

SPLITLOSS AND DEGENERACY ANALYSIS FOR QLDPC DECODING

Minjoo Kim
School of Electrical Engineering
Korea University
Seoul 02841, Republic of Korea
mindy0522@korea.ac.kr

Jun Heo
School of Electrical Engineering
Korea University
Seoul 02841, Republic of Korea
junheo@korea.ac.kr

Abstract—This study proposes a decoder that operates in a single stage without post-processing by redesigning the loss function of a Graph Neural Network (GNN)-based decoder. Decoding exhibits two failure modes: syndrome mismatch and logical sector error. We propose SplitLoss, which decomposes the constraint into a stabilizer block and a logical block, assigning separate weights to each. Furthermore, we introduce the operational degeneracy fraction, a metric that quantifies the severity of code degeneracy, and evaluate it across code families with different structures, namely Bivariate Bicycle and Hypergraph Product codes. Experimental results show that (i) using both blocks jointly with a higher emphasis on the logical block consistently yields the best performance, (ii) the proposed SplitLoss-based GNN outperforms other decoders without post-processing, and (iii) the improvement in logical error rate (LER) increases as the severity of degeneracy becomes greater.

Keywords—Quantum error correction, Quantum low-density parity-check codes, Graph neural network decoder, Degeneracy-aware decoding, Logical error rate

I. INTRODUCTION

Large-scale Fault-Tolerant Quantum Computing (FTQC) requires Quantum Error Correction (QEC) to detect and correct errors in real time during computation. The performance of a QEC code depends heavily on the choice of decoder.

A variety of QEC decoding algorithms have been developed. For topological codes such as surface codes, Minimum Weight Perfect Matching (MWPM) is widely used, while for quantum low-density parity-check (QLDPC) codes, belief propagation combined with ordered statistics decoding (BP+OSD) is one of the standard approaches. Belief propagation (BP) is a message-passing algorithm originally developed for classical low-density parity-check (LDPC) codes, and many techniques from classical LDPC decoding have been adapted to the QLDPC setting.

Quantum codes exhibit degeneracy, a property absent in classical codes, where multiple distinct physical errors can produce the same syndrome. Combined with short cycles in the Tanner graph, this leads to convergence issues in belief propagation (BP), which assumes a unique minimum-weight solution.

In practice, BP often fails to produce a valid syndrome-matching estimate, requiring post-processing methods such as ordered statistics decoding (OSD). OSD has cubic complexity and does not fundamentally resolve degeneracy-induced failures, as incorrect logical sector selection among syndrome-equivalent errors can still lead to logical errors. Therefore, finding a method to successfully decode QLDPC codes in a single step without post-processing has been a long-standing open problem.

This limitation has motivated neural-network-based decoders, particularly graph neural network (GNN) decoders, which perform QLDPC decoding using learned message-passing rules with greater expressive power than conventional BP. Despite this progress, most neural decoders still rely on post-processing or secondary decoders to supplement their performance, and more fundamentally, they do not explicitly incorporate degeneracy into the learning objective.

Astra [1] improves decoding performance by learning a message-passing algorithm on the Tanner graph using graph neural networks (GNNs), achieving LER superior to BP+OSD without post-processing. It also introduces a degeneracy-aware loss inspired by neural belief propagation (NBP), making it one of the first GNN-based decoders to incorporate degeneracy into training.

Astra's degeneracy loss handles the two failure conditions of decoding, syndrome mismatch and logical sector error, as a single term. Since logical sector errors arise within the syndrome space, we propose SplitLoss, which separates these two terms and assigns independent weights α and β to each. We additionally introduce a warm-up training procedure that prioritizes syndrome learning in the early stage. Beyond the loss redesign, we introduce an operational degeneracy fraction that quantifies a code's degree of degeneracy, measure it across code families of differing structure, and analyze the relationship between this metric and the effectiveness of the proposed loss modification.

II. PRELIMINARIES

A. Algebraic structure of Calderbank-Shor-Steane codes

Calderbank-Shor-Steane (CSS) codes are defined by two parity-check matrices, H_X and H_Z satisfying the commutation condition $H_X H_Z^T = 0 \pmod{2}$. From a linear algebraic perspective, this condition implies that the row space of one code is contained in the kernel (codeword space) of the other: $rs(H_Z) \subseteq \ker(H_X)$, $rs(H_X) \subseteq \ker(H_Z)$.

This condition guarantees that the X-stabilizers and Z-stabilizers commute with each other.

As shown in Fig. 1, $\ker(H_Z)$ contains all X-type operators that produce the trivial syndrome and decomposes into the X-stabilizer space $rs(H_X)$ and the quotient space $\ker(H_Z)/rs(H_X)$, which represents the logical X operators.

Decoding X-errors consists of estimating the error e from the observed syndrome $\sigma = H_Z e$. However, the observed syndrome does not uniquely specify which error occurred (Section II.B), and the decoder must identify the correct logical class among all errors sharing the same syndrome.

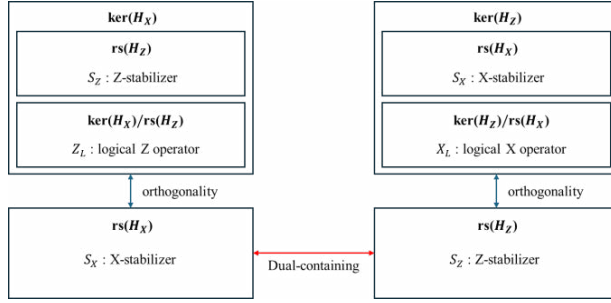


Fig. 1. Algebraic structure of CSS codes. $\ker(H_Z)$ contains all X-type operators with trivial syndrome and decomposes into the X-stabilizer space $rs(H_X)$ and the logical X operator space $\ker(H_Z)/rs(H_X)$ of dimension k . Symmetrically, the Z-sector decomposes $\ker(H_X)$ into $rs(H_Z)$ and the logical Z operator space. The dual-containing condition connects the stabilizer spaces of the two sectors.

B. Degeneracy

Unlike classical codes, quantum codes possess a fundamental property known as degeneracy. Multiple distinct errors can produce the same syndrome, and errors differing only by a stabilizer have identical logical effect and are therefore equivalent. Conversely, two errors sharing the same syndrome but differing by a nontrivial logical operator are not equivalent. Correcting one as the other introduces a logical error. Therefore, the success of quantum decoding depends not on identifying the exact error that occurred, but on determining the logical class to which the error belongs.

C. Coset and Subcoset

The error space is partitioned into 2^{n-k} syndrome cosets [2], each containing all errors producing a given syndrome. Within each coset, errors are further subdivided into subcosets (logical classes), which are sets of errors with the same logical effect. Any error E_S associated with syndrome S can be written as

$$E_S = T_S \cdot S \cdot L \quad (1)$$

where T_S is a representative error, S is a stabilizer, and L is a logical operator.

The goal of degeneracy-aware decoding is not to find the single most probable error, but to find the subcoset (i.e., the logical operator induced by the error) that maximizes the probability sum considering degeneracy.

D. Belief Propagation and GNN Decoding

BP performs iterative message passing on the Tanner graph to find a variable assignment satisfying as many syndrome constraints as possible. However, standard BP estimates only the single most-likely error given the observed syndrome, rather than directly optimizing over subcosets. Short cycles and local minima in the Tanner graph, compounded by degeneracy, cause BP to fail frequently on QLDPC codes, necessitating costly post-processing such as OSD.

GNN decoders replace the fixed BP update rules with learned message-passing functions. Check nodes and variable nodes correspond to nodes in the Tanner graph. The parity-check matrix H defines the message passing topology. Unlike BP, the GNN learns message generation and node update rules, providing greater expressive power to handle degeneracy and short cycles.

E. Loss Functions

The loss function serves as the training signal that guides the GNN's learned message-passing and node-update functions toward producing error probability estimates consistent with both the observed syndrome and the code's degeneracy structure.

Cross-entropy is the standard loss function for training neural networks, including GNNs, on binary or multi-class prediction tasks, and is likewise the natural choice for a decoder that outputs a per-qubit error probability.

The cross-entropy loss penalizes the discrepancy between the predicted error probability p_i and the true error label $y_i \in [0,1]$:

$$\mathcal{L}_{CE} = -\sum_i [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)] \quad (2)$$

Cross-entropy alone does not account for degeneracy: training the model to match a single error pattern is misleading when many stabilizer-equivalent errors share the same syndrome. To address this, [1] augments the loss with a degeneracy term $\mathcal{L}_{deg} = M \cdot r \pmod{2}$, where M is a basis matrix of $\ker(H_X)$ that spans both the stabilizer and logical constraints, guiding the residual $r = e + \hat{e} \pmod{2}$ to satisfy both constraints, using a differentiable approximation $|\sin(\pi x/2)|$ in place of the mod-2 operation.

III. SPLITLOSS AND DEGENERACY-AWARE DECODING

A. SplitLoss and warm-up

The success condition for X-error decoding is that the residual $r = e + \hat{e}$ lies in $rs(H_X)$, which requires two conditions to hold simultaneously: $H_Z \cdot r = 0$ (syndrome match, same coset) and $Z_L \cdot r = 0$ (logical sector match, same subcoset). The existing degeneracy loss [1] uses M , a basis of $rs([H_Z; Z_L])$ as a single block, handling both conditions within a single term. However, the two conditions correspond to qualitatively different failure modes. Syndrome mismatch ($H_Z \cdot r \neq 0$) is a failure in which the prediction does not reproduce the observed syndrome, whereas logical sector error ($Z_L \cdot r \neq 0$) is a failure in which the syndrome is satisfied but the wrong subcoset is selected. Since logical sector distinction occurs within the syndrome ($\ker(H_Z)/rs(H_X)$), separating the two terms and weighting them independently enables direct alignment with the evaluation criteria.

We propose SplitLoss, which decomposes the degeneracy constraint into a stabilizer block and a logical block and assigns independent weights to each:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda(\alpha \cdot \mathcal{L}_{stab} + \beta \cdot \mathcal{L}_{logical}) \quad (3)$$

where the success conditions are

$$H_Z \cdot r = 0 \text{ and } Z_L \cdot r = 0 \quad (4)$$

and the mod 2 operation is replaced by the differentiable approximation $|\sin(\pi x/2)|$ for use in a machine learning setting. $\mathcal{L}_{stab}$ penalizes syndrome mismatch via H_Z , and $\mathcal{L}_{logical}$ directly penalizes incorrect logical sector selection via Z_L . The hyperparameters α and β control the relative contribution of each term.

Applying the degeneracy terms from the start of training may interfere with the model's ability to learn basic error patterns before degeneracy constraints are introduced. To address this, we adopt a warm-up procedure in which λ is increased linearly over the course of training. For a total of n training epochs, λ follows the schedule below:

- $[0, n/4)$: $\lambda = 0$. Training on cross entropy only to learn basic error patterns.
- $[n/4, n/2)$: λ increases linearly from 0 to 1. Degeneracy terms are introduced gradually.
- $[n/2, n]$: $\lambda = 1$. Full SplitLoss is applied.

At epoch $n/2$, when λ reaches 1, the model selection criterion and learning rate scheduler are reset to prevent checkpoints from the warm-up phase from persisting as the final model. This procedure explicitly separates an early phase that prioritizes syndrome learning from a later phase that reinforces degeneracy alignment.

B. Operational Degeneracy Fraction

We introduce a quantitative metric that directly reflects how frequently degeneracy influences the decoder under an actual error distribution.

For a given physical error rate p , we sample n_{samp} error instances e . For each error, we check whether there exists a stabilizer $s \in \text{rs}(H_X)$ such that $\text{wt}(e \oplus s) < \text{wt}(e)$. The operational degeneracy fraction is defined as:

$$f_{\text{deg}} = \frac{|\{e \neq 0 : \exists s \in \text{rs}(H_X), \text{wt}(e \oplus s) < \text{wt}(e)\}|}{|\{e \neq 0\}|} \quad (5)$$

A larger f_{deg} indicates that minimum-weight representatives are frequently non-unique, making the target of cross-entropy-based training increasingly ambiguous.

IV. EXPERIMENTAL SETUP

A. Code Families

- BB [[72, 12, 6]]
- HGP [[58, 16, 3]]
- HGP [[129, 28, 3]]

B. Decoders

- BP+OSD(0): A classical decoder combining BP with min-sum scheduling and a maximum of 12 iterations, followed by OSD-0 post-processing.
- GNN-CE: A GNN trained with cross entropy loss only.
- GNN-DegLoss: A GNN with the degeneracy loss [1].
- GNN-SplitLoss: A GNN trained with the proposed SplitLoss, with a separate model for each of four (α, β) settings: (0.5, 1.5), (1.5, 0.5), (0.0, 1.0), and (1.0, 0.0).

C. GNN Decoder Architecture and Training Setup

The model learns three local functions: a message generation function f implemented as an MLP, a sum aggregation that collects messages from neighboring nodes, and a node update function g implemented as a GRU. This process is repeated for a fixed number of iterations. Finally, the output is computed from the final node state vectors using

a separate MLP, where the output of each error node corresponds to the error probability of the associated qubit.

Training data are regenerated at every epoch. Each epoch's training set combines two parts. The first is 15,000 error patterns whose per-sample physical error rate p is drawn independently from a uniform distribution $U(0, p_{\text{max}})$ with $p_{\text{max}} = 0.15$, excluding all-zero patterns. The second is the n single-qubit (weight-one) error patterns, one per qubit, which guarantee coverage of the low-weight regime. The main hyperparameters are node/edge feature dimension 128, number of iterations 24, and message-network size 256. We train for $n = 150$ epochs using the AdamW optimizer with learning rate 10^{-4} and a ReduceLROnPlateau scheduler. Decoding targets X errors only; by the CSS Z -symmetry of the codes the Z sector behaves identically, so we report the X -sector logical error rate.

V. RESULTS

A. Operational degeneracy across code families

We quantify the degeneracy of each code by measuring the operational degeneracy fraction f_{deg} defined in (5), evaluated at $p = 0.05$ with $n_{\text{samp}} = 3000$. As shown in Table 1, the two HGP codes considered here exhibit substantially higher f_{deg} than BB [[72, 12, 6]]. Among the HGP codes, f_{deg} increases from 1.1% for HGP [[58, 16, 3]] to 1.5% for HGP [[129, 28, 3]], indicating that the larger HGP code tends to exhibit stronger degeneracy in our experiments.

A larger f_{deg} indicates that minimum-weight representatives are non-unique more frequently, meaning that a decoder trained to fit a single error pattern under cross-entropy loss is more often exposed to degeneracy. The three codes therefore span a spectrum of degeneracy difficulty—BB [[72, 12, 6]] (low), HGP [[58, 16, 3]] (moderate), and HGP [[129, 28, 3]] (high)—which we use in the following sections as the axis along which the effect of SplitLoss is analyzed.

B. Effect of stabilizer-logical weighting

To isolate the effect of the two block weights in SplitLoss, we compare the LER improvement over GNN-CE for DegLoss and the four (α, β) configurations. Improvement is defined as the relative reduction in LER with respect to GNN-CE, so that a positive value denotes a gain over CE.

As shown in Fig. 2, for HGP [[129, 28]], the logical-weighted configuration $(\alpha, \beta) = (0.5, 1.5)$ yields the largest improvement at +26%, exceeding both DegLoss (+13%), which couples the two constraints in a single block, and the stabilizer-weighted (1.5, 0.5) (+11%). It also surpasses the logical-only extreme (0, 1) (+22%), indicating that retaining a small stabilizer contribution while emphasizing the logical term is more effective than discarding the stabilizer constraint entirely. The stabilizer-only setting (1, 0) reaches +17%, below the configurations that include a stronger logical term.

TABLE I. OPERATIONAL DEGENERACY FRACTION f_{deg}

Code	f_{deg} (%)
BB [[72, 12, 6]]	0.3
HGP [[58, 16, 3]]	1.1
HGP [[129, 28, 3]]	1.5

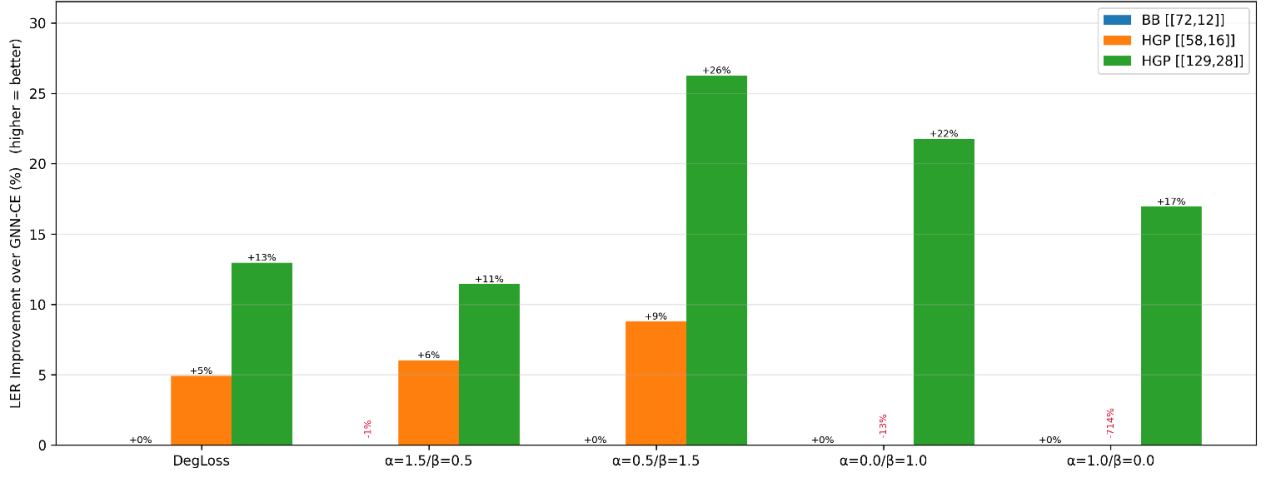


Fig. 2. LER improvement over GNN-CE (%) for the three codes (BB [[72,12]], HGP [[58,16]], HGP [[129,28]]), comparing GNN-DegLoss and SplitLoss under four (α, β) configurations. A higher value indicates a larger improvement over GNN-CE, while a negative value denotes performance below GNN-CE. For HGP [[129,28]], the code with the highest degeneracy, all configurations yield improvements, with the largest obtained at $(\alpha, \beta) = (0.5, 1.5)$, which weights the logical block more heavily.

The same ordering holds for HGP [[58,16]], where $(0.5, 1.5)$ again gives the largest improvement (+9%), above DegLoss (+5%) and $(1.5, 0.5)$ (+6%). The two extreme settings, by contrast, yield no improvement. For the logical-only setting $(\alpha, \beta) = (0, 1)$, the stabilizer constraint is not enforced at all, so syndrome matching is not directly imposed and the improvement becomes negative (-13%). For the stabilizer-only setting $(1, 0)$, the subcoset distinction is not explicitly enforced, so no logical-sector alignment is performed and the improvement drops sharply. This indicates that using both blocks together with an appropriate balance is essential.

For BB [[72,12]], whose operational degeneracy fraction is lowest, the improvement is 0% for most settings and only -1% for $(1.5, 0.5)$, so the change in loss function has essentially no effect. The effect of the degeneracy-aware terms is thus negligible for this code, and the advantage of weighting the logical block more heavily is most evident on the HGP codes, which exhibit higher degeneracy.

This is consistent with the analysis in Section III: decoding failure separates into syndrome mismatch and logical sector error, and since the latter occurs within the syndrome space, weighting the logical term more heavily aligns the objective directly with the evaluation criterion of selecting the correct subcoset. We therefore adopt $(\alpha, \beta) = (0.5, 1.5)$ as the representative GNN-SplitLoss model in the remaining analysis.

C. Logical Error Rate Comparison

We compare GNN-SplitLoss with the best configuration $(\alpha, \beta) = (0.5, 1.5)$ identified in Section V.B against BP+OSD(0), GNN-CE, and GNN-DegLoss. Fig. 3 shows the X-sector LER as a function of the physical error rate p_f for the three codes.

For BB [[72,12]], the four decoders nearly coincide across the measured range, and GNN-SplitLoss is essentially identical to GNN-CE. For this low-degeneracy code, SplitLoss yields no additional gain.

For HGP [[58,16]], BP+OSD(0) exhibits the highest LER throughout, whereas the GNN-based decoders substantially

outperform it. GNN-SplitLoss performs comparably to or slightly better than GNN-CE and GNN-DegLoss.

For HGP [[129,28]], the code with the highest degeneracy, GNN-SplitLoss achieves the lowest LER among all compared decoders at lower p_f , including BP+OSD(0), while requiring no post-processing. It also exhibits the largest improvement over GNN-CE among the codes considered.

Overall, GNN-SplitLoss is competitive with or superior to the baseline decoders without any post-processing, and its advantage becomes more pronounced for codes with higher degeneracy. This trend is consistent with the premise of this work that degeneracy is a primary cause of failure for cross-entropy-based learning and that a degeneracy-aware loss targets it directly. When the influence of degeneracy is weak, there is little room for correction and the additional gain is limited; when degeneracy is strong, the SplitLoss objective that reinforces logical-sector alignment becomes more effective. In other words, the LER improvement grows as the operational degeneracy fraction measured in Section V.A increases.

VI. CONCLUSION

The first contribution of this work is a decoder-level improvement. We introduce a logical-sector-aware loss into a GNN-based QLDPC decoder that operates in a single stage without post-processing, directly suppressing the logical component of the residual error. Specifically, we propose SplitLoss, which decomposes the degeneracy constraint into a stabilizer block and a logical block and assigns each an independent weight, α and β . Our experiments confirm that the logical-weighted configuration $(\alpha, \beta) = (0.5, 1.5)$ yields the largest LER improvement among the compared settings, including the degeneracy loss of the prior work.

The second contribution is an analysis of how code construction influences degeneracy. We quantify an operational degeneracy metric for code families built on different construction principles and analyze the relationship between this metric and the effectiveness of the degeneracy-aware loss.

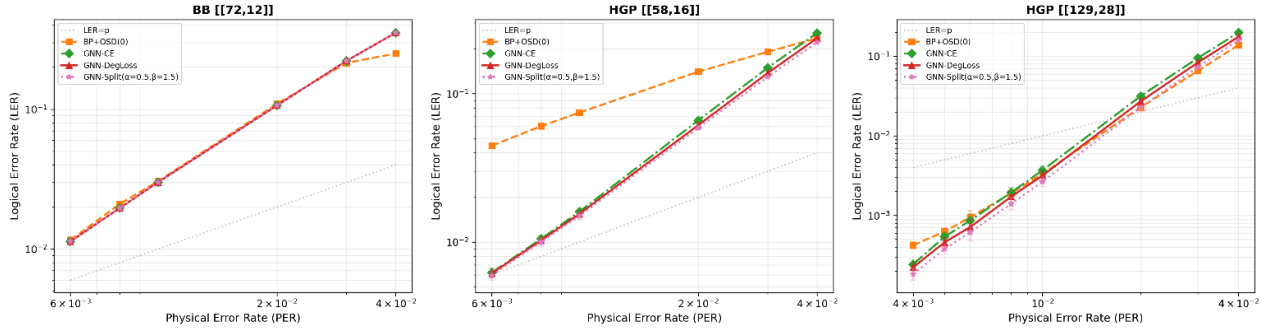


Fig. 3. X-sector logical error rate (LER) versus physical error rate p_f for the three codes (BB [[72,12]], HGP [[58,16]], HGP [[129,28]]). Each panel compares BP+OSD(0), GNN-CE, GNN-DegLoss, and GNN-SplitLoss with $(\alpha, \beta) = (0.5, 1.5)$. The dotted line indicates the $LER = p_f$ reference.

We find that the degeneracy-aware loss provides little benefit for BB [[72,12]], which exhibits a low operational degeneracy fraction. In contrast, the same loss is consistently beneficial for the HGP codes, where the metric is substantially larger. The magnitude of the LER improvement tends to increase with code degeneracy.

The proposed method offers three practical advantages. First, it operates in a single stage, without a post-processing step such as OSD. Second, by separating the degeneracy constraint into a stabilizer block and a logical block and assigning each an independent weight, it can isolate logical-sector errors during training. Third, the warm-up procedure allows the model to learn basic error patterns before gradually incorporating degeneracy constraints.

In future work, it will be important to evaluate SplitLoss under more realistic circuit-level noise models that include measurement errors and syndrome-extraction circuits. Another direction is to extend the analysis to a broader range of code families and code distances in order to further investigate the relationship between code degeneracy and

decoding performance. Such experiments would also enable evaluation of how well SplitLoss generalizes to larger-distance codes than those used during training.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00242396).

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00463, Development of a quantum repeater in optical fiber networks for quantum internet).

REFERENCES

- [1] Maan, Arshpreet Singh, and Alexandru Paler. "Machine learning message-passing for the scalable decoding of QLDPC codes." *npj Quantum Information* 11.1 (2025): 78.
- [2] Fuentes, Patricio, et al. "Degeneracy and its impact on the decoding of sparse quantum codes." *IEEE Access* 9 (2021): 89093-89119.