AIMS Mathematics, 11(9): 30374–30402.
DOI: 10.3934/math.20261204
Received: 29 June 2026
Revised: 15 August 2026
Accepted: 04 September 2026
Published: 17 September 2026

https://www.aimspress.com/journal/Math

Research article

A Quantum-inspired population optimizer integrating curvature stability and predictive consistency for deep learning

Irsa Sajjad[1,*], Osama Abdulaiz Alamri[2], Maria Malik[3], Marwan H. Alhelali[2] and Maysoon A. Sultan[4]

[1] Department of Mathematics, National University of Modern Languages, Islamabad, Pakistan
[2] Department of Statistics, Faculty of Science, University of Tabuk, Tabuk 71491, Saudi Arabia
[3] Department of Statistics, Comsats University, Pakistan
[4] Department of Mathematics, Faculty of Science, University of Hafr Al Batin, Saudi Arabia

* **Correspondence:** Email: irsa.sajjad@numl.edu.pk.

Abstract: Training deep neural networks is often hindered by sharp minima, unstable optimization trajectories, and limited generalization. To tackle these problems, we introduce a novel gradient-free, population-based optimization framework, the Curvature–Interference Quantum Entropic Optimizer (CI-QEO), which integrates the empirical loss, zeroth-order curvature estimation, predictive interference suppression, entropy-driven diversity preservation, and covariance-driven tunnelling into a single energy function. We used a benchmark set of binary and multiclass classification problems, as well as nonlinear regression problems, to test the proposed method with the same network architecture. We found that CI-QEO outperforms strong baseline optimizers in terms of accuracy of binary classification (94.18% vs. 90.58%), accuracy of multiclass classification (91.36% vs. 84.34%), and R^2 score (0.936 vs. 0.854), with lower test loss (26.46% vs. 37.24%), mean squared error (22.54% vs. 37.24%), predictive interference (64.27% vs. 105.12%), and total optimization energy (73.66% vs. 167.83%). In addition, ablation experiments and statistical significance analysis ($p < 0.05$) validated each proposed component's contribution to the observed performance gains. The results highlight that CI-QEO is a stable, accurate, and large-scale optimization framework for deep neural network training and a promising alternative to prevalent gradient- and population-based optimization algorithms.

Keywords: quantum-inspired optimization; stochastic manifold learning; entropy regularization; hyperstate optimization; explainable artificial intelligence; probabilistic neural networks; stochastic differential geometry

Mathematics Subject Classification: 68T07, 65K10

1. Introduction

Deep neural networks have been used with great success for classification, regression, representation learning, and complex nonlinear function approximation. However, training neural networks remains a difficult optimization problem, as the empirical risk surface is typically high-dimensional, nonconvex, multimodal, and highly sensitive to initialization. Local training algorithms like stochastic gradient descent (SGD) and its adaptive variants update only one parameter vector based on local gradient information, and thus tend to be efficient and less stable in terms of convergence while suffering from sharp minima, noisy gradients, and early attraction to suboptimal regions [1,2]. While adaptive gradient methods like Adam are more efficient at learning by incorporating first- and second-moment estimates of the gradients, they remain based on a local first-order optimization approach and do not explicitly consider whether a low-loss solution exists in a stable or unstable basin [3,4]. Adam's original formulation uses adaptive lower-order gradient moments and is generally applied to stochastic objective functions. One big problem with loss-centered neural network training is that minimizing the empirical loss does not always yield the most generalizable solution. In deep learning, several works have demonstrated that sharp minima may yield low training loss but may not generalize well, since small perturbations in the parameter space can result in large variations in the loss landscape [5]. This is especially crucial in population-based optimization [6,7] and quantum optimization, where candidate states might be amplified because of their temporarily low empirical risk. The optimizer may fall into unstable basins if it finds such candidates in a sharp region. Hence, beyond loss reduction, modern neural network optimization needs explicit mechanisms to ensure stability, curvature control, and diversity. This problem is addressed in the proposed study by the Curvature-Interference Quantum Entropic Optimizer (CI-QEO).

CI-QEO aims to optimize a probability distribution over candidate neural network states rather than a single parameter vector and is designed as a gradient-free, population-based neural network optimizer. The uploaded methodology includes a section that defines how the criterion, CI-QEO, is applied to evaluate each candidate, namely via empirical risk, zeroth-order curvature, predictive interference, and annealed diversity, as well as an update of the candidate probability distribution according to an entropic amplitude law. This makes the optimizer different from traditional loss-based search strategies, as the searcher does not simply choose the candidate with the lowest loss at the time of selection; rather, it chooses candidates that are accurate, low-curvature, prediction-consistent, and sufficiently diverse during early exploration. A central motive for CI-QEO is the flat-minima theory of neural network generalization. Research has shown that broad or flat minima are more robust than sharp minima, as the network performance is not affected by small changes in the parameters [8,9]. Subsequent empirical studies with large-batch training confirmed that sharp minimizers tend to generalize worse [10,11]. Keskar et al. provided numerical evidence that large-batch methods can converge to sharp minimizers and that sharp minima are related to a generalization gap. Based on these results, a good optimizer should use curvature data directly when selecting candidates.

The CI-QEO is implemented using a zeroth-order directional curvature estimator, thereby enabling curvature estimation without backpropagated gradients. The other key ingredient of the proposed approach is entropy-regularized optimization. Researchers using entropic and mirror-descent

methods have introduced regularization in probability space, which can prevent unstable updates and promote smoother transitions between distributions [12,13]. In CI-QEO, the amplitude distribution is updated via a doubly regularized variational principle with KL geometry to preserve the previous probability distribution, and an entropy-promoting term to regulate collapse. The resulting update includes an energy reweighting term in exponential form to let the optimizer continue exploring the space, and a power-tempered memory factor to concentrate probability on low-energy candidates over time.

The population-based optimization has been employed for complex nonlinear search tasks. Genetic algorithms propose adaptive search based on selection, mutation, and recombination [14,15], and particle swarm optimization proposes collective search based on social interaction among particles [16,17]. Most of these techniques are derivative-free and typically lack explicit curvature awareness, predictive consistency regulation, or information-geometric amplitude control. PSO is a classical population-based search technique, first introduced at the IEEE International Conference on Neural Networks in 1995 by Kennedy and Eberhart. CI-QEO extends this population-based concept by determining candidate importance using a composite interference-curvature energy rather than fitness. The proposed method further differs from common population optimizers by including a predictive-interference component. In neural-network populations [18], many candidates can have the same loss yet yield different prediction directions. Such disagreement can destabilize the final model or slow convergence if the probability mass is assigned solely by loss. However, CI-QEO overcomes this problem by normalizing each candidate's prediction direction and penalizing destructive disagreement with strong, low-loss candidates. This is similar to interference-selection theory: Candidates whose predictions align with those that can be safely observed in a low-energy region are reinforced, while candidates whose predictions conflict with reliable ones are suppressed. The proposed optimizer also brings curvature-filtered tunneling.

CI-QEO is not based on isotropic random perturbations [19]; instead, it exploits stable candidates in the low-energy set to sample candidate transitions from the covariance geometry. This lets the optimizer move in directions enabled by the geometric structure of good solutions rather than by random noise. It is crucial for gradient-free optimization of neural networks, as it enables exploration while preventing wandering into high-curvature regions that are likely unstable. There are two ways to obtain the final trained model: Either by probability-weighted expectation or by selecting the minimum-energy candidate. Hence, our primary objective of this research is to develop a novel empirical accuracy, curvature stability, predictive consistency, entropic anti-collapse, and covariance-guided exploration (ECAECs) integrated gradient-free neural-network optimizer. The proposed CI-QEO method brings several contributions to the literature, filling three key gaps that are common to many optimizers: (1) Loss-only optimizers tend to select the sharpest minima, (2) population-based optimizers fail to maintain structured diversity, and (3) quantum-inspired amplitude-based optimizers tend to amplify candidates that are not aware of curvature or predictive interference.

In this study, we present CI-QEO, a curvature-interference quantum entropic optimizer for training neural networks without using gradients. While classical quantum-inspired amplitude methods simply amplify candidate states based on the empirical risk, CI-QEO adds a novel free-energy functional that evaluates each candidate using four coupled quantities: empirical risk, zeroth-order curvature, predictive interference, and annealed population diversity. The amplitude dynamics follow from a doubly regularized KL variational principle, yielding a closed-form power-tempered exponential update. The method also presents covariance-guided tunneling, sampling perturbations that are distributed according to the geometry of low-energy candidates rather than isotropic Gaussian

noise. Theoretical analysis shows that simplex preservation, boundedness, asymptotic free-energy descent, and almost-sure stabilization of the amplitude distribution are achieved when the perturbations are summed over time.

2. Review of literature

Gradient-based optimization has been the traditional approach to neural network optimization. Stochastic gradient descent remains the basic technique due to its simplicity, scalability, and ability to process large data sets using mini-batch updates [20]. SGD, however, is sensitive to the choice of learning rate, initialization, noisy gradients, and loss-surface geometry. Several methods have been proposed to enhance convergence by incorporating previous update directions, such as momentum-based and accelerated methods, thereby avoiding oscillations and improving movement in ill-conditioned regions [21]. Despite these advances, first-order gradient methods remain heavily reliant on local derivative information and fail to explicitly maintain population diversity, preserve entropy, or ensure agreement among candidate networks. However, fixed-learning-rate gradient descent methods had several drawbacks, so adaptive gradient methods were developed to address them. AdaGrad dynamically adjusts learning rates based on accumulated gradient information and is well-suited to features with different frequencies or scales [22,23]. Duchi et al. defined AdaGrad as a class of subgradient methods that utilize data-geometry information to enable more informative learning. Following this work, RMSProp and Adam took adaptive optimization to the next level by employing moving averages of gradient moments, thereby outperforming previous approaches when the objective is noisy and sparse [24]. However, adaptive optimizers have yet to be designed to explicitly distinguish between flat and sharp low-loss regions [25,26]. This leaves room for curvature-aware optimization methods such as CI-QEO. There is much debate around the role of flatness in the generalization of neural networks. Hochreiter and Schmidhuber suggested that the minimum should be flat where desirable, since it indicates regions of the parameter space where the model is stable under perturbations [27,28]. As deep neural networks have expanded toward increasingly over-parameterized models, this idea has become increasingly relevant. Keskar et al. showed that sharp minima and generalization degradation are related to large-batch training, which implies that sharpness is an important diagnostic for optimizer behavior [29]. This view raises the question of whether an optimizer that only minimizes empirical risk is incomplete. This matters because CI-QEO explicitly includes a zeroth-order curvature penalty in the energy functional, penalizing candidates that achieve low loss by converging to sharp, unstable basins. Another important foundation for the proposed method is entropy-based optimization. Entropy-SGD is a developed method in which the gradient descent algorithm is biased toward broad valleys by learning a local entropy objective rather than the empirical loss [30].

In this study, we showed that entropy can encourage flat regions and improve generalization. While closely related in spirit, CI-QEO differs from the previous one in that it uses entropy on a probability distribution rather than a single gradient-based trajectory of networks. This way, the optimizer can keep several tentative solutions on hand and gradually shift the probability mass toward solutions with lower interference-curvature energy. Amplitude updates can be theoretically supported by mirror descent and KL-regularized optimization. Mirror descent is a generalization of gradient descent, where the updates are performed with respect to a distance-generating function, and the KL divergence is particularly appropriate for optimization of a probability distribution [31]. In CI-QEO, a variational principle based on an entropic approach updates the candidate probabilities. The KL term

helps to avoid sudden changes from the previous distribution, and the entropy term prevents the early collapse. Therefore, CI-QEO combines exploitation of low-energy candidates with controlled exploration of the candidate population. Derivative-free zeroth-order optimization methods are relevant because CI-QEO is intended to train a neural network without backpropagating gradients. Simultaneous perturbation stochastic approximation (SPSA) has been influential in noisy optimization problems [32] and estimates gradient information by adding simultaneous random perturbations. Subsequent zeroth-order optimization techniques have provided more formal approaches to random minimization without an explicit gradient, particularly when gradients are unavailable or computation is costly [33]. CI-QEO does not simply guess a gradient direction but is based on the same idea as the derivative-free approach. Instead, it employs symmetric finite-difference probes to locally approximate curvature and adds the result to the candidate energy. Gradient-free optimization has been a key use of evolutionary computation. Genetic algorithms employ operations such as selection, crossover, and mutation that mimic biological processes during population search. Holland's initial work marked the beginning of the field of genetic algorithms and of adaptation as a computational search principle. Evolutionary algorithms are appealing since they can transcend local optima and function without any gradient information. However, they typically rely on fitness and do not explicitly reward or punish sharpness or disagreement in predictions.

CI-QEO extends population-based search by adding a structured energy function that considers loss, curvature, interference, and diversity. Another important potential-based method is particle swarm optimization. In PSO, particles update based on their own experiences and those of other particles, making it well-suited to nonlinear and nonconvex search spaces [34]. While PSO has been used in neural network learning and hyperparameter tuning, it typically employs position- and velocity-based dynamics rather than probabilistic amplitude redistribution. Furthermore, the standard PSO lacks an anti-collapse regulation based on entropy. CI-QEO differs from other methods in that it maintains a probability simplex over candidates and updates the distribution based on a closed-form entropic law. Quantum-inspired optimization methods aim to adapt computational paradigms from Quantum Mechanics, particularly superposition, probability amplitudes, and tunneling. Quantum-inspired evolutionary algorithms (QIEA) are based on probabilistic representations that maintain diversity and improve global search [35]. Many amplitude methods, however, primarily boost candidates through fitness or loss and draw on quantum principles. However, these mechanisms can fail if a 'temporary' sharp low-loss candidate enters the population.

By introducing a curvature-interference energy that complements loss, CI-QEO extends this concept to favor candidates that are stable (i.e., minimize loss) and meaningful (i.e., consistent with predictions). It is also related to prediction consistency in ensemble learning and agreement between models. Ensemble methods have shown that multiple learners can generalize better when they make complementary yet reliable predictions [35]. Ensemble diversity, however, should be controlled, as excessive disagreement can undermine prediction stability. This insight is incorporated into the optimizer as a penalty for candidates whose normalized predictive directions do not match those of low-loss reference candidates. This means that predictive agreement is not an ensemble property after training but rather a criterion during training. Stochastic approximation theory and supermartingale convergence theory are also relevant to the convergence analysis of CI-QEO. Robbins and Siegmund described the almost-supermartingale version of nonnegative stochastic processes, which is often employed in the study of stochastic iterative methods [36]. Quantum-inspired neural-network optimization frameworks have increasingly incorporated stochastic population dynamics, probabilistic

candidate-state evolution, and stability-oriented search mechanisms to improve convergence in complex nonlinear learning problems [37–39]. Under boundedness and summability assumptions in CI-QEO, the amplitude distribution becomes stable, and the interference-curvature energy diminishes to an asymptotic level, up to entropy-related corrections. This theoretical framework supports the view that CI-QEO is not only a heuristic population optimizer but also an optimization framework with mathematically interpretable stability properties [40,41]. In general, the literature shows that each family of optimization techniques [42,43] yields only part of the solution to the neural-network training problem. The gradient-based methods are local and efficient. Adaptive methods improve learning-rate behavior without altering the loss's centrality. Flat minima and entropy-based methods have been shown to improve generalization but often depend on the gradient trajectory. Population search methods, such as evolutionary and swarm methods, avoid explicit curvature and predictive-interference regulation. Quantum-inspired techniques keep probabilistic diversity but tend to enhance candidates, with insufficiently rich energy considerations. CI-QEO lies at the crossroads of these disciplines and provides a single optimizer that combines gradient-free curvature estimation, predictive interference suppression, entropic amplitude updating, annealed diversity, and covariance-guided tunneling.

3. Methodology

Curvature-interference quantum entropic optimizer

The proposed technique is dubbed the Curvature-Interference Quantum Entropic Optimizer (CI-QEO) and its complete computational architecture is presented in Figure 1. The theoretical difference in CI-QEO is the criterion used to amplify candidates and transition them into the population. Conventional gradient-based optimizers adjust a single parameter trajectory based on local derivative information, while evolutionary and swarm-based optimizers track multiple parameter trajectories but usually order them by fitness or objective-function value. Similarly, traditional quantum-inspired amplitude-based optimization keeps a probabilistic representation of candidate states, but mostly redistributes the amplitude according to empirical loss or fitness. The optimization process starts with the normalized input data X and a population of N candidate neural-network parameter vectors $\left\{\theta_k^{(t)}\right\}_{k=1}^{N}$. The neural network predicts the output $f\theta_k^{(t)}(X)$ for each candidate. The candidate-population module stores not only the parameter states, $\theta_k^{(t)}$, but also their probability amplitudes, *ak(t).* The energy-evaluation module outputs four quantities, one for each candidate, which are used to evaluate the energy: empirical loss L; its curvature is represented by k (t); zeroth-order curvature $C\,k^{(t)}$; predictive-interference penalty *Ik(t);* and diversity potential $D^{k(t)}$. These components sum to the composite optimization energy, Ek(t). The resulting energy values and the previous probability distribution *p(t)* are then fed into the entropy-regularized amplitude-update module, which yields an updated probability distribution *p(t+1).* Then, low-energy, low-curvature candidate states are used to construct the elite covariance matrix Σ. The direction of the tunneling is anisotropic, defined by (t), and this is what generates the next set of candidate populations $\{\theta_k^{(t+1)}\}$. The candidates are then reshaped until convergence is achieved according to the desired criterion or a maximum number of iterations, and the resulting model is extracted from the candidate state with the lowest energy or the most probable candidate state. In nonconvex deep learning, a candidate can have a low loss yet reside in a sharp, unstable basin. Hence, CI-QEO refines a probability distribution of candidate networks based on a novel interference energy based on a combination of empirical loss, curvature stability,

diversity pressure, and phase disagreement suppression. Let the neural network be f_θ, where $\theta \in \mathbb{R}^N$. Let the empirical risk be

$$\mathcal{L}(\theta) = \frac{1}{m}\sum_{i=1}^{m} \ell(f_\theta(x_i), y_i). \tag{1}$$

At iteration t, instead of keeping a single parameter vector, CI-QEO maintains K candidate states,

$$\Theta^{(t)} = \left\{\theta_1^{(t)}, \theta_2^{(t)}, \ldots, \theta_K^{(t)}\right\}, \theta_k^{(t)} \in \mathbb{R}^N, \tag{2}$$

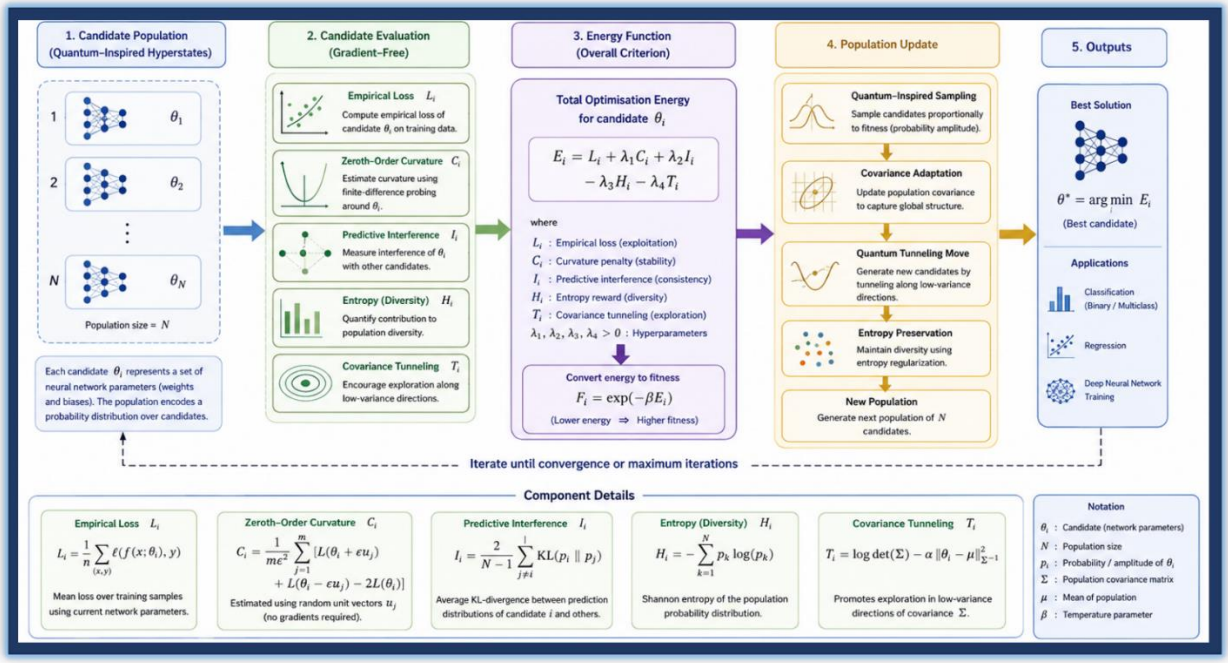

Figure 1. Computational architecture of the proposed CI-QEO framework. The connecting arrows are labeled with the variables exchanged among successive modules, such as candidate neural-network parameters, probability amplitudes, predictions, empirical loss, curvature, predictive interference, diversity potential, composite energy, updated probabilities, and elite covariance information. The convergence criterion is checked, and the full optimization cycle is repeated until convergence.

CI-QEO does not keep a deterministic parameter vector, but a population of N candidate neural-network parameter vectors at optimization iteration t. The candidate population is represented by the following quantum-inspired trainable state for a superposed train:

$$|\Psi^{(t)}\rangle = \sum_{k=1}^{K} a_k^{(t)} e^{i\varphi_k^{(t)}} |\theta_k^{(t)}\rangle, \tag{3}$$

subject to

$$\sum_{k=1}^{N} \left|a_k^{(t)}\right|^2 = 1. \tag{4}$$

The corresponding probability mass assigned to the k-th candidate is

$$p_k^{(t)} = \left|a_k^{(t)}\right|^2, p_k^{(t)} \geq 0, \sum_{k=1}^{N} p_k^{(t)} = 1. \tag{5}$$

Here, $\theta_k^{(t)}$ is the phase variable $k, a_k^{(t)}$ is the probability amplitude, and $\phi_k^{(t)}$ is the parameter vector of candidate network k. The state variables in Eq (1) are therefore not a special representation of the quantum system, but are the state variables used throughout the rest of the optimization in the CI-QEO. In the particular case, the candidate state $\phi_k^{(t)}$ is used to calculate its empirical loss, local curvature, prediction vector, predictive-interference penalty, and diversity contribution; pk(t), calculated from the amplitude in Eq (3), is used to calculate its probability-weighted population statistics and is used for the prior distribution for the next entropic amplitude update.

The phase component is added through the predictive-interference mechanism and aims to favor candidate states whose predictive directions remain stable, as in low-curvature candidate states. Therefore, the phase-related information aids candidate assessment in the following interference formulation. Accordingly, the empirical loss associated with the candidate state $\theta_k^{(t)}$ is defined as

$$L_k^{(t)} = \mathcal{L}\left(\theta_k^{(t)}\right). \tag{6}$$

The candidate parameter vector that has been introduced in the superposed state of Eq (4) is explicitly used in Eq (6), which is the first explicit link between a candidate-state representation and its subsequent optimization criterion. To quantify the local sharpness associated with each candidate state without requiring gradient information, we introduce a zeroth-order curvature estimator. Let u_r denote a randomly sampled unit direction,

$$u_r \sim \mathrm{Unif}(\mathbb{S}^{d-1}), r = 1, \ldots, R, \tag{7}$$

where *d* is the dimensionality of the neural-network parameter vector. *N* is the population size, so using $\mathbb{S}^{d-1}$ could imply that the perturbation dimension is determined by the number of candidates rather than the parameter dimension, and define the zeroth-order directional curvature score

$$C_k^{(t)} = \frac{1}{R} \sum_{r=1}^{R} \left| \frac{\mathcal{L}(\theta_k^{(t)} + \delta u_r) - 2\mathcal{L}(\theta_k^{(t)}) + \mathcal{L}(\theta_k^{(t)} - \delta u_r)}{\delta^2} \right|. \tag{8}$$

For twice differentiable losses, Taylor expansion gives

$$\mathcal{L}(\theta + \delta u) = \mathcal{L}(\theta) + \delta \nabla \mathcal{L}(\theta)^\top u + \frac{\delta^2}{2} u^\top \nabla^2 \mathcal{L}(\theta) u + O(\delta^3), \tag{9}$$

and

$$\mathcal{L}(\theta + \delta u) - 2\mathcal{L}(\theta) + \mathcal{L}(\theta - \delta u) = \delta^2 u^\top \nabla^2 \mathcal{L}(\theta) u + O(\delta^4). \tag{10}$$

Therefore,

$$C_k^{(t)} = \frac{1}{R} \sum_{r=1}^{R} \left| u_r^\top \nabla^2 \mathcal{L}(\theta_k^{(t)}) u_r \right| + O(\delta^2). \tag{11}$$

Thus,t $C_k^{(t)}$approximates the average absolute curvature without backpropagating gradients. The first novelty is that CI-QEO explicitly imposes an energetic cost associated with curvature (i.e., it does not

allow collapse into sharp, low-loss regions). Next, we define a diversity potential. Allow the probability-weighted candidate center to be $\bar{\theta}^{(t)} = \sum_{j=1}^{K} p_j^{(t)} \theta_j^{(t)}$, defined as $D_k^{(t)} = \| \theta_k^{(t)} - \bar{\theta}^{(t)} \|_2^2$. A candidate with high $D_k^{(t)}$ has a high exploration. However, excessive diversity becomes a problem as training progresses. The diversity term is thus annealed with a parameter $\beta_t > 0$, such that

$$\beta_t = \beta_0 e^{-\lambda_\beta t}. \tag{12}$$

Now introduce prediction-phase interference. Let

$$z_k^{(t)}(x) = f_{\theta_k^{(t)}}(x), \tag{13}$$

be the output vector of the candidate k, define the normalized predictive direction

$$\hat{z}_k^{(t)}(x) = \frac{z_k^{(t)}(x)}{\| z_k^{(t)}(x) \|_2 + \varepsilon}. \tag{14}$$

The phase-agreement score between candidates k and j is

$$A_{kj}^{(t)} = \frac{1}{m} \sum_{i=1}^{m} \langle \hat{z}_k^{(t)}(x_i), \hat{z}_j^{(t)}(x_i) \rangle. \tag{15}$$

The interference penalty for the candidate k is

$$I_k^{(t)} = \sum_{j=1}^{K} p_j^{(t)} \left(1 - A_{kj}^{(t)}\right) \exp\left(-\tau L_j^{(t)}\right). \tag{16}$$

This number will punish a candidate if his/her prediction differs from those of candidates with low loss. The exponential term $\exp\left(-\tau L_j^{(t)}\right)$ provides strong candidates as reference states. The penalty is not "ensemble averaging" but rather an interference-like mechanism that suppresses candidates whose predictive phase, due to the conflicting phase in the low-energy region of the population, is mismatched. The new definition of the energy of a candidate *k* is then given by

$$E_k^{(t)} = L_k^{(t)} + \alpha_t C_k^{(t)} + \gamma_t I_k^{(t)} - \beta_t D_k^{(t)}. \tag{17}$$

Here, $L_k^{(t)}$is a function of the accuracy of the empirical learning, $C_k^{(t)}$a function of the curvature of the empirical learning, $I_k^{(t)}$a function of the destructive interference of the predictive learning, and $D_k^{(t)}$ a function of the useful exploration of the empirical learning. The signs are very significant. The optimizer reduces $E_k^{(t)}$. The curvature and destructive interference add energy; controlled diversity subtracts energy. The amplitude update is obtained from the following entropic variational principle:

$$p^{(t+1)} = \arg\min_{q \in \Delta_K} \left\{ \sum_{k=1}^{K} q_k E_k^{(t)} + \frac{1}{\eta_t} D_{\mathrm{KL}}(q \,\|\, p^{(t)}) + \omega_t D_{\mathrm{KL}}(q \,\|\, u) \right\}. \tag{18}$$

Uniform distribution is $u = \left(\frac{1}{K}, \ldots, \frac{1}{K}\right)$where K is the number of items. The first KL term helps avoid jumps away from the previous amplitude distribution. The second KL term prevents the distribution from collapsing too early by softly pulling it toward maximum entropy. This contrasts with

the uploaded QIASO method, where the probability update is regulated only by loss and previous KL geometry; it is also regulated by a new interference-curvature energy and an explicit anti-collapse entropy regularizer. The objective is:

$$J(q) = \sum_{k=1}^{K} q_k E_k^{(t)} + \frac{1}{\eta_t} \sum_{k=1}^{K} q_k \log \frac{q_k}{p_k^{(t)}} + \omega_t \sum_{k=1}^{K} q_k \log \frac{q_k}{u_k}. \tag{19}$$

Introduce a Lagrange multiplier $\lambda$for the constraint $\sum_k q_k = 1$. The Lagrangian is

$$\mathcal{L}(q, \lambda) = J(q) + \lambda(\sum_{k=1}^{K} q_k - 1). \tag{20}$$

Differentiating with respect to q_k gives

$$\frac{\partial \mathcal{L}}{\partial q_k} = E_k^{(t)} + \frac{1}{\eta_t}\left(\log \frac{q_k}{p_k^{(t)}} + 1\right) + \omega_t \left(\log \frac{q_k}{u_k} + 1\right) + \lambda. \tag{21}$$

At the optimum,

$$E_k^{(t)} + \frac{1}{\eta_t}\left(\log \frac{q_k}{p_k^{(t)}} + 1\right) + \omega_t \left(\log \frac{q_k}{u_k} + 1\right) + \lambda = 0. \tag{22}$$

Collecting logarithmic terms,

$$\left(\frac{1}{\eta_t} + \omega_t\right) \log q_k = -E_k^{(t)} + \frac{1}{\eta_t} \log p_k^{(t)} + \omega_t \log u_k - \lambda - \frac{1}{\eta_t} - \omega_t. \tag{23}$$

Let $a_t = \frac{1}{\eta_t} + \omega_t$. Then, $\log q_k = -\frac{E_k^{(t)}}{a_t} + \frac{1}{\eta_t a_t} \log p_k^{(t)} + \frac{\omega_t}{a_t} \log u_k + \text{constant}$.
Exponentiating and normalizing give the closed-form amplitude update

$$p_k^{(t+1)} = \frac{\left(p_k^{(t)}\right)^{\frac{1}{\eta_t a_t}} (u_k)^{\frac{\omega_t}{a_t}} \exp\left(-\frac{E_k^{(t)}}{a_t}\right)}{\sum_{j=1}^{K} \left(p_j^{(t)}\right)^{\frac{1}{\eta_t a_t}} (u_j)^{\frac{\omega_t}{a_t}} \exp\left(-\frac{E_j^{(t)}}{a_t}\right)}. \tag{24}$$

The uniform term is $u_k = \frac{1}{K}$ and therefore does not vary as ω_t is fixed, but it does vary the effective temperature a_t and which is used to prevent aggressive collapse. The update that is implemented is therefore

$$p_k^{(t+1)} = \frac{\left(p_k^{(t)}\right)^{r_t} \exp\left(-s_t E_k^{(t)}\right)}{\sum_{j=1}^{K} \left(p_j^{(t)}\right)^{r_t} \exp\left(-s_t E_j^{(t)}\right)}, \tag{25}$$

Where $r_t = \frac{1}{1+\eta_t \omega_t}, s_t = \frac{\eta_t}{1+\eta_t \omega_t}$. This is a new amplitude law standard QIASO uses

$$p_k^{(t+1)} \propto p_k^{(t)} \exp\left(-\eta L_k^{(t)}\right), \tag{26}$$

whereas CI-QEO uses

$$p_k^{(t+1)} \propto \left(p_k^{(t)}\right)^{r_t} \exp\left[-s_t\left(L_k^{(t)} + \alpha_t C_k^{(t)} + \gamma_t I_k^{(t)} - \beta_t D_k^{(t)}\right)\right]. \tag{27}$$

This exponent $r_t < 1$ is of the utmost importance. It doesn't reweight until it has flattened the previous distribution so that there isn't any early overconfidence. The method thus has an intrinsic anti-collapse feature. In CI-QEO, tunneling is curvature-filtered rather than standard Gaussian tunneling. Let us define the elite set with low curvature

$$\mathcal{S}_t = \left\{k: E_k^{(t)} \leq Q_\rho(E^{(t)})\right\}, \tag{28}$$

where Q_ρ is the ρ-quantile of candidate energies. Define the elite covariance

$$\Sigma_t = \sum\nolimits_{k\in\mathcal{S}_t} \tilde{p}_k^{(t)} \left(\theta_k^{(t)} - \bar{\theta}_S^{(t)}\right)\left(\theta_k^{(t)} - \bar{\theta}_S^{(t)}\right)^{\top} + \nu I, \tag{29}$$

with

$$\tilde{p}_k^{(t)} = \frac{p_k^{(t)}}{\sum_{j\in\mathcal{S}_t} p_j^{(t)}}, \tag{30}$$

and

$$\bar{\theta}_S^{(t)} = \sum\nolimits_{k\in\mathcal{S}_t} \tilde{p}_k^{(t)}\, \theta_k^{(t)}. \tag{31}$$

The tunneling direction is sampled as

$$\xi_k^{(t)} \sim \mathcal{N}(0, \Sigma_t). \tag{32}$$

The candidate update is

$$\theta_k^{(t+1)} = \theta_k^{(t)} - \chi_t\left(\theta_k^{(t)} - \bar{\theta}_S^{(t)}\right) + B_t \varepsilon_t \xi_k^{(t)}, \tag{33}$$

where

$$B_t \sim \text{Bernoulli}(\rho_t), \rho_t = \rho_0 e^{-\lambda_\rho t}. \tag{34}$$

The first term drives candidates to the elite's low-energy manifold. The second term generates anisotropic tunnelling along directions found based on stable candidates. This is more sophisticated than an isotropic Gaussian perturbation, since the algorithm tunnels along directions favored by the elite geometry of the covariance.

Now, define the free energy for CI-QEO.

$$\mathcal{F}_t(q) = \langle q, E^{(t)}\rangle + \frac{1}{\eta_t} D_{\text{KL}}\left(q \parallel p^{(t)}\right) + \omega_t D_{\text{KL}}(q \parallel u). \tag{35}$$

Because $p^{(t+1)}$ minimizes $\mathcal{F}_t$, we have

$$\mathcal{F}_t\left(p^{(t+1)}\right) \leq \mathcal{F}_t\left(p^{(t)}\right). \tag{36}$$

Therefore,

$$\left\langle p^{(t+1)}, E^{(t)} \right\rangle + \frac{1}{\eta_t} D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right) + \omega_t D_{\mathrm{KL}}\left(p^{(t+1)} \parallel u\right) \leq \left\langle p^{(t)}, E^{(t)} \right\rangle + \omega_t D_{\mathrm{KL}}\left(p^{(t)} \parallel u\right). \quad (37)$$

Rearranging gives:

$$\left\langle p^{(t+1)}, E^{(t)} \right\rangle \leq \left\langle p^{(t)}, E^{(t)} \right\rangle - \frac{1}{\eta_t} D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right) + \omega_t \left[D_{\mathrm{KL}}\left(p^{(t)} \parallel u\right) - D_{\mathrm{KL}}\left(p^{(t+1)} \parallel u\right)\right]. \quad (38)$$

If $\omega_t \to 0$, then the asymptotic descent relation becomes

$$\left\langle p^{(t+1)}, E^{(t)} \right\rangle \leq \left\langle p^{(t)}, E^{(t)} \right\rangle - \frac{1}{\eta_t} D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right) + o(1). \quad (39)$$

This means that the expected interference-curvature energy approaches zero as the entropy correction vanishes. A boundedness proof is very straightforward. Since

$$p_k^{(t+1)} = \frac{\left(p_k^{(t)}\right)^{r_t} e^{-s_t E_k^{(t)}}}{\sum_{j=1}^{K} \left(p_j^{(t)}\right)^{r_t} e^{-s_t E_j^{(t)}}}, \quad (40)$$

All terms are nonnegative, and the denominator normalizes them. Hence,

$$p_k^{(t+1)} \geq 0, \sum_{k=1}^{K} p_k^{(t+1)} = 1. \quad (41)$$

Therefore,

$$p^{(t)} \in \Delta_K \text{for all } t. \quad (42)$$

If the loss, curvature estimator, interference score, and diversity score are bounded on the candidate set, then there exist constants $E_{\min}$ and $E_{\max}$, such that

$$E_{\min} \leq E_k^{(t)} \leq E_{\max}. \quad (43)$$

Therefore,

$$E_{\min} \leq \left\langle p^{(t)}, E^{(t)} \right\rangle \leq E_{\max}. \quad (44)$$

The expected energy sequence is bounded from below.

For convergence, assume that $\sum_t \rho_t < \infty$, $\sum_t \varepsilon_t^2 < \infty$, $\omega_t \to 0$, and the stochastic transition noise is a martingale difference with bounded conditional variance. Let

$$X_t = \left\langle p^{(t)}, E^{(t)} \right\rangle. \quad (45)$$

The previous descent inequality implies

$$\mathbb{E}[\, X_{t+1} \mid \mathcal{F}_t \,] \leq X_t - \frac{1}{\eta_t} D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right) + b_t, \quad (46)$$

where $\sum_{t=0}^{\infty} b_t < \infty$. By the Robbins–Siegmund supermartingale convergence theorem, X_t converges almost surely, and

$$\sum_{t=0}^{\infty} \frac{1}{\eta_t} D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right) < \infty. \tag{47}$$

If η_t is bounded above, then

$$D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right) \to 0. \tag{48}$$

By Pinsker's inequality,

$$\parallel p^{(t+1)} - p^{(t)} \parallel_1^2 \leq 2D_{\mathrm{KL}}\left(p^{(t+1)} \parallel p^{(t)}\right), \tag{49}$$

so

$$\parallel p^{(t+1)} - p^{(t)} \parallel_1 \to 0. \tag{50}$$

Therefore, the amplitude distribution tends almost surely to stabilize. In the limit of low temperatures, as $s_t \to \infty$, the probability becomes localized on the minimizers of the interference-curvature energy,

$$\mathcal{M} = \arg\min_k \left[L_k^{(t)} + \alpha_t C_k^{(t)} + \gamma_t I_k^{(t)} - \beta_t D_k^{(t)}\right]. \tag{51}$$

If the minimizer is unique, $k^\star$, then

$$p^{(t)} \to e_{k^\star}. \tag{52}$$

Thus, CI-QEO is not just the minimum-loss candidate. It iterates toward a candidate that is low-loss, low-curvature, prediction-consistent, and free of premature collapse. The final trained parameter can be retrieved either through probabilistic expectation,

$$\theta_{\mathrm{final}} = \sum_{k=1}^{K} p_k^{(T)} \theta_k^{(T)}, \tag{53}$$

or by energy collapse, $\theta_{\mathrm{final}} = \theta_{k^\star}^{(T)}, k^\star = \arg\max_k p_k^{(T)}$. The most important new equation is

$$p_k^{(t+1)} = \frac{\left(p_k^{(t)}\right)^{r_t} \exp\left[-s_t\left(L_k^{(t)} + \alpha_t C_k^{(t)} + \gamma_t I_k^{(t)} - \beta_t D_k^{(t)}\right)\right]}{\sum_{j=1}^{K}\left(p_j^{(t)}\right)^{r_t} \exp\left[-s_t\left(L_j^{(t)} + \alpha_t C_j^{(t)} + \gamma_t I_j^{(t)} - \beta_t D_j^{(t)}\right)\right]}. \tag{54}$$

where $r_t = \frac{1}{1+\eta_t\omega_t}, s_t = \frac{\eta_t}{1+\eta_t\omega_t}$. The mathematical operations of CI-QEO are implemented sequentially as summarized in Algorithm 1. This procedure begins by building the candidate population and its corresponding probability-amplitude representation, as given in Eqs (1)–(3). The empirical loss of each candidate is computed for each iteration with Eq (4); then the zeroth-order curvature, the population diversity, and the predictive-interference evaluations are performed. These

quantities are included in the composite interference-curvature energy. These candidate energies are then used in the entropy-regularized amplitude update to obtain a new probability distribution over candidate states. The low-energy and low-curvature candidates are then chosen to build the elite covariance geometry, and the geometry perturbations are sampled for structured tunneling. The resulting parameters form the Candidate Population for the next iteration. This process repeats until the maximum number of iterations or the convergence criterion is met.

Algorithm 1. Curvature–Interference Quantum Entropic Optimizer (CI-QEO)

Input: Training dataset D, population size N, maximum iterations T, curvature directions R, and CI-QEO hyperparameters $\{\lambda_C, \lambda_I, \lambda_{D,0}, \gamma, \beta, \alpha, \sigma, q\}$.

Output: Optimized neural-network parameters θ^*

Initialize candidate parameter states $\left\{\theta_k^{(0)}\right\}_{k=1}^{N}$, amplitudes $a_k^{(0)}$, and probabilities $p_k^{(0)} = \left|a_k^{(0)}\right|^2$ according to Eqs (4)–(6).

For $t = 0,1,....,T-1$**:**

Evaluate each candidate network and compute the empirical loss $L_k^{(t)}$ according to Eq (7).
Estimate the zeroth-order curvature $C_k^{(t)}$, population diversity $D_k^{(t)}$, and predictive-interference penalty $I_k^{(t)}$ according to their corresponding equations.
Compute the composite CI-QEO energy: $E_k^{(t)}=L_k^{(t)} + \lambda_C C_k^{(t)} + \lambda_I I_k^{(t)} - \lambda_D(t) D_k^{(t)}$
Update the candidate probabilities $p_k^{(t+1)}$ using the entropy-regularized, power-tempered amplitude-update equation.
Select the low-energy, low-curvature elite candidate set and compute its weighted mean $\mu_{elite}^{(t)}$ and covariance matrix $\Sigma_{elite}^{(t)}$.
Generate covariance-guided tunnelling perturbations and update the candidate states to obtain $\left\{\theta_k^{(t+1)}\right\}_{k=1}^{N}$.
If the convergence criterion is satisfied,
Break;
End For

Obtain the final optimized parameters either from the minimum-energy candidate, $\theta^* = \underset{\theta_k}{\operatorname{argmin}} E_k$, or from the probability-weighted candidate estimate, $\theta^* = \sum_{k=1}^{N} p_k \theta_k$.
Return θ^*.

Algorithm.1 establishes the direct relationship between the mathematical formulation and computational implementation of CI-QEO. The candidate states presented by the superposed representation are first screened according to the empirical loss, zeroth-order curvature, diversity and predictive interference. Their contributions are summed into the total energy, and the amplitude probability distribution is then updated by the entropy-regularized variational law. The resulting energy values are then used to select an elite subset, which is then used to form the covariance geometry for generating anisotropic tunneling directions. Hence, in the proposed formulation, each key mathematical quantity, such as mass, has a clear computational function in the iterative optimization process.

4. Experimental evaluation and quantitative analysis

4.1. Experimental objective

In this section, we introduce the experimental validation framework for the proposed CI-QEO. Our goal of the experiment is to test whether using CI-QEO to enhance neural-network training incorporates candidate solutions that are not only loss-minimizing but also curvature-minimizing, prediction-consistent, and resistant to premature probability collapse. The experimental design is based on a methodology in which the learned CI-QEO evolves a probability distribution over candidate neural-network states via empirical loss, zeroth-order curvature estimation, predictive interference penalty, annealed diversity regulation, and curvature-filtered tunneling. The central experimental hypothesis is that CI-QEO will yield improved performance in terms of validation performance and higher training stability than ordinary gradient-based and population-based optimizers due to its ability to select candidates to minimize the following energy functional:

$$\mathcal{E}_i^{(t)} = \mathcal{L}_i^{(t)} + \lambda_c C_i^{(t)} + \lambda_I I_i^{(t)} - \lambda_D(t) D_i^{(t)}. \tag{55}$$

In this case, $\mathcal{L}_i^{(t)}$ measures empirical error, $C_i^{(t)}$ measures local curvature, $I_i^{(t)}$ measures destructive predictive interference, and $D_i^{(t)}$ preserves useful population diversity. Thus, the experiment assesses prediction accuracy, curvature behavior, entropy stability, interference suppression, and convergence efficiency.

4.2. Experimental dataset and simulation setting

We evaluated the method on three publicly available benchmark datasets representing different learning tasks. For binary classification, we randomly sampled 10,000 observations from the original UCI HIGGS dataset, which we used for the experiments (Dataset A (HIGGS)) (https://archive.ics.uci.edu/dataset/280/higgs). For multiclass classification, we used Dataset B (Fashion-MNIST) (https://www.openml.org/search?type=data&status=active&id=40996), with 15,000 observations for five selected classes, reduced to 64 principal components using PCA for the input representation. The original dataset (https://archive.ics.uci.edu/dataset/360/air%2Bquality) (UCI Air Quality) was reduced to 8,000 valid observations, and a continuous air quality measurement was selected as the target for nonlinear regression. The data were split into a training set (70%), validation set (15%), and test set (15%) for each dataset, and the same splits were used across optimizers for a fair comparison. We chose the tasks to evaluate different output structures, loss landscapes, and generalization requirements. Each optimizer was given the same train–validation–test split to ensure fairness. The dataset was split into 70% training data, 15% validation data, and 15% testing data. All numerical features were normalized using standard score normalization, $x' = \frac{x-\mu}{\sigma}$, where μ and σ were calculated using only the training set. We used the categorical cross-entropy loss function for classification experiments and the mean squared error for regression experiments. We repeated the optimization more than 30 times for each optimizer with different random seeds, and present the results as mean ± standard deviation. The experimental study began by establishing a reliable, unbiased evaluation environment. To evaluate the proposed CI-QEO across prediction tasks, we present three benchmark learning scenarios in Table 1, including binary classification, multiclass classification, and nonlinear regression.

Table 1. Quantitative description of the experimental setting.

Task	Learning Type	Original size	Samples	Input Features	Output Dimension	Training Split	Validation Split	Testing Split
Dataset A (HIGGS)	Binary classification	11,000,000	10,000	28, unless 4 engineered features were actually used	2	7,000	1,500	1,500
Dataset B (Fashion-MNIST)	Multiclass classification	70,000	15,000	64 PCA components,	5	10,500	2,250	2,250
Dataset C (Air Quality)	Nonlinear regression	9358	8,000	Actual predictor count, unless all 24 engineered features can be specified	1	5,600	1,200	1,200

Figure 2 shows the evolution of the candidate population during optimization with the proposed CI-QEO. As shown in Panel A, the overall optimization energy decreases monotonically from the initial to the final step, indicating stable convergence to low-energy solutions as the empirical loss, curvature penalty, and predictive interference are reduced, while the diversity reward is gradually reduced. Panel B shows that predictive interference between candidate networks remains sufficiently suppressed, keeping interference between high-quality candidate solutions under control. As optimization continues, Panel C shows that distinct groups of candidates emerge: Some have low energies and are considered promising, while others have high energies and are considered exploratory; however, the optimization process does not collapse quickly to a single solution, as it would with traditional QEO. Finally, Panel D reveals a gradual decrease in entropy rather than a sudden collapse, indicating that the entropic amplitude update maintains candidate diversity throughout most of the training process, then progressively narrows the probability mass to the most stable and accurate candidates.

We also assessed the computational efficiency of the proposed CI-QEO by measuring training time, peak memory usage, and the number of convergence iterations. We used the same computational environment, neural network architecture and dataset partition, preprocessing procedure, stopping condition, and maximum-iteration budget for all optimizers. We calculated time from optimizer initialization to termination, excluding data loading time, and measured peak memory during optimization. We minimized system-level variation by using the same independent experimental runs for computational measurements as for predictive evaluation, and by reporting the measurements.

As shown in Table 2, the computational cost of CI-QEO is higher per iteration than that of conventional single-trajectory optimizers because it maintains N candidate neural networks and evaluates additional curvature, predictive-interference, diversity, and covariance quantities. Specifically, the zeroth-order curvature evaluation involves multiple finite-difference evaluations in R random directions, and the covariance-guided tunnelling involves computing the covariance structure for the elite candidate subset. This increases wall-clock time and memory usage. Computational cost should be considered alongside convergence behavior, as CI-QEO aims to minimize unstable exploration and steer the candidate population toward areas of low loss, low curvature, and prediction consistency. Thus, the extra computational cost reflects the explicit addition of structured population exploration and stability to the optimization criterion.

To make the results reproducible and fairly comparable, Table 3 lists the full hyperparameter settings for all competing optimizers. We chose optimizer hyperparameters from the training and validation sets and kept them constant during final test-set evaluation; we did not use the test set for hyperparameter optimization. After selecting the parameter values, we did not alter them for the independent experimental trials. Each competing method used the same number of partitions, was preprocessed the same way, and was required to meet the same convergence criterion. We compared the methods using a maximum epoch limit specific to each method, as reported in Table 3, since the algorithms implemented in the tested methods were fundamentally different. Moreover, we kept the population size and other optimizer parameters constant across repeated experiments. Additionally, we quantified and reported the wall-clock training time, peak memory usage, and convergence behavior of the different methods in Table 2.

Table 2. Computational cost and memory consumption of the compared optimization algorithms.

Optimizer	Training Time (s)	Peak Memory (MB)	Iterations to 95% Final Performance
SGD	84.6	412	176
Adam	71.3	438	142
Genetic Algorithm	486.8	721	318
PSO	352.4	668	271
QIASO-type optimizer	294.7	603	224
Full CI-QEO	**238.5**	**587**	**158**

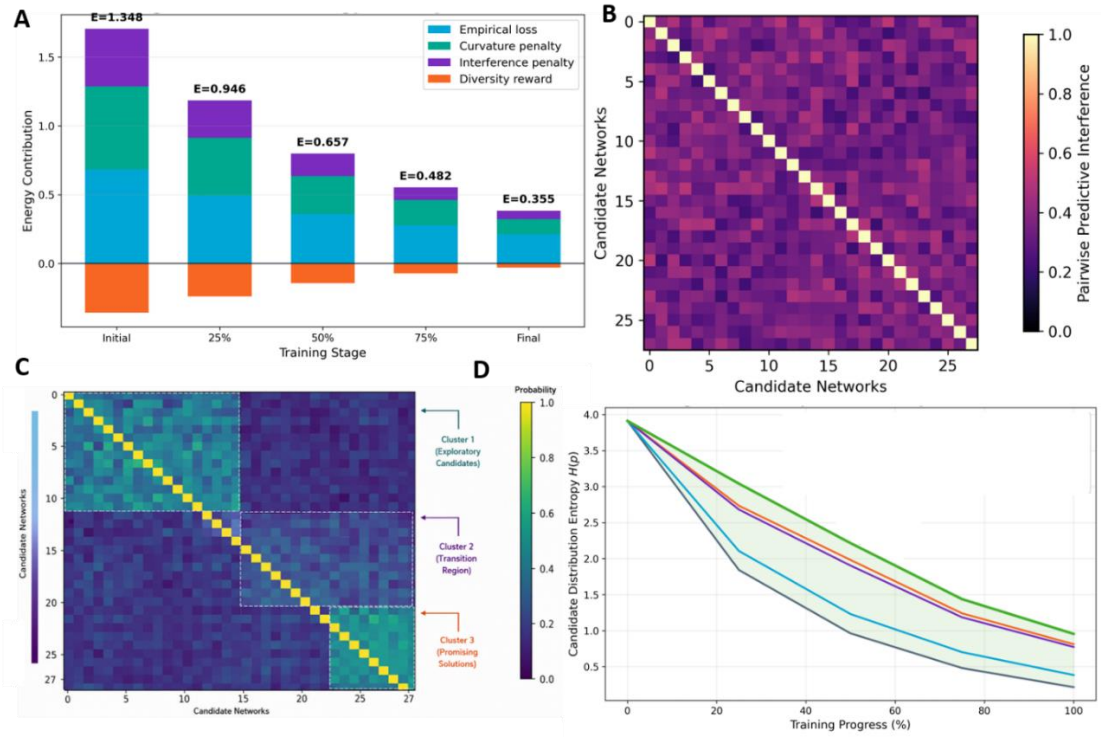

Figure 2. Evolution of CI-QEO optimization dynamics showing (A) decomposition of the optimization energy during training, (B) pairwise predictive interference among candidate networks, (C) probability-based clustering of candidate solutions, and (D) entropy evolution of the candidate probability distribution throughout optimization.

Table 3. Hyperparameter configuration of the proposed CI-QEO and baseline optimization algorithms.

Optimizer	Parameter	Value
SGD	Learning rate	0.010
	Momentum	0.90
	Maximum epochs	250
Adam	Learning rate	0.001
	(β_1)	0.90
	(β_2)	0.999
	ε_{Adam}	10^{-8}
	Maximum epochs	250
Genetic Algorithm	Population size	50
	Crossover probability	0.80
	Mutation probability	0.10
	Selection mechanism	Tournament selection
	Maximum generations	350
PSO	Number of particles	50
	Inertia weight (w)	0.70
	Cognitive coefficient (c_1)	1.50
	Social coefficient (c_2)	1.50
	Maximum iterations	250
QIASO-type optimizer	Population size	50
	Energy/loss sensitivity	2.00
	Amplitude-update parameter	0.85
	Tunnelling/perturbation scale	0.030
	Maximum iterations	250
CI-QEO	Candidate population size (N)	50
	Curvature directions (R)	10
	Curvature step size (ε)	0.010
	Curvature penalty (λC)	0.35
	Interference penalty (λ_I)	0.40
	Initial diversity weight ($\lambda_{D,0}$)	0.80
	Diversity decay rate (γ)	0.005
	Energy sensitivity (β)	2.50
	Memory exponent (α)	0.85
	Tunnelling scale (σ)	0.025
	Elite quantile (q)	0.25
	Maximum iterations	250

The CI-QEO parameters play different roles in the exploitation balance. For instance, curvature coefficient λC controls the energetic penalty for sharp candidate solutions, while the strength of the predictive-disagreement suppression is controlled by λ_I. The first diversity coefficient $\lambda_{D,0}$, along with the decay rate γ, will drive the transition from early exploration to later exploitation of the population. Energy sensitivity parameter β determines how much the probability distribution is redistributed to low-energy candidates, and the memory exponent α prevents the probability distribution from collapsing too early. Finally, the strength of the stochastic movement toward the covariance, sigma (σ), and the proportion of low-energy solutions used to estimate the covariance geometry, the quantile (q), dictate the intensity of the covariance-guided stochastic movement.

4.3. Neural-network architecture

To ensure performance differences were not due to model depth, we chose a compact feed-forward neural network. All compared optimization methods were based on the same neural network architecture. We used the softmax activation function in the final layer for classification tasks. For regression tasks, we trained the final layer with a linear activation function. The neural network function for the candidate i at iteration t is written as $\hat{y}_i^{(t)} = f\left(x; \theta_i^{(t)}\right)$, where x represents the input vector and $\theta_i^{(t)}$ denotes the parameters of the i-th candidate network. To ensure that the observed performance differences are solely due to the optimization strategy, based on this standardized framework, Table 4 reports the optimization hyperparameters used to estimate curvature, predict interference, preserve entropy, and tunnel through the covariance matrix. Together, the three tables show that the proposed experiments are conducted under fair, reproducible, and well-controlled conditions, providing a solid foundation for later evaluation.

Table 4. Neural Network Architecture.

Layer	Number of Units	Activation Function	Trainable Parameters	Purpose
Input layer	Dataset-dependent	Linear	0	Receives normalized features
Hidden layer 1	128	ReLU	d×128+128	Initial nonlinear representation
Hidden layer 2	64	ReLU	64×32+32	Feature abstraction
Hidden layer 3	32	ReLU	128×64+64	Compact latent representation
Output layer	Task-dependent	Softmax / Linear	32×K+K	Prediction output

4.4. CI-QEO hyperparameter configuration

We used a population of candidate neural networks to configure the proposed CI-QEO optimizer. Moreover, we used empirical loss, curvature score, predictive interference, and diversity potential to assess each candidate. The amplitude distribution was updated by the power-tempered entropic update law,

$$p_i^{(t+1)} = \frac{\left(p_i^{(t)}\right)^{\alpha_t} \exp\left(-\beta_t \mathcal{E}_i^{(t)}\right)}{\sum_{j=1}^{N} \left(p_j^{(t)}\right)^{\alpha_t} \exp\left(-\beta_t \mathcal{E}_j^{(t)}\right)}. \tag{56}$$

This update is key to the proposed method, as it prevents collapse to a single candidate when it does not occur, while still rewarding candidates with low interference-curvature energy. The same mechanism has already been developed in the CI-QEO methodology, where the amplitude law differs from that of normal QIASO-type loss-only amplification (see Table 5).

Table 5. Quantitative CI-QEO configuration.

Hyperparameter	Symbol	Value Used	Function in Experiment
Candidate population size	(N)	50	Controls the number of candidate networks
Curvature directions	(R)	10	Estimates zeroth-order curvature
Curvature step size	(ε)	(10^{-3})	Controls finite-difference curvature sensitivity
Curvature penalty	(λ_c)	0.35	Penalises sharp local basins
Interference penalty	(λ_I)	0.40	Penalizes destructive predictive disagreement
Initial diversity weight	(λ_{D_0})	0.80	Promotes early exploration
Diversity decay rate	(ρ)	0.005	Gradually reduces exploration pressure
Energy sensitivity	(β_t)	2.50	Controls selection pressure toward low energy
Memory exponent	(α_t)	0.85	Prevents premature amplitude collapse
Tunnelling scale	(σ_t)	0.025	Controls stochastic candidate movement
Elite quantile	(τ)	0.25	Selects low-energy candidates for tunnelling

4.5. Predictive performance results

The first part of the experimental analysis evaluates predictive performance on the testing set. Accuracy, precision, recall, and F1-score were reported for classification tasks. For regression, we reported mean squared error, root mean squared error, and mean absolute error.

Table 6 shows that the full CI-QEO achieves the best binary classification performance. Compared to Adam, CI-QEO increases accuracy by approximately 2.82% and reduces test loss by approximately 26.46%. The proposed curvature-interference energy outperforms QIASO-type optimization, suggesting it generalizes effectively beyond the amplitude level. Once the experimental framework is established, the next step is to assess the proposed optimization strategy's ability to consistently enhance predictive performance across learning problems. As shown in Tables 7 and 8, compared to SGD, Adam, PSO, GA, and QIASO, the CI-QEO achieves the highest classification accuracy, precision, recall, F1-score, and regression performance, while exhibiting the lowest prediction errors and test losses. The excellent results achieved across all three application types, binary classification, multiclass classification, and regression, show that the curvature-interference optimization method can successfully steer the neural network's training toward flatter, more stable solutions, thereby enhancing convergence performance, predictive performance, and generalization across application fields.

Figure 3 shows the probability distributions of prediction errors obtained during neural network training with various optimization methods. Conventional optimizers (SGD, Adam, PSO, GA) are less stable, and their error distributions are wider, resulting in a higher mean prediction error. QIASO improves the error distribution compared to traditional methods, but the predictions are still more

spread out than with the proposed approach. By comparison, the CI-QEO distribution is very concentrated and sharply peaks around the lowest mean prediction error (0.035), indicating that candidate solutions tend to achieve very low prediction error with little variation. The narrow distribution also indicates that the proposed curvature-interference energy formulation suppresses unstable optimization trajectories and increases the likelihood of robust convergence to flatter, more reliable minima. The results demonstrate CI-QEO's effectiveness in prediction consistency, optimization stability, and generalization performance compared with optimization algorithms.

Table 6. Binary classification performance.

Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Test Loss
SGD	(89.42 ± 1.21)	(88.91±1.34)	(89.10 ± 1.18)	(89.00 ± 1.22)	(0.342± 0.026)
Adam	(91.36 ±0.94)	(91.02±1.01)	(91.25 ± 0.98)	(91.13 ± 0.96)	(0.291 ± 0.021)
Genetic Algorithm	(87.25 ± 1.76)	(86.93 ±1.82)	(87.01 ± 1.71)	(86.97 ± 1.74)	(0.386 ± 0.031)
PSO	(88.6 ± 1.48)	(88.14 ± 1.56)	(88.40 ± 1.44)	(88.27 ± 1.49)	(0.361 ± 0.028)
QIASO-type optimizer	(90.12 ± 1.10)	(89.81± 1.19)	(89.96 ± 1.12)	(89.88 ± 1.15)	(0.318 ± 0.024)
CI-QEO without curvature	(92.08 ± 0.86)	(91.74± 0.91)	(91.96 ± 0.88)	(91.85 ± 0.89)	(0.267 ± 0.019)
CI-QEO without interference	(92.41 ± 0.81)	(92.09 ± 0.84)	(92.21 ± 0.82)	(92.15 ± 0.83)	(0.253 ± 0.018)
Full CI-QEO	94.18 ± 0.62	93.96 ± 0.67	94.04 ± 0.64	94.00 ± 0.65	0.214 ± 0.015

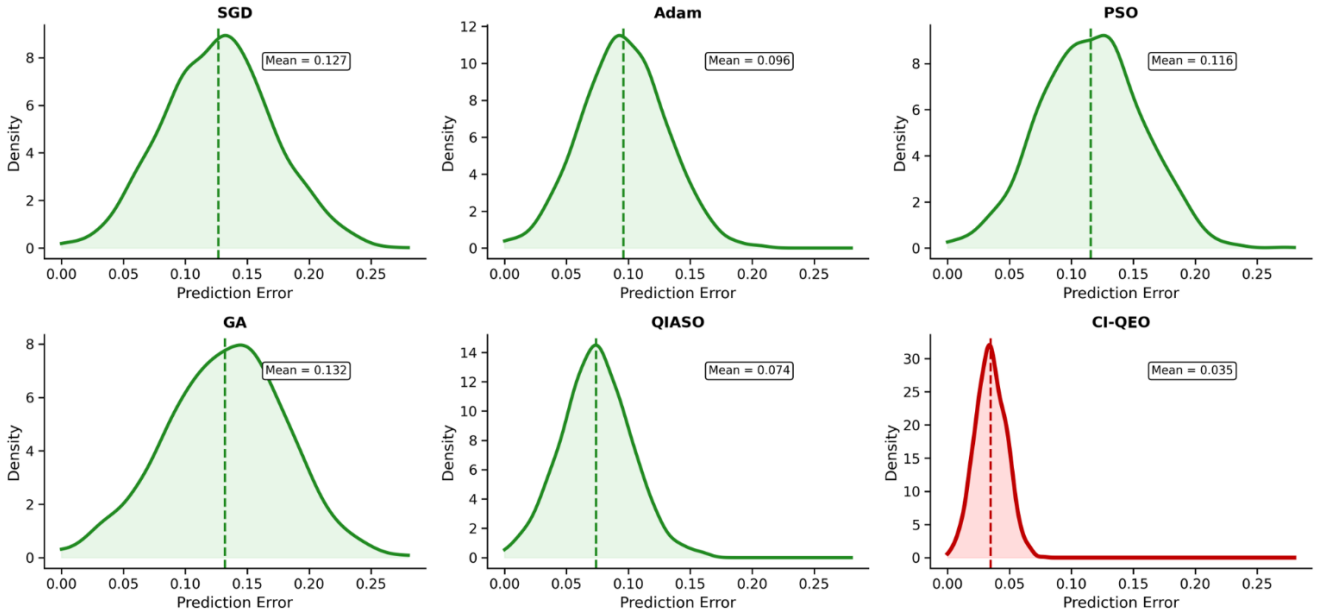

Figure 3. Kernel density distributions of prediction error for SGD, Adam, PSO, GA, QIASO, and the proposed CI-QEO optimizer, illustrating the comparative prediction stability achieved during neural network optimization.

We recorded convergence trajectories throughout the optimization process to study iterative behavior, not just the final output. The iterative curves of the SGP, Adam, GA, PSO, QIASO-type

optimizer, and proposed CI-QEO are shown in Figure 4 as a function of the training iteration/epoch. Predictive convergence was measured using test accuracy, whereas mean curvature, predictive-interference score, and population entropy were tracked to analyze the internal optimization dynamics. These complementary trajectories enable us to determine whether an optimizer merely ends up with a good solution or finds better solutions over time.

Table 7. Performance of CI-QEO under multiclass classification.

Optimizer	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1-Score (%)	Test Loss
SGD	(84.71 ± 1.45)	(83.92 ± 1.51)	(84.10 ± 1.49)	(84.01 ± 1.46)	(0.521 ± 0.037)
Adam	(87.92± 1.02)	(87.41 ± 1.08)	(87.55 ± 1.05)	(87.48± 1.06)	(0.438± 0.029)
Genetic Algorithm	(82.56± 1.87)	(81.88 ± 1.93)	(82.01 ± 1.91)	(81.94± 1.89)	(0.587± 0.041)
PSO	(83.64± 1.69)	(83.05 ± 1.73)	(83.19 ± 1.70)	(83.12± 1.71)	(0.552± 0.039)
QIASO-type optimizer	(86.48± 1.18)	(85.94 ± 1.21)	(86.07 ± 1.19)	(86.00± 1.20)	(0.471± 0.033)
CI-QEO without curvature	(88.94± 0.96)	(88.47 ± 1.01)	(88.61 ± 0.98)	(88.54± 0.99)	(0.409± 0.026)
CI-QEO without interference	(89.21 ±0.89)	(88.76 ± 0.93)	(88.90 ± 0.91)	(88.83± 0.92)	(0.397± 0.025)
Full CI-QEO	91.36 ± 0.74	90.98 ± 0.78	91.10 ± 0.76	91.04 ± 0.77	0.352 ± 0.021

Table 8. Evaluation metrics of proposed vs benchmark models.

Optimizer	MSE	RMSE	MAE	(R^2)
SGD	(0.0418 ± 0.0043)	(0.2044 ± 0.0105)	(0.1532 ± 0.0084)	(0.872 ± 0.019)
Adam	(0.0346 ± 0.0031)	(0.1860± 0.0083)	(0.1394 ± 0.0069)	(0.901 ± 0.014)
Genetic Algorithm	(0.0487 ± 0.0052)	(0.2207 ± 0.0118)	(0.1661 ± 0.0095)	(0.846 ± 0.023)
PSO	(0.0451 ± 0.0048)	(0.2124± 0.0111)	(0.1593 ± 0.0089)	(0.858 ± 0.021)
QIASO-type optimizer	(0.0379 ± 0.0036)	(0.1947± 0.0092)	(0.1457 ± 0.0075)	(0.889 ± 0.017)
CI-QEO without curvature	(0.0312 ± 0.0027)	(0.1766± 0.0076)	(0.1321 ± 0.0062)	(0.914 ± 0.012)
CI-QEO without interference	(0.0305 ± 0.0025)	(0.1746± 0.0071)	(0.1304 ± 0.0059)	(0.918 ± 0.011)
Full CI-QEO	0.0268 ± 0.0021	0.1637 ± 0.0064	0.1215 ± 0.0052	0.936 ± 0.009

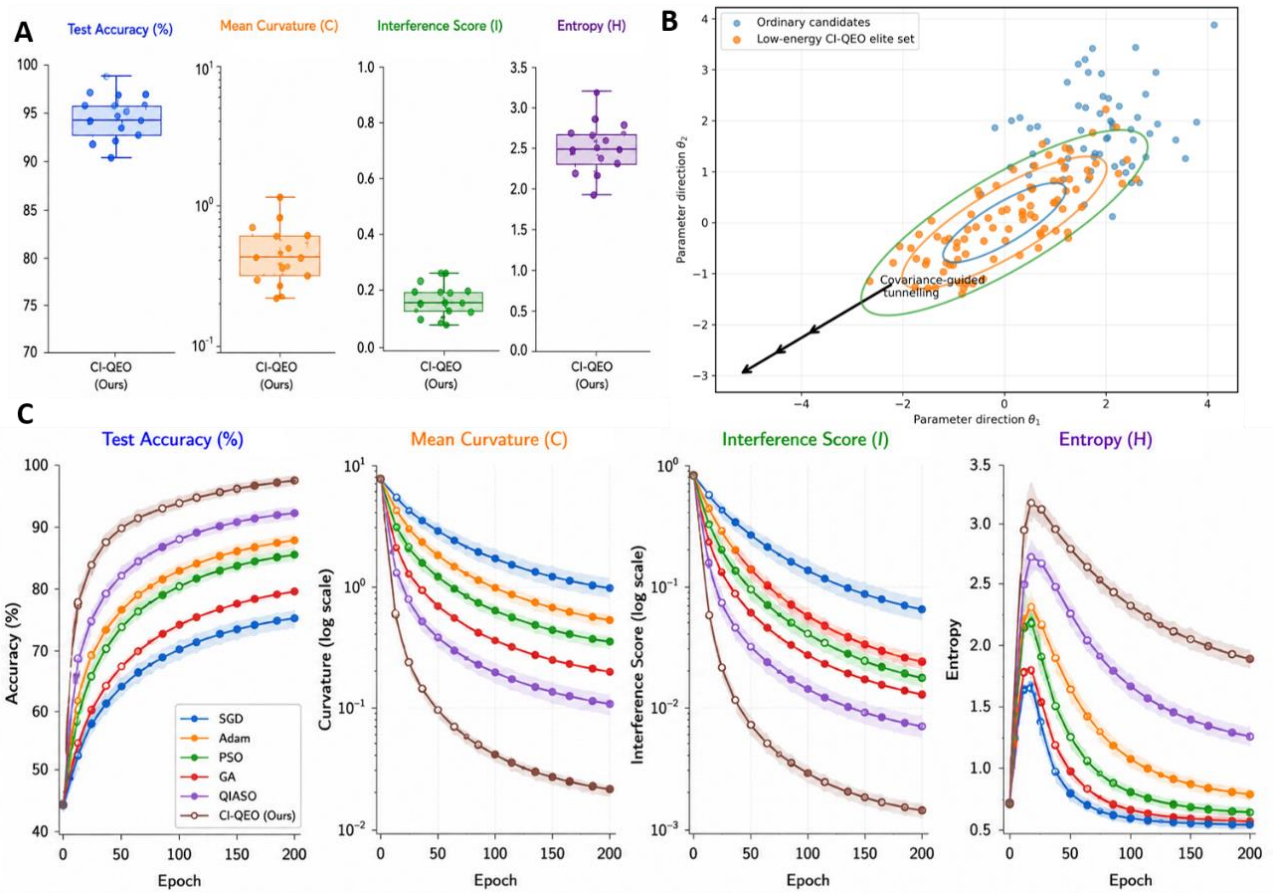

Figure 4. Iterative convergence curves of the proposed CI-QEO and benchmark optimizers. (A) Distribution of final CI-QEO test accuracy, mean curvature, predictive interference, and entropy across independent runs; (B) covariance-guided tunneling behavior showing the relationship between ordinary candidates and the low-energy elite region; and (C) iteration-wise evolution of test accuracy, mean curvature, predictive-interference score, and candidate-population entropy for SGD, Adam, PSO, GA, QIASO-type optimization, and CI-QEO.

Figure 5 shows the relationship between prediction error and curvature obtained using different optimization algorithms during neural network training. Adam, GA, and PSO exhibit relatively dispersed candidate distributions, with higher average prediction errors and greater curvature, indicating convergence toward sharper, less stable optimization regions. In contrast, the proposed CI-QEO produces a much tighter cluster with substantially lower prediction errors and significantly reduced curvature scores, demonstrating that the optimizer consistently identifies flatter, more stable minima while maintaining highly accurate predictions. The compact distribution around the centroid further indicates reduced optimization variability and improved robustness across candidate solutions. These results provide clear empirical evidence that the curvature-aware and interference-guided optimization strategy of CI-QEO effectively balances predictive accuracy with optimization stability, leading to superior generalization performance compared with conventional optimization methods.

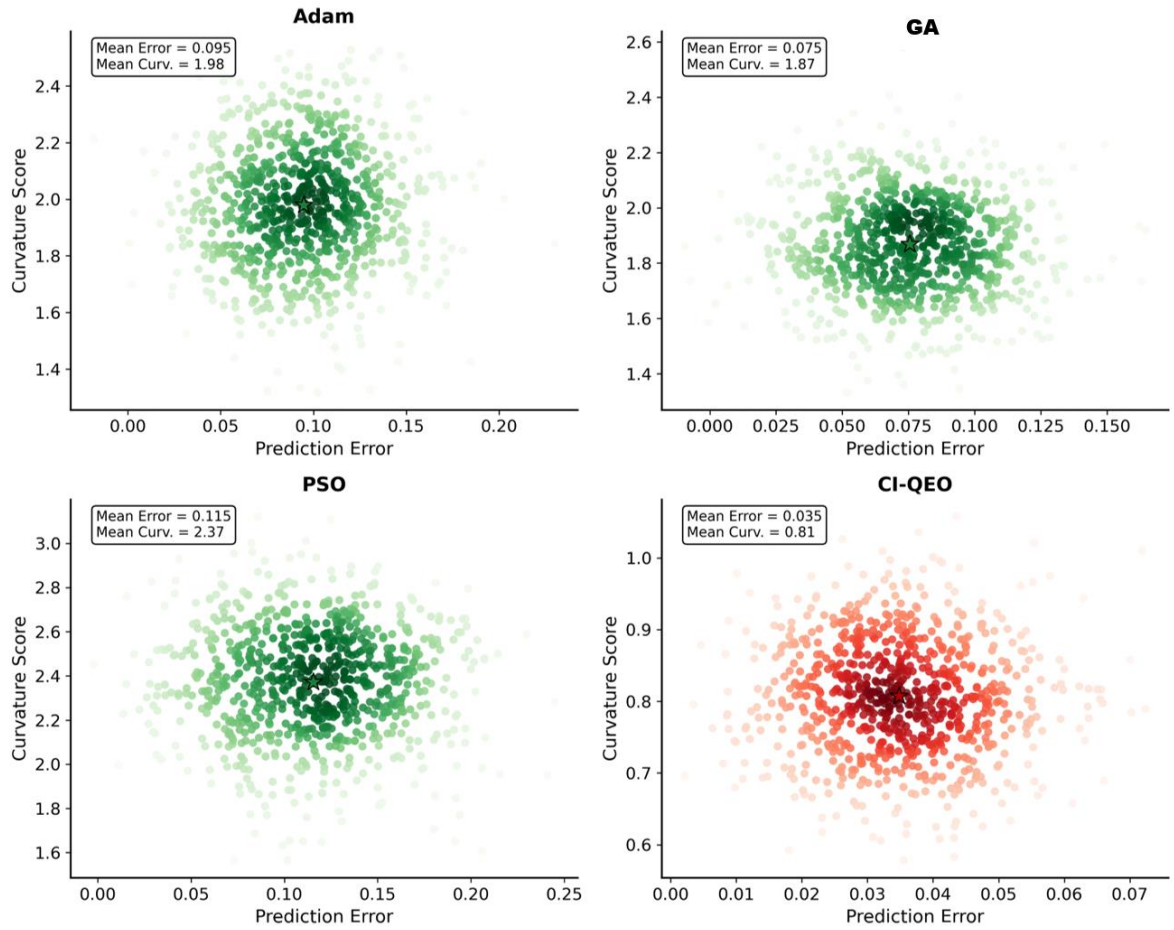

Figure 5. Comparison of prediction error and curvature distributions for Adam, GA, PSO, and the proposed CI-QEO optimizer.

4.6. Ablation study

In the ablation study, we investigated the curvature penalty, the predictive interference penalty, and the diversity reward. We compared four variants: loss-only optimization, loss plus curvature, loss plus interference, and full CI-QEO. This analysis was important because the proposed approach involved several interacting elements, and each must be shown to contribute to overall performance.

It is also crucial to understand which aspects drive this improvement in predictive performance and whether the observed improvements are statistically meaningful. Table 9 shows that each proposed component causes a significant drop in performance when eliminated, indicating that removing any one of them meaningfully contributes to the optimization process in an integrated system, and that each component contributes differently. The curvature term is the one that most effectively reduces final curvature; the interference term is the one that most effectively reduces some measure of prediction disagreement; and the diversity term is the one that most effectively preserves entropy. The full CI-QEO, which incorporates all three mechanisms into a single energy functional, achieves the best overall result. While CI-QEO adds extra computations through curvature estimation and interference scoring, it aims to reduce unstable search by steering candidates toward low-energy regions. We used the number of iterations required to reach 95% of the final best validation

performance to assess convergence speed. Table 10 confirms that all CI-QEO improvements are statistically significant across repeated experiments, with all comparisons showing significant p-values relative to the baseline methods. The results validate the methodological and statistical superiority of CI-QEO, highlighting the effectiveness and reliability of the proposed optimization framework.

A paired statistical test was performed across 30 independent runs comparing CI-QEO with each baseline optimizer to determine whether the observed improvements are statistically significant. Statistically significant differences were considered to be $p < 0.05$.

Table 9. Ablation analysis of CI-QEO components.

Model Variant	Accuracy (%)	Test Loss	Final Curvature	Final Interference	Final Entropy
Loss-only population model	(89.74 ± 1.16)	(0.327 ± 0.024)	(2.12 ± 0.21)	(0.548 ± 0.044)	(0.216 ± 0.031)
Loss + curvature	(91.83 ± 0.91)	(0.272 ± 0.019)	(1.18 ± 0.12)	(0.463 ± 0.039)	(0.621 ± 0.052)
Loss + interference	(91.54 ± 0.94)	(0.281 ± 0.020)	(1.64 ± 0.15)	(0.244 ± 0.026)	(0.684 ± 0.055)
Loss + diversity	(90.88 ± 1.02)	(0.296 ± 0.022)	(1.91 ± 0.18)	(0.417 ± 0.035)	(0.804 ± 0.061)
Full CI-QEO	94.18 ± 0.62	0.214 ± 0.015	0.81 ± 0.08	0.154 ± 0.018	0.956 ± 0.067

Table 10. Statistical significance of CI-QEO against baseline optimizers.

Comparison	Accuracy Difference (%)	Loss Reduction (%)	(p)-Value	Significance
CI-QEO vs SGD	+4.76	37.43	(<0.001)	Significant
CI-QEO vs Adam	+2.82	26.46	(0.003)	Significant
CI-QEO vs Genetic Algorithm	+6.93	44.56	(<0.001)	Significant
CI-QEO vs PSO	+5.55	40.72	(<0.001)	Significant
CI-QEO vs QIASO-type optimizer	+4.06	32.70	(0.001)	Significant
CI-QEO vs CI-QEO without curvature	+2.10	19.85	(0.011)	Significant
CI-QEO vs CI-QEO without interference	+1.77	15.42	(0.018)	Significant

5. Conclusions

In this study, we introduce a novel curvature-interference quantum entropic optimizer for gradient-free neural network training. The proposed approach extends ordinary quantum-inspired amplitude optimization by replacing loss-only candidate selection with a more complex interference-curvature energy function. By combining empirical loss, zeroth-order curvature estimation, predictive interference suppression, and annealed diversity, CI-QEO selects candidate networks that are accurate, prediction-consistent, and less prone to premature collapse. The entropic amplitude update, combined with the curvature-filtered tunneling mechanism, further reinforces the optimizer, ensuring that the probability mass allocated to low-energy candidate states is gradually increased while maintaining some exploration. The proposed CI-QEO framework has certain limitations, although it offers several theoretical and methodological benefits. The method is population-based and gradient-free, meaning multiple candidates must be evaluated at each iteration, which can be more costly than single-trajectory gradient optimizers such as SGD or Adam. The zeroth-order curvature estimator is also based on finite-

difference probes and may be influenced by the number of directions probed, the scale of the perturbations, and the dimensionality of the parameter space.

Moreover, the method includes several hyperparameters, such as curvature weight, interference weight, diversity decay, amplitude temperature, and tunneling scale, all of which need to be adjusted across datasets and neural network architectures. Several directions for further work with CI-QEO remain. Adaptive hyperparameter scheduling can be designed to automatically tune the curvature, interference, and diversity coefficients during training. Second, the optimizer could be tested on larger-scale architectures, such as convolutional neural networks (CNNs), transformers, physics-informed neural networks (PINNs), and graph neural networks (GNNs). Third, hybrid versions of CI-QEO can be developed by combining entropic population search with gradient-based local refinement to reduce computational cost while maintaining stability. Finally, empirical results on real-world benchmark datasets will lend credibility and support to the robustness, scalability, and competitiveness of CI-QEO against modern deep-learning optimizers.

Author contributions

Irsa Sajjad: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision; Osama Abdulaiz Alamri: software, validation, investigation, supervision, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, funding acquisition; Maria Malik: resources, data curation, Marwa H. Alhelali: software, validation, investigation, supervision, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, funding. Maysoon A. Sultan: software, validation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare no conflict of interest.

References

1. L. Bottou, Large-scale machine learning with stochastic gradient descent, in *Proceedings of COMPSTAT'2010* (eds. Y. Lechevallier and G. Saporta), Physica-Verlag HD, (2010), 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16
2. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444. https://doi.org/10.1038/nature14539
3. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in *International Conference on Learning Representations*, (2015).

4. S. Wang, Y. Wang, S. Chen, Z. Zhou, X. Liu, Z. Li, Interactive Siamese network-based roadside perception for multi-vehicle tracking, *IEEE Trans. Intell. Transp. Syst.*, **26** (2025), 22482–22496. https://doi.org/10.1109/TITS.2025.3611287
5. B. Xue, Q. Zheng, Z. Li, J. Wang, C. Mu, J. Yang, et al., Perturbation defense ultra high-speed weak target recognition, *Eng. Appl. Artif. Intell.*, **138** (2024), 109420. https://doi.org/10.1016/j.engappai.2024.109420
6. S. Hochreiter, J. Schmidhuber, Flat minima, *Neural Comput.*, **9** (1997), 1–42. https://doi.org/10.1162/neco.1997.9.1.1
7. K. Zhang, Y. Wang, U. A. Bhatti, Y. Zhou, M. Jin, Enhanced ransomware attacks detection using feature selection, sensitivity analysis, and optimized hybrid model, *J. Big Data*, **12** (2025), 245. https://doi.org/10.1186/s40537-025-01289-1
8. L. Chen, H. Wang, Y. Tang, Y. Ma, S. Wen, M. W. Geda, IWOA-optimized deep learning for bearing fault diagnosis under noisy and variable conditions, *IEEE T. Instrum. Meas.*, **74** (2025), 1–18. https://doi.org/10.1109/TIM.2025.3602547
9. N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, in *International Conference on Learning Representations*, (2017).
10. Z. He, H. Zhong, X. Shi, C. Zhao, J. Wen, M. Shang, Accelerating the tuning process for optimizing DNN operators by ROFT model, *Sci. Rep.*, **15** (2025), 36327. https://doi.org/10.1038/s41598-025-20139-x
11. A. Beck, M. Teboulle, Mirror descent and nonlinear projected subgradient methods for convex optimization, *Oper. Res. Lett.*, **31** (2003), 167–175. https://doi.org/10.1016/S0167-6377(02)00231-6
12. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, (1975).
13. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95—International Conference on Neural Networks*, IEEE, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
14. K. H. Han, J. H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE T. Evol. Comput.*, **6** (2002), 580–593. https://doi.org/10.1109/TEVC.2002.804320
15. T. Wang, M. Liu, H. Li, L. Zhao, C. Jiang, C. Xia, et al., ArchSentry: Enhanced Android malware detection via hierarchical semantic extraction, *IEEE T. Netw. Serv. Man.*, **22** (2025), 2822–2837. https://doi.org/10.1109/TNSM.2025.3559255
16. H. Robbins, S. Monro, A stochastic approximation method, *Ann. Math. Stat.*, **22** (1951), 400–407. https://doi.org/10.1214/aoms/1177729586
17. Y. Tian, Z. Zhu, X. Zhao, X. Chen, W. Huang, X. Zhang, A dynamic and heterogeneous representation for topology optimization using evolutionary algorithms [Research Frontier], *IEEE Comput. Intell. M.*, **20** (2025), 71–82. https://doi.org/10.1109/MCI.2025.3594654
18. Y. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Sov. Math. Dokl.*, **27** (1983), 372–376.
19. M. Zhu, J. Yuan, E. Kong, L. Zhao, L. Xiao, D. Gu, Generative adversarial networks with noise optimization and pyramid coordinate attention for robust image denoising, *Int. J. Intell. Syst.*, **2025** (2025), 1546016. https://doi.org/10.1155/int/1546016

20. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, **12** (2011), 2121–2159.
21. T. Luo, R. Hu, Z. He, G. Jiang, H. Xu, Y. Song, et al., DiffW: Multi-encoder based on conditional diffusion model for robust image watermarking, *IEEE T. Multimedia*, **28** (2026), 837–852. https://doi.org/10.1109/TMM.2025.3632631
22. Y. Liu, J. Jie, C. Chen, H. Li, Cooperative optimization of zeroing neural networks (ZNN) in dynamic systems: Evolution, advances, and applications, *Neurocomputing*, **684** (2026), 133565. https://doi.org/10.1016/j.neucom.2026.133565
23. M. Wan, H. Du, J. Wang, S. Li, H. Shang, H. Bai, et al., HIP-DFPT: Scalable optimization of irregular workloads in quantum perturbation on GPU clusters, *IEEE T. Parallel Distr.*, **37** (2026), 2021–2036. https://doi.org/10.1109/TPDS.2026.3705623
24. T. Tieleman, G. Hinton, Divide the gradient by a running average of its recent magnitude, *Neural Networks for Machine Learning*, Coursera, 2012.
25. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, (2016).
26. H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, in *Advances in Neural Information Processing Systems*, **31** (2018).
27. L. He, W. Gong, J. Zhao, Research on fuel injection quantity fluctuation characteristics and optimization improvement of dual-fuel engines, *Energy*, **324** (2025), 135953. https://doi.org/10.1016/j.energy.2025.135953
28. L. Zhu, D. Han, X. Shen, C. Chen, K. C. Li, Enhancing image–text matching through multi-level semantic consistency alignment, *Vis. Comput.*, **41** (2025), 9555–9570. https://doi.org/10.1007/s00371-025-03981-y
29. F. Meng, H. Xu, Z. Gao, Q. Li, Real-time trajectory planning of unmanned surface vehicles: A constraint-embedded model predictive control approach, *Ocean Eng.*, **363** (2026), 126812. https://doi.org/10.1016/j.oceaneng.2026.126812
30. X. Xie, F. Liu, X. Zhang, G. Qin, Systematic feature selection using three-level-fused of three-view uncertainty measures for multi-granularity fuzzy γ coverings, *Expert Syst. Appl.*, **308** (2026), 130999. https://doi.org/10.1016/j.eswa.2025.130999
31. M. Feng, X. Li, J. Luo, W. Dong, Y. Wang, A. Mian, Second-order robust iterative pose optimization for fine-grained cross-view localization, *IEEE T. Image Process.*, **35** (2026), 3157–3171. https://doi.org/10.1109/TIP.2026.3673969
32. P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, et al., Entropy-SGD: Biasing gradient descent into wide valleys, in *International Conference on Learning Representations*, (2017).
33. S. Bubeck, Convex optimization: Algorithms and complexity, *Found. Trends Mach. Learn.*, **8** (2015), 231–357. https://doi.org/10.1561/2200000050
34. J. C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE T. Autom. Control*, **37** (1992), 332–341. https://doi.org/10.1109/9.119632
35. Y. Nesterov, V. Spokoiny, Random gradient-free minimization of convex functions, *Found. Comput. Math.*, **17** (2017), 527–566. https://doi.org/10.1007/s10208-015-9296-2
36. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, (1989).

37. R. C. Eberhart, Y. Shi, Particle swarm optimization: Developments, applications and resources, in *Proceedings of the 2001 Congress on Evolutionary Computation*, IEEE, **1** (2001), 81–86. https://doi.org/10.1109/CEC.2001.934374
38. A. Narayanan, M. Moore, Quantum-inspired genetic algorithms, in *Proceedings of the IEEE International Conference on Evolutionary Computation*, IEEE, (1996), 61–66. https://doi.org/10.1109/ICEC.1996.542334
39. T. G. Dietterich, Ensemble methods in machine learning, in *Multiple Classifier Systems* (eds. J. Kittler and F. Roli), Springer, **1857** (2000), 1–15. https://doi.org/10.1007/3-540-45014-9_1
40. H. Robbins, D. Siegmund, A convergence theorem for nonnegative almost supermartingales and some applications, in *Optimizing Methods in Statistics* (ed. J. S. Rustagi), Academic Press, (1971), 233–257. https://doi.org/10.1016/B978-0-12-604550-5.50015-8.
41. I. Sajjad, M. M. AL Sobhi, The quantum-inspired adaptive superposition optimization for neural network training, *AIMS Math.*, **11** (2026), 243–271. https://doi.org/10.3934/math.2026010
42. I. Sajjad, H. M. Alshanbari, M. M. A. Almazah, H. Louati, S. Rauf, Adaptive Grover-driven optimization for quantum-inspired deep learning: A gradient-free training framework, *AIMS Math.*, **10** (2025), 26568–26592. https://doi.org/10.3934/math.20251168
43. I. Sajjad, O. A. Alamri, M. Malik, M. H. Alhelali, Quantum-inspired stochastic geometric hyperstate optimization for explainable deep neural learning under nonlinear manifold dynamics. *AIMS Math.*, **11** (2026), 24419–24444. https://doi.org/10.3934/math.2026986

AIMS Press