New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

PAPER

Back-propagation neural network based optimization for high secret key rate continuous-variable quantum key distribution systems

Yang Liu[1], **Jiangliang Jin**[1], **Xue-Qin Jiang**[1,2,3,*], **Han Hai**[1], **Jisheng Dai**[1] and **Yinghua Xie**[1,*]

[1] College of Information and Intelligent Science, Donghua University, Shanghai 201620, People's Republic of China
[2] Shanghai Research Center for Quantum Sciences, Shanghai 201315, People's Republic of China
[3] Hefei National Laboratory, Hefei 230088, People's Republic of China
[*] Authors to whom any correspondence should be addressed.

E-mail: xqjiang@dhu.edu.cn and yh_xie@dhu.edu.cn

Keywords: continuous-variable quantum key distribution, back-propagation (BP) neural network, secure key rate, throughput of secure keys, parameter optimization

OPEN ACCESS

RECEIVED
18 December 2025

REVISED
24 March 2026

ACCEPTED FOR PUBLICATION
27 March 2026

PUBLISHED
8 April 2026

Abstract

In the continuous-variable quantum key distribution (CV-QKD) system, the secure key rate and the throughput of secure keys are important factors in measuring system quality. However, the secure key rate and the throughput of secure keys depend on various nonlinearly related parameters, which makes it difficult to effectively find the optimal parameters. In this paper, we introduce machine learning into the CV-QKD system and propose a parameter optimization scheme for the CV-QKD system based on the back-propagation (BP) neural network. The proposed scheme improves the secure key rate and throughput of secure keys by optimizing the code rate and the number of decoding iterations in the error correction process. Furthermore, under some reasonable assumptions, we prove that the optimization objective is a concave function of the optimization parameters under the BP neural network, which ensures that there exists a global optimal solution in a certain domain and provides a theoretical basis for the optimization based on BP neural networks. Simulation results show that compared with traditional optimization schemes, the proposed scheme can improve the secure key rate and transmission distance.

1. Introduction

As a crucial element of contemporary civilization, communication security makes it possible to share confidential information while at the same time reducing the likelihood of leaks that could occur. However, as technology for quantum computing continues to advance, conventional encryption techniques that rely on the intricacies of mathematical problems can no longer offer sufficient assurances of the security of information transmission. Under such circumstances, quantum key distribution (QKD) technology—grounded in the fundamental laws of quantum physics—has been developed. QKD exploits the fundamental principles of quantum mechanics to allow two users to establish keys that are secure in the information-theoretic sense [1–3] and encodes information on photons, enabling two distant legitimate users, usually referred to as Alice and Bob, to share the secret key in the presence of an eavesdropper (Eve) [4, 5]. Based on variations in the carriers used for information encoding and the approaches employed for measurement, QKD technology can be typically classified into two types: discrete-variable QKD (DV-QKD) and continuous-variable QKD (CV-QKD) [6, 7]. DV-QKD protocols encode information on discrete variables such as the phase or the polarization of single photons. The efficiency and speed of photon detectors in the single photon regime are the primary technical challenges and limits of

DV-QKD systems. In CV-QKD systems, information is encoded on continuous variables [8]. The modulation and decoding of the CV-QKD scenario can be effectively executed using contemporary technologies, enabling researchers to alleviate the limitations associated with DV-QKD. Consequently, investigating CV-QKD systems plays a crucial role in advancing the practical application and progression of quantum cryptography [9].

The CV-QKD system primarily comprises two procedures: quantum transmission [10] and post-processing. Within post-processing, the typical steps involve base sifting [11], parameter estimation [12], reconciliation [13, 14], and privacy amplification [15, 16]. Following the quantum information transmission procedure, noise leads to discrepancies in the raw key between Alice and Bob [17]. For this reason, the raw key transmitted between Alice and Bob needs to be corrected and this correction is accomplished through the use of reconciliation during the post-processing procedure [18]. The reconciliation stage, which is an essential component of the CV-QKD post-processing, makes use of strong error-correction codes in order to process information and generate the secret key. As a result, the overall effectiveness of the CV-QKD system is constrained by the performance of the reconciliation process.

The secret key rate and throughput of secret keys serve as crucial indicators for evaluating the performance of the CV-QKD system. These indicators are closely related to several system parameters involved in the process of information reconciliation (IR). For instance, parameters such as the code rate and the number of decoding iterations during the error correction process all have important effects on system performance. In recent decades, researchers have mostly adopted numerical analysis methods [19–21] to analyze the parameters in the CV-QKD system. Nevertheless, because of the nonlinear correlation that exists between the parameters, it is not effective to get the optimal values that are required to enhance the key rate by employing conventional techniques. As a result, it is extremely vital to discover a more efficient approach to parameter optimization.

Machine learning provides powerful and versatile tools for addressing complex challenges across various fields, with applications spanning critical areas such as parameter estimation, output prediction based on historical input data, high-dimensional data classification, and complex pattern recognition [22]. In CV-QKD systems, the optimization of the performance and practical security of QKD systems by integrating support vector regression models is first proposed in [23]. This innovative advancement not only greatly improves the accuracy of parameter estimation, but also eliminates the need for additional monitoring modules. Additionally, a low-complexity quantum k-nearest neighbor classifier utilizing machine learning techniques was proposed in [24], which improves the key rate of DV-QKD by forecasting the lossy discrete modulated coherent states on Bob's side. The proposed scheme points out a new direction for improving the performance of the QKD system. Moreover, in [25], the back-propagation (BP) neural network is applied to QKD based on the measurement-device-independent protocol to optimize parameters, which proves the effectiveness of the BP neural network. The BP neural network realizes automatic gradient calculation based on the chain rule. When used for parameter optimization, it eliminates the need for manual gradient calculation, which significantly reduces computational complexity and labor costs. Moreover, it can adapt to network structures of different depths [26]. Meanwhile, it can automatically adapt to parameter optimization requirements based on data characteristics without the need for manual presetting of complex mapping relationships. It has strong versatility and is suitable for parameter optimization in various scenarios, making it highly suitable for complex systems such as CV-QKD.

In this paper, we propose an approach based on BP neural network, which improves the secret key rate and the throughput of secret keys by optimizing the code rate and the number of decoding iterations in the error correction process.

This paper makes three main contributions. First, the trained BP neural network can efficiently derive the secret key rate $\tilde{g}(R)$ given the code rate R and the throughput of secret keys $\tilde{\varphi}(I_{\text{ave}})$ given the number of decoding iterations I_{ave}. Second, we prove that, under certain reasonable assumptions, if the secret key rate $\tilde{g}(R)$ and the throughput of secret keys $\tilde{\varphi}(I_{\text{ave}})$ are obtained from the BP neural network, then $\tilde{g}(R)$ is concave in R, and $\tilde{\varphi}(I_{\text{ave}})$ is concave in I_{ave}. This theoretical result allows us to efficiently find the maximum secret key rate K_{finite} and the maximum throughput of secret keys K_{secret} given the BP neural network. Third, compared with existing schemes, the simulation results show that the proposed approach achieves a higher secret key rate and throughput of secret keys at the same transmission distance.

The structure of the subsequent sections of this paper is arranged as follows: section 2 provides a concise overview of IR in CV-QKD systems, as well as the detailed workflow of the BP neural network. In section 3, we propose the BP neural network based optimization for CV-QKD systems. In section 4, a number of simulations are carried out in order to demonstrate the performance of the optimization that

is based on the BP neural network. In section 5, we summarize the primary findings of this paper and draw conclusions correlated with these findings.

2. Preliminaries

In this section, we first present a concise overview of IR within CV-QKD systems. After that, we provide a brief description of the parameters that affect the secret key rate and the throughput of secret keys. Finally, we elaborate on the detailed workflow of the BP neural network. Section 3 will apply this BP neural network to the optimization of parameters for CV-QKD systems.

2.1. IR

IR constitutes a critical step in CV-QKD systems, in which error correction is achieved while minimizing the leakage of secret key information, thereby enabling two communicating parties to obtain symmetric secret keys. IR is predominantly classified into two categories: direct reconciliation and reverse reconciliation. When the signal-to-noise ratio is low, reverse reconciliation is more effective than direct reconciliation at maintaining a certain secret key rate in fiber optic communications, and it also provides longer transmission distances. In reverse reconciliation, multidimensional reverse reconciliation is a mainstream reconciliation protocol. The core idea of the multi-dimensional reconciliation is to utilize the property that the sum of the squares of multiple Gaussian random variables follows the chi-square distribution, thereby enabling the reconciliation to overcome the difficulty of data being difficult to be correctly binarized. Both communicating parties, Alice and Bob, use the rotation mapping operation in the multi-dimensional space to map the quantum channel to a virtual channel. This method successfully converts the continuous Gaussian variables on the quantum channel into uniformly distributed binary variables on the virtual channel, thereby being able to use channel coding to solve the problem of secret key inconsistency [27, 28].

The schematic diagram of multidimensional reverse reconciliation [28–30] based CV-QKD systems is shown in figure 1. As is shown in figure 1, X and $Y = X + Z$ are the relevant Gaussian sequences, with Z being the noise of the quantum channel. Alice and Bob are responsible for selecting the dimension d, which is the dimensionality of multidimensional reconciliation, in order to split these sequences. Subsequently, Alice and Bob normalize the sequences X and Y to x and y, respectively. Utilizing a quantum random number generator, Bob produces a random binary sequence u, and uses it as a secret key. The encoder uses the error-correction code to encode the secret key u as c, and outputs the side information. From the encoded sequence c and normalized sequence y, one can calculate the mapping function $M(y, c)$ that is a part of the side information [28]. Bob transmits the side information to Alice through the use of a classical channel. Alice utilizes side information and normalized sequence x to obtain $\hat{u}$ by data mapping. In this paper, we adopt eight-dimensional reconciliation ($d = 8$) because it has the highest performance compared with other dimensions ($d = 1, 2, 4$) [28]. We use Raptor codes for error correction. Meanwhile, Raptor decoding adopts the belief propagation decoding algorithm.

2.2. Parameters of the IR

When the reconciliation frame error rate P_e and the finite-size effect are taken into account, the secret key rate is given by [31]

$$K_{\text{finite}} = \frac{n}{N}(1 - P_e)\left[\beta I_{AB} - \chi_{BE} - \Delta(n)\right], \tag{1}$$

where I_{AB} denotes the mutual information between Alice and Bob, χ_{BE} denotes the maximum of the Holevo information that Eve can obtain, N denotes the total number of data exchanged by Alice and Bob, n denotes the number of data used for secret key extraction, the other $m = N - n$ data is used for parameter estimation, and $\Delta(n)$ denotes the finite-size offset factor. According to equation (1), it is clear that the secret key rate mainly depends on the reconciliation efficiency β and the reconciliation frame error rate P_e. To maximize the secret key rate of CV-QKD, achieving high reconciliation efficiency while sustaining a low P_e is of paramount importance. The reconciliation efficiency β can be written as

$$\beta = \frac{R}{C}, \tag{2}$$

where C denotes the channel capacity, R denotes the error-correction code rate used for IR, given by

$$R = \frac{k}{r}, \tag{3}$$

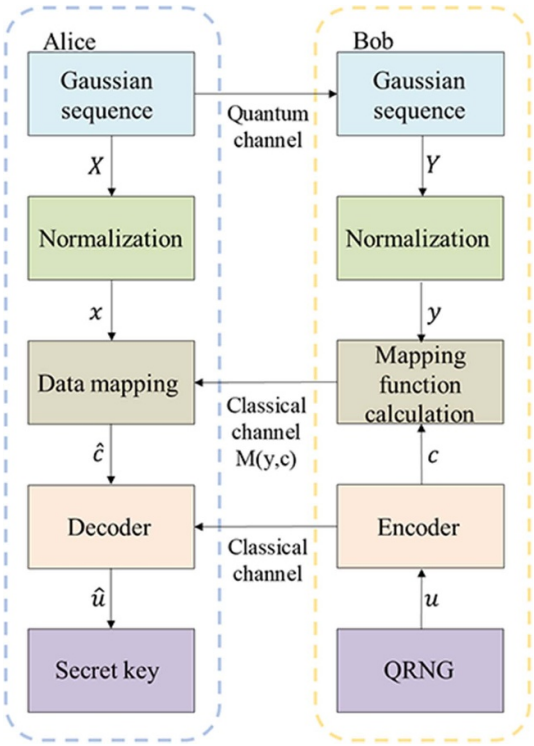

Figure 1. Schematic of postprocessing with reverse reconciliation in the CV-QKD system using Raptor codes. X and Y are two Gaussian sequences that are correlated with one another, and x and y are the normalized sequences, respectively. The mapping function that is conveyed from Bob to Alice is represented by the $M(y, c)$. Furthermore, u stands for the initial sequence that is produced by QRNG, and c refers to the result of the encoding of u. In addition, the sequence that exists prior to decoding is represented by the $\hat{c}$, whereas the symbol $\hat{u}$ is used to describe the outcome of the decoding process.

where k denotes the number of information bits, and r denotes the code length. When R increases, the reconciliation efficiency increases. At the same time, the redundancy decreases, resulting in degraded error correction performance and an increased P_e. According to equation (1), an increase in P_e will lead to a decrease in the secret key rate, while an increase in the reconciliation efficiency will result in an increase in the secret key rate. Therefore, it is essential to optimize the code rate R in the error correction process of IR.

The throughput of secret keys can be expressed as [32–34]

$$K_{\text{secret}} = \frac{nR(1-P_e)}{T_{\text{IR}}}, \tag{4}$$

where T_{IR} is the processing time of IR. According to equation (4), the throughput of secret keys is primarily determined by the processing time of IR and the reconciliation frame error rate P_e. To maximize the throughput of secret keys of CV-QKD, it is essential to minimize the processing time of IR and P_e.

As previously described, IR contains two processes: multidimensional reconciliation and error correction with Raptor codes. The computational complexity of the multidimensional reconciliation is low, which can achieve high speed on central processing unit (CPU). However, for decoding with CPU, the speed of the error correction process will be quite slow [35]. Therefore, the main reason for the excessively long IR time is the long decoding time. The decoding time of Raptor can be expressed as

$$T_{\text{dec}} = \tau I_{\text{ave}}, \tag{5}$$

where τ denotes the average decoding time and I_{ave} denotes the number of decoding iterations. Increasing I_{ave} usually improves the reliability of decoding and reduces the P_e because it makes it easier for the decoder to reach the convergence state [9]. However, this also increases the computational complexity and decoding time T_{dec}. As T_{dec} increases, T_{IR} increases accordingly. According to equation (4), an increase in T_{IR} will lead to a decrease in the throughput of secret keys, while a decrease in P_e will result in an increase in the throughput of secret keys. Therefore, during the error correction process of IR, optimizing the number of decoding iterations I_{ave} is of crucial importance for enhancing the throughput of secret keys.

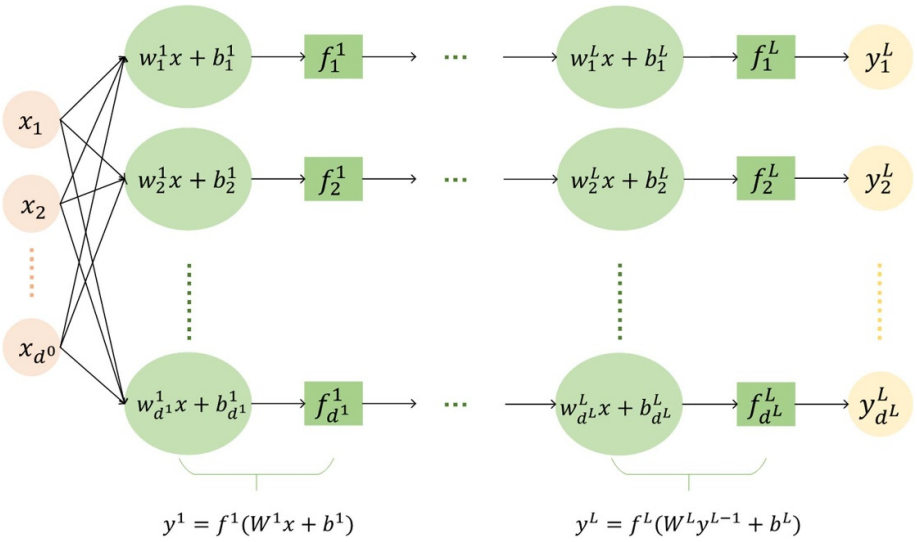

Figure 2. Structure of BP neural network.

2.3. BP neural network

The BP neural network [36, 37] is a type of artificial neural network that consists of multiple layers and is trained using the BP algorithm. It is a feed-forward neural network. Two different parts are generally included in the self-learning process of the BP neural network. The initial step is the forward transfer of information. The other is the reverse transfer of error, which occurs between the expected output and the output that is actually produced. The architecture of the BP neural network comprises three components: the input layer, the hidden layer [38], and the output layer. Figure 2 illustrates the L-layer BP neural network with d^0 -dimensional inputs and the neural network output represented as $y = y^{\mathrm{L}}$. At the first layer, we first apply a linear transformation with weight W^1 and threshold b^1 to the input x, followed by a non-linear transformation with the activation function f^1. Consequently, the output y^1 can be expressed as

$$y^1 = f^1\left(W^1x + b^1\right) = \begin{pmatrix} f_1^1\left(w_1^1x + b_{d^1}^1\right) \\ \vdots \\ f_{d^1}^1\left(w_{d^1}^1x + b_{d^1}^1\right) \end{pmatrix}, \tag{6}$$

where d^1 denotes the dimension of the output vector, and for $\forall i \in \left[1, d^1\right] \cap \mathbb{Z}^+$, f_i^1 denotes the ith activation function at the first layer, w_i^1 denotes the ith weight at the first layer, b_i^1 denotes the ith threshold at the first layer.

For $\forall l \in [2, L] \cap \mathbb{Z}^+$, y^{l-1} is the output of $(l-1)$th layer, and the input of the lth layer. At the lth layer, y^l is obtained from y^{l-1} through a linear transformation including weight W^l and threshold b^l, followed by a non-linear transformation with the activation function f^l. Consequently, the output y^l can be expressed as

$$y^l = f^l\left(W^ly^{l-1} + b^l\right) = \begin{pmatrix} f_1^l\left(w_1^ly^{l-1} + b_{d^1}^l\right) \\ \vdots \\ f_{d^l}^l\left(w_{d^l}^ly^{l-1} + b_{d^l}^l\right) \end{pmatrix}, \tag{7}$$

where d^l denotes the dimension of the output vector, and for $\forall j \in \left[1, d^l\right] \cap \mathbb{Z}^+$, f_j^l denotes the jth activation function at the lth layer, w_j^l denotes the jth weight at the lth layer, b_j^l denotes the jth threshold at the lth layer.

To derive the subsequent theoretical results, we introduce the following assumption.

Assumption 1. In the BP neural network, all the activation functions $f_j^l, \forall l, j$ are supposed to be monotonic concave functions and second-order differentiable in the positive domain.

Remark 1. The majority of commonly used activation functions, including the ReLU function, the hyperbolic tangent function, and the sigmoid function are monotonic concave functions and second-order differentiable in the positive domain, hence the assumption 1 can hold.

Assumption 2. In the BP neural network, we assume that for any $\forall l \in [2, L] \cap \mathbb{Z}^+$, b^l is a positive vector, with each element being a positive number, and W^l is a positive matrix, with each element being a positive number.

Assuming that the above assumptions hold, we will obtain the following theorem.

Theorem 1. *Assuming that assumptions 1 and 2 hold. If x is a positive vector and y(x) is derived from the BP neural network, then y(x) is a concave function of x.*

Assuming that the above assumptions hold, we will prove below that $y(x)$ is a concave function of x if $y(x)$ is acquired by the BP neural network in figure 2 and x is a positive vector.

The proof of theorem 1 is provided in appendix A.

Remark 2. It should be noted that theorem 1 holds only when it is assumed that 1 and 2 hold and x is a positive vector. Moreover, in continuous and discrete optimization problems, the local optimality of concave/-convex analysis ensures global optimality [39, 40]. As a result, the concavity result in theorem 1 enables us to find the global optimal solution of $y(x)$, ensuring that any local optimal solution obtained by the BP neural network is essentially the global optimal solution within the feasible region. This lays a crucial theoretical foundation for the BP neural network optimization proposed in the CV-QKD system.

3. BP neural network based optimization for CV-QKD systems

To maximize the secret key rate and throughput of secret keys, selecting the appropriate error-correction code rate and the number of decoding iterations is essential. The BP neural network is applicable to CV-QKD systems for accurately finding the optimal secret key rate and throughput of the secret keys. The secret key rate K_{finite} is a function of error-correction code rate R, and the throughput of secret keys K_{secret} is a function of the number of decoding iterations I_{ave}. To simplify, the relationship between the secret key rate K_{finite} and the error-correction code rate R is expressed mathematically as follows [22]:

$$K_{\text{finite}} = g(R). \tag{8}$$

The relationship of the throughput of secret keys K_{secret} and the number of decoding iterations I_{ave} is expressed mathematically as follows:

$$K_{\text{secret}} = \varphi(I_{\text{ave}}). \tag{9}$$

The purpose of parameter optimization is to find the most suitable code rate to maximize the secret key rate K_{finite}, then based on the optimal code rate, find the most suitable number of decoding iterations to maximize the throughput of secret keys K_{secret} , that are

$$R^{\text{opt}} = \arg\max_{R \in G} [K_{\text{finite}} = g(R)], \tag{10}$$

$$I_{\text{ave}}^{\text{opt}} = \arg\max_{I_{\text{ave}} \in \Phi} [K_{\text{secret}} = \varphi(I_{\text{ave}})], \tag{11}$$

where G denotes the domain of R, and Φ denotes the domain of I_{ave}.

The scheme of parameter optimization based on BP neural network is illustrated in figure 3. As shown in figure 3, first, we simulate in the CV-QKD system according to the sampling range and step size of the code rate to obtain the relevant training data about the code rate and the secret key rate. The obtained data is divided into training sets and testing sets for training the BP neural network to learn the mapping relationship between parameter R and secret key rate K_{finite}. Once this mapping relationship is obtained, we use the bisection method to find the optimal code rate and the corresponding optimal secret key rate. Then, we input the optimal code rate into the CV-QKD system and simulate based on the sampling range and step size of the number of decoding iterations, obtaining the relevant training data about the number of decoding iterations and the throughput of secret keys. The data is divided into training sets and testing sets for training the BP neural network to learn the mapping relationship between the parameter I_{ave} and the throughput of secret keys K_{secret}. Once this mapping relationship is obtained, we use the bisection method to find the optimal decoding iteration number and the corresponding optimal key throughput.

When training the BP neural network, the error-correction coding rate is fed into an input layer of a single neuron initially. After that, the input layer is linked to a hidden layer that has six neurons.

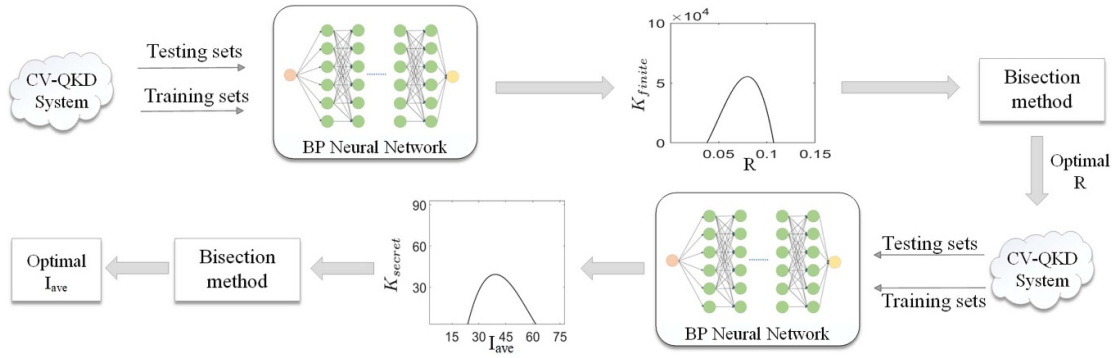

Figure 3. Parameter optimization diagram of CV-QKD based on BP neural network.

To accelerate model convergence and improve training stability, normalized experimental parameters is essential [41]. The parameters from the CV-QKD system are normalized with a normalization interval of $(-1,1)$ and the normalized parameters are used in the BP neural network. In order to generate more scientific and reliable training sets $(R^{\text{train}}, K^{\text{train}}_{\text{finite}})$ and testing sets $(R^{\text{test}}, K^{\text{test}}_{\text{finite}})$, the code rate training sets R^{train} are derived from the CV-QKD system and the corresponding secret key rate training sets $K^{\text{train}}_{\text{finite}}$ are calculated. The data different from the training sets $(R^{\text{train}}, K^{\text{train}}_{\text{finite}})$ are used as the testing sets $(R^{\text{test}}, K^{\text{test}}_{\text{finite}})$.

The training sets $(R^{\text{train}}, K^{\text{train}}_{\text{finite}})$ are used to train the BP neural network. Following training, the testing sets $(R^{\text{test}}, K^{\text{test}}_{\text{finite}})$ are used to assess the BP neural network. The secret key rate K_{finite} is predicted by feeding the testing sets R^{test} into the trained BP neural network. Consequently, the relationship between code rate R and secret key rate K_{finite} can be analyzed. After that, the optimal code rate can be obtained by bisection method. The optimal code rate is input into the CV-QKD system to get the iteration number and throughput parameters, and the parameters are normalized to get the training sets $(I^{\text{train}}_{\text{ave}}, K^{\text{train}}_{\text{secret}})$ and testing sets $(I^{\text{test}}_{\text{ave}}, K^{\text{test}}_{\text{secret}})$. The training sets $(I^{\text{train}}_{\text{ave}}, K^{\text{train}}_{\text{secret}})$ are used to train the BP neural network. Following training, the testing sets $(I^{\text{test}}_{\text{ave}}, K^{\text{test}}_{\text{secret}})$ are used to assess the BP neural network. The throughput of secret keys K_{secret} is predicted by feeding $I^{\text{test}}_{\text{ave}}$ into the trained BP neural network. Consequently, the relationship between the number of iterations I_{ave} and the throughput of secret keys K_{secret} can be analyzed. After that, the optimal number of iterations can be obtained by bisection method.

The BP neural network depicted in figure 3 is a specific instance of figure 2, where assumption 2 holds, the code rate R and number of iterations I_{ave} are the inputs and all the activation functions, f^l_j, $\forall l, j$, are chosen as hyperbolic tangent (tanh) functions. For notational simplicity, in propositions 1 and 2 and their proofs, we will use $\tilde{g}(R)$ and $\tilde{\varphi}(I_{\text{ave}})$ to emphasize that K and K_{secret} are determined by R and I_{ave} via the neural networks in figure 2. It is easy to show that $\tilde{g}(R)$ is a concave function of R and $\tilde{\varphi}(I_{\text{ave}})$ is a concave function of I_{ave} because R and I_{ave} are both positive by definition and the *tanh* function is a monotonic concave function that is second-order differentiable in the positive domain.

Proposition 1. *Suppose that assumptions 1 and 2 hold and all the activation functions, f^l_j, $\forall l, j$ are hyperbolic tangent (tanh) functions. If $\tilde{g}(R)$ and $\tilde{\varphi}(I_{\text{ave}})$ are derived from a BP neural network where the code rate satisfies $R > 0$ and the number of decoding iterations satisfies $I_{\text{ave}} > 0$, then $\tilde{g}(R)$ is a concave function of R and $\tilde{\varphi}(I_{\text{ave}})$ is a concave function of I_{ave}.*

Proposition 2. *Suppose that $\tilde{\varphi}(I_{\text{ave}})$ is derived from a BP neural network where the number of decoding iterations satisfies $I_{\text{ave}} \in \mathbb{R}^+$ and $\tilde{\varphi}(I_{\text{ave}})$ is a concave function of I_{ave}. If the number of decoding iterations satisfies $I_{\text{ave}} \in \mathbb{Z}^+$, then $\tilde{\varphi}(I_{\text{ave}})$ is a discrete concave function I_{ave}.*

The proof of proposition 1 is provided in appendix B, and the proof of proposition 2 is provided in appendix C.

Propositions 1 and 2 show that the bisection method can be used to determine the optimal code rate R and its corresponding maximum secret key rate K_{finite}. The optimal number of iterations I_{ave} and its corresponding throughput of secret keys can also be determined at the optimal code rate by bisection method. The procedure involves the following steps: first, initialize a search range for the code rate R and the number of iterations I_{ave}. Then, compute the midpoint of this range and evaluate the secret key rate K_{finite} and the throughput of secret keys K_{secret} at the midpoint and its adjacent points. Next, narrow the search range by selecting the subinterval with the higher value, and iterate this process until convergence to the target value.

Remark 3. It is worth noting that since the number of decoding iterations is a positive integer, if the target number of decoding iterations obtained after the bisection method is not a integer, the throughput values of its left and right adjacent points are compared. The number of decoding iterations corresponding to the larger throughput of secret keys is the optimal number of iterations in the final output.

4. Simulation results

In this section, assumptions 1 and 2 always hold and K_{finite} is a concave function of R and K_{secret} is a concave function of I_{ave} according to proposition 1. On this basis, we specify the system parameters for CV-QKD, namely the channel excess noise ε, the electrical noise v_{el}, the detection efficiency η, and the quantum channel transmission efficiency T. The values of these parameters are fixed at $\varepsilon = 0.01$, $v_{\text{el}} = 0.015$, $\eta = 0.6$, which are typical values in CV-QKD. In addition, the quantum channel transmission efficiency is calculated by $T = 10^{\alpha l/10}$. The training sets for the code rate R and the number of decoding iterations I_{ave} are both generated by the CV-QKD system simulation. The value range of R is $[0.01, 0.9]$ with a sampling step size of 0.001, and the value range of I_{ave} is $[1, 300]$ with a sampling step size of 1. The training set and test set are randomly divided in a ratio of 7:3. The training settings of the BP neural network are presented in table 1.

The core idea of this paper is to introduce the BP neural network into the CV-QKD system for parameter optimization. Specifically, this paper first optimizes the code rate used in the error correction process and then further optimizes the number of decoding iterations in the error correction process after obtaining the optimal code rate. The proposed approach is compared with traditional CV-QKD optimization approaches based on MR-Raptor [42] and Raptor codes [43] in order to assess its performance. Figure 4 illustrates the secret key rate of a CV-QKD system without considering the throughput of secret keys. To make a fair comparison, all approaches use the same number of decoding iterations. The black solid line represents the proposed approach, where the code rate is derived from the BP neural network. The red dash-dot line and green dashed line represent the traditional CV-QKD optimization approaches based on MR-Raptor codes [42] and Raptor codes [43], respectively. As shown in figure 4, the black solid line is always positioned above the red dash-dot line and green dashed line. This indicates that the proposed approach performs better than traditional CV-QKD optimization approaches based on MR-Raptor codes and Raptor codes in terms of the secret key rate and transmission distances.

Figure 5 illustrates the secret key rate after considering the throughput of secret keys and how the secret key rate varies with transmission distance in the CV-QKD system. Because the number of decoding iterations mainly affects system performance through the throughput of secret keys, all the approaches calculate the throughput of secret keys using equation (4) and take it into account. The blue solid line represents the proposed approach, where the code rate and the number of decoding iterations are derived from the BP neural network. The red dash-dot line and green dashed line represent the traditional CV-QKD optimization approaches based on MR-Raptor codes [42] and Raptor codes [43], respectively. As shown in figure 5, the blue solid line is always positioned above the red dash-dot line and green dashed line. This indicates that, when compared to the traditional CV-QKD optimization approaches, the proposed approach outperforms them in terms of both key rate and transmission distance. The black dotted line represents the proposed approach, where the code rate is derived from the BP neural network but the number of decoding iterations is consistent with the traditional CV-QKD optimization approaches based on MR-Raptor codes [42] and Raptor codes [43]. The blue solid line in figure 5 is always positioned above the black dotted line. This indicates that optimizing both the code rate and the number of decoding iterations at the same time is more effective at improving the secret key rate and the distance over which it is transmitted than optimizing solely the code rate.

Table 1. The training settings of the BP based approach.

Items	Settings
Training goal	0.000 01
Training method	Gradient descent
Network learning rate	0.01
Number of hidden layers	1
Maximum number of training	1000

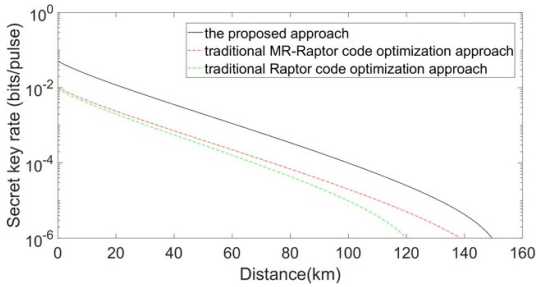

Figure 4. Secret key rate versus transmission distance for the CV-QKD system at a signal-to-noise ratio of −8 dB. All approaches use the same number of decoding iterations and do not take into account the impact of the throughput of secret keys. The black solid line represents the proposed approach, where the code rate is the optimal code rate derived by the trained BP neural network, and the number of decoding iterations is consistent with the traditional CV-QKD optimization approaches based on MR-Raptor codes and Raptor codes. The total number of data used for key extraction and parameter estimation, denoted by N, is 10^{12}.

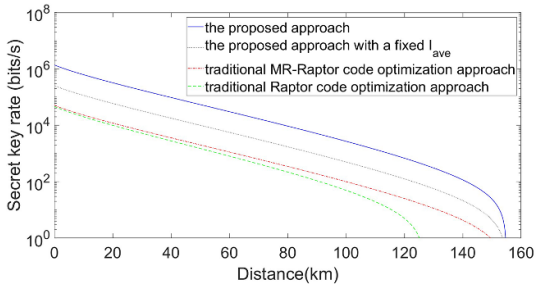

Figure 5. Secret key rate versus transmission distance for the CV-QKD system considering the throughput of secret keys. All the approaches in the figure calculate the throughput of secret keys based on equation (4) and consider its impact on the transmission distance and the secret key rate. The blue solid line represents the proposed approach, where the code rate and the number of decoding iterations are derived by the trained BP neural network. The black dotted line represents the proposed approach, where the code rate is the optimal code rate derived by the trained BP neural network but the number of decoding iterations is consistent with the traditional CV-QKD optimization approaches. The signal-to-noise ratio is −8 dB.

5. Conclusion

In this paper, we examined the factors affecting the secret key rate and the throughput of secret keys of the CV-QKD system and proposed a parameter optimization scheme. In the past, parameters were typically selected based on experience or numerical analysis. However, these two approaches cannot efficiently obtain the optimal parameters. We introduced the BP neural network into the CV-QKD system, which can efficiently and quickly find the optimal parameters to achieve the maximum secret key rate and the maximum throughput of secret keys. Under certain assumptions, we further proved that $\tilde{g}(R)$ is concave in R and $\tilde{\varphi}(I_{\text{ave}})$ is concave in I_{ave} when derived from the BP neural network. This theory allowed us to use the bisection method to obtain the optimal R and I_{ave} and the corresponding maximum secret key rate and throughput of secret key. Simulation results indicated that the proposed scheme had a longer transmission distance and a higher secret key rate and throughput of secret keys for the same transmission distance, providing new opportunities for the development of efficient CV-QKD systems.

Acknowledgments

This work is supported by Innovation Program for Quantum Science and Technology (Grant no. 2021ZD0300703); Shanghai Municipal Science and Technology Major Project (Grant no. 2019SHZDZX01); the State Key Laboratory of Advanced Optical Communication Systems and Networks(Grant no.2024GZKF006), China.

Data availability statement

The data cannot be made publicly available upon publication because they contain sensitive personal information. The data that support the findings of this study are available upon reasonable request from the authors.

Conflict of interest

The authors declare no conflicts of interest.

Appendix A. Proof of theorem 1

This appendix provides the proof that the function $y(x)$ generated by the BP neural network is concave in x when x is a positive vector. The proof of theorem 1 will be demonstrated by using of mathematical induction under assumptions 1 and 2 , which includes the following two main steps: step 1 will prove that y^1 is a concave function of x; step 2 will prove that for any $l \geqslant 2$, if $y^{l-1}$is a concave function of x, then y^l is a concave function of x.

Step 1 The BP neural network that has been designed for d^0-dimensional inputs and has L layers is illustrated in figure 2. The output of this neural network is represented by the $y = y^l$. At the first layer, the input x is first applied a linear transformation including weight W^1 and threshold b^1, followed by a non-linear transformation with the activation function f^1. For the sake of convenience, the map form x to y^1 is symbolized by the $H^1(X)$. Consequently, $H^1(X)$ can be written as

$$\begin{aligned} H^1(x) = y^1 &= f^1\left(W^1 x + b^1\right) \\ &= \begin{pmatrix} f_1^1\left(w_1^1 x + b_1^1\right) \\ \vdots \\ f_{d^1}^1\left(w_{d^1}^1 x + b_{d^1}^1\right) \end{pmatrix}. \end{aligned} \tag{12}$$

The first-order derivative of $H^1(x)$ with respect to x is:

$$\begin{aligned} \frac{\mathrm{d}H^1(x)}{\mathrm{d}x} = y^{1'} &= f^{1'}\left(W^1 x + b^1\right) \cdot W^1 \\ &= \begin{pmatrix} f_1^{1'}\left(w_1^1 x + b_1^1\right) \cdot w_1^1 \\ \vdots \\ f_{d^1}^{1'}\left(w_{d^1}^1 x + b_{d^1}^1\right) \cdot w_{d^1}^1 \end{pmatrix}. \end{aligned} \tag{13}$$

The second-order derivative of $H^1(x)$ with respect to x is:

$$\begin{aligned} \frac{\mathrm{d}^2 H^1(x)}{\mathrm{d}^2 x} = y^{1'} &= f^{1''}\left(W^1 x + b\right) \cdot W^{1^2} \\ &= \begin{pmatrix} f_1^{1''}\left(w_1^1 x + b_1^1\right) \cdot w_1^1 \cdot w_1^1 \\ \vdots \\ f_{d^1}^{1''}\left(w_{d^1}^1 x + b_{d^1}^1\right) \cdot w_{d^1}^1 \cdot w_{d^1}^1 \end{pmatrix}, \end{aligned} \tag{14}$$

where d^1 denotes the dimension of output vector y^1, and for $\forall i \in \left[1, d^1\right] \cap \mathbb{Z}^+$, f_i^1 denotes the ith activation function at the first layer, w_i^1 denotes the ith weight at the first layer, and b_i^1 denotes the ith threshold at the first layer.

According to equation (14), the sign of the second-order derivative depends on $f^{1''}\left(W^1 x + b^1\right)$ and W^{1^2}. Suppose that assumption 1 holds, by the definition of a concave function, we have $f^{1''}(x) \leqslant 0, \forall x \in \mathbb{R}^+$. And because $W^{1^2} > 0$, so we have

$$\frac{\mathrm{d}^2 H^1(x)}{\mathrm{d}^2 x} \leqslant 0, \forall x \in \mathbb{R}^+. \tag{15}$$

The aforementioned inequality holds for all positive vectors x, which provides evidence that $H^1(x)$ is a concave function of x. In accordance with equation (12), this also demonstrates that y^1 is a concave function of x.

Step 2 For $\forall l \in [2, L] \cap \mathbb{Z}^+$, y^{l-1} is the output of $(l-1)$th layer, and the input of the lth layer. At the lth layer, y^l is obtained from y^{l-1} through a linear transformation involving the weight matrix W^l and threshold b^l, followed by a nonlinear transformation using the function f^l. Consequently, $H^l(x)$ can be expressed as

$$\begin{aligned} H^l(x) = y^l &= f^l\left(W^l y^{l-1} + b^l\right) \\ &= \begin{pmatrix} f_1^l\left(w_1^l y^{l-1} + b_1^l\right) \\ \vdots \\ f_{d^l}^l\left(w_{d^l}^l y^{l-1} + b_{d^l}^l\right) \end{pmatrix} \\ &= \begin{pmatrix} f_1^l\left(w_1^l H^{l-1}(x) + b_1^l\right) \\ \vdots \\ f_{d^l}^l\left(w_{d^l}^l H^{l-1}(x) + b_{d^l}^l\right) \end{pmatrix}. \end{aligned} \tag{16}$$

The first-order derivative of $H^l(x)$ with respect to x is:

$$\begin{aligned} H^{l'}(x) = y^{l'} &= f^{l'}\left(W^l y^{l-1} + b^l\right) \cdot W^l \cdot y^{l-1'} \\ &= f^{l'}\left(W^l y^{l-1} + b^l\right) \cdot W^l \cdot H^{l-1'}(x). \end{aligned} \tag{17}$$

According to equation (17), the sign of the first-order derivatives are determined by $f^{l'}(W^l y^{l-1} + b^l)$, W^l, and $H^{l-1'}(x)$. Since $f(x)$ is a monotonic function, $f^{l'}(x)$ only exists in two cases: $f^{l'}(x) \geqslant 0, \forall x \in \mathbb{R}^+$ or $f^{l'}(x) < 0, \forall x \in \mathbb{R}^+$.

If the first-order derivative of the activation function $f^{l'}(x) \geqslant 0, \forall x \in \mathbb{R}^+$, we have $f^{l'}(W^l y^{l-1} + b^l) \geqslant 0$ and $H^{l-1'}(x) \geqslant 0$. Since $W^l \geqslant 0$, it follows that the first-order derivative $H^{l'}(x) \geqslant 0, \forall x \in \mathbb{R}^+$.

If the first-order derivative of the activation function $f^{l'}(x) < 0, \forall x \in \mathbb{R}^+$, then we have $f^{l'}(W^l y^{l-1} + b^l) < 0$ and $H^{l-1'}(x) < 0$. Since $W^l > 0$, it follows that the first-order derivative $H^{l'}(x) > 0, \forall x \in \mathbb{R}^+$.

In summary, for any positive vector, the first-order derivative satisfies $H^{l'}(x) \geqslant 0$.

The second-order derivative of $H^l(x)$ with respect to x is:

$$\begin{aligned} H^{l''}(x) = y^{l''} &= f^{l''}\left(W^l y^{l-1} + b^l\right) \cdot W^{l^2} \cdot y^{l-1'} + f^{l'}\left(W^l y^{l-1} + b^l\right) \cdot W^l \cdot y^{l-1''} \\ &= f^{l''}\left(W^l y^{l-1} + b^l\right) \cdot W^{l^2} \cdot H^{l-1'}(x) + f^{l'}\left(W^l y^{l-1} + b^l\right) \cdot W^l \cdot H^{l-1''}(x). \end{aligned} \tag{18}$$

For convenience, we note that

$$u = f^{l''}\left(W^l y^{l-1} + b^l\right) W^{l^2} H^{l-1'}(x), \tag{19}$$

$$v = f^{l'}\left(W^l y^{l-1} + b^l\right) \cdot W^l H^{l-1''}(x). \tag{20}$$

Assuming that assumption 1 is valid, according to the definition of a concave function, for each positive vector x, the second order derivative of the activation function satisfies $f^{l''}(x) \leqslant 0$, then we have $f^{l''}(W^l y^{l-1} + b^l) \leqslant 0$ and $H^{l-1''}(x) \leqslant 0$. And because the first-order derivative satisfies $H^{l-1}(x)' \geqslant 0$ and $H^{l-1}(x)' \geqslant 0$, it follows that the $u \leqslant 0$ and $v \leqslant 0$. So we have:

$$\frac{\mathrm{d}^2 H^l(x)}{\mathrm{d}^2 x} \leqslant 0, \forall x \in \mathbb{R}^+. \tag{21}$$

Because the aforementioned inequality is satisfied for all positive vectors x, it can be proven that $H^l(x)$ is a concave function of x. According to equation (16), it can be deduced that y^l is also a concave function of x.

The proof of theorem 1 is completed by the step 1 and step 2.

Appendix B. Proof of proposition 1

Since all the activation function of the BP neural network are hyperbolic tangent (tanh) functions and the hyperbolic tangent function is a monotonic concave function in the positive domain and is second-order differentiable, the assumptions 1 holds. In the BP neural network, $b^l, \forall l \in [2, L] \cap \mathbb{Z}^+$ is a positive vector and $W^l, \forall l \in [2, L] \cap \mathbb{Z}^+$ is a positive matrix. Therefore, the assumptions 2 holds. Since $R > 0, I_{\text{ave}} > 0$, and $\tilde{g}(R)$ and $\tilde{\varphi}(I_{\text{ave}})$ are derived from the BP neural network where the code rate satisfies $R > 0$ and the number of decoding iterations satisfies $I_{\text{ave}} > 0$, it follows from theorem 1 that proposition 1 holds.

Appendix C. Proof of proposition 2

For the discrete function $f(x), \forall x \in \mathbb{Z}^+$, it is concave-extensible if there exists a concave function $g(x), \forall x \in \mathbb{R}^+$ such that $f(x) = g(x), \forall x \in \mathbb{Z}^+$ [40]. If a discrete function is concave-extensible, it is a discrete concave function [40]. $\tilde{\varphi}(I_{\text{ave}})$ is derived from a BP neural network where the number of decoding iterations satisfies $I_{\text{ave}} \in \mathbb{R}^+$ and $\tilde{\varphi}(I_{\text{ave}})$ is a concave function of I_{ave}. When a subset of the domain of the function is taken, the value of the function does not change in that subset [44]. Therefore, when the number of decoding iterations satisfies $I_{\text{ave}} \in \mathbb{Z}^+$, $\tilde{\varphi}(I_{\text{ave}})$ is a concave-extensible function and satisfies the definition of a discrete concave function. As a result, proposition 2 holds.

ORCID iDs

Yang Liu 0009-0008-8573-7576
Jiangliang Jin 0000-0003-0077-8624
Xue-Qin Jiang 0000-0002-0414-4349
Jisheng Dai 0000-0002-0462-4414

References

[1] Xu F, Ma X, Zhang Q, Lo H-K and Pan J-W 2020 Secure quantum key distribution with realistic devices *Rev. Mod. Phys.* **92** 025002
[2] Pirandola S *et al* 2020 Advances in quantum cryptography *Adv. Opt. Photon.* **12** 1012–236
[3] Portmann C and Renner R 2022 Security in quantum cryptography *Rev. Mod. Phys.* **94** 025008
[4] Wootters W K and Zurek W H 1982 A single quantum cannot be cloned *Nature* **299** 802–3
[5] Pan Y *et al* 2025 High-rate 16-node quantum access network based on a passive optical network *Optica* **12** 953–60
[6] Lucamarini M, Yuan Z L, Dynes J F and Shields A J 2018 Overcoming the rate—distance limit of quantum key distribution without quantum repeaters *Nature* **557** 7705
[7] Zhong X, Hu J, Curty M, Qian L and Lo H-K 2019 Proof-of-principle experimental demonstration of twin-field type quantum key distribution *Phys. Rev. Lett.* **123** 100506
[8] Diamanti E, Lo H K, Qi B and Yuan Z 2016 Practical challenges in quantum key distribution *npj Quantum Inf.* **2** 1
[9] Gümüs K, Eriksson T, Takeoka M, Fujiwara M, Sasaki M, Schmalen L and Alvarado A 2021 A novel error correction protocol for continuous variable quantum key distribution *Sci. Rep.* **11** 10465
[10] Xu Y, Wang T, Zhao H, Huang P and Zeng G 2023 Round-trip multi-band quantum access network *Photon. Res.* **11** 1449–64
[11] Grosshans F and Grangier P 2002 Continuous variable quantum cryptography using coherent states *Phys. Rev. Lett.* **88** 057902
[12] Leverrier A, Grosshans F and Grangier P 2010 Finite-size analysis of a continuous-variable quantum key distribution *Phys. Rev. A* **81** 36–43
[13] Van Assche G, Cardinal J and Cerf N J 2012 Reconciliation of a quantum-distributed Gaussian key *Inf. Theory IEEE Trans.* **50** 394–400
[14] Wang X, Wang H, Zhou C, Chen Z, Yu S and Guo H 2022 Continuous-variable quantum key distribution with low-complexity information reconciliation *Opt. Express* **30** 30455–65
[15] Deutsch D, Ekert A, Jozsa R, Macchiavello C, Popescu S and Sanpera A 1998 Quantum privacy amplification and the security of quantum cryptography over noisy channels (vol 77, pg 2818, 1996) *Phys. Rev. Lett.* **9** 80
[16] Zhang Y, Chen Z, Pirandola S, Wang X, Zhou C, Chu B, Zhao Y, Xu B, Yu S and Guo H 2020 Long-distance continuous-variable quantum key distribution over 202.81 km of fiber *Phys. Rev. Lett.* **125** 010502
[17] Tang G-Z, Sun S-H and Li C-Y 2019 Experimental point-to-multipoint plug-and-play measurement-device-independent quantum key distribution network* *Chin. Phys. Lett.* **36** 070301
[18] Huang D, Lin D, Wang C, Liu W, Fang S, Peng J, Huang P and Zeng G 2015 Continuous-variable quantum key distribution with 1 Mbps secure key rate *Opt. Express* **23** 17511–9
[19] Zhang H, Mao Y, Huang D, Guo Y, Wu X and Zhang L 2018 Finite-size analysis of eight-state continuous-variable quantum key distribution with the linear optics cloning machine *Chin. Phys. B* **27** 090307
[20] Gan Y-H, Wang Y, Bao W-S, He R-S, Zhou C and Jiang M-S 2019 Finite-key analysis for a practical high-dimensional quantum key distribution system based on time-phase states *Chin. Phys. Lett.* **36** 040301
[21] Bunandar D, Govia L C G, Krovi H and Englund D 2020 Numerical finite-key analysis of quantum key distribution *npj Quantum Inf.* **6** 104

[22] Jiang X-Q, Wang N, Jin J, Feng Y, Hai H, Dai J and Huang P 2025 A machine learning based optimization approach for continuous-variable quantum key distribution *Adv. Quantum Technol.* **8** e00269
[23] Liu W, Huang P, Peng J, Fan J and Zeng G 2018 Integrating machine learning to achieve an automatic parameter prediction for practical continuous-variable quantum key distribution *Phys. Rev. A* **97** 022316
[24] Liao Q, Fei Z, Liu J, Huang A, Huang L and Wang Y 2025 High-rate discretely-modulated continuous-variable quantum key distribution using quantum machine learning *Chaos, Solitons Fractals* **196** 116331
[25] Wang W and Lo H-K 2019 Machine learning for optimal parameter prediction in quantum key distribution *Phys. Rev. A* **100** 062334
[26] Heaton J B, Polson N G and Witte J H 2016 Deep learning for finance: deep portfolios: J. B. Heaton, N. G. Polson and J. H. Witte *Appl. Stoch. Models Bus. Ind.* **33** 3–12
[27] Jouguet P, Kunz-Jacques S and Leverrier A 2011 Long-distance continuous-variable quantum key distribution with a Gaussian modulation *Phys. Rev. A* **84** 062317
[28] Leverrier A, Alléaume R, Boutros J, Zémor G and Grangier P 2008 Multidimensional reconciliation for a continuous-variable quantum key distribution *Phys. Rev. A* **77** 042325
[29] Shengmei Z, Zhigang S, Hong X and Le W 2018 Multidimensional reconciliation protocol for continuous-variable quantum key agreement with polar coding *Sci. China Phys. Mech. Astron.* **61** 90323
[30] Cao Z, Chen X, Chai G and Peng J 2023 IC-LDPC polar codes-based reconciliation for continuous-variable quantum key distribution at low signal-to-noise ratio *Laser Phys. Lett.* **20** 045201
[31] Johnson S J, Chandrasetty V A and Lance A M 2016 Repeat-accumulate codes for reconciliation in continuous variable quantum key distribution *Australian Communications Theory Workshop (AusCTW)* (*Melbourne, VIC, Australia*) (IEEE) pp 18–23
[32] Zhang K, Hou J, Jiang X Q, Bai E, Huang P and Zeng G 2023 High-speed information reconciliation with syndrome-based early termination for continuous-variable quantum key distribution *Opt. Express* **31** 11
[33] Liu Y, Jiang X Q, Dai J, Hai H and Huang P 2025 Lossy compression based on polar codes for high throughput information reconciliation in CV-QKD systems *Quantum Sci. Technol.* **10** 025043
[34] Milicevic M, Feng C, Zhang L M and Gulak P G 2018 Quasi-cyclic multi-edge LDPC codes for long-distance quantum cryptography *npj Quantum Inf.* **4** 21
[35] Wang X, Zhang Y, Yu S and Guo H 2018 High speed error correction for continuous-variable quantum key distribution with multi-edge type LDPC code *Sci. Rep.* **8** 10543
[36] Mouloodi S, Rahmanpanah H, Gohery S, Burvill C and Davies H M 2022 Feedforward backpropagation artificial neural networks for predicting mechanical responses in complex nonlinear structures: a study on a long bone *J. Mech. Behav. Biomed. Mater.* **128** 105079
[37] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back propagating errors *Nature* **323** 533–6
[38] Stathakis D 2009 How many hidden layers and nodes? *Int. J. Remote Sens.* **30** 2133–47
[39] Nocedal J, Wright S J, Mikosch T V, Resnick S I and Robinson S M 1999 *Numerical Optimization* (Springer)
[40] Murota K and Vygen J 2004 Discrete convex analysis *Math. Intell.* **26** 74–76
[41] Huang L, Qin J, Zhou Y, Zhu F, Liu L and Shao L 2023 Normalization techniques in training dnns: methodology, analysis and application *IEEE Trans. Pattern Anal. Mach. Intell.* **45** 10173–96
[42] Jiang X-Q, Xue S, Tang J, Huang P and Zeng G 2024 Low-complexity adaptive reconciliation protocol for continuous-variable quantum key distribution *Quantum Sci. Technol.* **9** 025008
[43] Zhou C, Wang X, Zhang Y, Zhang Z, Yu S and Guo H 2019 Continuous-variable quantum key distribution with rateless reconciliation protocol *Phys. Rev. Appl.* **12** 054013
[44] Rudin W 1976 *Principles of Mathematical Analysis 3rd* edn (McGraw-Hill)