SCIENCE CHINA
Information Sciences

• Supplementary File •

Effective rate-adaptive reconciliation for CV-QKD using QC-MET-LDPC codes

Kun ZHANG[1], Jia HOU[1,2*], Xue-Qin JIANG[3,4,5*], Jisheng DAI[3], Peng HUANG[4,5,6] & Guihua ZENG[4,5,6]

[1]*School of Electronic and Information Engineering, Soochow University, Suzhou 215006, China;*
[2]*Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou 324000, China;*
[3]*College of Information Science and Technology, Donghua University, Shanghai 201620, China;*
[4]*Hefei National Laboratory, Hefei 230088, China;*
[5]*Shanghai Research Center for Quantum Sciences, Shanghai 201315, China;*
[6]*State Key Laboratory of Advanced Optical Communication Systems and Networks, and Center of Quantum Sensing and Information Processing (QSIP), Shanghai Jiao Tong University, Shanghai 200240, China*

Appendix A Construction of seed matrix

Multi-edge type (MET) low-density parity-check (LDPC) codes extend traditional LDPC codes by introducing multiple types of edges and node connections. This allows for a more flexible and optimized code design, where different edge types can have different degree distributions and connection properties. MET-LDPC codes with a Raptor-like structure not only maintain the high error-correcting performance of MET-LDPC codes but also can be used to design rate-adaptive codes. The goal is to enhance the performance and adaptability of the codes in various scenarios. Figure A1 shows the parity-check matrix of MET-LDPC codes with a Raptor-like structure.

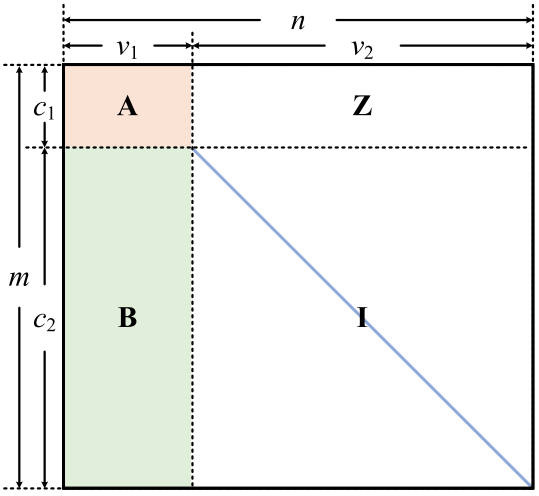

Figure A1 Parity-check matrix of MET-LDPC codes with a Raptor-like structure.

The structure of this parity-check matrix is composed of four concatenated submatrices: **A**, **B**, **Z**, and **I**. Here, **A** and **B** are sparse submatrices of size $c_1 \times v_1$ and $c_2 \times v_1$, respectively, **Z** is an all-zero matrix of size $c_1 \times v_2$, and **I** is an identity matrix of size $c_2 \times v_2$. Typically, **A** is a high-rate submatrix, and the concatenation of **B** and **I** allows **A** to extend to lower code rates. MET-LDPC codes with a Raptor-like structure are well-suited for long-distance continuous variable quantum key distribution (CV-QKD) systems with fluctuating signal-to-noise ratios (SNRs), due to their excellent error-correcting performance in the low code rate range and rate adaptive ability.

In MET-LDPC codes, different types of edges have independent degree distribution functions. Here, the degree refers to the number of edges connected to variable nodes or check nodes. To simplify mathematical expressions and computations,

* Corresponding author (email: houjia@suda.edu.cn, xqjiang@dhu.edu.cn)

degree distribution functions are typically represented in polynomial form. Each type of edge in MET-LDPC codes has corresponding degree distribution polynomials for variable nodes and check nodes. For instance, if there are two types of edges, at least four polynomials are required to describe the degree distributions.

From the perspective of nodes, a pair of polynomials is used to represent unpunctured ensemble of MET-LDPC codes, as defined below:

$$v(x) \triangleq \sum_{i=1}^{n_v} v_i \mathbf{x}^{\mathbf{d}_i}, \text{ and } \mu(x) \triangleq \sum_{i=1}^{n_c} \mu_i \mathbf{x}^{\mathbf{d}_i}, \tag{A1}$$

where $v(x)$ and $\mu(x)$ represent the degree distributions of variable nodes and check nodes, respectively. Here, n_v and n_c denote the number of terms in the degree distribution polynomials for variable nodes and check nodes, respectively. The coefficients v_i and μ_i represent the proportion of variable nodes and check nodes with degree $\mathbf{d}_i$, respectively. Furthermore, $\mathbf{x}^{\mathbf{d}_i} \triangleq \prod_{j=1}^{n_e} x_j^{d_{ij}}$, where n_e is the number of edge types in MET-LDPC codes, x_j represents the node for the edge type-j, and d_{ij} represents the degree of the i-th node for the edge type-j.

Each type of edge introduces a constraint: the degree of the variable nodes corresponding to this edge type must match the degree of the check nodes. The code rate of MET-LDPC codes can be expressed as:

$$R = \sum_{i=1}^{n_v} v_i \mathbf{x} - \sum_{i=1}^{n_c} \mu_i \mathbf{x}. \tag{A2}$$

In order to maintain consistent multi-edge degree distribution while accounting for varying lifting sizes [1], we adjust the code rate to $R = 0.01995$. The degree distribution of the MET-LDPC code with a code rate of $R = 0.01995$ is designed by modifying a code rate $R = 0.02$ multi-edge degree structure [2], which is shown in Table A1.

Table A1 Degree distribution of the MET-LDPC codes for $R = 0.01995$.

Code rate	Degree distribution
0.01995	$\nu(x) = 0.0225x_1^2x_2^{57} + 0.0175x_1^3x_2^{57} + 0.96x_3^1$
	$\mu(x) = 0.00005x_1^2 + 0.01065x_1^3 + 0.00935x_1^7 + 0.6x_2^2x_3^1 + 0.36x_2^3x_3^1$

Subsequently, the seed matrix of the MET-LDPC code with a Raptor-like structure is constructed using the degree distribution shown in Table A1. In Table A1, the variable nodes and check nodes connected by edge type-1 form submatrix $\mathbf{A}$ in Figure A1, which is represented by the degree distributions $v_{t1}(x) = 0.0225x_1^2 + 0.0175x_1^3$ and $\mu_{t1}(x) = 0.00005x_1^2 + 0.01065x_1^3 + 0.00935x_1^7$. Similarly, the variable nodes and check nodes connected by edge type-2 form submatrix $\mathbf{B}$, which is represented by the degree distributions $v_{t2}(x) = 0.04x_2^{57}$ and $\mu_{t2}(x) = 0.6x_2^2 + 0.36x_2^3$. The variable nodes and check nodes connected by edge type-3 form submatrix $\mathbf{I}$, which is also represented by the degree distributions $v_{t3}(x) = 0.96x_3^1$ and $\mu_{t3}(x) = 0.96x_3^1$.

We consider designing a seed matrix for a Raptor-like MET-LDPC code with a block length of $n = 20000$ and a code rate of 0.01995. Based on the degree distribution shown in Table A1, the submatrix $\mathbf{A}$ in the seed matrix has a size of 401×800, submatrix $\mathbf{B}$ has a size of 19200×800, submatrix $\mathbf{I}$ has a size of 19200×19200, and submatrix $\mathbf{Z}$ has a size of 401×19200. Since submatrix $\mathbf{Z}$ is an all-zero matrix and $\mathbf{I}$ is an identity matrix, we only need to consider how to construct submatrices $\mathbf{A}$ and $\mathbf{B}$. We use the progressive edge-growth (PEG) algorithm to construct submatrices $\mathbf{A}$ and $\mathbf{B}$ by following their specified degree distributions. We can obtain a seed matrix for the MET-LDPC code with a Raptor-like structure and a code rate of 0.01995 by assembling submatrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Z}$, and $\mathbf{I}$ according to the structure shown in Figure A1.

Appendix B Implementation of rate adaptivity

To maintain a high reconciliation efficiency in a CV-QKD system under fluctuating SNR values, it is essential to adopt a rate-adaptive LDPC code. For the MET-LDPC code with a Raptor-like structure, the rate adaptivity can be achieved by cutting or extending its seed matrix.

Figure B1 illustrates the principle of increasing the code rate by cutting the seed matrix of the MET-LDPC code with a Raptor-like structure. By cutting the same number of rows and columns from the bottom-right corner of the seed matrix along a defined cutting direction, the remaining part constitutes the target matrix for achieving the required code rate. Let n_{cut} be the cut number of the rows and columns, and the code rate R_{cut} of the resulting target matrix is calculated as:

$$R_{\text{cut}} = \frac{(n - n_{\text{cut}}) - (m - n_{\text{cut}})}{n - n_{\text{cut}}} = \frac{n - m}{n - n_{\text{cut}}}. \tag{B1}$$

The greater the cut number of rows and columns, the higher the code rate of the resulting target matrix. However, this also degrades the error-correcting performance of the MET-LDPC code, as increasing the cut number of rows and columns disrupts a larger proportion of the well-designed degree distribution. Therefore, when the target code rate significantly differs from that of the selected MET-LDPC code, it is advisable to choose MET-LDPC codes with a code rate closer to the target code rate, such as 0.05 or 0.1, and then perform code rate increase operations. Theoretically, this method of

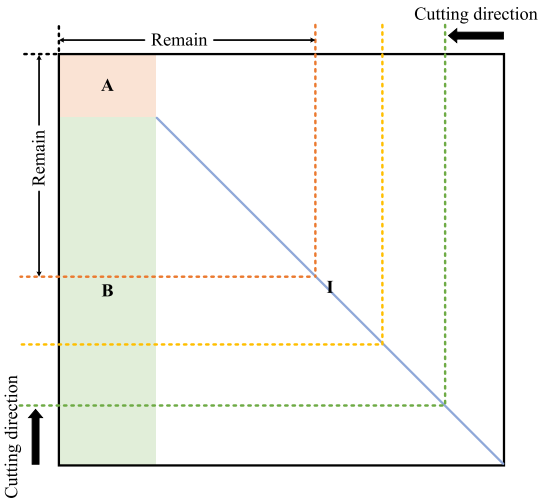

Figure B1 Schematic diagram of cutting a seed matrix of the MET-LDPC code with a Raptor-like structure.

increasing the code rate is sufficient to address the scenario where the modulation variance V_A remains constant while the SNR changes due to fluctuations in channel transmittance.

Figure B2 illustrates the principle of decreasing the code rate by extending the seed matrix of the MET-LDPC code with a Raptor-like structure. In contrast to the simple cut operation, the code rate reduction operation requires adding the same number of rows and columns to the seed matrix along the specified extending direction. Let n_{extend} be the added number of rows and columns, then the code rate R_{extend} of the target matrix obtained by extending the seed matrix can be calculated as:

$$R_{\text{extend}} = \frac{(n + n_{\text{extend}}) - (m + n_{\text{extend}})}{n + n_{\text{extend}}} = \frac{n - m}{n + n_{\text{extend}}}. \tag{B2}$$

The greater the added number of rows and columns, the lower the code rate of the target matrix.

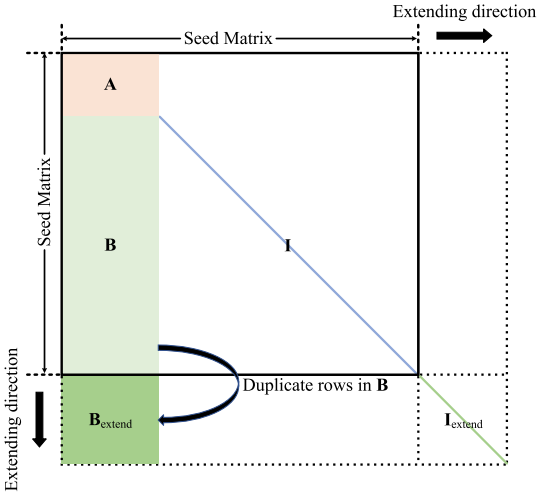

Figure B2 Schematic diagram of extending a seed matrix of the MET-LDPC code with a Raptor-like structure.

Given the unique structure of the Raptor-like design, the extended bottom-right $\mathbf{I}_{\text{extend}}$ remains an identity matrix, which can be viewed as an extension of the identity matrix $\mathbf{I}$. The elements below $\mathbf{I}$ and above $\mathbf{I}_{\text{extend}}$ are all zeros. Therefore, we only need to design the extended bottom-left $\mathbf{B}_{\text{extend}}$ to obtain a target matrix with a reduced code rate.

Unlike the conventional methods of constructing $\mathbf{B}_{\text{extend}}$ using random sampling Tanner graphs or the PEG algorithm according to a specified degree distribution, we consider directly utilizing the existing elements in the seed matrix to obtain $\mathbf{B}_{\text{extend}}$. In this way, the code rate reduction operation of the seed matrix can be achieved in real-time CV-QKD systems without additional computational resources or the time overhead associated with constructing the matrix.

Similar to the role of submatrices $\mathbf{B}$ and $\mathbf{I}$ in the seed matrix, the concatenation of $\mathbf{B}_{\text{extend}}$ and $\mathbf{I}_{\text{extend}}$ during the seed matrix extension process leads to a reduction in the code rate. Therefore, the elements in $\mathbf{B}_{\text{extend}}$ should be derived from $\mathbf{B}$ to maintain the consistency of the Raptor-like structure of the seed matrix. To ensure that the degree distribution of the seed matrix remains unchanged, elements in $\mathbf{B}$ cannot be removed to $\mathbf{B}_{\text{extend}}$. Therefore, we consider duplicating rows in

$\mathbf{B}$ into $\mathbf{B}_{\text{extend}}$. Specifically, whenever the seed matrix needs to be extended by a row, a random row from $\mathbf{B}$ is duplicated into $\mathbf{B}_{\text{extend}}$.

The duplication operation preserves the degree distribution characteristics without constructing a new matrix, significantly reducing the resources and time required for the code rate reduction preparation process. Directly duplicating rows may lead to cycles of length 4, which could affect the error-floor performance. However, this does not degrade the performance of MET-LDPC codes in CV-QKD systems with high target FER, such as $p_e = 0.5$, 0.1, 0.01.

Appendix C Utilization of QC-LDPC construction techniques

Quasi-cyclic (QC) LDPC codes are a class of LDPC codes with a QC structure, offering advantages in hardware efficiency, scalability, and performance. The structural properties of QC-LDPC codes enable efficient encoding and decoding algorithms, often utilizing hardware with parallel execution capabilities, such as the field-programmable gate array (FPGA) and the graphics processing unit (GPU), to enhance the throughput of information reconciliation in CV-QKD systems. Additionally, by selecting appropriate lifting sizes for the circulant permutation matrices (CPMs), QC-LDPC codes can be designed to meet the specific block length requirements of CV-QKD systems. Furthermore, QC-LDPC codes provide strong error-correcting capabilities across various code rates and block lengths.

Generally, the construction of a QC-LDPC code start with a small base matrix $\mathbf{H}_b$,

$$\mathbf{H}_b = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mn} \end{bmatrix}_{m \times n}, \tag{C1}$$

where each element represents a cyclic shift value of a corresponding CPM. These values indicate the number of positions that the 1 elements of the identity matrix are shifted to the right in each row. Then, the base matrix $\mathbf{H}_b$ is lifted to form the full parity-check matrix, where each element in the base matrix translates to a submatrix of size $z \times z$ in the parity-check matrix. Here, z is referred to as the lifting size of the QC-LDPC code. Specifically, if the cyclic shift value in the base matrix is equal to -1, it represents a zero submatrix of size $z \times z$.

For example, if the cyclic shift value p_{11} in the base matrix $\mathbf{H}_b$ of Eq. (C1) is equal to 1, it corresponds to a CPM obtained by shifting all the 1 elements of an identity matrix of size $z \times z$ one position to the right, represented as follows:

$$\mathbf{P}_{11} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}_{z \times z}. \tag{C2}$$

In this manner, each element p_{ij} in the base matrix $\mathbf{H}_b$, where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, is transformed into the corresponding CPM $\mathbf{P}_{ij}$. Thus, the structure of a QC-LDPC code can be expressed as:

$$\mathbf{H}_{\text{QC-LDPC}} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \mathbf{P}_{13} & \cdots & \mathbf{P}_{1n} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \mathbf{P}_{23} & \cdots & \mathbf{P}_{2n} \\ \mathbf{P}_{31} & \mathbf{P}_{32} & \mathbf{P}_{33} & \cdots & \mathbf{P}_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{m1} & \mathbf{P}_{m2} & \mathbf{P}_{m3} & \cdots & \mathbf{P}_{mn} \end{bmatrix}_{mz \times nz}, \tag{C3}$$

and its size is lifted from $m \times n$ for the base matrix to $mz \times nz$ for the parity-check matrix.

By utilizing QC-LDPC construction techniques, CV-QKD systems gain several advantages: 1) information reconciliation throughput can be significantly enhanced through the use of FPGA and GPU hardware; 2) block length requirements specific to CV-QKD systems are effectively addressed; and 3) superior error-correcting performance is consistently maintained.

Provided that the degree distribution is preserved, QC-LDPC construction techniques can be employed on the target matrix to derive the parity-check matrix of a QC-MET-LDPC code with the desired block length. Each element in the target matrix is replaced with its corresponding cyclic shift value; specifically, all 0 elements are replaced with -1, and each 1 element is replaced with a random number between 1 and z. In this way, the target matrix is transformed into the target cyclic shift matrix. Finally, each element in the target cyclic shift matrix is converted into the corresponding CPM based on its cyclic shift value to obtain the final parity-check matrix of the QC-MET-LDPC code.

While QC-LDPC construction techniques offer flexible block lengths for CV-QKD systems to satisfy system requirements for variable block lengths, they also present challenges such as high memory demands and long construction latency. In Section Appendix A, we design a seed matrix of size 19601×20000, where the number of 1 elements is 66750 based on its

degree distribution (corresponding to 66750 edges in its Tanner graph) . By replacing each 1 element with a corresponding 50×50 CPM and each 0 element with a 50×50 all-zero matrix, a QC-MET-LDPC code parity-check matrix of size 980050×1000000 is derived. Clearly, storing a matrix of this or larger size is challenging. For a system that requires real-time block length adjustments, the construction latency must be minimized.

The LLR-BP decoding transmits messages through the positions of 1 elements in the parity-check matrix of QC-MET-LDPC code. This is because 1 elements in the parity-check matrix indicate the connection between variable nodes and check nodes, which are the only positions involved in the message passing process. Conversely, positions of 0 elements indicate no direct connection, and thus do not participate in message passing. Given the sparse nature of the LDPC code (the parity-check matrix contains relatively few non-zero elements), we can significantly reduce memory requirements by recording only the indices of the 1 elements in the parity-check matrix, specifically their row and column positions, utilizing QC-LDPC construction techniques. In this manner, the required memory is considerably minimized. Obviously, storing just the 66750 and 3337500 row and column indices uses far less memory compared to keeping the full seed matrix of size 19601×20000 and the parity-check matrix of size 980050×1000000.

Similarly, the CPM can also be represented using the row and column indices of 1 elements. For instance, if the lifting size $z = 5$ as specified in Eq. (C2), there is no need to store a full 5×5 matrix:

$$\mathbf{P}_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}_{5\times 5} . \tag{C4}$$

Instead, it suffices to store the corresponding row indices $d_{\text{row}} = 1, 2, 3, 4, 5$ and column indices $d_{\text{col}} = 2, 3, 4, 5, 1$. Furthermore, leveraging the powerful parallel processing capabilities of GPUs, independent 1 elements in the base matrix can be simultaneously transformed into CPMs. This significantly reduces the construction delay, meeting the high-speed post-processing demands of real-time CV-QKD systems.

Appendix D Experimental results

In this section, we demonstrate the performance of QC-MET-LDPC codes in CV-QKD systems. The raw keys processed by our rate-adaptive reconciliation are obtained from the CV-QKD system described in Ref. [3], which employs a local local oscillator configuration. The error-correcting performance is analyzed based on the rate-adaptive reconciliation using LLR-BP decoding. To showcase the best performance, the dimension of the multidimensional reconciliation method is set to 8 [4].

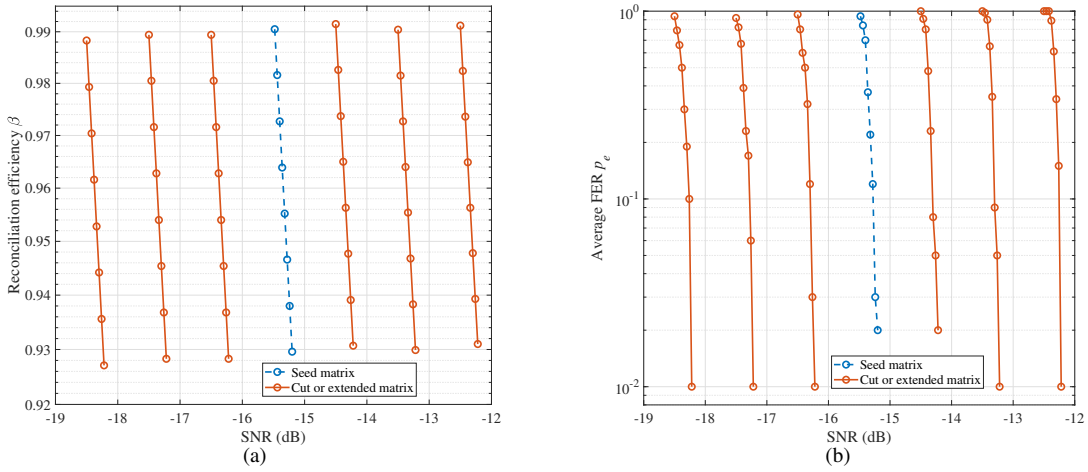

Figure D1 (a) Reconciliation efficiency and (b) FER performance of the seed matrix and its target matrices under different SNR values. To ensure similar block lengths for all matrices, the lifting sizes for the target matrices are set to 26, 32, 40, 63, 80, and 100, with higher code rates corresponding to larger lifting sizes, respectively. The lifting size for the seed matrix is set to 50. In both D1(a) and D1(b), the red solid lines from left to right represent target matrices with code rates of 0.01, 0.0126, 0.0158, 0.02494, 0.03122, 0.0391, and block lengths of 1037400, 1013120, 1010000, 1008000, 1022400, 1020000, respectively. The blue dashed line represents the seed matrix with a code rate of 0.01995 and a block length of 1000000. Each line from left to right in sub-figure D1(a), corresponds one-to-one with the lines from left to right in sub-figure D1(b). Furthermore, each point from top to bottom on a line in sub-figure D1(a), representing reconciliation efficiency β, corresponds exactly to a point from top to bottom on its corresponding line in sub-figure D1(b), representing average FER p_e. The maximum preset decoding iteration is 500.

Figure D1 illustrates two sub-figures, labeled D1(a) and D1(b). Sub-figure D1(a) shows the reconciliation efficiency, while Sub-figure D1(b) illustrates the FER performance of both the seed matrix and target matrices under varying SNR values.

We validate the performance of QC-MET-LDPC codes (including seed and target matrices) across code rates ranging from 0.01 to 0.0391. Notably, the proposed rate-adaptive scheme exhibits extensibility to both lower code rates (e.g., below 0.01) and higher code rates (e.g., above 0.0391). Conventional cut operations tend to degrade error-correcting performance as the cut number n_{cut} increases. While our proposed matrix extension method, which performs a rate reduction operation, does not suffer from degraded error-correcting performance with an increasing added number n_{extend}, it maintains the same error-floor ($p_e = 0.01$). Based on the rate-adaptive reconciliation using the LLR-BP decoding, the QC-MET-LDPC code achieves excellent error-correcting performance over a wide SNR range from -18.5 dB to -12.22 dB and can achieve a maximum reconciliation efficiency of higher than 98%.

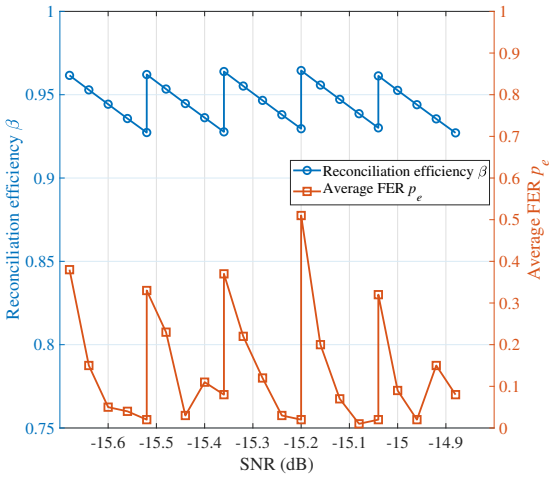

Figure D2 Reconciliation efficiency of the seed matrix with block length of 1000000 and its target matrices with block lengths of 1034880, 1039000, 1079400, 1044400 when the FER is below 0.5. The code rates of the target matrices are adjusted with variations in SNR to 0.0185, 0.0192, 0.0207, and 0.0214, respectively. The solid blue line with circle markers represents the reconciliation efficiency achieved by the seed matrix and its target matrices at the current SNR values, while the solid red line with square markers indicates their FER performance at the corresponding reconciliation efficiency values. The maximum preset decoding iteration is 500.

Table D1 GPU-accelerated decoding comparison with different block lengths and code rates.

Code rate	Block length	GPU threads	Construction delay (s)	β(%)	FER	Max iterations	Decoding speed (Mbits/s)
0.01	1037400	256	0.1907	98	0.78	400	1.0165
					0.92	200	1.9786
				96	0.4	400	1.0161
					0.56	200	1.9763
0.01995	1000000	256	0.1863	98	0.8	400	1.0758
					0.94	200	2.0907
				96	0.29	400	1.0758
					0.5	200	2.0916
0.0391	1020000	256	0.1836	96	0.7	400	1.0799
					0.91	200	2.0973
				94	0.06	400	1.08
					0.15	200	2.099

In Figure D2, to present the experimental data in detail, we fixed the SNR range between -15.68 dB and -14.88 dB. The FER performance exhibits greater fluctuations than the reconciliation efficiency in response to changes in SNR. This is because the performance of fixed-rate LDPC codes is notably sensitive to variations in SNR and primarily effective in error-correcting within a limited SNR range. To avoid situations where fluctuations in SNR could lead to a FER of 1, resulting in a secret key rate (SKR) of 0, the code rate is dynamically adjusted by modifying the size of the seed matrix, enabling effective error-correcting across different SNR ranges. Specifically, preset thresholds for reconciliation efficiency and FER are defined. When the actual reconciliation efficiency falls below the preset threshold, the code rate of the current MET-LDPC code is increased via the cut operation to restore the reconciliation efficiency to the threshold. Conversely,

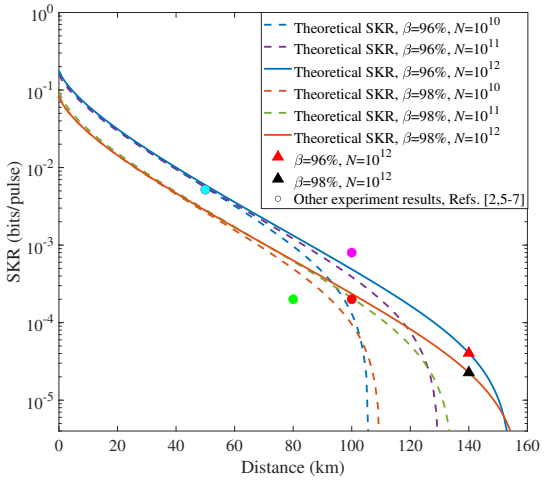

Figure D3 SKR vs. distance. The lines represent the finite-size theoretical SKRs for $N = 10^{10}$, 10^{11}, and 10^{12} with reconciliation efficiency values of $\beta = 96\%$ and 98%, respectively. The triangles represent the seed matrix and its target matrices at reconciliation efficiency values $\beta = 96\%$, and 98% with $N = 10^{12}$, respectively. The modulation variance V_A is maintained at its optimal value. The other experimental parameters are set as follows: excess noise $\varepsilon = 0.01$, electric noise $\nu_{el} = 0.015$, and homodyne detector efficiency $\eta_{det} = 0.6$. For comparison, we also present the experimental results in Refs. [2, 5–7].

when the actual FER exceeds the preset threshold, the code rate is reduced via the extend operation to enhance error correction capability and decrease FER. The rate-adaptive target matrix generated by cut or extend operations utilizes QC-LDPC construction techniques to derive a QC-MET-LDPC parity-check matrix with the required block length for error correction. For example, as shown in Figure D2, the preset reconciliation efficiency threshold is 92.5%, the FER threshold is 0.5, and the block length requirement is approximately 10^6. The results demonstrate that the proposed rate-adaptive reconciliation scheme consistently maintains performance above the reconciliation efficiency threshold and below the FER threshold.

The QC-LDPC construction delay and the decoding speed of three different code rates of QC-MET-LDPC codes on a GPU platform are tested, with results presented in Table D1. We utilize a single NVIDIA GeForce RTX 4060 Laptop GPU. GPU threads refer to the number of edges being processed in parallel during QC-LDPC construction and LLR-BP decoding. The efficient parallel processing capability of the GPU significantly reduces QC-LDPC construction delay and enhances decoding speed. Reducing the maximum number of iterations improves decoding speed but may decrease error-correcting performance. Notably, we ensure that the GPU remains fully utilized. A more powerful GPU platform can further reduce QC-LDPC construction delay and improve decoding speed.

Figure D3 shows the finite-size SKRs of the designed QC-MET-LDPC codes with respect to transmission distance without considering the light source repetition rate. For a given N, higher reconciliation efficiency often results in longer transmission distances and higher SKR for the CV-QKD system. With a constant reconciliation efficiency β, increasing N not only enhances the SKR of the CV-QKD system but also extends its transmission distance. Therefore, employing LDPC codes that allow for rate adaptivity and flexible block lengths to maintain high reconciliation efficiency under varying SNR values holds significant practical importance for CV-QKD systems. Compared to the experimental results in Ref. [2,5–7], the proposed adaptive reconciliation scheme, combined with QC-MET-LDPC codes, demonstrates a comparatively advantage. Overall, our work has positive implications for the practical implementation of CV-QKD systems.

References

1 Milicevic M, Feng C, Zhang L-M, et al. Quasi-cyclic multi-edge LDPC codes for long-distance quantum cryptography. npj Quantum Inform, 2018, 4: 21
2 Zhou C, Wang X, Zhang Z, et al. Rate compatible reconciliation for continuous-variable quantum key distribution using Raptor-like LDPC codes. Sci China-Phys Mech Astron, 2021, 64: 260311
3 Xu Y, Wang T, Liao X, et al. Robust continuous-variable quantum key distribution in the finite-size regime. Photonics Res, 2024, 12(11): 2549-2558
4 Leverrier A, Grosshans F, Grangier P. Finite-size analysis of a continuous-variable quantum key distribution. Phys Rev A, 2010, 81: 062343
5 Jouguet P, Kunz-Jacques S, Leverrier A, et al. Experimental demonstration of long-distance continuous-variable quantum key distribution. Nat Photon, 2013, 7: 378
6 Wang C, Huang D, Huang P, et al. 25 MHz clock continuous-variable quantum key distribution system over 50 km fiber channel. Sci Rep, 2015, 5: 14607
7 Huang D, Huang P, Lin D, et al. Long-distance continuous-variable quantum key distribution by controlling excess noise. Sci Rep, 2016, 6: 19201