Neural-Enhanced Quantized Min-Sum Successive Cancellation Decoding for Polar Codes

Wen Yang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 2375204942@qq.com

Guiping Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: liguiping@xatu.edu.cn

***Abstract*—Polar codes have been adopted as the channel coding scheme for the 5G NR control channel, yet achieving hardware-efficient decoding without sacrificing error-correction performance remains a challenge. The Min-Sum(MS) approximation simplifies the successive cancellation (SC) f-function at a cost of systematic magnitude overestimation, conventional correction schemes—Normalized Min-Sum(NMS) and Offset Min-Sum(OMS)—rely on fixed parameters optimized under full-precision conditions, rendering them structurally unable to compensate for the nonlinear distortions introduced by ultra-low bit-width quantization. This paper proposes the Neural-network-enhanced Quantized Min-Sum SC (NQ-MS-SC) decoder, in which a lightweight, bit-width-adaptive neural network predicts input-dependent normalization and offset parameters directly from quantized LLRs, jointly trained with a quantization-aware strategy using the Straight Through Estimator to simultaneously address both Min-Sum approximation error and quantization induced distortion. Simulation results demonstrate that NQ-MS-SC outperforms conventional quantized Min-Sum decoders, with gains that intensify at higher SNR, coarser quantization, and longer code lengths.**

Keywords-Successive; Cancellation; Decoding; Fixed-Point Quantization; Neural Network

I. INTRODUCTION

Polar codes, introduced by Arikan in 2009 [1], are the first class of channel codes with a provably capacity-achieving construction and have been adopted as the channel coding scheme for the control channel in the 5G New Radio standard [2]. Despite strong theoretical foundations, achieving hardware-efficient decoding while maintaining competitive error-correction performance remains a central challenge in the practical deployment of polar codes.

At the algorithmic level, SC decoding [1] recursively computes LLRs through two elementary operations: the f-function and the g-function. The exact f-function involves hyperbolic tangent computations with substantial hardware cost, which the Min-Sum (MS) approximation addresses by replacing these transcendental operations with simple sign and comparison operations. To compensate for the systematic magnitude overestimation inherent to MS, the Normalized Min-Sum (NMS) [3] and Offset Min-Sum (OMS) [4] algorithms apply a fixed scaling factor and a fixed offset, respectively, both of which effectively narrow the performance gap with exact SC decoding under floating-point precision. However, these correction parameters are determined offline and applied uniformly across all nodes, without adapting to the distinct LLR distributions at individual computation stages—a fundamental inflexibility that limits their effectiveness under low-precision quantization.

Fixed-point quantization is another critical technique for hardware-efficient decoder design, as reducing the bit-width of LLR representations directly lowers memory requirements and arithmetic unit complexity. Prior work has established that 6-bit quantization is sufficient for SC decoders to achieve near-floating-point performance, and the 4–6-bit range represents the practical sweet spot that balances decoding quality against implementation cost [5,6]. Nevertheless, at ultra-low bit-widths of 3–4 bits, the nonlinear

distortion introduced by quantization compounds with the inherent approximation error of the Min-Sum algorithm, resulting in severe performance degradation. More critically, the fixed correction parameters of NMS and OMS are optimized under floating-point or high-precision conditions and are therefore unable to adapt to the nonlinear distortion that characterizes low-precision quantization. This mismatch renders conventional Min-Sum enhancement schemes largely ineffective in the ultra-low bit-width regime.

Recent advances in deep learning have opened new avenues for optimizing Min-Sum decoding parameters. Lugosch and Gross [7] proposed the Neural Offset Min-Sum (NOMS) scheme, which assigns independently learned offset parameters to individual message-passing nodes, yielding clear gains over fixed-offset methods. Xu et al. [8] further introduced the 2D-OMS algorithm with per-layer and per-node parameter optimization, effectively closing the performance gap with exact BP decoding on length-1024 polar codes. Lyu et al. [9] extended this line of work to jointly optimize both the scaling factor and the offset in the SOMS algorithm, outperforming all existing BP-based decoders with FPGA validation. However, a common limitation across these approaches is that training and inference are conducted entirely under full-precision conditions, without incorporating quantization effects into the optimization objective. When subsequently quantized to 3–4 bits, the correction parameters learned under full precision become mismatched to the quantization-induced nonlinear distortions, and the learned compensation can no longer function as intended.

To address this gap, this paper proposes the Neural-network-enhanced Quantized Min-Sum SC (NQ-MS-SC) decoder, which employs a lightweight neural network to adaptively predict the normalization factor and offset of the f-function from quantized input LLRs, trained jointly with a quantization-aware training (QAT) strategy to simultaneously compensate for both Min-Sum approximation error and quantization-induced distortion. An adaptive LLR scaling mechanism is further introduced to optimize dynamic range utilization under low bit-width conditions. Simulation results demonstrate that NQ-MS-SC achieves significant performance gains over conventional quantized Min-Sum decoders at 3–4 bits, while introducing no additional penalty at 5–6 bits.

II. Basic Theory

A. Polar Codes and SC Decoding

Polar codes are a class of linear block codes constructed by exploiting the channel polarization phenomenon. Given N ($N = 2^n$) independent copies of a binary-input memoryless channel $W : x \to y$, a recursive transformation synthesizes N polarized synthetic channels. As N grows large, the symmetric capacities of these synthetic channels polarize toward two extremes: a fraction $I(W)$ of channels approach capacity one, while the remaining fraction1 $1-I(W)$ approach capacity zero. The symmetric capacity, which characterizes the maximum reliable transmission rate of the channel, is defined as:

$$I(W) \triangleq \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y \mid x) \log \frac{W(y \mid x)}{\frac{1}{2} W(y \mid 0) + \frac{1}{2} W(y \mid 1)} \quad (1)$$

Where X and Y denote the input and output alphabets, respectively, an W(x|y) is the channel transition probability. The polarization process proceeds through two stages: channel combining, which recursively constructs the vector channel W_n and channel splitting, which decomposes W_n into N coordinate channels $W_n^{(i)}$ with transition probabilities:

$$W_n^{(i)}(y_1^N, u_1^{i-1} \mid u_i) \triangleq \sum_{u=i+1 \in x}^{N} \frac{N-1}{2^{N-1}} W_N(y_1^N \mid u_1^N) \quad (2)$$

For a polar code of length $N = 2^n$ and K information bits, the encoding operation is expressed as:

$$x = uG_n \quad (3)$$

Where $u=(u_0,u_1,\cdots u_{N-1})$ is the input bit vector, $x=(x_0,x_1,\cdots x_{N-1})$ is the encoded codeword, and $G_N=B_NF^{\otimes n}$ is the generator matrix.$B_N$denotes the bit-reversal permutation matrix, $F=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$ is the polarization kernel, and $F^{\otimes n}$ denotes its n-th Kronecker power. The K most reliable bit positions in u carry information bits, while the remaining N-K positions are fixed to zero and referred to as frozen bits. Channel reliability in this work is estimated via the Gaussian Approximation (GA) method.

SC decoding sequentially estimates $u_0,u_1,\cdots u_{N-1}$ from left to right. For the i-th bit, the decoder computes the LLR based on the channel output y and previously decoded bits $\hat{u}_0^{i-1}=(\hat{u}_0,\hat{u}_1,\cdots\hat{u}_{i-1})$

$$\begin{cases} L_i=\ln\dfrac{p(y_0^{N-1},\hat{u}_0^{i-1}\mid u_i=0)}{p(y_0^{N-1},\hat{u}_1^{i-1}\mid u_i=1)} \\ N=2^n\ \beta>0\alpha\in(0,1) \end{cases} \tag{4}$$

The hard decision rule is then applied as:

$$\hat{u}_i=\begin{cases} 0,if & i\in \mathrm{F}(frozen\ \mathrm{bit}) \\ 0,if & i\notin \mathrm{F}\ \text{and}\ \mathrm{L}_i\geq 0 \\ 1,if & i\notin \mathrm{F}\ \text{and}\ \mathrm{L}_i\leq 0 \end{cases} \tag{5}$$

Where F denotes the frozen bit index set. The decoded bit estimate is determined by a three-case rule, if the index i belongs to the frozen bit set, the bit is directly set to 0 without computation; otherwise, the sign of the log-likelihood ratio determines the hard decision. This sequential left-to-right decision process is the defining characteristic of SC decoding, where each bit estimate depends on all previously decoded bits through the recursive LLR computation.

The LLR computations in SC decoding are efficiently implemented through a recursive decomposition. As illustrated in Fig 1, a length-N decoding problem is split into two length-N/2 subproblems. Denoting the left and right LLR vectors as $L_{l=}(L_l^0,L_l^1\ldots L_l^{N/2-l})$ and $L_{r=}(L_r^0,L_r^1\ldots L_r^{N/2-l})$,the two core update functions are defined as follows.

The f-function, used to compute the LLRs of the upper sub-bits:

$$f(a,b)=2\tanh^{-1}[\tanh(\frac{a}{2})\tanh(\frac{b}{2})] \tag{6}$$

which admits the equivalent expression:

$$f(a,b)=sign(a)\cdot sigh(b)\cdot\min(|a|,|b|)+\ln\frac{1+e^{-|a+b|}}{1+e^{-|a+b|}} \tag{7}$$

The g-function, used to compute the LLRs of the lower sub-bits:

$$g(a,b,\hat{u})=(1-2\hat{u})\cdot a+b \tag{8}$$

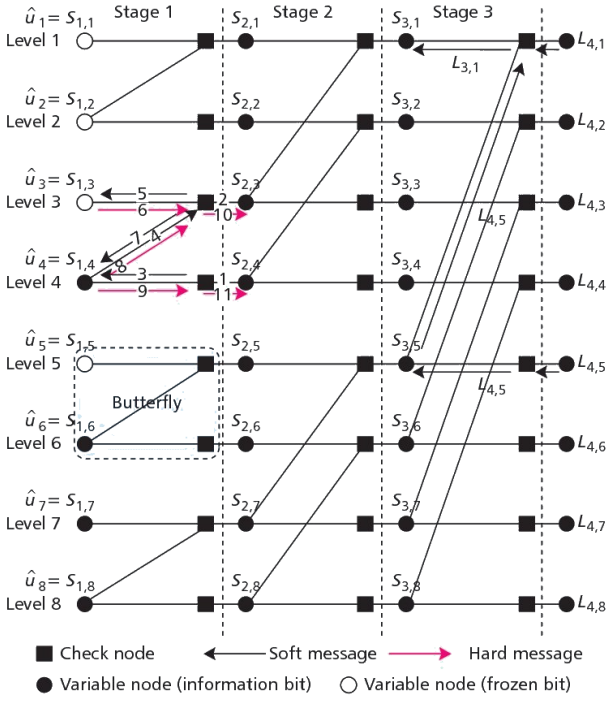


Figure 1. Trellis representation of polar codes

where $\hat{u}$ is the hard decision of the corresponding upper sub-bit. Since the g-function involves only addition and conditional negation, its hardware implementation is straightforward.

The f-function, by contrast, requires hyperbolic tangent evaluations in its exact form, making its simplification the primary target for reducing SC decoder complexity.

B. Min-Sum Approximation

The Min-Sum approximation reduces the complexity of the f-function by discarding the logarithmic correction term, yielding:

$$f_{MS}(a,b) = sign(a) \cdot sign(b) \cdot \min(|a|,|b|) \quad (9)$$

This reduces the transcendental computation to sign extraction, absolute value, and comparison operations, greatly simplifying hardware implementation. However, the Min-Sum approximation is known to systematically overestimate the magnitude of the f-function output $|f_{MS}(a,b)| \geq |f(a,b)|$ for all a,b, which leads to degraded decoding performance. Two classical corrections have been proposed to address this bias:

- Normalized Min-Sum (NMS):

$$f(a,b) = a \cdot sign(a) \cdot sigh(b) \cdot \min(|a|,|b|) \quad (10)$$

where $\alpha \in (0,1)$ is a fixed scaling factor, with typical values in the range 0.75-0.875[10].

- Offset Min-Sum (OMS):

$$f_{OMS}(a,b) = sign(a) \cdot sign(b) \cdot \min(|a|,|b|) - (\beta, 0) \quad (11)$$

where $\beta > 0$ is a fixed offset, with typical values in the range 0.25-0.5[11].

Both NMS and OMS achieve near-ideal f-function performance under floating-point precision. Their correction parameters, however, are determined offline through numerical simulation and applied uniformly to all nodes, making them unable to accommodate the nonlinear distortions that arise under low-precision quantization.

C. Quantization Effects in SC Decoding

While the Min-Sum approximation eliminates transcendental function evaluations, practical hardware implementations do not require the infinite-precision floating-point arithmetic that the MS-SC decoder still implicitly assumes. Introducing an all-integer fixed-point quantization strategy yields the Quantized Min-Sum SC (QMS-SC) decoder [12], in which all LLR values and intermediate results are represented as fixed-point integers. The uniform quantization function $Q(x,y)$ is defined as:

Where x is the input value, $T(q) = 2q - 1$ is the quantization boundary, and q denotes the number of quantization bits. It has been established that 6-bit quantization is sufficient for SC decoders to achieve near-floating-point performance; beyond this point, further increases in bit-width yield negligible gains. Accordingly, this work focuses on the practically relevant range of 3–6 bits, which represents the most challenging and hardware-critical quantization regime.

$$Q(x,q) = \begin{cases} |x + 0.5|, T(q) + 1 < x < T(q) - 1 \\ sigh(x)(T(q) - 1), otherwise \end{cases} \quad (12)$$

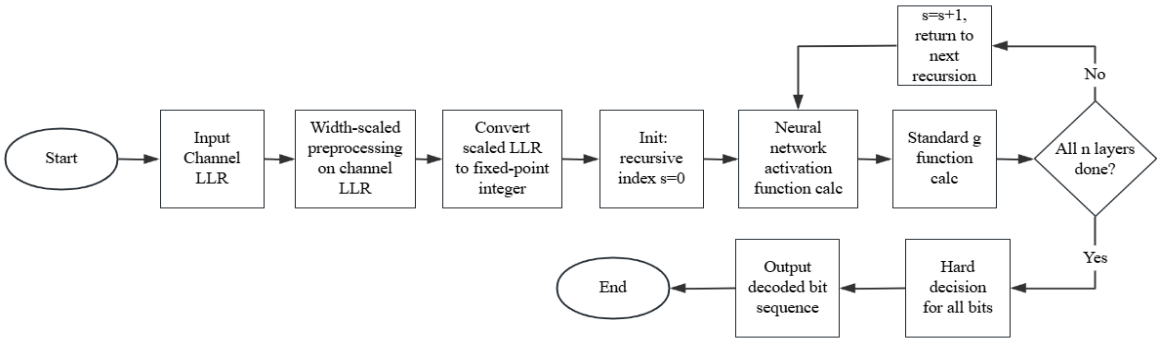


Figure 2. Overall architecture of the NQ-MS-SC decoder

III. Neural-Network-Enhanced Quantized SC Decoder

This section presents the proposed NQ-MS-SC decoder in detail. We begin with an overview of the system architecture, followed by descriptions of the neural network design, the adaptive LLR scaling mechanism, and the quantization-aware training strategy.

A. System Architecture

The overall architecture of NQ-MS-SC decoder is illustrated in Fig 2. Compared with conventional quantized Min-Sum SC decoder, the primary enhancement lies in the introduction of a neural network module to augment the f-function computation. The system comprises the following components:

- Step 1: Applies a bit-width-dependent scaling factor to the channel LLRs prior to quantization, optimizing the utilization of the available dynamic range.
- Step 2: Converts the scaled LLRs into fixed-point integer representations at the target bit-width.
- Step3: Computes the corrected f-function output by predicting optimal correction parameters from the quantized input LLRs via a lightweight neural network.
- Step 4: Retains the conventional g-function computation without modification.
- Step 5: Performs bit decisions based on the sign of the computed LLRs.

The decoding procedure follows the recursive structure of standard SC decoding over nlayers. For the $\mathrm{i}-\mathrm{th}$ node at layers $(s=0,1,\cdots n-1)$, with left and right input LLRs $L_l^{(s,i)}$ and $L_r^{(s,i)}$,the neural network enhanced f-function output is given by:

$$L_f^{(s,i)} = f_{neural}(L_l^{(s,i)}, L_r^{(s,i)}) \tag{13}$$

and the g-function output is:

$$L_g^{(s,i)} = (1-2\hat{u}^{(s,i)}) \cdot L_l^{(s,i)} + L_r^{(s,i)} \tag{14}$$

In Equation (13) and (14), the superscript (s,i) identifies the specific node in the SC decoding tree: s denotes the layer index (ranging from 0 to n-1, where n=log2(N)), and i denotes the node position within that layer. The left and right input LLRs, denoted by the subscripts l and r respectively, correspond to the two child branches of the binary tree at each decomposition stage. The neural-enhanced f-function f_{nerual} replaces the conventional Min-Sum f-function at every node, while the g-function retains its standard linear form. The hard decision value at each node, determined by the sign of the corresponding LLR, propagates upward through the tree to inform subsequent g-function computations.

B. Bit-Width-Adaptive Neural f-Function Design

The dominant sources of f-function error differ fundamentally between low and high bit-widths, motivating distinct neural network architectures for the two regimes.

At low bit-widths (3–4 bits), the limited magnitude resolution of quantized LLRs causes the Min-Sum approximation error and quantization error to couple into a compound nonlinear distortion. To address this, the Adaptive Normalized Offset Min-Sum (Adaptive NOMS) architecture is adopted, which jointly corrects both the gain and the bias of the f-function output. At high bit-widths (5–6 bits), quantization error is relatively minor, and the dominant error source is the logarithmic correction term $\ln\frac{1+e^{-|a+b|}}{1+e^{-|a+b|}}$ discarded by the Min-Sum approximation, whose magnitude follows a relatively stable pattern as a function of the inputs. In this regime, the Adaptive Correction architecture is employed, which augments the Min-Sum output with a learned additive correction term.

1) Adaptive NOMS Architecture

The neural f-function for low bit-widths takes the following Adaptive NOMS form:

Unlike NMS and OMS, which apply fixed correction parameters uniformly, the proposed scheme uses a neural network to adaptively predict the normalization factorαand the offsetβfrom the

magnitude characteristics of the quantized input LLRs, yielding input-dependent optimal corrections. This design offers three key properties：

$$f_{neural}(a,b) = sigh(a)\cdot sigh(b)\cdot \max(\alpha(a,b) \cdot \min(|a|,|b|-\beta(a,b),0) \quad (15)$$

When α=1 and β=0 the formulation reduces exactly to standard Min-Sum, guaranteeing a performance lower bound; the jointly predicted α and β simultaneously compensate for Min-Sum magnitude overestimation (via β) and quantization-induced nonlinear distortion (via α); the max(0) constraint ensures a non-negative output, preserving the physical interpretation of the f-function.

A key distinction of the proposed Adaptive NOMS formulation over conventional fixed-parameter correction schemes lies in its multi-level adaptiveness.

The neural network predicts distinct normalizatio and offset values for each pair of quantized LLR inputs based on their magnitude characteristics, thereby providing node-specific compensation that accounts for the local error distribution at each f-function computation. Beyond input-level adaptation, the framework is inherently bit-width-adaptive through the use of separate, capacity-scaled neural networks for each quantization configuration.

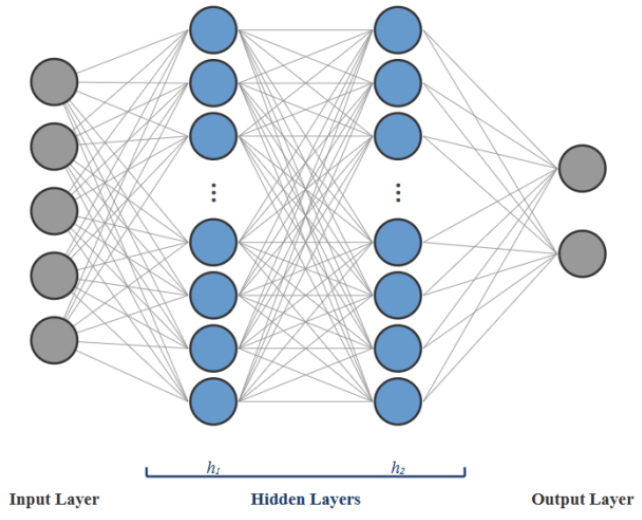


Figure 3. Neural Network Architecture for f-Function Parameter Prediction

Furthermore, since the neural network operates directly on quantized LLR magnitudes whose statistical distribution is governed by the channel SNR, at low SNR, where LLR magnitudes are small and concentrate near the quantization origin, the network learns to apply conservative corrections; at high SNR, where LLR magnitudes span a wider range and quantization clipping becomes more prevalent, the network shifts toward more aggressive compensation.

As illustrated in Fig 3, The input feature vector z is constructed from the absolute values of the two quantized LLR inputs (a,b) at each f-function node. During SC decoding, whenever the decoder reaches an f-function node in the binary tree, it has already computed the two input LLRs from the previous recursion level. These two values are the raw data from which z is derived—no additional data collection or sensor measurement is required, as all inputs originate from the channel observations and prior decoding computations within the SC algorithm itself. The five features are designed to capture the essential magnitude characteristics that determine the optimal correction:

$$z=\left[\frac{|a|}{M},\frac{|b|}{M},\frac{\min(|a|,|b|)}{M},\frac{\max(|a|,|b|)}{M},\frac{|a|,|b|}{M}\right]^T \quad (16)$$

Where $M = 2(2^{q-1}-1)$ is a normalization factor determined by the quantization range. The network consists of two fully connected hidden layers with LeakyReLU activations. The output layer is a two-neuron linear layer producing the raw predictions α_{raw} and β_{raw},which are mapped to bounded parameters via:

$$\alpha = 0.9 + 0.2\cdot \tanh(\alpha_{raw}) \in [0.7, 1.1]$$
$$\beta = 0.25\cdot sighmoid(\beta_{raw}) \in [0, 0.25] \quad (17)$$

The hidden layers employ LeakyReLU activations, which preserve gradient flow for negative inputs and thereby avoid the dying neuron problem that standard ReLU may introduce during training, while remaining computationally lightweight for potential hardware deployment.

The output layer applies tanh and sigmoid activations to constrain the predicted parameters within their valid ranges as defined in Equation (17).

To ensure stable optimization, the network weights are initialized such that the output biases correspond to near-standard Min-Sum behavior (approximately 1 for normalization and 0 for offset), allowing the decoder to start from a known-good baseline and progressively learn input-dependent corrections. The training loss typically converges within 200-300 epochs.the normalization factor decreases for input pairs with similar magnitudes, where the Min-Sum overestimation is most pronounced, and approaches unity when the magnitude difference is large and the approximation is already accurate. This behavior aligns with the theoretical structure of the logarithmic correction term in Equation (7), confirming that the network has captured the underlying error mechanism.

The five components of z in Equation (16) are designed to jointly characterize the magnitude profile of the input pair. The normalized individual magnitudes $|a|/M$ and $|b|/M$ reflect the reliability of each input LLR, while $min(|a|,|b|)/M$ directly corresponds to the Min-Sum output magnitude before sign multiplication, serving as the baseline estimate. The remaining two features, $min(|a|,|b|)/M$ and $(|a|-|b|)/M$, capture the magnitude asymmetry between the inputs. All features are normalized by $M=2(2^{(q-1)}-1)$, bounding them within [-1,1] to stabilize training across different bit-widths. Notably, these features require only absolute value, comparison, subtraction, and division by a constant, introducing negligible computational overhead.

2) Adaptive Correction Architecture

The neural f-function for high bit-widths augments the Min-Sum output with a learned additive correction:

$$f_{neural}(a,b)=sigh(a)\cdot sigh(b)\cdot \max(\min(|a|,|b|) +\Delta(a,b),0) \quad (18)$$

where the correction term $\Delta(a,b)\in[-1.5,0.5]$ is predicted by a neural network. The input feature construction follows the same procedure as the low bit-width scheme, with the normalization factor set to $M=2^{q-1}-1$.The network likewise consists of two fully connected hidden layers, but the output layer is a single-neuron linear layer that directly produces Δ.

To account for the varying correction complexity across bit-widths, a bit-width-adaptive network capacity design is adopted. Lower bit-widths introduce more severe nonlinear distortion and therefore require greater network expressiveness. Table I summarizes the network configuration for each bit-width, including hidden layer width, parameter count, and normalization factor.

For both architectures, the sign of the f-function output is determined entirely by the product of input signs:

$$signf(a,b)=sign(a)\cdot sign(b) \quad (19)$$

This mathematical property allows the sign computation to be separated from the network, so that the neural network is responsible solely for magnitude correction, reducing the learning burden and improving training stability.

TABLE I. NETWORK CONFIGURATION AT DIFFERENT BIT-WIDTHS

Bit-Width	Network Structure	Parameters	Normalization Factor M
3	4→32→24→2	1002	6
4	5→48→32→2	1922	14
5	5→48→48→1	2689	30
6	5→32→32→1	1281	62

C. Adaptive LLR Scaling Mechanism

A fundamental challenge in ultra-low bit-width quantization is the mismatch between the limited dynamic range of the quantizer and the wide amplitude distribution of channel LLRs. For a BPSK-AWGN channel at $E_b/N_0=4dB$, the

typical magnitude of the channel LLR $|L_{ch}|$ can exceed 20, whereas a 3-bit quantizer covers only the range [-3,+3].Without preprocessing, the vast majority of channel LLRs would saturate at the boundary values, rendering the soft information indistinguishable across different reliability levels and severely degrading decoding performance. Appropriate LLR scaling is therefore a necessary prerequisite for effective low bit-width quantization.

This work assigns a fixed scaling factor γ_q to each bit-width, mapping the LLR distribution over the target SNR range into the effective interval of the quantizer. Specifically, letting $L_{\max} = 2^{q-1} - 1$ denote the maximum representable value for q-bit quantization, the scaling is designed to keep the magnitude of the scaled LLR within $0.8\,L_{\max}$, reserving a 20% margin for extreme values. The scaling factors are determined through theoretical estimation and simulation-based fine-tuning: 3-bit quantization requires the most aggressive compression ($\gamma_3 = 0.4$), 6-bit quantization requires virtually no scaling $(\gamma_6 = 1.0)$.

D. Quantization-Aware Training Strategy

The quantization operator $Q(\bullet)$ is a non-differentiable staircase function and therefore cannot be directly incorporated into gradient-based backpropagation. To enable end-to-end training, the Straight-Through Estimator (STE) is adopted to approximate the gradient through the quantization operation: in the forward pass, the true quantized value $\hat{x} = Q(x,q)$ is computed; in the backward pass, the quantization operation is treated as the identity mapping. This work further employs a modified STE in which gradients in the saturation region ($|x| > L_{\max} + 0.5$) are multiplied by a factor of 0.1 rather than being set to zero. This preserves a weak gradient signal that encourages the network to avoid saturating outputs, while preventing gradient explosion.

Training data are generated from a BPSK-AWGN channel model over the target SNR range of 1-4dB. Input pairs (a,b)are uniformly sampled at each SNR point and passed through the scaling and quantization pipeline before being fed to the network.

Specifically, for each SNR point in the range1-4dB, the channel LLR values are computed as $L_{ch} = \frac{2y}{\sigma^2}$, where y is the received signal sample and σ^2 is the noise variance. Input pairs (a,b) are then formed by randomly selecting two LLR values from the channel output distribution, simulating the pairs that would appear at f-function nodes during actual SC decoding. Each input pair is passed through the adaptive scaling and quantization (Equation (12)) before being fed to the neural network. The corresponding training target for each pair is computed from the exact f-function in Equation (6) evaluated on the original unquantized inputs, providing a ground-truth reference. A total of approximately 500,000 training samples are generated per bit-width configuration, with equal representation across all SNR points in the target range.

The training target is defined as a weighted combination:

$$y_{target} = 0.4Q(f_{neural}(a,b),q) + 0.6 \cdot f_{MS}(a,b) \quad (20)$$

These weighting balances two objectives, approximating the ideal f-function (40% weight) and preserving the stable Min-Sum baseline (60% weight). The conservative weighting toward the Min-Sum baseline is particularly important at 3 bits, where an overly aggressive training target can destabilize optimization. In addition, a supplementary dataset uniformly covering all integer-valued quantization points is included to ensure adequate coverage of all possible inputs.

The loss function jointly addresses quantization accuracy, sign consistency, and small-value region fidelity:

$$\mathrm{L} \equiv \mathrm{L}_{\overline{MES}} + \lambda_1 \mathrm{L}_{\overline{sign}} \quad (21)$$

where$\mathcal{L}_{\mathrm{MES}}$is the mean squared error between the network output and the quantized ideal target,

and L_{sign} is a sign consistency loss with $\lambda_1 = 0.2$.The network is trained with the AdamW optimizer at an initial learning rate of 0.003, with a cosine annealing with warm restarts schedule $T_0 = 100$ and an early stopping patience of P=60. The complete training procedure is summarized in Algorithm1.

IV. Simulation Results and Analysis

A. Performance Analysis of 3-bit NQ-MS-SC Decoder

Figs 4(a) and 4(b) present the BER and FER performance of the proposed 3-bit NQ-MS-SC decoder alongside the Float SC baseline and the 3-bit QMS-SC scheme for a polar code with N = 512, K = 256, and R = 0.5. All simulations are conducted over an AWGN channel with BPSK modulation, and the SNR range extends from 0 to 4 dB.

Algorithm 1: Quantization-Aware Neural Network Training

Input：Bit-width q，training epochs E，batch size B，learning rate η，SNR range

Output：Optimized network parametersΘ^*

1:Determine network capacity (hidden layer width) and normalization factor M based on q

2:Initialize network parametersΘ，with output biases set such that$\alpha\approx1$, $\beta\approx0$（Min-Sum initialization）

3:Generate training dataset$D=\{(a_i,b_i,y_i)\}_{i=1}^{N_{samples}}$

4:**for** epoch = 1 to E **do**

5: **for** each mini-batch（a_B,b_B,y_B) **do**

6: Forward pass:$\alpha_B,\beta_B=param_{net}(features(a_B,b_B))$

7: Compute output：$\hat{y}_B=sign\cdot max(\alpha\cdot min(|a|,|b|)-\beta,0)$

8: STE quantization：$\hat{y}_{B_q}=STE_{Quantize}(\hat{y}_B)$

9: Compute loss：$L=L_{MSE}+0.2\cdot L_{sign}$

10: Gradient clipping +AdamWparameter update

11:**end for**

12: Cosine annealing learning rate update + early stopping check

13:**end for**

14:**return** Best parametersΘ^*

In the low-SNR regime (0-1.5 dB), both 3-bit schemes exhibit a performance gap relative to the floating-point baseline. Specifically, at $E_b / N_0 = 1.0dB$, 3-bit QMS-SC representing a reduction of approximately 10.9% over the QMS-SC baseline. As the SNR increases, the neural-enhanced decoder demonstrates increasingly pronounced gains. at $E_b / N_0 = 2.5dB$, the BER of NQ-MS-SC compared to QMS-SC, yielding a gain of approximately 1.64 dB in effective coding gain. The FER curves in Fig 4(b) exhibit a similar trend. These results confirm that the neural network-based scaling factor optimization effectively compensates for the quantization-induced information loss inherent in 3-bit representation.

Figs 5(a) and 5(b) illustrate the BER and FER performance for N = 1024. The overall performance gap between floating-point and 3-bit schemes widens compared to the N = 512 case, which is consistent with the increased sensitivity of longer polar codes to quantization noise at moderate code lengths. Nevertheless, the proposed NQ-MS-SC consistently outperforms QMS-SC across the entire SNR range. The NQ-MS-SC decoder demonstrates a sustained advantage over QMS-SC. At $E_b / N_0 = 2.0dB$, NQ-MS-SC achieves a BER compared to QMS-SC, a relative improvement of 15.7%. More significantly, in the high-SNR regime at $E_b / N_0 = 4.0dB$, representing a gain of 29.2%.

The FER curves in Fig 6(b) exhibit analogous behavior. These results indicate that the advantage of neural network-aided scaling factor optimization is amplified at larger code lengths, where the per-node quantization error accumulates more significantly across the successive cancellation tree.

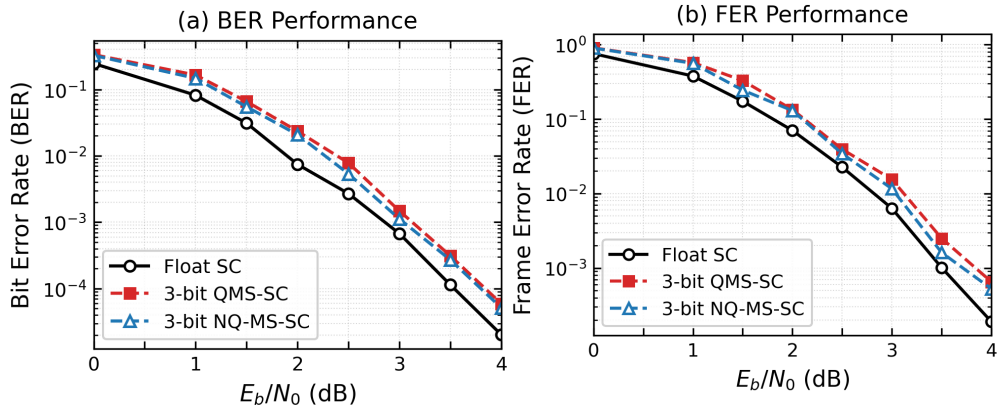


Figure 4. BER and FER Performance of Float SC, QMS-SC, and NQ-MS-SC Decoders for Polar Codes (N=512, R=0.5) under 3-bit Quantization over AWGN Channel (a)BER;(b)FER.

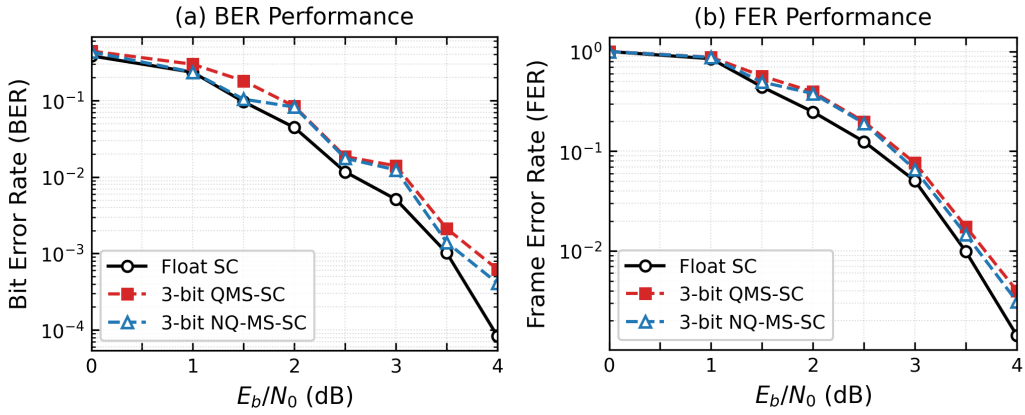


Figure 5. BER and FER Performance of Float SC, QMS-SC, and NQ-MS-SC Decoders for Polar Codes (N=1024, R=0.5) under 3-bit Quantization over AWGN Channel (a)BER;(b)FER.

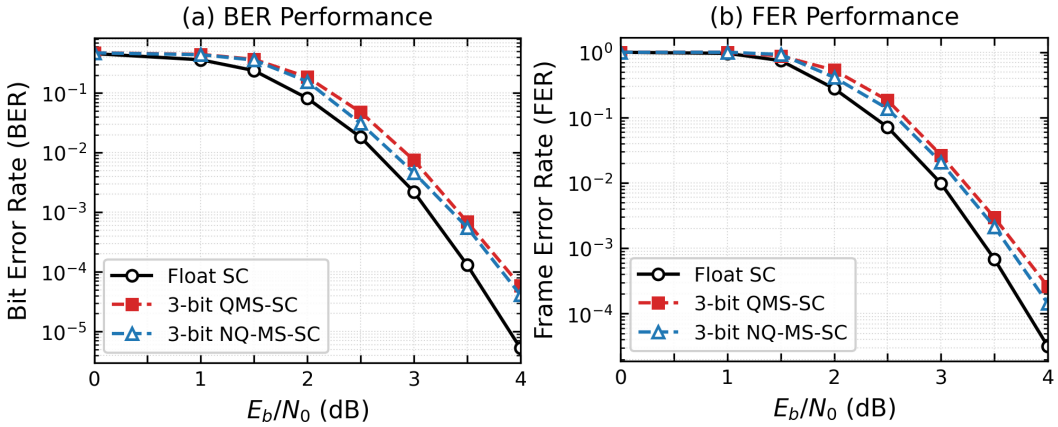


Figure 6. BER and FER Performance of Float SC, QMS-SC, and NQ-MS-SC Decoders for Polar Codes (N=2048, R=0.5) under 3-bit Quantization over AWGN Channel (a)BER; (b)FER.

Figs 6(a) and 6(b) present results for N = 2048, which represents the longest code length evaluated in this study. As code length increases, the quantization degradation of 3-bit schemes becomes more pronounced, and the gap from the floating-point baseline widens in the low-SNR region. Both quantized decoders exhibit BER values near 0.5 at 0 dB, reflecting near-random decoding performance at very low SNR, which is consistent with observations in the literature for

coarsely quantized SC decoders at such code lengths.

B. *Performance Analysis of 4/5/6-bit NQ-MS-SC Decoder*

Fig 7 presents the BER performance for N = 512 across 4-bit, 5-bit, and 6-bit quantization schemes, comparing QMS-SC and NQ-MS-SC for each bit-width.

As the quantization resolution increases from 4 to 6 bits, both QMS-SC and NQ-MS-SC converge toward the Float SC performance, which is expected since higher bit-width reduces the quantization granularity. For each bit-width, NQ-MS-SC consistently outperforms its QMS-SC counterpart in the moderate-to-high SNR range. For 4-bit quantization at $E_b / N_0 = 3.0dB$, NQ-MS-SC achieves BER lower than QMS-SC. For 5-bit quantization, the NQ-MS-SC achieves a 9.2% improvement compared to QMS-SC at the same SNR point, suggesting that the learned scaling factors have successfully compensated for quantization distortion and even introduced a marginal stochastic regularization effect at high SNR. It is observed that the performance ordering among bit-widths is not strictly monotonic at very low SNR (1.0 – 1.5 dB). This phenomenon is attributed to the noise-dominated regime where the benefit of finer quantization resolution is partially offset by the greater sensitivity of higher-bit representations to channel noise, consistent with prior findings on finite-alphabet decoding of polar codes.

For 6-bit quantization, the BER of NQ-MS-SC at $E_b / N_0 = 2.0dB$ closely approaching the Float SC value, confirming that 6-bit NQ-MS-SC effectively eliminates most quantization-induced performance loss at this code length. In the FER domain, the performance hierarchy at $E_b / N_0 = 3.5dB$ is as follows: Float SC< 6-bit NQ-MS-SC < 5-bit NQ-MS-SC< 4-bit NQ-MS-SC< 5-bit QMS-SC <4-bit QMS-SC , confirming that the neural-enhanced decoder uniformly outperforms its quantized Min-Sum counterpart across all bit-widths.

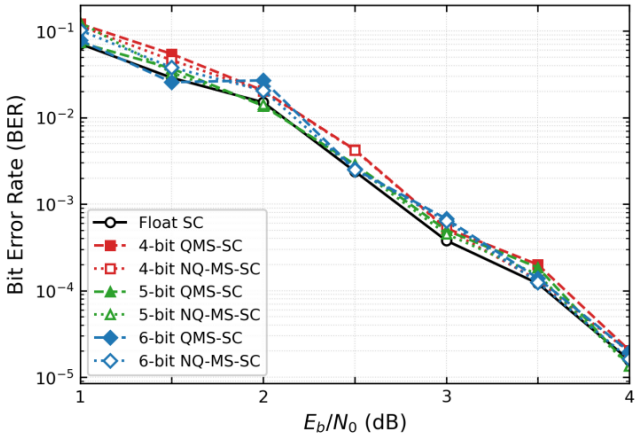
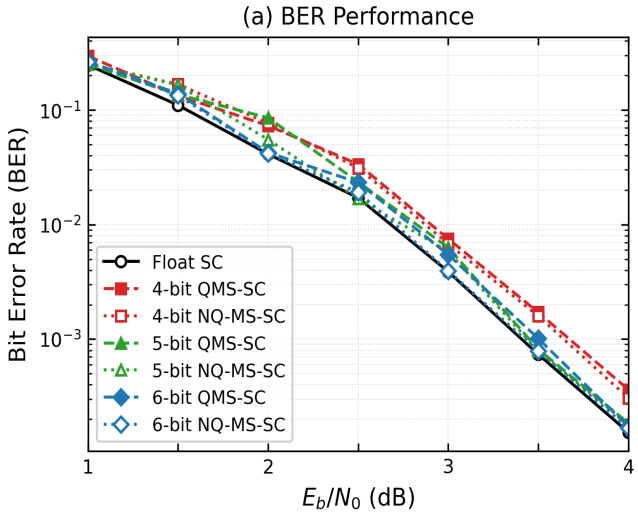


Figure 7. BER Performance of Float SC, QMS-SC, and NQ-MS-SC Decoders for Polar Codes (N=512, R=0.5) under 4/5/6-bit Quantization over AWGN Channel

These results collectively demonstrate that the proposed neural network-based framework provides a consistent and scalable performance gain over the quantized Min-Sum baseline, with the improvement becoming more significant as SNR increases and more pronounced for lower bit-widths where quantization distortion is more severe.

Figs 8(a) and 8(b) present the BER and FER results for N = 1024 across the three bit-widths. The general trend is consistent with the N = 512 case, however, several notable features specific to N = 1024 merit discussion.



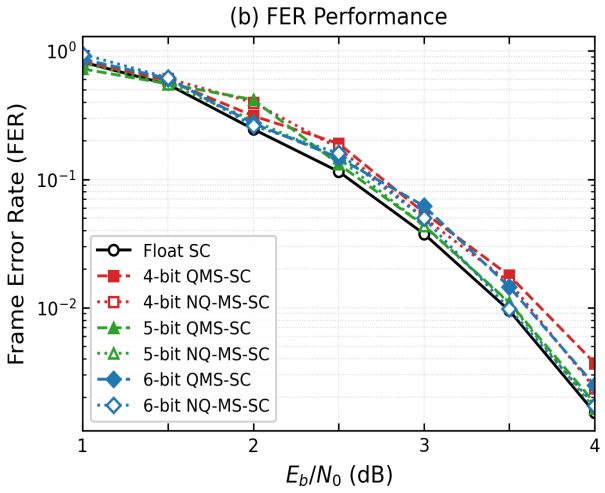


Figure 8. BER and FER Performance of Float SC, QMS-SC, and NQ-MS-SC Decoders for Polar Codes (N=1024, R=0.5) under 4/5/6-bit Quantization over AWGN Channel (a)BER;(b)FER.

C. Discussion

The simulation results across all tested configurations consistently demonstrate that the proposed NQ-MS-SC decoder bridges the performance gap between coarsely quantized Min-Sum decoding and the floating-point upper bound. A closer examination of the comparative analysis reveals that the effectiveness of the neural-enhanced framework is governed by three interacting factors: SNR regime, code length, and quantization bit-width

The improvement contributed by neural-enhanced scaling factor optimization is most pronounced in the moderate-to-high SNR regime (SNR⩾2.0 dB). At lower SNR, channel noise overwhelms quantization distortion as the dominant source of decoding error, leaving limited room for quantization-specific compensation to manifest. As SNR increases and channel noise recedes, the residual quantization distortion becomes the binding constraint on decoder performance, and the learned adaptive parameters — calibrated precisely to this distortion — yield increasingly significant gains. This SNR-dependent behavior is consistent with the fundamental operating principle of quantization-aware learning, the network can only recover what quantization has selectively destroyed, not what the channel has erased.

This bit-width dependency interacts closely with code length. As the successive cancellation tree deepens from N = 512 to N = 2048, per-level quantization errors propagate and accumulate across more decoding stages, amplifying the sensitivity of final bit decisions to f-function approximation quality. Consequently, the FER improvement of NQ-MS-SC over QMS-SC at $E_b / N_0 = 4.0dB$ grows substantially with code length — from 19.3% at N = 512 and 21.8% at N = 1024 to 45.0% at N = 2048—confirming that the adaptive scaling mechanism becomes progressively more impactful as the error accumulation path lengthens.

Viewed through the lens of hardware implementation, these trends converge on a clear practical conclusion. At 6-bit resolution, both NQ-MS-SC and QMS-SC closely approach the floating-point baseline, leaving little headroom for further improvement; the neural enhancement is thus most valuable precisely where hardware constraints are most severe, namely in the 3 to 4-bit regime, where it delivers meaningful error-rate reductions at no additional memory cost beyond storing the compact set of trained network parameters. This positions the proposed framework as a naturally targeted solution for resource-constrained deployment scenarios, where ultra-low bit-width quantization is not a design choice but a practical necessity.

V. Conclusions

This paper presented the NQ-MS-SC decoder, a lightweight framework that jointly compensates for Min-Sum approximation error and quantization-induced distortion through quantization-aware neural network training. The fundamental limitation of conventional NMS and OMS schemes lies in their fixed, uniformly applied correction parameters optimized under full-precision conditions, making them structurally unable to adapt to the nonlinear distortions inherent in ultra-low bit-width quantization. The proposed decoder overcomes this by employing a bit-width-adaptive neural f-function that predicts input-dependent correction parameters directly from quantized LLRs, complemented by an adaptive LLR scaling mechanism that maximizes dynamic range utilization prior to quantization.

Across code lengths from N = 512 to 2048 and bit-widths from 3 to 6 bits, NQ-MS-SC consistently outperforms the quantized Min-Sum baseline, with gains that grow more pronounced at higher SNR, longer code lengths, and coarser quantization — precisely the conditions where residual quantization distortion dominates decoding error. At N = 2048 with 3-bit resolution, an FER reduction of 29.2% is achieved at 4.0 dB, and the margin amplifies with code length as per-stage quantization errors accumulate along the deeper SC tree. At 5 to 6-bit resolution, no performance penalty is introduced, establishing NQ-MS-SC as a non-degrading replacement for conventional quantized decoders across all practical bit-widths. With fewer than 2,000 trainable parameters per configuration and no modifications to the g-function or SC scheduling, the proposed framework offers a scalable and hardware-compatible path toward high-quality polar code decoding under stringent resource constraints.

REFERENCES

[1] E. Arıkan, Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] V. Bioglio, C. Condo, and I. Land, Design of polar codes in 5G new radio, IEEE Commun. Surv. Tutor., vol. 23, no. 1, pp. 29–40, First quarter, 2021.

[3] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, Reduced-complexity decoding of LDPC codes, IEEE Trans. Commun., vol. 53, no. 8, pp. 1288–1299, Aug. 2005.

[4] J. Chen and M. P. C. Fossorier, Near optimum universal belief propagation based decoding of low-density parity-check codes,IEEE Trans. Commun., vol. 50, no. 3, pp. 406–414, Mar. 2002.

[5] Q. Zhang, B. Li, and K. Niu, "On optimized uniform quantization for SC decoder of polar codes," in Proc. IEEE Int. Conf. Commun. (ICC), Sydney, Australia, Jun. 2014.

[6] B. Yuan and K. K. Parhi, Low-latency successive-cancellation polar decoder architectures using 2-bit decoding, IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 1241–1254, Apr. 2014.

[7] L. Lugosch and W. J. Gross, Neural offset min-sum decoding, in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Aachen, Germany, Jun. 2017, pp. 1361–1365.

[8] W. Xu, X. Tan, Y. Be'ery, Y.-L. Ueng, Y. Huang, X. You, and C. Zhang, Deep learning-aided belief propagation decoder for polar codes,IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 10, no. 2, pp. 189–203, Jun. 2020.

[9] N. Lyu, J. Yan, and X. You, Optimization and hardware implementation of learning assisted min-sum decoders for polar codes, J. Signal Process. Syst., vol. 93, no. 2–3, pp. 287–301, Mar. 2021.

[10] J. Zhang, M. P. C. Fossorier, and D. Gu, "Two-dimensional correction for min-sum decoding of irregular LDPC codes," IEEE Commun. Lett., vol. 10, no. 3, pp. 180–182, Mar. 2006.

[11] M. Jiang, C. Zhao, L. Zhang, and E. Xu, Adaptive offset min-sum algorithm for low-density parity check codes,IEEE Commun. Lett., vol. 10, no. 6, pp. 483–485, Jun. 2006.

[12] Pimsri, W.; Muangkammuen, P.; Suthisopapan, P.; Imtawil, V. All-Integer Quantization for Low-Complexity Min-Sum Successive Cancellation Polar Decoder. Appl. Sci. 2025, 15, 3241.