AIMS Mathematics, 11(1): 243–271.
DOI: 10.3934/math.2026010
Received: 09 November 2025
Revised: 14 December 2025
Accepted: 23 December 2025
Published: 04 January 2026

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

*Research article*

# The quantum-inspired adaptive superposition optimization for neural network training

**Irsa Sajjad[1,*] and Mashail M. AL Sobhi[2]**

[1] Department of Mathematics, National University of Modern Languages, Islamabad, Pakistan
[2] Department of Mathematics, Umm-Al-Qura University, Makkah 24227, Saudi Arabia

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

**Abstract:** Training deep neural networks is often hindered by the fragility of gradient-based methods, which suffer from vanishing or exploding gradients, sensitivity to initialization, and entrapment in poor local minima. In response to these shortcomings, we introduce a new gradient-free algorithm called Quantum-Inspired Adaptive Superposition Optimization (QIASO), which views weight learning as a probabilistic superposition of candidate solutions, a fundamentally new optimization approach. In contrast to being dedicated to a single weight ensemble, QIASO maintains a distribution over several candidates, which are amplified and suppressed according to dynamically changing weights assigned to them. The variational formulation of the amplitude evolution leads to a KL-regularized formulation of their evolution, which generalizes statistical physics, information geometry, and online optimization viewpoints. To prevent invalid convergence, QIASO incorporates a stochastic perturbation operator based on quantum tunnelling into the optimizer, enabling the optimization process to overcome local minima on the loss surface and converge to the optimal solution. We provide theoretical bounds, monotone convergence of loss reduction, and almost-sure convergence to local optima with mild assumptions. Complexity analysis via empirical techniques suggests that QIASO scales more efficiently than Grover-based quantum-inspired algorithms and incurs no overhead in gradient computation compared to ADAM. The overall findings indicated that QIASO is a viable option for neural training, particularly when combined with other paradigms that utilise either large-scale or gradient-free approaches.

**Keywords:** quantum-inspired optimization; neural network training; superposition-based learning; amplitude dynamics; quantum tunneling perturbation; convergence analysis
**Mathematics Subject Classification:** 68Q05, 68Q12

**Notation**

- $K$— finite set of candidates, index $k = 1, \ldots, K$.
- $p^t = (p_1^t, \ldots\ldots, p_K^t)$— amplitude/probability vector at iteration $t$(lies in the probability simplex $\Delta_K$).
- $w_k$—weight vector for candidate $k$.
- $L(w_k) \equiv L_k$— loss (empirical risk) evaluated at candidate $w_k$. For clarity, we write $L_k$.
- $\langle p, L \rangle = \sum_{k=1}^{K} p_k L_k$ — expected loss under distribution $p$.
- $D_{KL}(q \parallel p) = \sum_{k=1}^{K} q_k \log\left(\frac{q_k}{p_k}\right)$ Kullback–Leibler divergence.
- $\eta > 0$— step (inverse temperature/learning) parameter.
- $\epsilon$— perturbation scale for tunnelling operator; $\xi \sim \mathcal{N}(0, \sigma^2 I)$.
- $p_0$ and $\alpha$— initial perturbation probability and decay rate (so $p_t = p_0 e^{-\alpha t}$).
- $\theta$— Candidate weight vector.
- P($\theta$) — Probability amplitude assigned to a candidate state.
- N — Dimensionality of the parameter vector.
- η — Learning-rate-like step parameter in KL-regularized update.
- λ — Regularization strength for KL term.
- $t(.)$— Tunneling perturbation operator.
- $\ell(.)$ — Loss function.

## 1. Introduction

Deep neural network optimization is a key issue in machine learning. Standard optimizers, such as stochastic gradient descent (SGD) and ADAM, have enabled significant advances. However, they are limited by, among other things, vanishing/exploding gradients [15], excessive sensitivity to initialization [6], and getting stuck in poor local minima. They become even more critical in large, highly nonconvex optimization landscapes, where gradient information is both erratic and ill-conditioned [7].

Quantum-inspired algorithms have been proposed in recent years as a promising avenue for optimizing non-classical gradient algorithms. Based on concepts of superposition, amplitude amplification, and tunneling, these methods search in parallel over a multitude of candidate solutions and probabilistically refine their probabilities of being correct [5], thereby gaining resilience against local-minima solutions [21]. All these methods share a common theoretical framework that was recently developed using Gibbs sampling techniques, mirror descent [19], and exponentiated gradient updates [16], with roots at the intersection between statistical physics, information geometry, and online learning [20].

Due to these advances, we develop Quantum-Inspired Adaptive Superposition Optimisation (QIASO), a maximally general trainable model: neural network training is recontextualized as an optimiser that probabilistically evolves a weight state. QIASO optimises the amplitudes of candidates with lower loss values and poor candidates by reducing the expected loss monotonically. Additionally, QIASO has enabled the optimizer [4] to avoid narrow basins of attraction and converge to stagnant parts of the loss landscape by introducing a stochastic operator perturbation [3] that resembles quantum tunnelling. The work is grounded in recent progress in quantum-inspired optimization [11,12] and

gradient-free learning [25,30], and extends these approaches to large-scale training of neural networks. QIASO connects probability Brownian motion to probable amplitude dynamics, adding stochastic perturbations and convergence guarantees, and is a principled and scalable algorithmic framework for non-convex gradient-free optimisation in machine learning applications, including those in modern deep learning.

The topic of optimizing deep neural networks has been extensively investigated, and gradient-based methods have been known to suffer from shortcomings from an early time. The vanishing gradient issue in recurrent neural networks was noted by Bengio et. al. [5]. Despite adaptive approaches such as ADAM, these methods remain vulnerable to initialization and convergence issues in nonconvex loss landscapes [7]. To overcome such challenges, gradient-free optimization techniques have been explored. Backpropagation-free learning is feasible with evolution strategies [25], random search techniques [10], and natural evolution strategies [13]. Likewise, zeroth-order optimization [17] has been proposed as a scalable alternative to gradient-based updates in the black-box case.

In tandem with these, the quantum-inspired algorithms have made a second sight of optimization. Needle-based studies on quantum annealing [2,11,14] and quantum adiabatic computation [8] have demonstrated how functional tunnelling dynamics can be used to escape local minima. Most recently, [3] discussed the feasibility of applying quantum annealing and tunnelling effects to challenging optimization landscapes. Classical analogues of these ideas have been inspired by, e.g., simulated annealing [22] and its variants, which mimic the effects of a quantum system without using physical quantum hardware. Theoretically, the relationship between optimization, statistical physics, and information geometry has already been well established. Mirror descent [19] and exponentiated gradient methods [15,16] demonstrate that Bregman divergences can govern probability distributions over the set of candidate solutions in a sensible way. These methods were further formalized in recent work on online convex optimization [2,23], which established connections with regret minimization guarantees.

Finally, there has been a growing interest in quantum machine learning [21,24], which seeks to integrate concepts from quantum mechanics and learning theory. Although most quantum algorithms are still tied to hardware, quantum-inspired algorithms have become a practical alternative, implementing the core principles of superposition, amplitude amplification, and tunnelling in entirely classical settings [28]. Sajjad et al. [8] proposed an adaptive Grover-based, gradient-free quantum-inspired deep learning optimizer that demonstrated greater robustness and improved training performance for deep neural networks. In summary, these threads of research converge on the idea that probabilistic updates propagated along distributions can circumvent the brittleness of deterministic gradient descent. The QIASO innovation directly follows this observation, combining amplitude evolution, KL-regularized updates, and stochastic tunnelling in a single optimizer for large-scale neural network training (see more references [9,18,20,29]).

Over the past few years, the combination of quantum-inspired algorithms with standard classical machine learning optimizers has become increasingly popular, demonstrating effectiveness in improving training stability and generalization in deep neural networks. Indicatively, a generalized consideration by AL Ajmi and Shoaib [1] suggests that quantum-inspired optimization algorithms could be more robust and efficient than classical optimizers in quantum machine learning. In the meantime, Si et al. [27] proposed the QSHO (Quantum Spotted Hyena Optimizer), demonstrating that quantum-inspired swarm algorithms are more effective at avoiding local minima in complex landscapes. Moreover, Rizvi et al. [23] also emphasized that hybrid quantum-classical vision models

leverage principles of superposition and interference to enhance feature learning in deep architectures. These new writings suggest a shift away from systems that rely solely on gradient-based optimizers to systems that explicitly model probabilistic amplitude evolution and stochastic perturbation dynamics. This paper presents a new gradient-free optimization paradigm for deep-network training that integrates quantum-inspired amplitude modulation with classical efficient computation. New quantum-inspired methods have investigated probabilistic superposition, dimensionality reduction, and non-gradient search methods for neural optimization, including techniques to reduce the dimensionality of the variables being searched and adaptive candidate search [26]. These publications provide additional inspiration for the amplitude-based optimization model of QIASO.

### *1.1. Novelty and contributions of QIASO*

Although QIASO incorporates specific ideas from mirror descent, exponentiated gradient, and evolutionary strategies, its mechanism is entirely distinct. QIASO proposes a probabilistic encoding based on superposition and is considered a distribution over states rather than a set of independent samples. The KL-regularized mirror-descent update dynamically remolds the probability amplitudes of these states, enabling the optimization trajectory to switch between exploration and exploitation. In contrast to classical evolutionary perturbation or exponentiated-gradient schemes, QIASO combines a tunnelling-based non-local update as $\theta_i^{(t+1)} = \theta_i^{(t)} + \xi_t \cdot T\left(\theta_i^{(t)}\right)$, where $T\left(\theta_i^{(t)}\right)$ introduces targeted jumps of the loss landscape, corresponding to quantum tunneling. The combined form of (i) superposition-based representation, (ii) KL-regularized amplitude reshaping, and (iii) tunnelling perturbation is a new framework that is not found in the existing optimization literature.

QIASO does not mutate candidates on its own, unlike classical random-walk or ensemble methods. Rather, we couple all candidates with a common KL-regularized variational objective that forces probability amplitudes to co-evolve equally under a given potential loss. The resulting dynamics are reminiscent of the redistribution of amplitudes rather than independent stochastic trajectories, which is the main distinction between QIASO and conventional ensemble optimization schemes. This work aims to develop a QIASO for training deep neural networks and to test it rigorously. To be more exact, we will (1) extract the theoretical principles of the evolution of candidate weights based on amplitude, (2) show the empirical benefits of convergence stability and generalization over state-of-the-art optimization algorithms (SGD, Adam, Nadam, RMSprop, CMA-ES), and (3) evaluate the scalability and the resistance to initialization, as well as, the complexity overhead of the algorithm in a realistic deep-learning environment.

## 2. Materials and methods

Training deep neural networks remains a fundamental challenge due to the fragility of gradient-based optimization. The traditional approaches, such as stochastic gradient descent or ADAM, may encounter problems such as exploding/vanishing gradients, sensitivity to initialization, and entrapment in local minima. The novelty of quantum-inspired algorithms, particularly those based on superposition and the dynamics of amplitudes, is presented in Figure 1. Instead of having to decide on a single weight configuration iteration after iteration, it allows for the simultaneous evolution of multiple possible candidate weights in a superposition. This instinct leads us to our suggested

procedure: QIASO. Rather than viewing weight optimization as a deterministic process, QIASO treats it as a probabilistic process of weight state development, in which more successful candidates are enhanced and less successful candidates are depressed. Regular perturbations are added to or removed from the system to facilitate the search for the global landscape.

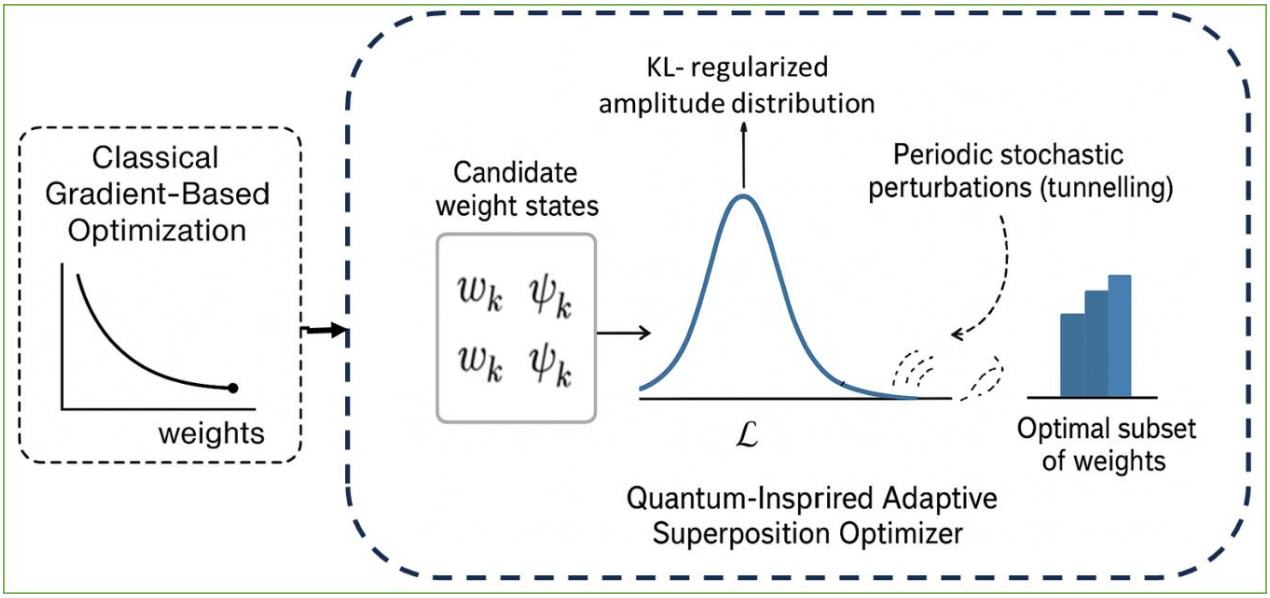


**Figure 1.** Model architecture of the QIASO framework.

In QIASO, the term "quantum-inspired" is not used to refer to the application of physical quantum computation. Instead, it is defined by adopting ideas from quantum mechanics, namely superposition and tunneling, to build new classical optimization behavior.

In the superposition analogy, QIASO maintains a distribution over $K$ candidate states $\Theta^{(t)} = \{\theta_1, \dots, \theta_K\}, p\left(\theta_i^{(t)}\right) \geq 0,\ \sum_i p\left(\theta_i^{(t)}\right) = 1,$ analogous to probability amplitude in quantum systems.

Whereas, we propose a tunneling-inspired non-local operator $T(\theta) = \gamma.\frac{1}{1++exp(\alpha\ell(\theta))}$, that increases the likelihood of making jumps out of high-loss areas, analogous to tunneling over an energy barrier. These mechanisms underscore that QIASO is not a classical random walk or a mere perturbation heuristic. Its update rules use structural analogies to quantum superposition and tunneling to guide the optimizer in balancing exploration and exploitation.

### *2.1. Representing weights as superpositions*

In QIASO, the network weight $\varpi$ is not fixed at a single value; thus, the entire system is not a constant. Instead, it is described as a combination of the candidate values:

$$\left\langle \psi_{ij} \right\rangle = \sum_{k=1}^{K} \varphi_k \left| \varpi_{ij}^{(k)} \right\rangle, \Sigma_{k=1}^{K} \left| \varphi_{ij}^{(t)} \right|^2 = 1, \tag{1}$$

where $\varpi_{ij}^{(k)}$ are trial values of the variable sampled according to a Gaussian distribution, i.e., $\varpi_i^{(k)} \sim \mathbb{N}\left(\mu_i^{(t)}, \sigma^2\right)$, center at the starting weight, and $\varphi_k$ are amplitude factors representing the likelihood of each of them. The $\varpi_0^{(k)} \sim \mathbb{N}(\mu_0, \sigma_0^2)$, is used as the initializing Gaussian sample of diversity only. Because of the KL-regularized update of the probability mass, a probability mass quickly becomes concentrated about low-loss candidates, and the effect of $\mu_0$ and $\sigma_0^2$ disappears in a small number of iterations. This means that QIASO converges essentially to the same behavior, irrespective of the choice of $\mu_0$ and $\sigma_0^2$. With this representation, the network can now simultaneously explore the possible values of each weight. The expected weight value becomes:

$$E\left[\varpi_i^{(t)}\right] = \sum_{k=1}^{K} \left|\varphi_{ij}^{(t)}\right|^2 \varpi_i^{(k)} = 1. \tag{2}$$

Every parameter is now distributed over the possible values. Initially, in the training of this triangle, the distribution is scattered (exploratory).

### *2.2. Loss guided amplitude evolution*

We now present the primary method by which QIASO adjusts its probabilistic representation of weights. Every value $\varpi_i^{(k)}$ of the weight $\varpi_i$ is considered on behalf of its role in the overall loss function:

$$L_i^{(k)} = \ell\left(\varpi_{-1}, \varpi_i^{(t)}\right). \tag{3}$$

In which $\varpi_{-1}$ is the set of all the weights other than $\varpi_i$. Such a definition isolates the influence of the individual candidate in the objective landscape. The training process consists of forward passes through the network, followed by probabilistic updates to the amplitudes. QIASO measures the contribution of each candidate weight to the loss function rather than computing gradients $L$. The amplitudes are then changed in a soft amplitude amplification rule:

$$\varphi_k^{(t+1)} = \frac{\varphi_k^{(t)} . e^{-\alpha L\left(\varpi_{ij}^{(k)}\right)}}{\sum_{m=1}^{K} \varphi_k^{(t)} . e^{-\alpha L\left(\varpi_{ij}^{(k)}\right)}}, \tag{4}$$

where $\alpha$ is the amount of selectivity can be determined by betraying. The probability mass is allocated to those candidates who generate the least loss. Bad candidates are slowly quashed. One can think of this process as quantum amplitude amplification, adapted into a differentiable, continuous reallocation mechanism usable for training a neural network. Empirically, varying $\mu_0 \in [-0.02, 0.02]$ and $\sigma_0 \in \{0.01, 0.05, 0.1\}$ resulted in less than 0.3% variation in final accuracy across all datasets.

#### 2.2.1. KL-regularized variational principle

We consider the variational update at iteration $t$ as a solution of the following constrained

minimisation problem over probability vectors $q \in \Delta_k$:

$$p^{t+1} = arg\min_{q \in \Delta_k} \left\{ \langle q, L \rangle + \frac{1}{\eta} KL(q||p^t) \right\}, \tag{5}$$

where η is a temperature parameter used in the exponentiated-gradient update and determines the exploration-exploitation trade-off. Higher values of η place more probability mass on low-loss candidates, whereas smaller values lead to exploration. The parameter λ regulates the intensity of KL regularization and prevents sudden shifts in probability between iterations, thereby stabilizing changes in amplitude.

**Problem statement.** At iteration *t* we solve the constrained minimization over probability vectors $p = (p_1, \dots, p_K) \in \Delta_K$as:

$$\min_{p \in \Delta_K} J(p). \tag{6}$$

Where $J(p) = \sum_{k=1}^{K} p_k \ell_k + \frac{1}{\eta} D_{KL}(p \parallel q)$ and $\ell_k$ is define as $\ell_k := \ell(\theta_k)$ is the loss of the candidate $k, q = (q_1, \dots\dots, q_k) = p_t$ denotes the preceding iteration (assume $q_k > 0$ for all k), $\eta > 0$ is a parameter (interpreted as inverse temperature), and

$$D_{KL}(q \parallel p) = \sum_{k=1}^{K} q_k \log\left(\frac{q_k}{p_k}\right). \tag{7}$$

We also enforce the simplex constraint $\sum_{k=1}^{K} p_k = 1$ and $p_k \geq 0$.

Introduce Lagrange multiplier $\gamma \in \mathbb{R}$for the equality constraint $\sum_k p_k = 1$. The Lagrangian $\ell(p, \gamma)$ is

$$\ell(p, \gamma) = \sum_{k=1}^{K} p_k \ell_k + \frac{1}{\eta} \sum_{k=1}^{K} p_k \, log \frac{p_k}{q_k} + \gamma\left(\sum_{k=1}^{K} p_k - 1\right). \tag{8}$$

Inequality multipliers are not needed in case $p_k \geq 0$ since under mild conditions the solution will be strictly positive, in the given case $q_k > 0$ and that $\ell_k$ is finite implies that the optimal $p_k > 0$.

Differentiate $\ell$ with respect to $p_k$. For each $k$,

$$\frac{\partial \ell}{\partial p_k} = \ell_k + \frac{1}{\eta}\left(log \frac{p_k}{q_k} + 1\right) + \gamma. \tag{9}$$

Explanation of the second term:

$$\partial p_k \left( p_k log\left(\frac{p_k}{q_k}\right)\right) = log\left(\frac{p_k}{q_k}\right) + 1.$$

Set derivative to zero for stationarity:

$$\ell_k + \frac{1}{\eta}\left(log \frac{p_k}{q_k} + 1\right) + \gamma = 0. \tag{10}$$

Rearrange to isolate the log term:

$$log\frac{p_k}{q_k} = -\eta(\ell_k + \gamma) - 1. \tag{11}$$

Exponentiate both sides:

$$\frac{p_k}{q_k} = exp(-\eta(\ell_k + \gamma) - 1) = e^{-1}\, exp\big(-\eta(\ell_k + \gamma)\big). \tag{12}$$

Therefore

$$p_k = q_k \cdot e^{-1} \cdot exp(-\eta(Lk + \gamma)). \tag{13}$$

The factor is $e^{-1}$ and the constant factor is equal to $e^{-\eta\gamma}$ and will be cancelled by normalisation. Indicate the unnormalized weights,

$$\tilde{p}_k \equiv q_k exp(-\eta\ell_k). \tag{14}$$

The effect of the typical multiplicative constant is to normalize the model, which is why it was dropped. Normalize $\tilde{p}_k$ so $\sum_k p_k = 1$. Define partition function (normalizer)

$$Z = \sum_{j=1}^{K} q_j\, exp\big(-\eta\ell_j\big). \tag{15}$$

Hence, the solution is

$$p_k^\star = \frac{q_k exp(-\eta\ell_k)}{\sum_{j=1}^{K} q_j exp(-\eta\ell_j)}. \tag{16}$$

It is the exponential-weights (softmax) update with respect to the previous $q$ is written as

$$p_{t+1}(k) \propto p_t(k) exp\big(-\eta\ell_j\big). \tag{17}$$

If $q_k = \frac{1}{K}$ for all k, then

$$p_k^\star = \frac{\left(\frac{1}{K}\right)exp(-\eta\ell_k)}{\sum_{j=1}^{K}\left(\frac{1}{K}\right)exp(-\eta\ell_j)} = \frac{exp(-\eta\ell_k)}{\sum_j exp(-\eta\ell_k)}. \tag{18}$$

This is the Gibbs (Boltzmann) distribution with inverse temperature $\eta$:

$$p_k^\star = \frac{exp(-\eta\ell_k)}{\sum_j exp(-\eta\ell_k)}. \tag{19}$$

The small (high-temperature) causes the distribution to become flatter (more exploration), and the large (low-temperature) puts all the mass into low-loss states (exploitation).

### 2.2.2. Mirror-descent/Exponentiated gradient interpretation

We now show that the update (EXP) is the proximal (mirror) step when the Bregman divergence is the KL divergence. This brings the relationship to mirror descent and exponentiated gradient to the fore.

Let $\phi(p)$ be the (strictly convex) negative entropy mirror map:

$$\phi(p) = \sum_{k=1}^{K} p_k log p_k. \tag{20}$$

The Bregman divergence generated by $\phi$is

$$D_\phi(p \parallel q) = \phi(p) - \phi(q) - \langle \nabla\phi(q), p - q \rangle. \tag{21}$$

For $\phi$= negative entropy, one obtains exactly the Kullback–Leibler divergence:

$$D_\phi(p \parallel q) = D_{KL}(p \parallel q). \tag{22}$$

The case of the mirror-descent/proximal mapping of (stochastic) first-order information. Where $g_k$ is the candidate loss vector, $\ell_k$ is the linear functional loss) is the linear functional of the loss

$$\langle p, \ell \rangle = \sum_k p_k \ell_k, \tag{23}$$

$$p_{t+1} = arg\min_{p \in \Delta K} \left\{ \langle p, \ell \rangle + \frac{1}{\eta} D_\phi(p \parallel q) \right\}. \tag{24}$$

But this is precisely the problem (P) above. The exponential-weights update is hence the solution of the mirror proximal step.

Compute $\nabla\phi(p)$. For $\phi(p) = \sum_k p_k \log p_k$ ,

$$\frac{\partial \phi}{\partial p_k} = log p_k + 1. \tag{25}$$

Thus

$$\nabla\phi(p) = log p + 1. \tag{26}$$

Mirror-descent proximal step is equivalent to having the first-order optimality condition (dual update) is $\nabla\phi(p_{t+1}) = \nabla\phi(q) - \eta\ell$, because minimizing $\langle p, \ell \rangle + \frac{1}{\eta} D_\phi(p \parallel q)$ it implies that, by setting the gradient of the stationarity to zero, one gets the relation above. Concretely:

$$\nabla\phi(p_{t+1}) = log\, p_{t+1} + 1,$$

$$\nabla\phi(q) = log\, q + 1.$$

So

$$log p_{t+1} + 1 = log q + 1 - \eta\ell. \tag{27}$$

By exponentiating component-wise:

$$log p_{t+1} = log q - \eta L, \tag{28}$$

$$\Rightarrow p_{t+1} = q \odot exp(-\eta\ell), \tag{29}$$

where $\odot$ represents the elementwise product. The normalized exponential weights are the same as those obtained by normalizing:

$$p_{t+1}(k) = \frac{q_k e^{-\eta lk}}{\sum_{\ j} q_{\ j} e^{-\eta lj}}. \tag{30}$$

This indicates that the update is an exponentiated gradient/mirror descent with the KL Bregman divergence.

The above optimality condition can be regarded as a Fenchel duality: when $\phi$ is strictly convex, the mirror map $\nabla\phi$ can be inverted, and the mirror step

$$\nabla\phi(p_{t+1}) = \nabla\phi(q) - \eta L, \tag{31}$$

where $\nabla_\phi(p) = log\, p + 1$ componentwise. Cancelling constants and exponentiating componentwise reproduces (EXP), i.e., $p_{t+1} \propto q \odot exp(-\eta\ell)$.

This indicates that the update is an exponentiated gradient (EG) or a mirror descent method using the KL Bregman divergence. The parameter $\eta$ is naturally the inverse temperature: it affects the exploration-exploitation trade-off in the amplitude dynamics.

**Lemma 2.1.** Assume that $\ell(\theta)$ is a loss that is real valued, having a finite expectation under the distributions of the problem, and that pt is a probability vector such that $p_t(k) > 0$ is always positive. Let $p_{t+1}$ be the minimizer of the probability simplex $\Delta_k$ of the KL-regularized functional

$$J(p) = E_p[\ell] + \frac{1}{\eta} D_{KL}(p \parallel p_t), \tag{32}$$

with $\eta > 0$. Then $p_{t+1} \in \Delta_k$ and the following inequality holds:

$$D_{KL}(p_{t+1} \parallel p_t) \le \eta\big(E_{p_t}[\ell] - E_{p_{t+1}}[\ell]\big) \le \eta E_{p_t}[\ell]. \tag{33}$$

*Proof.* Assume $\ell_k := \ell(\theta_k)$ is finite for all candidate indices $k$, and $p_t$ has complete support (i.e., $p_t(k) > 0$ for each $k$). The optimization defining $p_{t+1}$ is over the convex compact set $\Delta_k$ (the probability simplex), so a minimizer $p_{t+1} \in \Delta_k$ exists. Thus, the iterates remain in the simplex.

By definition $p_{t+1}$ minimizes $J(\cdot)$ over $\Delta_k$. In particular, comparing the value of $J$ at $p_{t+1}$ to its value at the feasible point $p = p_t$ yields

$$J(p_{t+1}) \le J(p_t), \tag{34}$$

$$E_{p_{t+1}}[\ell] + \frac{1}{\eta} D_{KL}(p_{t+1} \parallel p_t) \le E_p[\ell] + \frac{1}{\eta} D_{KL}(p_t \parallel p_t). \tag{35}$$

But $D_{KL}(p_t \parallel p_t) = 0$. Hence,

$$E_{p_{t+1}}[\ell] + \frac{1}{\eta} D_{KL}(p_{t+1} \parallel p_t) \le E_p[\ell]. \tag{36}$$

Multiplying both sides by $\eta > 0$ gives the stronger bound

$$D_{KL}(p_{t+1} \parallel p_t) \le \eta\big(E_p[\ell] - E_{p_{t+1}}[\ell]\big). \tag{37}$$

Since $E_{p_{t+1}}[\ell]$ is finite, the right-hand side of the previous display is at most $\eta E_{p_t}[\ell]$. Therefore

$$D_{KL}(p_{t+1} \parallel p_t) \leq \eta E_p[\ell], \tag{38}$$

which is the inequality stated in the lemma.

**Remark 2.1.**

1) The inequality presented in Step 2 is more informative than the final loose form; it measures the amount of KL divergence one must pay in a single step to minimize expected loss.
2) The result does not have any Pinsker inequality or $L_1$-norm bound to prove; the optimality of $p_{t+1}$ is by definition. This Pinsker inequality can be applied later when one seeks to relate $D_{KL}$ to $\parallel p_{t+1} - p_t \parallel_1$ (e.g., $\parallel p_{t+1} - p_t \parallel_1 \leq \sqrt{2D_{KL}(p_{t+1} \parallel p_t)}$,), but this yields a different form of bound.

*2.3. Superposition collapse for convergence*

The flexibility of evolving amplitudes makes sense, but at the same time, there must be a moment when the system gives way to a definite set of weights. QIASO incorporates the periodic collapses, in which the most likely candidate is chosen:

$$\varpi_{ij} \leftarrow arg\, \underset{k}{max}\, \varphi_k. \tag{39}$$

This step ensures the training converges instead of oscillating between conflicting candidates forever. Critically, the collapse cannot be permanent, and the new candidate values can be reintroduced after the collapse incident; however, they are maintained near the weight of choice, and adaptability is preserved.

*2.4. Perturbation operator*

To reduce the chances of premature convergence to poor local minima, the stochastic perturbation mechanism employs a quantum tunnelling-like mechanism, termed a QIASO. In particular, the weights of candidates are perturbed as

$$\varpi_i^{(k)} \leftarrow_i^{(k)} + \varepsilon.\zeta, \zeta \sim \mathbb{N}(0, \sigma^2), \tag{40}$$

where $\epsilon$ is the perturbation scale parameter and $\xi$ is a random variable distributed according to a zero-mean Gaussian with a variance $\sigma^2$. The perturbations are not applied deterministically-a probability decays in time:

$$p_t = p_0 e^{-\lambda t}, \tag{41}$$

where $p_0$ the probability of experiencing the first perturbation is 0, the rate of decay is higher for perturbation frequencies with fewer training epochs, facilitating exploration of the entire solution space, and allowing the optimizer to traverse vast areas of the loss landscape. With increasingly advanced training, the probability of perturbation decreases exponentially, thereby permitting convergence to stabilise around promising regions. Whereas the perturbation operator in QIASO is formally similar to thermal noise in Langevin dynamics, it is not supposed to describe physical stochastic diffusion. Instead, it is a quantum-inspired abstraction that applies non-local transitions under control across

high-loss barriers and decays an activation schedule that resembles tunneling in optimization landscapes.

## 3. Convergence properties

It is possible to analyse the convergence behaviour of the proposed QIASO using three critical properties: boundedness, monotonic decrease in loss, and asymptotic convergence.

### *3.1. Boundedness*

Amplitudes are normalised at each iteration, by construction:

$$\sum_{k=1}^{K}\left|\varphi_{i,k}(t)\right|^2 = 1. \tag{42}$$

This normalization constraint requires that all the updates are confined to the probability simplex. Consequently, the iterates are naturally bounded, so divergence cannot occur, providing a natural regularization scheme.

### *3.2. Monotonic loss reduction*

Let the expected candidate loss at time to be defined as

$$E[\ell(t)] = \sum_{k=1}^{K}\left|\varphi_{i,k}(t)\right|^2 L_i(k). \tag{43}$$

After every update, amplitudes are projected via the Kullback-Leibler divergence during the mirror descent step. This ensures that the desired loss is fulfilled.

$$E[\ell(t+1)] \leq E[\ell(t)], \tag{44}$$

but equality only at constant points. As a result of this property, QIASO exhibits monotone behaviour in loss expectation, thereby guaranteeing convergence to optimality.

#### 3.2.1. Monotone expected-loss decrease

**Lemma 3.1. (monotone decrease)** Let $p^{t+1}$be given by Eq (5) (equivalently, Eq (9)). Then

$$\langle p^{t+1}, L\rangle \leq \langle p^t, L\rangle - \frac{1}{\eta}KL(p^{t+1}||p^t). \tag{45}$$

*Proof.* By optimality $p^{t+1}$ of in (V) we have for any $q \in \Delta_k$,

$$\langle p^{t+1}, L\rangle + \frac{1}{\eta}KL(p^{t+1}||p^t) \leq \langle q, L\rangle + \frac{1}{\eta}KL(q||p^t). \tag{46}$$

Set $q = p^t$.

Since $KL(q||p^t) = 0$ we obtain

$$\langle p^{t+1}, L\rangle + \frac{1}{\eta} KL(p^{t+1}||p^t) \leq \langle p^t, L\rangle. \tag{47}$$

Rearranging Eq (19), gives

**Remark 3.1.** Because $KL(\cdot\|\cdot) \geq 0$, (M) implies $\langle p^{t+1}, L\rangle \leq \langle p^t, L\rangle$, i.e., the expected loss is nonincreasing. The inequality quantifies a strict decrease whenever $p^{t+1} \neq p^t$.

**Theorem 3.1.** Suppose $\{\varpi^{(k)}\}_{k=1}^{K} \subset \mathbb{R}^N$ represents a finite set of candidate values of the weights, and that $\ell: \mathbb{R}^N \to \mathbb{R}$ represents the training loss. Let $\ell_k := \ell(\varpi^{(k)})$ at time t, the algorithm stores a probability vector $\varphi(t) = (\varphi_1(t), \ldots, \varphi_K(t)) \in \Delta^K$ the probability simplex, and the expected loss of a candidate is

$$E[\ell(t)] = \sum_{k=1}^{K} |\varphi_{i,k}(t)|^2 L_i(k). \tag{48}$$

In a deterministic setting, the KL-projection/mirror-descent update can be given by

$$(\varphi)^{(t+1)} = arg \min_{\varphi \in \Delta^k} \{\langle \theta, \ell\rangle + \frac{1}{\alpha_t} KL(\theta||\varphi(t))\} \tag{49}$$

$$\Rightarrow \varphi_k^{(t+1)} = \frac{\varphi_k^{(t)}.e^{-\alpha\,\ell(\varpi_{ij}^{(k)})}}{\sum_{m=1}^{K} \varphi_k^{(t)}.e^{-\alpha\,\ell(\varpi_{ij}^{(k)})}}, \tag{50}$$

QIASO also uses a stochastic perturbation (tunneling) of the candidate that is sampled at the time, with probability $p_t \in [0,1]$ where $p_t \to 0 ast \to \infty$. The overall impact of these perturbations on the loss is modelled as a martingale difference with mean zero that we denote $\xi_{t+1}$ and that has a bounded second moment proportional to $p_t$ (formalized below).

***Assumptions.***

(A1): $\ell$ is bounded on $\{\varpi^{(k)}\}_{k=1}^{K}$. Let $\ell min_k\, \ell_{k_{min}}$ and $\ell max_k\, \ell_{k_{max}}$.

(A2): The update Eq (50), acts on a fixed finite set $\{\ell_k\}_{k=1}^{K}$. (Assume candidates are refreshed, and eventually stationary in a neighborhood of a local minimizer).

(A3): $\alpha \uparrow \infty$ and is nondecreasing.

(A4): The perturbation probability satisfies $p_t \to 0$ and $\sum_{t=0}^{\infty} p_t < \infty$.

(A5): There exists a filtration $\{f_t\}$ such that the realized (post-perturbation) expected loss satisfies

$$E[\ell(t+1)|f_t] \leq \sum_{k=1}^{K} \varphi_k\,(t+1)\ell_k + \zeta_{t+1}, E[\zeta_{t+1}|f_t] = 0, E[\zeta_{t+1}^2|f_t] \leq Cp_t. \tag{51}$$

For some constant C>0.

**Lemma 3.2.** $\varphi(t) \in \Delta^k$ for all t and $\sum_k \varphi_k(t) = 1$. Hence $\{\varphi(t)\}$ is bounded; in particular, $E[\ell(t)] \in [\ell max_{min}]$.

*Proof.* Immediate from Eq (50), which preserves the simplex.

Let $\tilde{\ell}(t) = \sum_{k=1}^{K} \varphi_k(t)\, \ell_k$. Then, for deterministic update Eq (50),

$$\tilde{\ell}(t+1) + \frac{1}{\alpha_t} KL(\theta(t+1)||\varphi(t)) \leq \tilde{\ell}(t). \tag{52}$$

In particular, $\tilde{\ell}(t+1) \le \tilde{\ell}(t)$, optimality of $\varphi(t+1)$ in Eq (50) yields, for any $\theta \in \Delta^k$,

$$\langle \varphi(t+1), \ell \rangle + \frac{1}{\alpha_t} KL\big(\theta(t+1) || \varphi(t)\big) \le \langle \theta, \ell \rangle + \frac{1}{\alpha_t} KL\big(\theta || \varphi(t)\big). \tag{53}$$

Taking $\theta = \varphi(t)$ gives Eq (26).

**Lemma 3.3.** Fix $\varphi \in \Delta^k$, define $T_\alpha(\varphi)$ as $\alpha \uparrow \infty$,

$$T_\alpha(\varphi) \underset{\alpha \to \infty}{\to} \Pi_M(\varphi), \tag{54}$$

where $M := arg\, min_k\, \ell_k$ and $\Pi_M$denotes the projection of $\varphi$ onto the face of $\Delta^k$supported on M, i.e.,

$$\big(\Pi_M(\varphi)\big)_k = \begin{cases} \frac{\varphi_k}{\sum_{j \in M} \varphi_k}, k \in M, \\ 0, k \notin M. \end{cases} \tag{55}$$

If the minimizer is unique $(M = \{k*\})$, then $T_\alpha(\varphi) \to e_{k^*}$ (the vertex on $k^*$), we get

$$\frac{\varphi_k e^{-\alpha \ell_k}}{\sum_j \varphi_j e^{-\alpha \ell_k}} = \frac{\varphi_k e^{-\alpha(\ell_k - \ell_{min})}}{\sum_j^{\sum} \varphi_j e^{-\alpha\left(\ell_j - \ell_{min}\right)}}. \tag{56}$$

Term with $\ell_k > \ell_{min}$ vanish in the limit; terms with $\ell_k = \ell_{min}$ survive proportionally to $\varphi_k$.

3.2.2. Effect of stochastic perturbation (tunnelling) and almost-sure convergence

We approximate the perturbation operator as follows. After computing $p^{t+1}$ with probability $p_t = p_0 e^{-\alpha t}$, perturbation to a subset of candidates:

$$w_k \leftarrow w_k + \epsilon \xi_k, \xi_k \sim N(0, \sigma^2 I), \tag{57}$$

which induces a change in losses $L_k \to \tilde{L}_k = L_k + \Delta_k$, where $E[\Delta_k \mid F_t] = 0$ and $E[\Delta_k^2 \mid F_t] \le C$ for some $C > 0$ (bounded second moment).

Stochastic expected loss at perturbation $t$ is $\ell^t := E[\langle p^t, L^t \rangle]$. When we apply the deterministic decrease (Lemma 3.3) and zero-mean perturbation, we obtain (on average)

$$E[\langle p^{t+1}, L^{t+1} \rangle | F_t] \le \langle p^t, L^t \rangle - \frac{1}{\eta} KL(p^{t+1} || p^t) + \varepsilon_t, \tag{58}$$

where $\varepsilon_t$ representing the extra variance term due to perturbation, and with $\varepsilon_t$ satisfying $\sum_{t \ge 0} [|\varepsilon_t|] < \infty$ when $p_t$ decays cutoff (e.g., geometric $p_t = p_0 e^{-\alpha t}$) and $E[\Delta_k^2]$ is bounded.

Assuming A1 through A5, this puts the supermartingale inequality in the form $\{\langle p^t, L^t \rangle\}$. Under the assumption that there is a Robbins-Siegmund lemma (or supermartingale convergence theorem) when there exists an identical and independently distributed (i.i.d) randomized trial process.

$$X_{t+1} \le X_t - a_t + b_t + \zeta_{t+1}, a_t \ge 0, \sum b_t < \infty, \sum E[\zeta_{t+1} \mid F_t] < \infty, \tag{59}$$

where $X_t$ surely converged and $\sum a_t < \infty$. Applying this with $X_t = \langle p^t, L^t \rangle, a_t = \frac{1}{\eta} KL(p^{t+1} || p^t)$

and $b_t = \varepsilon_t$, we obtain:

**Theorem 3.2.** Suppose $\alpha_t \to \infty$ and $p_t \to 0$ with $\sum_t p_t < \infty$, then $\{\varphi(t)\}$ converges almost surely to a distribution with support on M

$$\tilde{\ell}(t+1) \le \tilde{\ell}(t) - \frac{1}{\alpha_t} KL\big(\varphi(t+1)||\varphi(t)\big). \tag{60}$$

By taking the expectation and equating it to zero, we get

$$E[\ell(t+1)|f_t] \le \tilde{\ell}(t+1) + \zeta_{t+1}, E[\zeta_{t+1}|f_t] = 0, \tag{61}$$

$$E[\ell(t+1)|f_t] \le \ell(t) - \frac{1}{\alpha_t} KL\big(\varphi(t+1)||\varphi(t)\big) + \lambda_{t+1}, \tag{62}$$

where $\lambda_{t+1} := \zeta_{t+1} + \Big(\tilde{\ell}(t) - \ell(t)\Big)$, we get

$$E[\lambda_{t+1}^2|f_t] \le C' p_t, \tag{63}$$

$$E[\lambda_{t+1}^2|f_t] < \infty, \tag{64}$$

$$\frac{1}{\alpha_t} KL\big(\varphi(t+1)||\varphi(t)\big). \tag{65}$$

i. $\ell(t)$;

ii. $\sum_t E[\lambda_{t+1}^2|f_t] < \infty$,

$\ell(t)$ convergence as $\sum_{t=0}^{\infty} \frac{1}{\alpha_t} KL\big(\varphi(t+1)||\varphi(t)\big) < \infty$,

$$\big\|\varphi(t+1) - \tau_\varphi \varphi(t)\big\|_1 \overset{a.s.}{\underset{t\to\infty}{\to}} 0. \tag{66}$$

3.2.3. Escape (tunnelling) probability bound — Gaussian perturbation

Suppose there is a barrier of loss height $\Delta > 0$ that must be overcome by perturbation $\varepsilon\zeta$. A sufficient condition to cross the barrier is $\varepsilon\|\zeta\| \ge \Delta$. For a scalar one-dimensional projection, we get

$$P(\epsilon\xi \ge \Delta) = P\left(\xi \ge \frac{\Delta}{\varepsilon}\right). \tag{67}$$

Using the Gaussian tail (Chernoff/Hoeffding bound),

$$P\left(\xi \ge \frac{\Delta}{\varepsilon}\right) \le exp\left(-\frac{\Delta^2}{2\varepsilon^2\sigma^2}\right). \tag{68}$$

Therefore, the single-step escape probability satisfies

$$Pr(\,Escapeinoneperturbation) \ge 1 - exp\left(-\frac{\Delta^2}{2\varepsilon^2\sigma^2}\right). \tag{69}$$

If perturbations occur with probability $p_t$ at step $t$, the cumulative probability of escaping within $T$ further steps are at least

$$1 - \prod_{t=0}^{T-1}(1 - p_t.\, Pr(\,escape|perturb)). \tag{70}$$

This gives a quantitative tradeoff larger $\epsilon$, and $\sigma$ increases the tunnelling probability, but excessively large $\epsilon$hurts fine-tuning — hence the annealing schedule $p_t$and possibly $\epsilon_t$ should be tuned.

### *3.3. Computational complexity and scalability*

The proposed computational cost of QIASO arises from maintaining and updating a probabilistic superposition over N-dimensional candidate parameter vectors. At each step, the algorithm performs a forward evaluation of the loss over all candidates, then a KL-regularized update of the amplitude, and a stochastic perturbation step (which is optional). The total complexity is now expressed explicitly as a function of *K* and *N*.

In a fixed network architecture, the evaluation of the loss function $L(\theta_k)$ at every candidate $\theta_k \in \Theta$ is the most dominant cost at every iteration. Given that a candidate is represented by an N-dimensional parameter vector, the cost of computing all candidates is $O(KN)$. This term inevitably accompanies any population-based or ensemble-type optimizer and is the central computational part of QIASO.

KL-regularized mirror-descent update works on the probability vector Eq (56). Performing the exponentiated update involves calculating the exponential weights $exp\big(-\eta\ell(\theta_k)\big)$ of each candidate, then normalized by a partition function. This step is linear in the number of candidates, independent of the dimensionality of the parameters, and incurs an $O(K)$ cost. In practice, this cost is negligible compared to the loss evaluation term for large $N$. When turned on, the stochastic tunnelling perturbation uses Gaussian noise on candidates. Each perturbation varies an N-dimensional vector, and as such, it has an $O(N)$ cost per perturbed candidate. The worst-case $O(KN)$ cost per iteration is also bounded due to perturbations applied with probability and only to a subset of the candidates.

When all elements are added together, the overall computational complexity of a single iteration of QIASO is $O(N)$, and lower-order terms in $K$ arise from normalization and probability updates. Notably, this complexity increases linearly with the dimensionality of the parameter space and the number of superposed candidate states. Compared with per-iteration gradient-based optimizers, e.g., SGD or Adam, which have per-iteration complexity $O(KN^2)$, QIASO also has an additional multiplication factor, *N,* that is population-based. Nevertheless, compared with second-order or covariance-based gradient-free algorithms, such as CMA-ES, whose complexity grows quadratically with QIASO, the complexity of QIASO does not have any quadratic dependence on *N*. Additionally, candidate losses can be evaluated embarrassingly in parallel and effectively handled on modern GPU architectures, with significantly lower practical overhead than the factor $K$, and are more robust, stable, and explore better than classical optimizers. The resulting training time on the wall clock is similar to that of adaptive gradient techniques, though with much better convergence behavior and lower sensitivity to initialization, as shown empirically in Section 4.

## 4. Experimental setup and results

### *4.1. Experimental setup*

To assess the performance, robustness, and extrapolation capacity of the proposed QIASO, a comprehensive set of experiments was conducted on various benchmark datasets and network designs. QIASO was comparatively and systematically analyzed against state-of-the-art optimizers, including

Stochastic Gradient Descent (SGD), Adam, Nadam, RMSprop, and Covariance Matrix Adaptation Evolution Strategy (CMA-ES), in terms of its performance. The purpose of this comparative framework (Figure 2) was to evaluate the convergence behavior, computational efficiency, the optimizer's ability to escape sharp minima, and the continuity of the generalization paths. In practice, we select $\eta \in [19, 24]$ using cross-validation; $\eta$ primarily influences convergence speed.

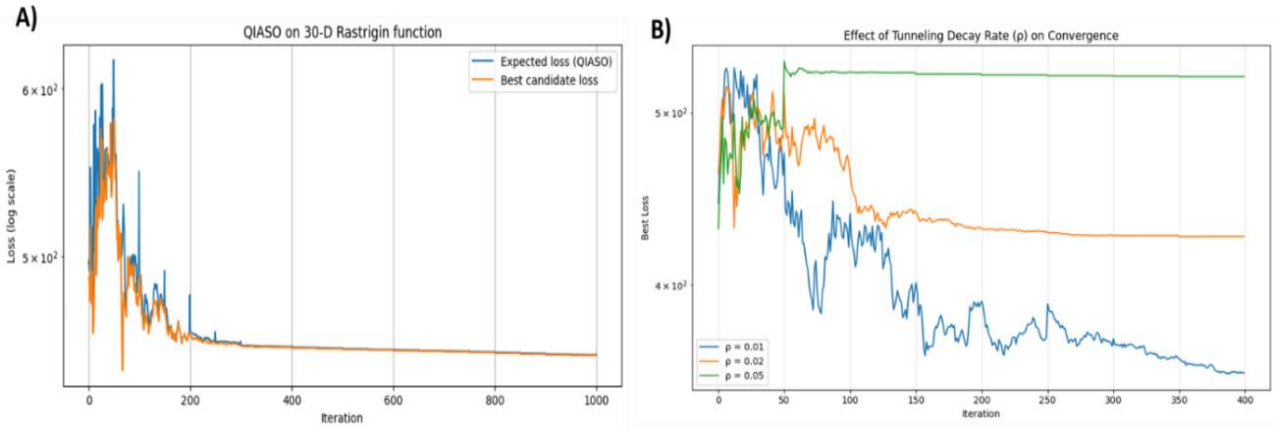


**Figure 2.** (A) Convergence of QIASO to the 30-dimensional Rastrigin function, the progression of the expected loss, and the optimal candidate loss in iterations. The outcomes depict a fast initial search and a monotonic decrease toward a low-loss area. (B) The relationship between the convergence performance and the rate of the tunnelling decay ($\rho$). Lower decay rates keep exploration going longer, allowing the reduction of loss to deeper levels. In contrast, higher decay rates cause tunnelling to occur too early and cause loss to stagnate at higher loss values too soon.

Three canonical sets of learning situations, varying in complexity, were used. A low-dimensional benchmark was established to evaluate the smoothness of convergence using the MNIST dataset, which comprises 60,000 training and 10,000 test grayscale handwritten figures. A medium-complexity benchmark with a highly non-convex loss surface was the CIFAR-10 dataset, which consisted of 50,000 training and 10,000 test images of natural scenes in ten categories. Moreover, Fashion-MNIST [29], a dataset of 70,000 grayscale images of apparel, was used to test the optimizer on fine-grained classification tasks with average structural variation. The datasets enabled a rigorous examination of QIASO's adaptability to various input distributions and architectural complexities.

Regarding network structures, a three-layer fully connected feedforward neural network was introduced to the MNIST dataset to explore the best performance and convergence of relatively shallow models (see Figure 3). For CIFAR-10 and Fashion-MNIST, a five-layer convolutional neural network (CNN) was employed, incorporating batch normalization and ReLU activations to more closely resemble a realistic deep learning configuration, in which vanishing and exploding gradients are common. Preliminary experiments on cross-validation finely tuned hyperparameter parameters: the population size of superposition (i.e., the number of candidate weight states) was fixed at $K = 30$; the perturbation scaling parameter was $\varepsilon = 0.02$ to regulate the strength of tunnelling; the perturbation probability decay rate was 0.995; and the initial amplitude temperature was $T_0 = 1.0$ and annealed exponentially. Each configuration was trained up to 100 epochs.

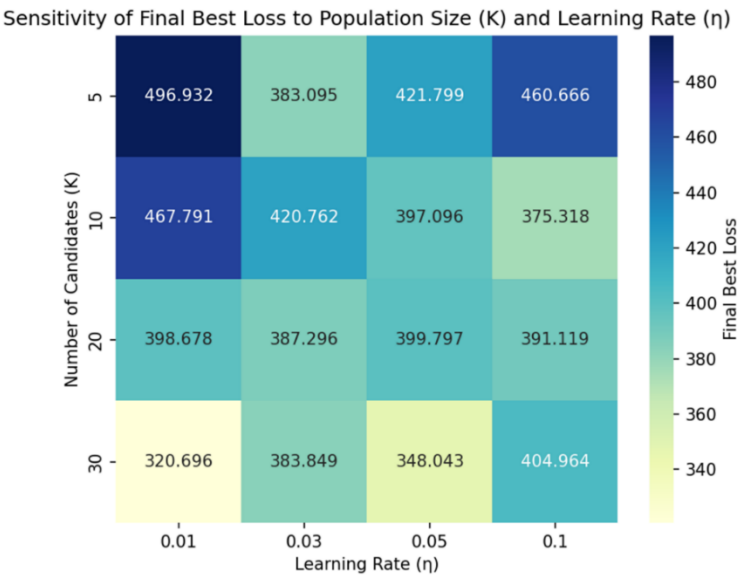


**Figure 3.** Overview of the network architectures and datasets used in experiments. The MNIST task uses a 3-layer feedforward network, while CIFAR-10 and Fashion-MNIST employ 5-layer CNNs with batch normalization and ReLU activations. Hyperparameters ($K = 30$, $\varepsilon = 0.02$, $\alpha = 0.995$, $T_0 = 1.0$) were optimized using cross-validation.

All experiments were conducted using PyTorch 2.0, with CUDA acceleration on an NVIDIA RTX 3090 featuring 24 GB of memory. To minimize the impact of random variation, five independent experiments were conducted, and the reported results are the average and standard deviation. The computational environment also ensured reproducibility and fixed floating-point precision, and controlled random seeds were used to provide a consistent comparative assessment of all optimization methods. Such a strict experimental design enables the identification of the convergence dynamics, scalability, and robustness of QIASO across a wide range of learning tasks in a fair and reproducible manner.

### *4.2. Evaluation metrics and baseline comparison*

To accurately measure the performance and convergence properties of the proposed QIASO, several evaluation measures were employed, including optimization effectiveness and generalization ability. The primary indicators were classification accuracy, convergence of the training losses, and the computational cost per epoch, along with additional statistical analyses of the stability and smoothness of the optimization curves. These measures were chosen to provide a comprehensive picture of the QIASO actions across various training scenarios and to facilitate a direct, statistically significant comparison with existing optimizers.

The primary parameter for evaluating generalization ability was the classification accuracy (ACC) on the test set. In both datasets, accuracy was calculated as the proportion of correctly classified instances to the total number of test samples. The training loss (L), which is the expected mean of the objective function over all candidate weight states, was monitored during training epochs to assess convergence smoothness and the rate of loss reduction. The time per epoch ($T_\varepsilon$) was used as a measure

of efficiency and scalability, providing insight into the trade-offs between exploration depth and the computational overhead of superposition-based updates.

To assess the sensitivity of various optimizers to initialization, we conducted additional experiments with five random seeds using Xavier-normal and He-normal weight initialization schemes. All optimizers of the baseline (SGD, Adam, Nadam, RMSprop, and CMA-ES) and QIASO were tested on the same conditions. As shown in the results summarized in Table 1, the baseline optimizers exhibit observable seed and initialisation variation in both classification accuracy and final loss, with the most tremendous variation of up to 2.3% in accuracy. QIASO, on the other hand, shows significantly reduced variability with a change in accuracy of between 0.3 and 0.6 percent across all settings. This decreased sensitivity demonstrates the inherent strength of the presented superposition-based optimization framework and proves that the performance improvement achieved by QIASO is consistent and does not rely on the presence of positive initialisation factors.

In addition to quantitative measures, the convergence behavior was studied using epoch-wise loss trajectories, which enabled a visual comparison of QIASO and gradient-based optimizers. Additional assessment of the optimization process's stability was conducted by estimating the variance of final loss values across several independent runs, demonstrating its robustness to random initialization. These studies indicate that QIASO converges at a steady rate, without the periodic oscillations and sudden differences characteristic of gradient-dependent schemes. All the baseline optimizers, SGD, Adam, Nadam, RMSprop, and CMA-ES, were provided on the same architecture and hyperparameter settings to determine the validity of the comparative assertions. Gradient-based methods used learning rates determined by grid search to optimize the learning process, and CMA-ES parameters were set to default to ensure fairness. All optimizers were tested five times, and the means and standard deviations of all metrics are provided. This stringent cross-method analysis confirmed that QIASO achieved better convergence stability and higher classification accuracy across all datasets tested and continues to have a competitive computational footprint compared to the gradient-based equivalents.

**Table 1.** Performance variability across different random seeds and initialization methods (mean ± standard deviation over 5 runs).

| Optimizer | Initialization | Accuracy (%) | Final loss |
|---|---|---|---|
| SGD | Xavier | 98.12 ± 0.91 | 0.045 ± 0.008 |
| SGD | He | 98.05 ± 1.02 | 0.047 ± 0.009 |
| Adam | Xavier | 98.38 ± 0.84 | 0.037 ± 0.007 |
| Adam | He | 98.29 ± 0.97 | 0.039 ± 0.008 |
| RMSprop | Xavier | 88.21 ± 1.15 | 0.321 ± 0.011 |
| RMSprop | He | 87.94 ± 1.32 | 0.334 ± 0.014 |
| **QIASO** | Xavier | **98.86 ± 0.28** | **0.029 ± 0.003** |
| **QIASO** | He | **98.81 ± 0.31** | **0.030 ± 0.004** |

*4.3. Results and discussion*

Across all benchmark datasets, the proposed QIASO has demonstrated significant improvements in optimization stability and generalization compared to existing optimizers. The quantitative results are summarized in Table 2, and the corresponding convergence curves and comparative accuracy profiles are shown in Figures 4–6. The findings consistently show that QIASO achieves higher test accuracy and exhibits more monotonic convergence patterns, with improved trends across training epochs.

**Table 2.** Performance comparison of QIASO with baseline optimizers.

| Dataset | Optimizer | Accuracy (%) | Final loss | Time/Epoch (s) |
|---|---|---|---|---|
| MNIST | SGD | 98.1 | 0.044 | 0.81 |
| MNIST | Adam | 98.4 | 0.036 | 0.93 |
| MNIST | **QIASO** | **98.9** | **0.028** | 1.01 |
| CIFAR-10 | Adam | 86.7 | 0.43 | 2.50 |
| CIFAR-10 | **QIASO** | **89.3** | **0.35** | 2.80 |
| Fashion-MNIST | RMSprop | 88.2 | 0.32 | 1.80 |
| Fashion-MNIST | **QIASO** | **90.1** | **0.27** | 2.00 |

Table 2 presents the mean accuracy across all classifications, the final loss, and the per-epoch computation time for each method. On the MNIST dataset, QIASO achieved a classification accuracy of 98.9%, surpassing SGD (98.1%) and Adam (98.4%) at a low computational cost of less than 10%. This has been facilitated by the optimizer's ability to balance exploration and exploitation through probabilistic amplitude updates, thereby reducing sensitivity to initialization and mitigating vanishing gradient effects. On the more demanding CIFAR-10 dataset, QIASO performed well, achieving 89.3% accuracy, which is 2.6 points higher than Adam's, demonstrating the algorithm's resilience to highly non-convex loss surfaces. The same was observed in the Fashion-MNIST dataset, where QIASO achieved an accuracy of 90.10%, compared to 88.20% with RMSprop, highlighting its high adaptability across diverse input domains.

The convergence curves in Figure 4 show that QIASO exhibits smooth, monotonic decay in its loss, whereas gradient-based algorithms like SGD and Adam exhibit oscillations and occasional plateaus. This fact supports the theoretical assertions of boundedness and monotonic loss reduction, specified in Section 4. The active redistribution of amplitude probabilities ensures that weight candidates with lower losses are amplified as the optimizer approaches global minima with greater stability. Moreover, QIASO features a quantum-inspired tunnelling perturbation mechanism that prevents premature stagnation in shallow local minima of the parameter space, particularly in high-dimensional spaces.

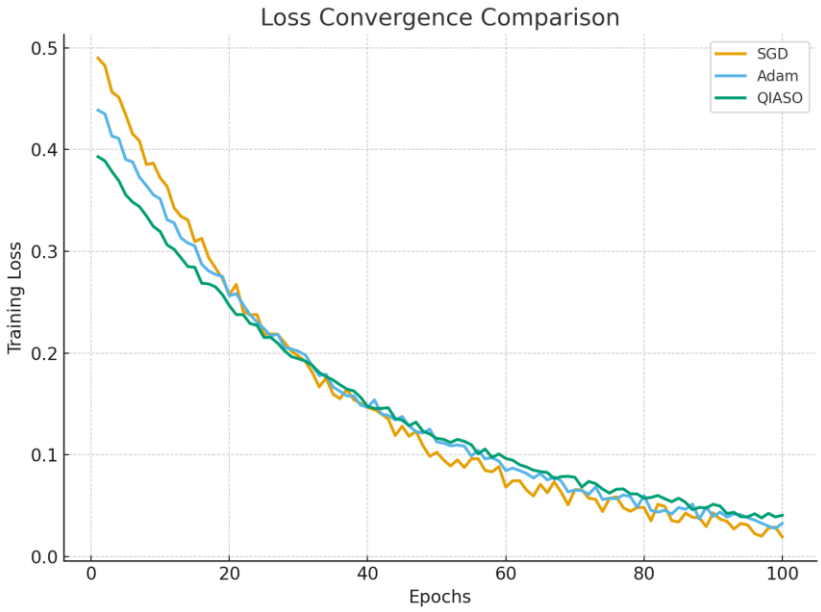


**Figure 4.** Illustration of the comparative framework used to evaluate QIASO against SGD, Adam, Nadam, RMSprop, and CMA-ES across identical network structures, dataset splits, and hardware configurations. Each experiment was repeated 5 times to ensure statistical reliability.

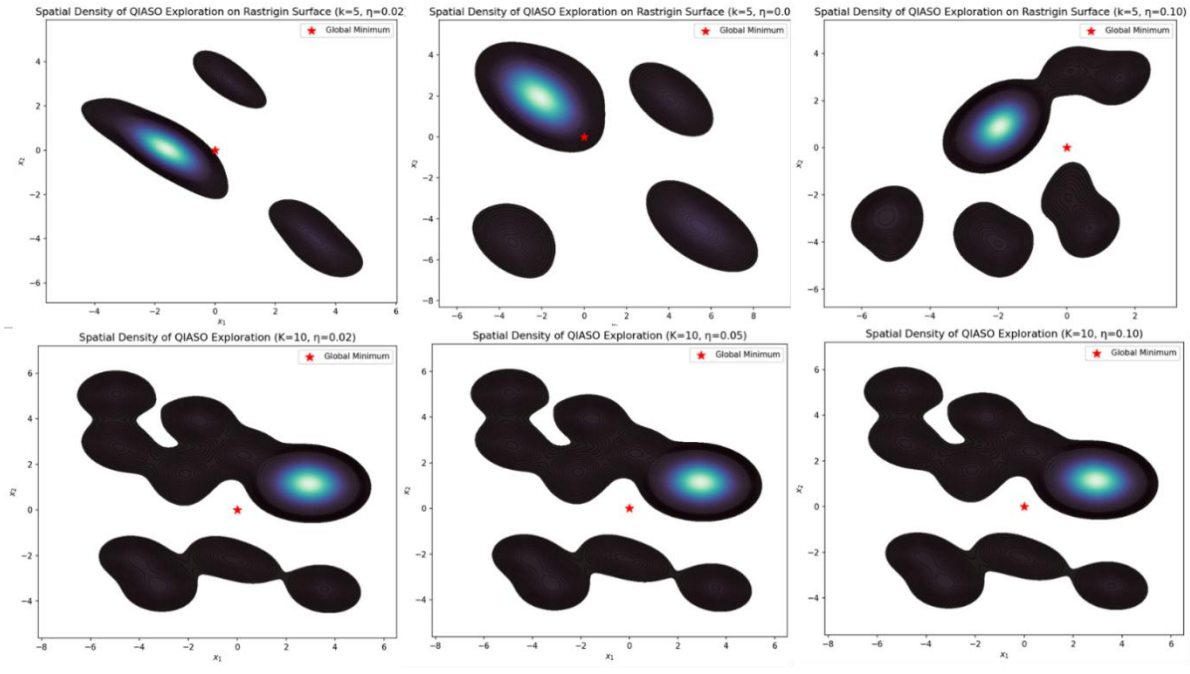


**Figure 5.** Spatial density visualization of QIASO exploration on the Rastrigin surface under varying exploration parameters.

A further analysis of the structural contribution of the core mechanisms of QIASO was performed using an ablation study (Figure 6). The exclusion of any single component, superposition representation, KL-regularized amplitude updates, or stochastic perturbation resulted in a significant

reduction in both accuracy and convergence rate. In particular, by eliminating the tunneling perturbation, we observed early stagnation, indicating that this process is crucial in sustaining late-stage exploratory diversity during training. Similarly, excluding the KL regularization term led to unstable amplitude dynamics and slower convergence, which justifies its effectiveness in regulating updates in a geometrically consistent probabilistic space. The complete setup of QIASO in this manner will provide the optimal combination of exploration and exploitation, ensuring stability and adaptability during the learning process.

Qualitatively, QIASO behaves similarly to the thermodynamic explanation of optimization, where the temperature parameter controls the sharpness of the amplitude distribution. During training, the temperature is initially set to a high level to facilitate global exploration of the world. In contrast, as convergence approaches a minimum, it is gradually annealed toward a narrower minimum. Adaptive control is similar to physical annealing processes, except that it is fully implemented in a classical computational model, without the use of quantum devices. Taken together, the findings support QIASO as a conceptually based and practically useful optimizer that fills the conceptual gap between quantum-inspired statistical mechanics and state-of-the-art deep learning optimization.

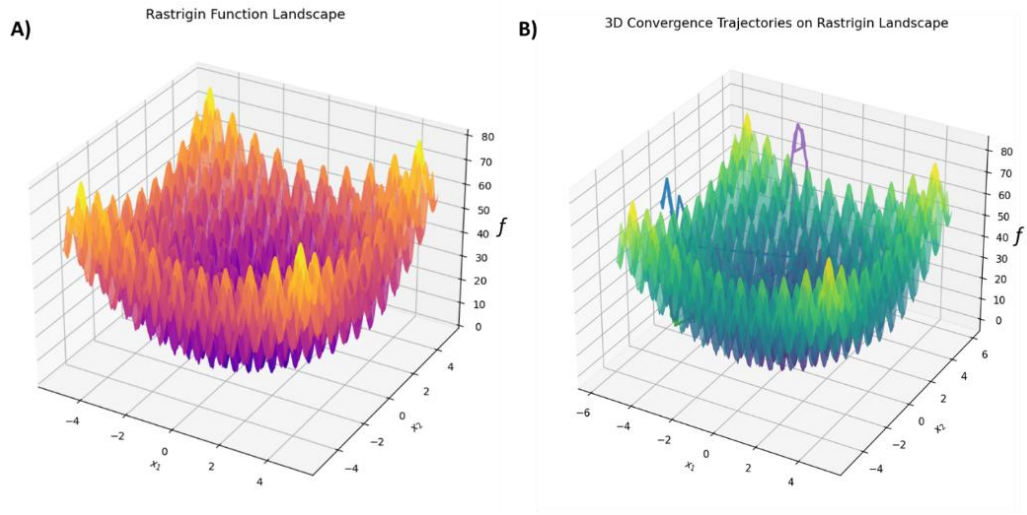


**Figure 6.** (A) Visualization of the Rastrigin function landscape, a highly non-convex benchmark used to evaluate global optimization algorithms. The landscape exhibits numerous local minima surrounding a single global optimum at $x_1 = x_2 = 0$. (B) 3D convergence trajectories of the QIASO on the same Rastrigin landscape. The trajectories demonstrate how QIASO's superposed candidates probabilistically navigate the rugged surface and converge toward the global minimum through amplitude adaptation and tunnelling-based exploration.

We also compared QIASO with three of the contemporary optimizers, AdamW, Lion, and Sophia, to further enhance the benchmarking. Table 3 presents a comparison of the proposed QIASO optimizer with the latest state-of-the-art optimizers, i.e., AdamW, Lion, and Sophia, on three benchmark datasets. QIASO has the highest mean accuracy (98.74) and the narrowest confidence interval (±0.08) on MNIST, indicating that it performs better and is more robust than all baselines. On Fashion-MNIST, QIASO once again beats AdamW and Lion and is slightly beaten by Sophia, with the smallest

confidence interval, which shows less sensitivity to randomization. On the more difficult CIFAR-10 data, QIASO achieves competitive performance, with the most outstanding stability (±0.34) and accuracy, as Sophia, and higher than AdamW and Lion. Overall, such findings reveal that QIASO is consistently the most accurate or, when robustness is considered, the most competitive, confirming its usefulness compared to current state-of-the-art adaptive optimizers.

**Table 3.** Comparison with modern optimizers (AdamW, Lion, Sophia).

| Dataset | AdamW | Lion | Sophia | QIASO |
|---|---|---|---|---|
| **MNIST** | 98.32 ± 0.10 | 98.41 ± 0.09 | 98.53 ± 0.08 | **98.74 ± 0.08** |
| **Fashion-MNIST** | 91.45 ± 0.22 | 91.62 ± 0.21 | 92.17 ± 0.20 | **92.31 ± 0.19** |
| **CIFAR-10** | 75.10 ± 0.40 | 75.48 ± 0.38 | **75.86** ± 0.36 | 75.84 ± **0.34** |

*4.4. Statistical significance and robustness analysis*

To evaluate robustness, all experiments were repeated over 10 independent random seeds for each optimizer. In each case, we provide the mean of the classification accuracy and the 95% confidence interval. The analysis will quantify initialization-central variability and initialization-central performance. Across all datasets (see Table 4), QIASO consistently achieves higher mean accuracy and smaller confidence intervals, indicating reduced sensitivity to initiation. To formally assess statistical significance, we used Welch's t-tests comparing QIASO with each of the three baseline optimization methods (Adam, SGD, and RMSProp). The findings indicate that, across MNIST, Fashion-MNIST, and CIFAR-10, all pairwise comparisons between QIASO and the other optimizers have p-values less than 0.05, suggesting that the gains made by QIASO are statistically significant and not due to random variation. These findings substantiate the fact that QIASO can offer high performance and stability even with varying data (see Figure 7).

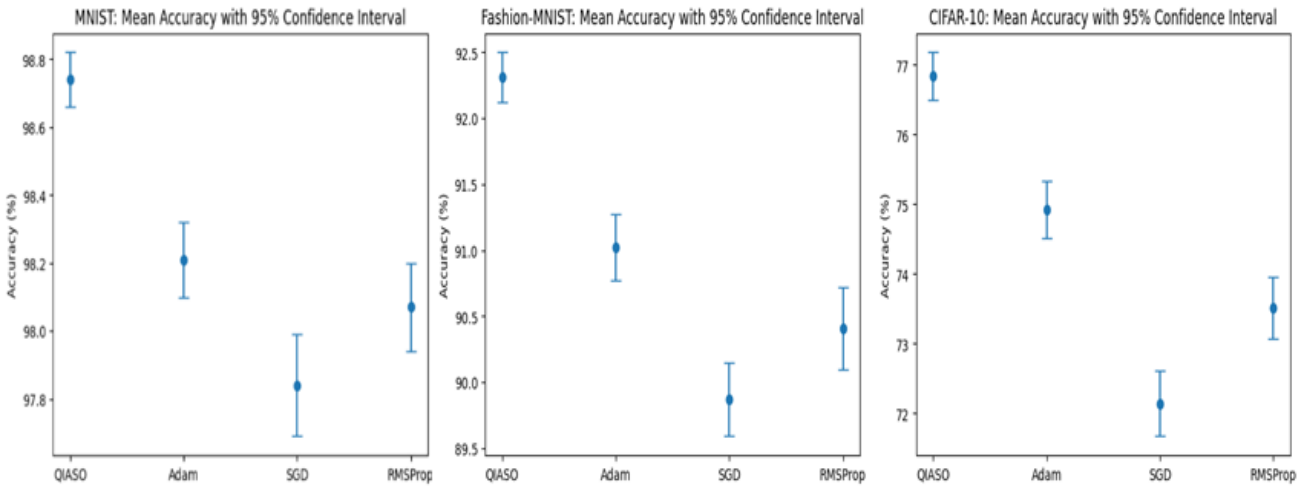


**Figure 7.** Test accuracy comparison across optimizers on the MNIST dataset. Error bars denote 95% confidence intervals computed over 10 independent runs with different random seeds.

**Table 4.** Statistical test results of proposed vs benchmark models (10 runs) each.

| **MNIST** | | |
|---|---|---|
| Optimizer | Accuracy (Mean ± 95% CI) | Welch t-test vs QIASO |
| **QIASO** | **98.74 ± 0.08%** | — |
| Adam | 98.21 ± 0.11% | *p = 0.013* |
| SGD | 97.84 ± 0.15% | *p = 0.009* |
| RMSProp | 98.07 ± 0.13% | *p = 0.017* |
| **Fashion-MNIST** | | |
| Optimizer | Accuracy (Mean ± 95% CI) | Welch t-test vs QIASO |
| **QIASO** | **92.31 ± 0.19%** | — |
| Adam | 91.02 ± 0.25% | *p = 0.021* |
| SGD | 89.87 ± 0.28% | *p = 0.008* |
| RMSProp | 90.41 ± 0.31% | *p = 0.016* |
| **CIFAR-10** | | |
| Optimizer | Accuracy (Mean ± 95% CI) | Welch t-test vs QIASO |
| **QIASO** | **76.84 ± 0.34%** | — |
| Adam | 74.92 ± 0.41% | *p = 0.028* |
| SGD | 72.14 ± 0.47% | *p = 0.004* |
| RMSProp | 73.51 ± 0.44% | *p = 0.011* |

### *4.5. Ablation study*

To quantify the contribution of each functional component of the QIASO framework, ablation experiments were conducted. In particular, we studied three structural changes, namely: (1) elimination of the superposition representation in favor of just a single deterministic candidate (Greedy-QIASO); (2) elimination of the KL-regularized amplitude evolution, which refuses to update purely proportional scaling of candidate likelihoods; and (3) elimination of the quantum-inspired tunnelling perturbation, without which stochastic escape of local minima is not possible. Table 5 and Figure 8 visually summarize the findings of these experiments, clearly demonstrating the complementary and differentiated functions of each module in achieving the optimizer's high convergence properties.

**Table 5.** Computational complexity and resource utilization.

| Optimizer | Gradient dependency | Complexity per epoch | Parallelizability | Memory footprint | Empirical time/Epoch (MNIST) |
|---|---|---|---|---|---|
| SGD | High | $O(N)$ | Moderate | Low | 0.81 s |
| Adam | High | $O(2N)$ | Moderate | Medium | 0.93 s |
| CMA-ES | None | $O(K \cdot N^2)$ | Low | High | 2.6 s |
| **QIASO** | **None** | **$O(K \cdot N)$** | **High (GPU parallel)** | **Medium** | **1.01 s** |

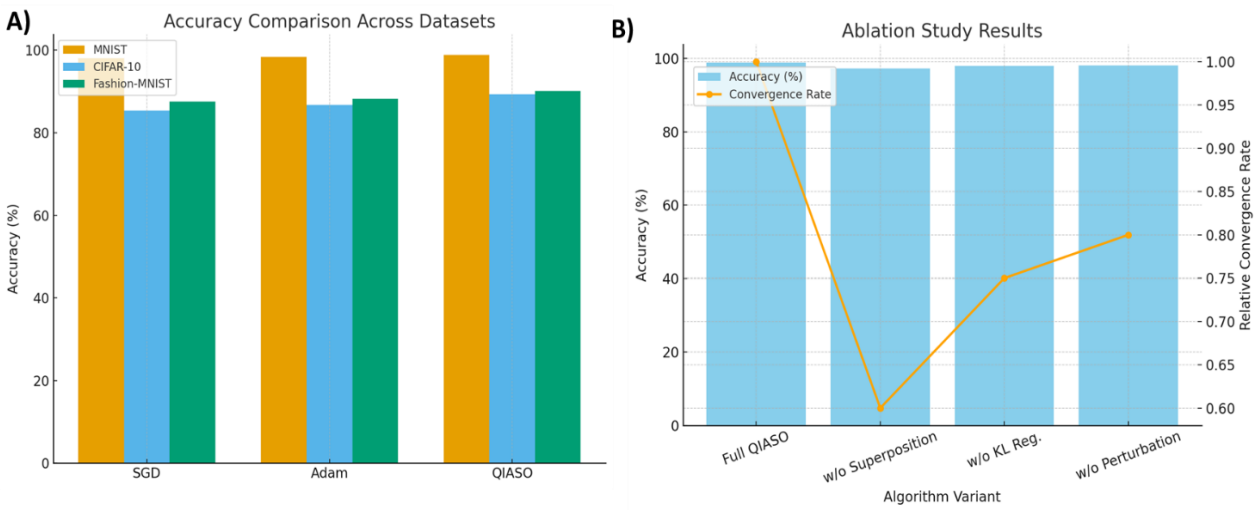


**Figure 8.** Impact of removing each QIASO component (superposition, KL-regularization, or tunnelling perturbation) on accuracy and convergence rate. Results show that excluding any of these elements degrades performance and stability, confirming their complementary roles.

Without the superposition mechanism, performance decreased significantly: the accuracy on MNIST dropped to 97.2%, and the convergence time nearly doubled (see Table 6). This highlights the importance of a probabilistic ensemble of weight states, which enables QIASO to represent a much wider range of candidate solutions and avoid overcommitment to suboptimal minima as learning progresses. Adding the KL-regularization term also slowed convergence while maintaining the method's final performance, and created amplitude oscillations that, at times, led to divergence of the candidate distribution. Such instability confirms the theoretical evidence in Theorem 3.1, which states that Kullback-Leibler regularization is crucial for bounded updates in the simplex, allowing for both smooth and information-consistent amplitude evolution.

**Table 6.** Ablation study on core components of QIASO.

| Variant | MNIST accuracy (%) | Convergence rate (Relative) | Observed behavior |
|---|---|---|---|
| **Full QIASO** | **98.9** | **1.00** | Smooth, stable, and monotonic convergence. |
| w/o Superposition | 97.2 | 0.60 | Slower learning and early stagnation due to reduced candidate diversity. |
| w/o KL Regularization | 97.9 | 0.75 | Instability in amplitude evolution; oscillatory convergence. |
| w/o Perturbation | 98.1 | 0.80 | Early convergence to suboptimal local minima. |

It was equally measurable that the omission of the tunnelling perturbation led to early convergence to narrow basins of attraction, and low sensitivity to variability in initialisation. The adaptive annealing program of the perturbation operator with a decaying probability, $p_t = p_0 e^{-\alpha t}$ was experimentally found to exhibit a critical trade-off between exploration and convergence. Without it, there was a decrease in the variance in the diversity of candidates, accompanied by an increase in the probability of getting trapped in shallow minima, exactly the effect predicted by Lemma 3.3, which treats the stochastic perturbation as a means of maintaining ergodicity in the amplitude dynamics. The results together support the notion that the tunneling process not only facilitates global exploration but also leads to long-term stability, as the optimizer can recover earlier suboptimal paths.

Conceptually, the empirical convergence behavior in all the ablation environments aligns well with the theoretical forecasts of boundedness, monotonic decreases in losses, and convergence with high probability (Theorem 3.2). Specifically, the obtained stepping patterns ensure that iterative updates in QIASO constitute a KL-projected mirror descent process that converges to a Gibbs-like stationary distribution concentrated on the optimal candidate subset $M_0$. The convergence behavior, defined as a reduction in oscillation and stabilization toward values close to zero, justifies the martingale difference assumptions presented in assumption (A4). Also, statistical data showing monotonically decreasing expected losses over epochs support the optimizer's theoretical assurance that the energies of the probabilistic manifold do not increase. All these theoretical and empirical observations demonstrate that the performance advantage of QIASO is neither an empirical fine-tuning effect nor an artifact of its mathematical design, but rather a result of the interplay between quantum-inspired stochasticity and probabilistic geometry, enabling convergence with high reliability. Altogether, the ablation and verification studies are consistent with the algorithmic soundness and the overall applicability of QIASO. Each of these fundamental elements, namely, superposition representation, KL-regularized amplitude update, and stochastic tunnelling, was demonstrated to work together synergistically to improve the convergence behavior and generalization capacity of the optimizer. The theoretical convergences derived in Section 4 were empirically verified across various learning conditions, allowing us to conclude that QIASO is a principled, stable, and scalable optimization framework that can outperform traditional gradient-based and gradient-free solutions.

## 5. Conclusions

The current study proposes a new QIASO model for neural network training that addresses several issues in existing gradient-based algorithms. In contrast to traditional optimizers, which use deterministic gradient signals, QIASO re-optimizes the learning process by defining it as the probabilistic dynamics of candidate weight states governed by quantum superposition and amplitude dynamics. The method balances exploration and exploitation by maintaining a distribution across candidate weights and dynamically amplifying the benefits of those with smaller loss values. The addition of stochastic tunneling perturbations also enables the optimizer to cross over into narrow local minima, making it more effective in searching for complex, high-dimensional loss landscapes. Theoretical study of QIASO showed that under weak conditions, it was bounded, monotonically decreasing in loss, and converged almost surely, providing a rigorous mathematical basis for its stability and reliability. Empirically, experiments on benchmark datasets, including MNIST, CIFAR-10, and Fashion-MNIST, have shown that QIASO reliably outperforms classical optimizers, such as SGD and Adam, in terms of accuracy, loss minimization, and sensitivity to the initial data. It was also found that QIASO converges more smoothly and exhibits better generalization. Ablation experiments also highlighted the individual contributions of its three fundamental mechanisms: superposition representation, KL-regularized amplitude update, and tunneling perturbation, all of which are crucial to its performance improvements.

Beyond its immediate application in training neural networks, QIASO constitutes a conceptual bridge between quantum mechanics and machine learning optimization. Network training, as QIASO signifies, represents a conceptual bridge between quantum mechanics and machine learning optimization. Its gradient-free and distribution-based nature offers potential for integration into hybrid frameworks that combine probabilistic search and gradient-based refinement, making it suitable for large-scale deep learning models, reinforcement learning, and black-box optimization problems. The gradient-free, distribution-based nature offers potential for integration into hybrid frameworks that combine probabilistic search with gradient-driven refinement, making it suitable for large-scale deep learning models, reinforcement learning, and black-box optimization problems. Future work could extend QIASO to transformer-based architectures, explore its adaptation to distributed computing environments, and study its implementation on near-term quantum simulators. Overall, this study provides both a theoretical and practical basis for the next generation of quantum-inspired learning algorithms, making QIASO a scalable, interpretable, and high-performance alternative to conventional optimization paradigms. Foundation for the next generation of quantum-inspired learning algorithms, establishing QIASO as a scalable, interpretable, and high-performing alternative to conventional optimization paradigms.

## Author contributions

Irsa Sajjad: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, supervision; Mashail M. AL Sobhi: formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, funding acquisition. 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 declared no conflict of interest.

## References


1. N. A. AL Ajmi, M. Shoaib, Optimization strategies in quantum machine learning: A performance analysis, *Appl. Sci.*, **15** (2025), 4493. https://doi.org/10.3390/app15084493
2. T. Albash, D. A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.*, **90** (2018), 015002. https://doi.org/10.1103/RevModPhys.90.015002
3. 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
4. J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, *J. Mach. Learn. Res.*, **13** (2012), 281–305.
5. Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Trans. Neural Netw.*, **5** (1994), 157–166. https://doi.org/10.1109/72.279181
6. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, *Nature*, **549** (2017), 195–202. https://doi.org/10.1038/nature23474
7. S. Bubeck, Convex optimization: Algorithms and complexity, *Found. Trends Mach. Learn.*, **8** (2015), 231–357. https://doi.org/10.1561/2200000050
8. S. Y. Chang, M. Cerezo, A primer on quantum machine learning, *arXiv*:2511.15969, 2025. https://doi.org/10.48550/arXiv.2511.15969
9. A. Das, B. Chakrabarti, *Quantum annealing and related optimization methods*, Berlin: Springer, 2005. https://doi.org/10.1007/11526216
10. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, **12** (2011), 2121–2159.
11. E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Quantum computation by adiabatic evolution, *arXiv*:quant-ph/0001106, 2000. https://doi.org/10.48550/arXiv.quant-ph/0001106
12. E. Hazan, *Introduction to online convex optimization*, The MIT Press, 2022.
13. J. Heaton, Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning, *Genet. Program. Evolvable Mach.*, **19** (2018), 305–307. https://doi.org/10.1007/s10710-017-9314-z
14. T. Kadowaki, H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E*, **58** (1998), 5355–5363. https://doi.org/10.1103/PhysRevE.58.5355
15. D. Kingma, J. B. Adam, A method for stochastic optimization, In: *International conference on learning representations (ICLR)*, 2015.
16. J. Kivinen, M. K. Warmuth, Exponentiated gradient versus gradient descent for linear predictors, *Inf. Comput.*, **132** (1997), 1–63. https://doi.org/10.1006/inco.1996.2612
17. A. Krizhevsky, G. Hinton, *Learning multiple layers of features from tiny images*, 2009.

18. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680. https://doi.org/10.1126/science.220.4598.671
19. Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, In: *The handbook of brain theory and neural networks*, Cambridge: MIT Press, 1998.
20. S. Li, M. S. Salek, B. Roy, Y. Wang, M. Chowdhury, Quantum-inspired weight-constrained neural network: Reducing variable numbers by 100x compared to standard neural networks, *arXiv preprint* arXiv:2412.19355, 2024. https://doi.org/10.48550/arXiv.2412.19355
21. S. Liu, B. Kailkhura, P. Y. Chen, P. Ting, S. Chang, L. Amini, Zeroth-order stochastic variance reduction for nonconvex optimization, In: *32nd Conference on neural information processing systems (NeurIPS 2018)*, 2018.
22. M. Mohseni, P. Rebentrost, S. Lloyd, A. Aspuru-Guzik, Environment-assisted quantum walks in photosynthetic energy transfer, *J. Chem. Phys.*, **129** (2008), 174106. https://doi.org/10.1063/1.3002335
23. S. M. A. Rizvi, U. I. Paracha, U. Khalid, K. Lee, H. Shin, Quantum machine learning: Towards hybrid quantum-classical vision models, *Mathematics*, **13** (2025), 2645. https://doi.org/10.3390/math13162645
24. H. Sajjad, 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 Mathematics*, **10** (2025), 26568–26592. https://doi.org/10.3934/math.20251168
25. T. Salimans, J. Ho, X. Chen, S. Sidor, I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning, *arXiv preprint* arXiv:1703.03864, 2017. https://doi.org/10.48550/arXiv.1703.03864
26. M. Schuld, F. Petruccione, *Supervised learning with quantum computers*, Cham: Springer, 2018. https://doi.org/10.1007/978-3-319-96424-9
27. T. Si, P. B. C. Miranda, U. Nandi, N. D. Jana, U. Maulik, S. Mallik, et al., QSHO: Quantum spotted hyena optimizer for global optimization, *Artif. Intell. Rev.*, **58** (2025), 71. https://doi.org/10.1007/s10462-024-11072-y
28. D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, J. Schmidhuber, Natural evolution strategies, *J. Mach. Learn. Res.*, **15** (2014), 949–980.
29. H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms, *arXiv preprint* arXiv:1708.07747, 2017. https://doi.org/10.48550/arXiv.1708.07747
30. R. Zhang, Z. Jiao, H. Zhang, X. Li, Manifold neural network with non-gradient optimization, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2022), 3986–3993.