UNIVERSITY OF ULSAN

Commun., vol. 70, no. 11, pp. 7250-7264, Nov. 2022.

[6] E. Nachmani and L. Wolf, ”Hyper-graph-network decoders for block codes, “ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[7] Q. Wang *et al.*, “Normalized min-sum neural network for LDPC decoding,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 1, pp. 70-81, Feb. 2023.

[8] D. Liu, M. Bober, and J. Kittler, “Neural belief propagation for scene graph generation,” *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 45, no. 8, pp. 10161-10172, Aug. 2023.

[9] E. Nachmani and L. Wolf, “Autoregressive belief propagation for decoding block codes,” 2021, *arXiv preprint arXiv:2103.11780*.

[10] I. Be’Ery, N. Raviv, T. Raviv, and Y. Be’Ery, “Active deep decoding of linear codes,” *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 728-736, Feb. 2020.

[11] F. Liang, C. Shen, and F. Wu, “An iterative BP-CNN architecture for channel decoding,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 144-159, Feb. 2018.

[12] T. J. O’Shea and J. Hoydis, “An introduction to machine learning communications systems”, 2017, *arXiv preprint arXiv:1702.00832*.

울산대학교
UNIVERSITY OF ULSAN

[13] F. A. Aoudia and J. Hoydis, “Model-free training of end-to-end communication systems,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503-2516, Nov. 2019.

[14] D. García *et al.*, “Model-free machine learning of wireless SISO/MIMO communications,” *Comp. Commun.*, vol. 181, pp. 192-202, Jan 2022.

[15] T. Gruber, S. Cammerer, J. Hoydis, and S. t. Brink, “On deep learning-based channel decoding,” in *Proc. 51st Annual Conf. Inf. Sciences Syst. (CISS)*, 2017, pp. 1-6.

[16] A. Bennatan, Y. Choukroun, and P. Kisilev, “Deep learning for decoding of linear codes - a syndrome-based approach,” in *Proc. 2018 IEEE Inter. Symp. Info. Theory (ISIT)*, Vail, CO, USA, 2018, pp. 1595-1599.

[17] Y. Choukroun and L. Wolf, “Error correction code transformer,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp.38695-38705.

[18] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol.30, 2017.

[19] E. Tjoa and C. Guan, ”A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 4793-4813, Nov. 2021.

울산대학교
UNIVERSITY OF ULSAN

[20] D.Misra, “Mish: A self regularized non-monotonic activation function,” 2019, *arXiv preprint arXiv:1908.08681*.

[21] Z. Fan *et al.*, “Mask attention networks: Rethinking and strengthen transformer,” 2021, *arXiv preprint arXiv:2103.13597*.

[22] A. Athar *et al.*, “Differentiable soft-masked attention,” 2022, *arXiv preprint arXiv:2206.00182*.

[23] S. Sukhbaatar *et al.*, “Adaptive attention span in transformers,” 2019, *arXiv preprint arXiv:1905.07799*

[24] T. Richardson and R. Urbanke, “The capacity of low-density parity check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. of the IEEE Conf. Comput. Vision Pattern Recognit.*, pp. 770-778, 2016.

[26] F. He, T. Liu, and D. Tao, “Why resnet works? Residuals generalize,” *IEEE Trans. Neural Netw. Learn. Syst.*, no. 12, vol. 31, pp. 5349-5362, 2020.

[27] W. Tong, W.Chen, W. Han, X. Li, and L. Wang, “Channel-attention-based DenseNet network for remote sensing image scene classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*”, vol. 13, pp. 4121-4132, 2020.

울산대학교
UNIVERSITY OF ULSAN

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, pp. 4700-4708, 2017.

[29] S. Qian, C. Ning, and Y. Hu, “MobileNetV3 for image classification.” in *Proc. IEEE 2nd Int. Conf. Big Data, Artif Intell. Internet of Things Eng. (ICBAIE).*, 2021.

[30] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proc. 18th Int. Conf Med. Image Comput. Comput.-Assist. Interv.*, Oct. 2015, pp. 234-241.

[31] J. Devlin *et al.*, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv preprint arXiv:1810.04805*.

[32] A. Galassi, M. Lippi and P. Torroni, ”Attention in Natural Language Processing,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291-4308, Oct. 2021.

[33] X. Liu, H. Lu, J. Yuan, and X. Li, “CAT: Causal audio transformer for audio classification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2023, pp. 1-5.

[34] B. Tang and D. S. Matteson, “Probabilistic transformer for time series analysis,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 23592-23608.

울산대학교
UNIVERSITY OF ULSAN

[35] Y. Liu *et al.*, ”A Survey of Visual Transformers,” *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1-21, Mar. 2023.

[36] H. Zhao, J. T. Zhou and Y. -S. Ong, ”Word2Pix: Word to Pixel Cross-Attention Transformer in Visual Grounding,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 1523-1533, Feb. 2024

[37] Y. Wang, Z. Gao, D. Zheng, S. Chen, D. Gunduz, and H. V. Poor, “Transformer-empowered 6G intelligent networks: From massive MIMO processing to semantic communication,” *IEEE Wireless Commun.*, pp. 1-9, Nov. 2022.

[38] 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 J. Sel. Areas Inf. Theory*, vol. 3, no. 3, pp. 587-602, Sept. 2022.

[39] Y. Shao, E. Ozfatura, A. G. Perotti, B. M. Popović, and D. Gündüz, “AttentionCode: Ultra-reliable feedback codes for short-packet communications,” *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4437-4452, Aug. 2023.

[40] I. O. Tolstikhin *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24261-24272.

[41] M. Sundararajan, A. Taly, and Q. Yan, Axiomatic attribution for deep networks,” in *Proc. Int. Conf. Mach. Learn.*, July 2017, pp. 3319-3328.

울산대학교
UNIVERSITY OF ULSAN

[42] C. -M. Fan, T. -J. Liu, and K. -H. Liu, "SUNet: Swin transformer UNet for image denoising," in *Proc. 2022 IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, 2022, pp. 2333-2337.

[43] D. H. Thai, X. Fei, M. T. Le, A. Züfle, and K. Wessels, "Riesz-Quincunx-UNet variational autoencoder for unsupervised satellite image denoising," *IEEE Trans. Geosci. Remote Sens.*, Art no. 5404519, vol. 61, pp. 1-19, 2023.

[44] A. Lin *et al.*, "DS-TransUNet: Dual swin transformer U-Net for medical image segmentation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1-15, 2022.

[45] M. Perslev *et al.*, "U-time: A fully convolutional network for time series segmentation applied to sleep staging." in *Proc. Adv. Neural Inf. Process. Syst.* vol. 32, 2019, pp. 4415-4426.

[46] D. Stoller, M. Tian, S. Ewert, and S. Dixon, "Seq-U-Net: A one-dimensional causal U-Net for efficient sequence modelling." *arXiv preprint arXiv:1911.06393*, 2019 .

[47] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[48] D. Liu and G. Liu, "A transformer-based variational autoencoder for sentence generation," in *Procs 2019 Int. Joint Conf. on Neural Netw. (IJCNN)*, 2019, pp. 1-7.

울산대학교
UNIVERSITY OF ULSAN

[49] V. Raj and S. Kalyani, “Design of communication systems using deep learning: A variational inference perspective,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1320-1334, Dec. 2020.

[50] M. A. Alawad, M. Q. Hamdan and K. A. Hamdi, “Innovative variational autoencoder for an end-to-end communication system,” *IEEE Access*, vol. 11, pp. 86834-86847, 2023.

[51] Y. M. Saidutta, A. Abdi and F. Fekri, “Joint source-channel coding over additive noise analog channels using mixture of variational autoencoders,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2000-2013, July 2021.

[52] M. Helmling *et al.*, “Database of channel codes and ML simulation results”, www.uni-kl.de/channel-codes, 2019.

[53] A. Paszke *et al.*, ”Pytorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8024–8035.

[54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv preprint arXiv:1412.6980*.

[55] J. Pearl, Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan kaufmann, 1988.

울산대학교
UNIVERSITY OF ULSAN

[56] I. Tal and A. Vardy, “List Decoding of Polar Codes,” *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213-2226, May 2015.

[57] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” 2016, *arXiv preprint arXiv:1608.03983*.

울산대학교
UNIVERSITY OF ULSAN

국문요약

본 연구에서는 ECCT(Error Correction Code Transformer)의 성능을 향상시키기 위해 DARM(Dynamic Adaptive Refinement Masking)과 Mirror-Sharing Variational U자형 아키텍처를 제안합니다. 고정된 바이너리 마스킹 대신 DARM 모듈은 주의 가중치 분포를 참조로 사용하는 동적 크기를 사용하여 주의 지도의 대비를 더욱 강화하여 자기 주의 메커니즘을 강화하는 각 주의 머리에 대한 고유한 마스킹을 생성하도록 설계되었습니다. 또한, 순차 신경 구조(Sequential Neural Architecture)를 활용하여 DARM 모듈이 순차적으로 연결되도록 설계하여 반복적인 정제 효과를 만들어냅니다. 적당한 코딩 길이의 경우 변환기 기반 디코더의 전반적인 효율성을 향상시키기 위해 미러 공유 변형 U자형 아키텍처가 도입되었습니다. 변형 자동 인코더와 같은 건너뛰기 연결을 갖춘 U자형 아키텍처는 중간 길이의 코드, 특히 낮은 코딩 속도에서 잘 작동하는 분할과 같은 동작을 제공합니다. U자형 모델은 바람직한 성능을 달성하기 위해 일정 수준의 깊이가 필요하므로 미러 공유라는 아키텍처 수준 매개변수 공유 방식이 도입되어 U자형 모델을 효과적으로 확장하여 더 나은 효율성과 성능을 달성할 수 있습니다. 실험 결과는 기본 ECCT에 비해 비트 오류율이 크게 향상되는 동시에 훈련 수렴 속도도 크게 향상되는 것을 보여줍니다.

울산대학교 UNIVERSITY OF ULSAN