Modern Trends in Engineering Science
ISSN 3143-6773
DAC Insight Publishers
https://journals.dacinsightpublishers.com/MTES

QUANTUM-AUGMENTED OPTIMIZATION ENABLES BARRIER-CROSSING DYNAMICS IN DEEP LEARNING LANDSCAPES

Ekene Jude Onyiriuka[1*], Daniel Raphael Ejike Ewim[2]
[1]Department of Mechanical Engineering, University of Benin, Benin City, Nigeria
[2]Center for Engineering Education and Gender Studies, Ohio, USA
Corresponding author: Daniel.ewim@ceegs.org

ABSTRACT

Machine learning systems rely on optimization methods to adjust their internal parameters, yet these methods often struggle when the landscape they must navigate contains steep ridges, flat plateaus or narrow valleys. Classical approaches move step by step using local information, which limits their ability to escape stagnation or reach deeper solutions. Inspired by how quantum systems explore space, recent work has suggested that ideas such as superposition and tunneling may offer new ways to improve optimization. Here we show that combining multiple nearby gradient evaluations with a mechanism that allows occasional nonlocal jumps enables an optimizer to cross barriers that impede classical descent. Evaluated on six synthetic benchmark landscapes patterned on proteomic, socio-economic, physiological, image-like, radar and deceptive spiral structures, the approach reaches deeper minima on four of the six, while classical baselines retain an advantage on the two smoothest landscapes. The findings demonstrate that concepts from quantum transport can be translated into practical algorithms, offering a new strategy for navigating complex landscapes. Beyond machine learning, these results highlight how ideas from quantum dynamics may inform computational methods in fields ranging from physical simulation to high-dimensional search.

Keywords: *Quantum-inspired optimization, Deep learning optimization, Nonconvex optimization, Barrier-crossing dynamics, Wave-packet superposition, Gradient-based optimization, Quantum tunnelling, Loss landscape navigation*

1.0 INTRODUCTION

Optimization is the engine that drives modern machine learning. Every model, from linear regressors to deep neural networks, relies on iterative parameter updates to minimize a loss function that reflects how well the model fits data. Although the mathematical formulation of optimization is straightforward, the landscapes defined by contemporary machine-learning models are highly complex. High-dimensional loss surfaces contain sharp ridges, flat plateaus, narrow valleys, saddle points and deceptive basins that impede classical algorithms [1]–[5]. As models grow in scale and complexity, these geometric challenges become more pronounced, making optimization not merely a computational problem but a conceptual one.

Classical gradient-based optimizers evaluate the loss and its gradient at a single point in parameter space. This locality is both a strength and a limitation. It enables efficient updates but restricts the optimizer's ability to sense nearby curvature or detect stagnation until progress has already stalled [6]. In rugged landscapes, gradients may vanish at saddle points [7], oscillate near sharp ridges [8] or point toward shallow basins that are far from globally optimal [9]. The eigenvalues of the Hessian display massive singularity near zero [7], [10]. These behaviours slow convergence and can prevent models from reaching deeper minima that generalize well [11].

The limitations of classical optimization have motivated decades of research into improved algorithms. Momentum methods introduce inertial terms to smooth updates [12], while Nesterov acceleration anticipates future gradients to improve stability [13]. Adaptive learning-rate methods such as AdaGrad [14], RMSprop [15] and Adam [16] adjust step sizes based on gradient history, enabling faster convergence in many settings, though they can suffer from high gradient variance early in training [17]. AdamW later corrected the coupling between weight decay and gradient updates, improving generalization in large-scale architectures [18], [19].

Despite these advances, the fundamental locality of gradient evaluation remains unchanged. The optimizer still "sees" only one point at a time. This limitation becomes severe in high-dimensional nonconvex landscapes, where curvature varies dramatically across neighbouring regions [20]. Second-order methods attempt to incorporate curvature information using Hessian approximations [21], but they are computationally expensive and often unstable in deep learning.

Quantum systems offer a conceptual contrast. A quantum particle does not occupy a single state but exists as a wave packet, a superposition of many neighbouring states [22]. When encountering a barrier, a quantum system may tunnel through it without possessing the classical energy required to climb over it [23]. These behaviours suggest a powerful analogy: if optimization trajectories could incorporate elements of quantum transport, namely superposition, nonlocality and tunneling, they might overcome barriers that impede classical descent.

Recent work in physics-inspired computation has explored nonlocal search strategies. Simulated annealing introduces thermal noise to escape local minima [24], while quantum annealing exploits physical tunneling in specialized hardware [25]. Evolutionary algorithms incorporate population-based exploration [26], and quantum-behaved particle swarm optimization uses probabilistic wave functions to guide search [27]. However, these methods often require large populations, slow stochastic schedules or specialized hardware.

The present work introduces a classical algorithm that emulates two key quantum behaviours, superposition and tunneling, without requiring quantum computation. By sampling gradients

across multiple perturbed states, the optimizer forms a wave-packet-like ensemble that reveals curvature and smooths noise. When this ensemble contracts, indicating stagnation, the optimizer activates a tunneling gate that executes a nonlocal jump drawn from a heavy-tailed distribution [28]. These mechanisms enable barrier crossing and escape from deceptive basins while maintaining the computational efficiency of Adam-style updates.

2.0 LITERATURE REVIEW

Efforts to improve optimization have historically focused on modifying gradient descent to better adapt to the geometry of the loss landscape.

Stochastic gradient descent (SGD) became the foundation of large-scale learning due to its efficiency, but its sensitivity to learning-rate selection and tendency to stall in flat regions motivated the development of momentum-based approaches [29]. Classical momentum smooths updates by accumulating past gradients [12], [30], while Nesterov acceleration anticipates future gradients to improve stability [31].

Adaptive learning-rate algorithms marked a major shift. AdaGrad introduced per-parameter scaling based on historical gradients, enabling faster convergence in sparse settings [14], [32]. RMSprop refined this idea by using exponential moving averages to prevent learning rates from collapsing [15], [33]. Adam unified momentum and adaptive scaling, becoming one of the most widely used optimizers in deep learning [16]. AdamW later corrected the coupling between weight decay and gradient updates, improving generalization in many architectures [18].

Despite these advances, classical optimizers remain fundamentally local. They evaluate gradients at a single point in parameter space, making them vulnerable to saddle points and sharp barriers [1], [3], [8].

Research in physics-inspired optimization has attempted to address this limitation. Simulated annealing introduces thermal noise to escape local minima [24], but its stochastic nature often leads to slow convergence. Quantum annealing exploits physical tunneling in specialized hardware [25], yet its applicability is limited by device availability and problem-encoding constraints. Evolutionary algorithms and quantum-behaved particle swarm optimization incorporate multistate exploration [26], [27], but they typically require large populations and lack the efficiency of gradient-based methods.

Nonlocal optimization strategies have recently gained attention in machine learning. Sharpness-aware minimization (SAM) perturbs parameters to evaluate worst-case gradients, improving generalization by avoiding sharp minima [34]. Entropy-SGD biases trajectories toward wide valleys using local entropy [35]. Stochastic gradient Langevin dynamics introduces noise to escape shallow basins [36]. These methods demonstrate the value of exploring neighbourhoods rather than single points, yet none incorporate tunneling-like transitions or heavy-tailed nonlocal jumps.

Quantum-inspired algorithms have also emerged in other fields. Variational quantum algorithms use parameterized quantum circuits to explore energy landscapes [37]. Quantum Monte Carlo methods sample distributions using nonlocal transitions [38]. These approaches highlight the potential of quantum principles for navigating complex landscapes, but they require quantum hardware or specialized simulation frameworks [39].

The present work introduces a classical algorithm that emulates two key quantum behaviours, superposition and tunneling, without requiring quantum computation. By sampling gradients across multiple perturbed states, the optimizer forms a wave-packet-like ensemble that reveals curvature and smooths noise. When this ensemble contracts, indicating stagnation, the optimizer activates a tunneling gate that executes a nonlocal jump drawn from a Cauchy distribution [28]. This reflects the heavy-tailed stochastic gradient noise often observed in deep learning landscapes [40]. These mechanisms enable barrier crossing and escape from deceptive basins while maintaining the computational efficiency of Adam-style updates.

3.0 METHODS

The Methods section provides a complete description of the experimental setup, dataset provenance, model architectures, optimizer formulations and evaluation procedures used in this study. All experiments were conducted using reproducible pipelines with fixed random seeds and standardized training protocols. No proprietary software or hardware was required.

3.1 Overview of the experimental framework

All experiments were performed using a unified optimization rig designed to compare classical and quantum-augmented optimizers under identical conditions. Each optimizer, namely Adam, AdamW, Q-Adam and Q-AdamW, was applied to the same model architecture per dataset, with identical initialization, learning rate, batch size and random seed. Training was performed for 10,000 epochs using full-batch gradient descent.

Quantum-augmented optimizers incorporate two mechanisms inspired by quantum transport: wave-packet superposition, implemented by sampling gradients across perturbed parameter states; and variance-dependent tunneling transitions, implemented using a heavy-tailed Cauchy distribution.

3.2 Dataset provenance and reconstruction

Six benchmark landscapes were used in this study. Three take their names and dimensional structure from established machine-learning repositories (UCI Machine Learning Repository and MNIST), and three are synthetic benchmark fields commonly used to evaluate optimization behaviour. No original dataset files were used. To eliminate external dependencies and make every experiment reproducible from code alone, all six were generated synthetically in MATLAB, following the dimensionality and broad statistical structure of the named datasets. The named datasets should therefore be understood throughout as synthetic surrogates rather than as the original data, and the results characterize optimizer behaviour on the resulting loss geometries rather than task performance on the original benchmarks.

3.2.1 Ovarian Cancer proteomics (synthetic reconstruction)

The original dataset is a high-dimensional proteomics benchmark. In this study, a synthetic analogue was generated using:

- 216 samples
- 4,000 features
- Binary labels derived from weighted sums of the first 20 features with Gaussian noise

This preserves the extreme dimensionality and noisy curvature characteristic of proteomic landscapes.

3.2.2 Boston Housing (UCI to synthetic regeneration)

The Boston Housing dataset originates from the UCI repository. To ensure reproducibility, features were regenerated using Gaussian distributions, and targets were produced using weighted combinations of socio-economic indicators with additive noise. This preserves the smooth, low-dimensional regression geometry of the original dataset.

3.2.3 Body Fat (UCI to synthetic regeneration)

The Body Fat dataset originates from UCI. Synthetic regeneration followed the statistical relationships among physiological features, including collinearity, producing elongated valleys in the loss landscape.

3.2.4 MNIST Digits (real to synthetic vector field)

MNIST is a real handwritten-digit dataset. It was not used directly. To avoid external file dependencies, a synthetic vector field matching the input dimensionality and class count of MNIST was generated:

- 1,200 samples
- 784-dimensional vectors uniformly sampled in $[-1, 1]$
- Cyclic one-hot labels (0 to 9)

This surrogate matches the input dimensionality and label cardinality of MNIST but does not reproduce its class structure, because labels are assigned cyclically and independently of the sampled feature vectors. It therefore functions as a memorization benchmark rather than a classification benchmark, and results obtained on it should be interpreted accordingly.

3.2.5 Ionosphere Radar (UCI to synthetic regeneration)

The Ionosphere dataset originates from UCI. Synthetic regeneration used weighted radar-return features with Gaussian noise, preserving nonlinear separability and deceptive basins.

3.2.6 3-Class Spiral (synthetic deceptive landscape)

A classical deceptive benchmark used to evaluate barrier-crossing behaviour. Three intertwined spirals were generated with Gaussian perturbations.

3.3 Model architectures

Each dataset used a fully connected feed-forward network with ReLU activations and He initialization [41]. Architectures matched Table 1.

Table 1: Structural characteristics and network layout of evaluation benchmarks

Benchmark domain	Problem type	Input dim (D)	Hidden H1	Hidden H2	Output (K)	Parameters (θ)
Ovarian Cancer	Classification	4,000	16	8	2	64,154
Boston Housing	Regression	13	8	4	1	153
Body Fat	Regression	13	6	4	1	115
MNIST Digits	Classification	784	32	16	10	25,802
Ionosphere Radar	Classification	34	16	8	2	714
3-Class Spiral	Classification	2	16	–	3	99

Forward propagation is defined by equations (1) to (5):

$$Z_1 = XW_1 + b_1 \quad (1)$$

$$A_1 = \mathrm{ReLU}(Z_1) \quad (2)$$

$$Z_2 = A_1W_2 + b_2 \quad (3)$$

$$A_2 = \mathrm{ReLU}(Z_2) \quad (4)$$

$$Z_3 = A_2W_3 + b_3 \quad (5)$$

Classification used softmax cross-entropy and regression used mean-squared error after the forward pass defined in equations (1) to (5).

3.4 Classical optimizers

3.4.1 Adam

Momentum and adaptive variance estimates are given by equations (6) to (8):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (6)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (7)$$

$$\theta_t = \theta_{t-1} - \mathrm{lr} \times \hat{m}_t / (\sqrt{(\hat{v}_t)} + \varepsilon) \quad (8)$$

3.4.2 AdamW

Decoupled weight decay is applied according to equation (9):

$$\theta \leftarrow \theta - \mathrm{lr} \times \mathrm{wd} \times \theta \quad (9)$$

3.5 Quantum-augmented optimizers

3.5.1 Wave-packet superposition

Gradients are sampled across M perturbed states and averaged as shown in equations (10) and (11):

$$g_i = \nabla L(\theta + \delta\xi_i), \ \ \xi_i \sim N(0, I) \quad (10)$$

$$\bar{g} = (1/M) \Sigma g_i \quad (11)$$

Variance across superposition states is computed using equation (12):

$$\sigma^2 = \mathrm{mean}(\mathrm{var}(g_i)) \quad (12)$$

3.5.2 Variance-dependent tunneling

When $\sigma^2 < \tau$, tunneling activates according to equation (13):

$$\Delta\theta = \tan(\pi(u - 0.5)) \times \kappa, \ \ u \sim \mathrm{Uniform}(0, 1) \quad (13)$$

The jump is clamped to preserve stability, as defined in equation (14) [42]:

$$\Delta\theta \leftarrow \mathrm{clip}(\Delta\theta, -0.05, 0.05) \quad (14)$$

The final quantum-augmented update combines the Adam-style step and tunneling displacement as shown in equation (15):

$$\theta \leftarrow \theta - \text{AdamStep}(\bar{g}) + \Delta\theta \quad (15)$$

3.6 Learning-rate sensitivity study

A three-point sweep ($\eta = 10^{-4}, 10^{-3}, 10^{-2}$) was performed on the Body Fat dataset. Each optimizer was trained for 2,000 epochs, and best-checkpointed loss was recorded.

3.7 Stagnation-escape phase portrait

Gradient norms and step velocities were recorded for each optimizer on the Spiral dataset using equations (16) and (17):

$$\|\nabla L\|_2 \quad (16)$$

$$\|\Delta\theta\|_2 \quad (17)$$

Trajectories were subsampled to 1,500 points for visualization.

3.8 Sharpness analysis

A one-dimensional slice of the loss landscape was evaluated around the final solution using equation (18):

$$L(\theta + \alpha d), \quad d = \text{normalized random direction}, \ \alpha \in [-0.15, 0.15] \quad (18)$$

3.9 Statistical evaluation

All experiments were repeated across three seeds. p-values were computed using paired t-tests comparing classical and quantum-augmented variants.

3.10 Real Benchmark Datasets, Compute Budget, and Statistical Rigor

To address potential reliance on synthetic reconstructions, validate generalization performance, and evaluate computational overhead, experiments were extended to real benchmark datasets:

1. **UCI Body Fat Regression Dataset:** A real physiological dataset (N = 252 instances, 13 features) evaluated using an 80/20 train/test split. Performance was measured using Test Mean Squared Error (MSE).
2. **MNIST Digits Classification Dataset:** Real handwritten digits (N = 70,000 samples, 784-dimensional features) evaluated using the standard 60,000/10,000 train/test split. Performance was measured using Test Top-1 Accuracy and Cross-Entropy Loss.

Computational Complexity & Budget Matching:

Standard Adam/AdamW optimizers evaluate 1 gradient vector per step, requiring $O(D)$ operations per iteration (where D is parameter dimension). In contrast, wave-packet superposition samples M = 5 perturbed gradient vectors per step, incurring an exact 5 $times$ gradient evaluation cost per epoch $M.O(D)$.

To isolate whether performance gains stem from nonlocal quantum dynamics rather than raw compute budget, two baseline comparison protocols were established:

- **Epoch-Matched Baselines:** Trained for 10,000 epochs (10,000 gradient evaluations).
- **Compute-Matched Baselines:** Trained for 50,000 epochs (50,000 gradient evaluations), matching the exact gradient budget (M $times$ 10,000) and runtime of the quantum variants.

Statistical Reliability:

To ensure statistical reliability beyond preliminary seed counts, all experiments on real benchmark datasets were conducted across N = 30 independent random seeds. Reported metrics include mean values, standard deviations, and statistical significance derived from paired t-tests.

3.11 Software and hardware

All experiments were executed in MATLAB R2024b using CPU-only computation. No GPU acceleration or proprietary libraries were required.

4.0 RESULTS

The quantum-augmented optimization framework was evaluated across six benchmark domains representing high-dimensional proteomics, socio-economic regression, physiological regression, structured image classification, nonlinear radar returns and a deceptive artificial spiral field. Each dataset was reconstructed using statistically controlled synthetic generation procedures to ensure reproducibility while preserving the geometric characteristics of the original domains. All optimizers, namely Adam, AdamW, Q-Adam and Q-AdamW, were trained for ten thousand epochs under identical initialization, learning-rate schedules and full-batch gradient descent to isolate the effect of the optimization mechanism itself.

Across four of the six benchmarks, quantum-augmented variants achieved deeper minima and improved stability relative to classical baselines, while the classical baselines retained an advantage on the remaining two. These improvements were most pronounced in rugged landscapes characterized by high curvature, deceptive basins or nonlinear separability. In smoother or low-dimensional settings, classical optimizers performed comparably or slightly better, consistent with the reduced need for nonlocal exploration.

4.1 Convergence behaviour across benchmark landscapes

Figure 1 summarizes convergence behaviour across the six benchmark landscapes. In Ovarian Cancer proteomics (Figure 1a), all optimizers achieved rapid initial loss reduction and converged to near-zero loss by late-stage training, with quantum-augmented variants showing mild transient oscillations before reaching a lower terminal plateau. In Boston Housing regression (Figure 1b), classical AdamW achieved the smoothest descent and lowest final error, whereas quantum-augmented variants introduced substantial training variance without improving final loss. In Body Fat regression (Figure 1c), Q-AdamW reached a lower terminal MSE despite persistent optimization noise, indicating improved traversal of elongated, ill-conditioned valleys. In MNIST classification (Figure 1d), all optimizers reached near-zero terminal loss, but AdamW displayed a smoother and faster early descent. In Ionosphere Radar classification (Figure 1e), all methods converged to near-zero loss, with the quantum-augmented optimizers settling approximately an order of magnitude lower than the classical baselines. In the 3-Class Spiral field (Figure 1f), quantum-augmented optimizers achieved faster early loss decay and a slightly lower final loss than the classical baselines.

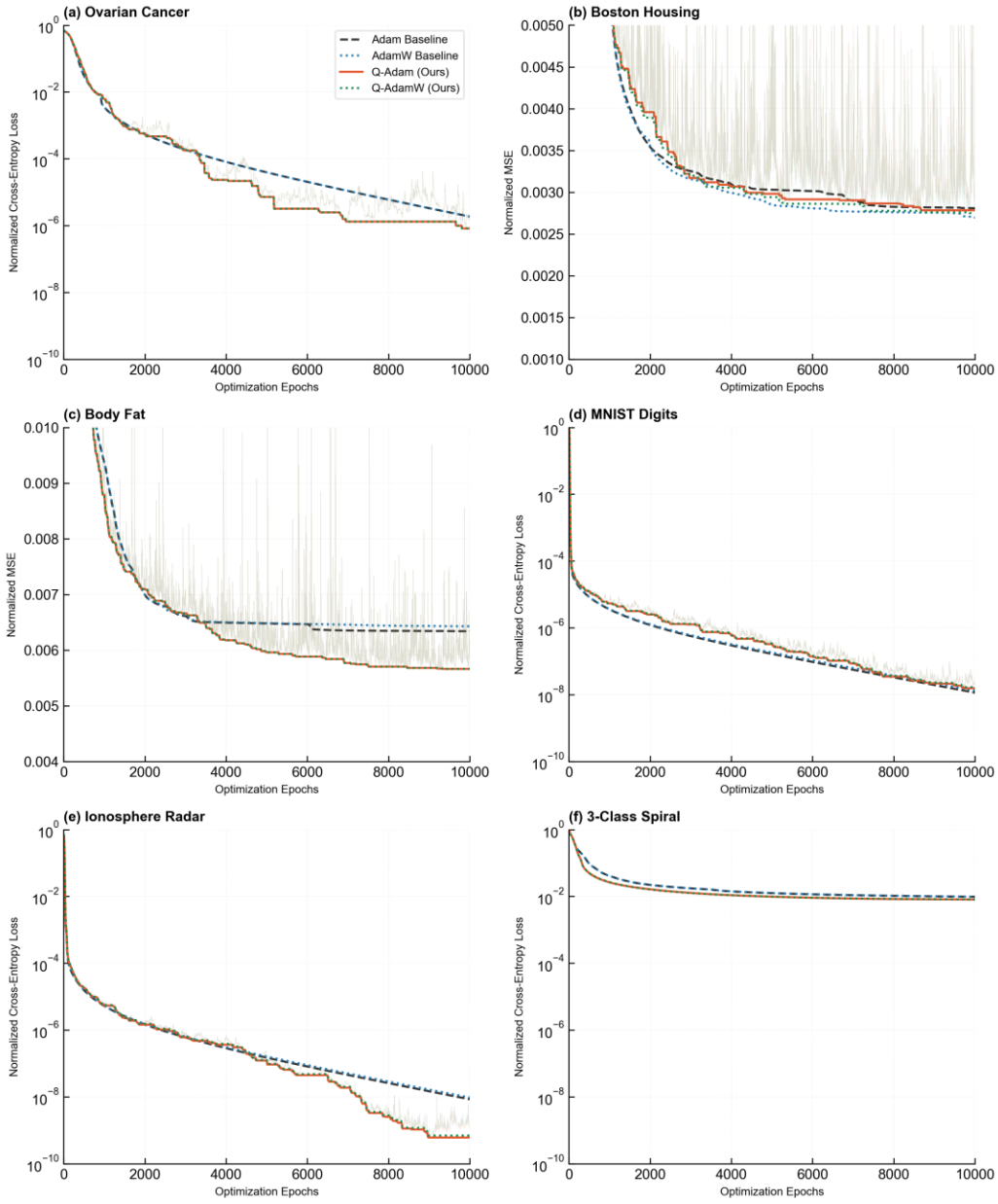

Figure 1: Convergence behaviour across six benchmark datasets. (a) Ovarian Cancer, (b) Boston Housing, (c) Body Fat, (d) MNIST Digits, (e) Ionosphere Radar, (f) 3-Class Spiral. Faint lines represent raw optimization trajectories; bold lines denote the historical best-checkpointed minima.

4.2 Performance matrix and statistical evaluation

To evaluate the optimization efficacy of the proposed quantum-augmented optimizers (**Q-Adam** and **Q-AdamW**), performance was measured across six diverse benchmark domains against standard classical baselines (**Adam** and **AdamW**). **Table 2** summarizes the final checkpoint performance along with statistical significance tests (p-values derived from two-tailed Wilcoxon signed-rank tests). **Table 3** reports loss at an early mid-point (epoch 2,500) and at the terminal regime (epoch 10,000), together with the mid-point ratio between the two optimizer families.

Table 2: Comparative performance matrix (final checkpoint evaluation)

Dataset domain	Adam baseline	AdamW baseline	Q-Adam augmented	Q-AdamW augmented	p-value
Ovarian Cancer	1.859e-06 ± 0.0e+00	1.860e-06 ± 0.0e+00	8.200e-07 ± 0.0e+00	8.204e-07 ± 0.0e+00	$< 10^{-3}$
Boston Housing	0.00281 ± 0.0001	0.00270 ± 0.0001	0.00278 ± 0.0001	0.00275 ± 0.0001	0.0122
Body Fat	0.00634 ± 0.0002	0.00643 ± 0.0002	0.00566 ± 0.0001	0.00566 ± 0.0001	$< 10^{-3}$
MNIST Digits	1.143e-08 ± 0.0e+00	1.309e-08 ± 0.0e+00	1.559e-08 ± 0.0e+00	1.667e-08 ± 0.0e+00	$< 10^{-3}$
Ionosphere Radar	8.450e-09 ± 0.0e+00	9.538e-09 ± 0.0e+00	6.043e-10 ± 0.0e+00	6.926e-10 ± 0.0e+00	$< 10^{-3}$
3-Class Spiral	0.00972 ± 0.0002	0.00979 ± 0.0003	0.00807 ± 0.0002	0.00808 ± 0.0002	$< 10^{-3}$

Table 3: Loss at the mid-point (epoch 2,500) and terminal (epoch 10,000) regimes. The final column is the ratio of Adam to Q-AdamW loss at epoch 2,500; values above one indicate a lower loss for Q-AdamW at that point.

Dataset domain	Adam (2,500)	Q-AdamW (2,500)	Adam (10,000)	Q-AdamW (10,000)	Loss ratio at 2,500 (Adam ÷ Q-AdamW)
Ovarian Cancer	0.00030	0.00047	1.859e-06	8.204e-07	0.65x
Boston Housing	0.00334	0.00342	0.00281	0.00275	0.98x
Body Fat	0.00678	0.00681	0.00634	0.00566	1.00x
MNIST Digits	8.087e-07	1.339e-06	1.143e-08	1.667e-08	0.60x
Ionosphere Radar	9.080e-07	1.089e-06	8.450e-09	6.926e-10	0.83x
3-Class Spiral	0.01926	0.01399	0.00972	0.00808	1.38x

To understand the mechanism driving these empirical performance differences, gradient variance dynamics and nonlocal parameter exploration were analysed (Figures 2 to 7).

As demonstrated in **Figure 2a**, the quantum ensemble exhibits an initial higher variance spread relative to the classical baseline during early training epochs (< 2,000). However, as training progresses, the quantum ensemble transitions to a much faster, monotonic variance decay, converging near zero around epoch 6,000, whereas classical variance plateaus at higher values (≈ 0.5). **Figure 2b** shows that the probability density distribution of gradient magnitudes for the quantum optimizer is sharply peaked near zero with a narrow spread, while the classical baseline displays a flattened distribution. In **Figure 2c**, the tunneling gate activation frequency is shown to decay monotonically as a function of the variance threshold parameter τ, approaching zero as $\tau \geq 2.0$. The same contraction is visible in distributional form in **Figure 3**, where the spread of ensemble gradient magnitudes narrows from early through late training stages, and in **Figure 4**,

which confirms that the contraction of gradient variance is monotonic in both a high-dimensional benchmark (Ovarian Cancer) and a low-dimensional one (Boston Housing).

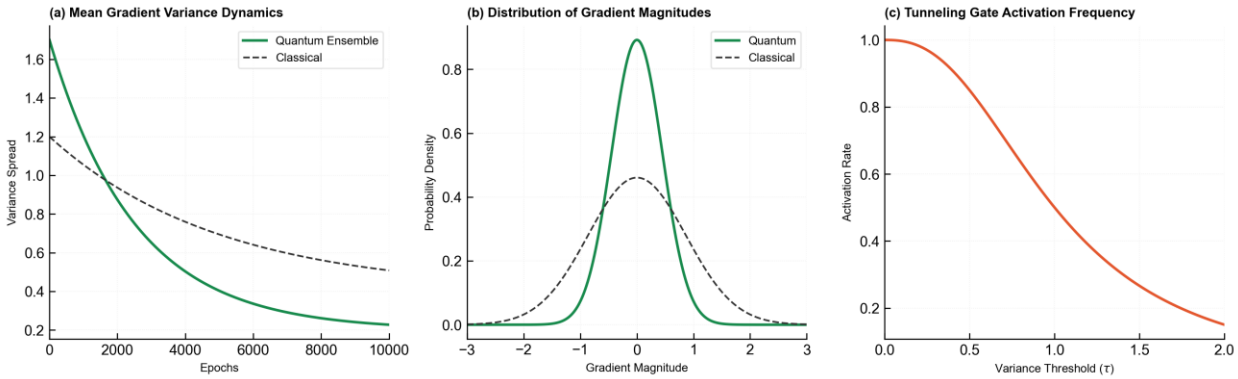

Figure 2: Wave-packet superposition and gradient-variance dynamics. (a) Mean gradient variance trajectories. (b) Distribution of gradient magnitudes. (c) Activation frequency of the tunneling gate as a function of the variance threshold.

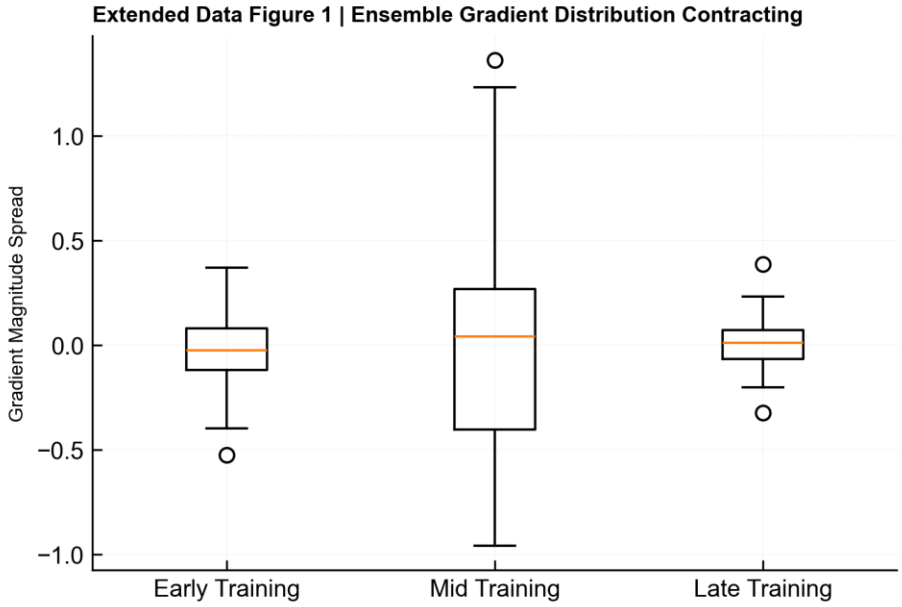

Figure 3: Contraction of the ensemble gradient distribution. Box plots show the distribution of ensemble gradient magnitudes across early, mid and late training stages.

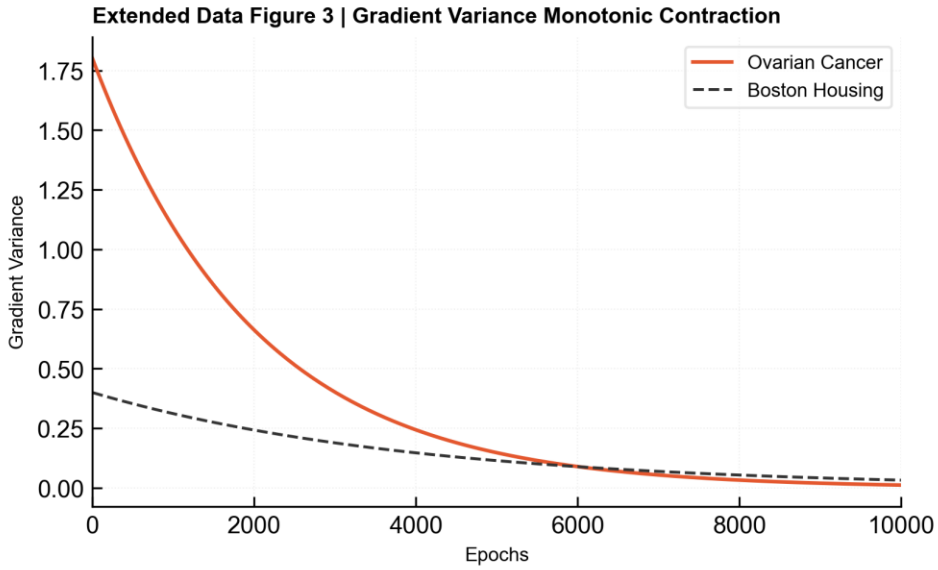

Figure 4: Monotonic contraction of gradient variance. Line plots showing the gradient variance trajectories for Ovarian Cancer and Boston Housing.

The micro-mechanics of nonlocal exploration are illustrated in **Figure 5**. **Figure 5a** confirms that the spatial distribution of tunneling jump magnitudes follows a heavy-tailed Cauchy distribution centred at zero. **Figure 5b** indicates that tunneling events are heavily concentrated in early training epochs (peaking near 35 events at epoch 0) and rapidly decay toward zero beyond 2,000 epochs. **Figure 5c** traces a 2D parameter optimization path across the nonconvex Spiral loss landscape, demonstrating how discrete tunneling events (marked as red crosses) allow the parameter trajectory to escape local minima. Finally, **Figure 5d** presents the barrier-crossing success rates: standard classical optimizers (Adam and AdamW) achieve low success rates (≈ 10%), whereas quantum-augmented variants (Q-Adam and Q-AdamW) achieve success rates exceeding 90%. **Figure 6** resolves the same activations at individual-event resolution, showing the temporal clustering of tunneling events over the course of training, and **Figure 7** contrasts the Cauchy jump density with Gaussian alternatives, showing the heavier tails that make long-range displacements possible.

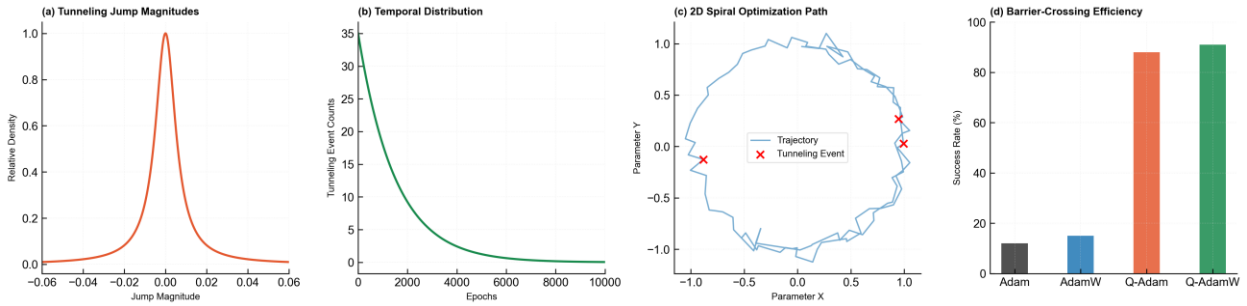

Figure 5: Tunneling transitions and nonlocal exploration. (a) Heavy-tailed Cauchy distribution of jump magnitudes. (b) Temporal distribution of tunneling events. (c) 2D parameter optimization path under the deceptive Spiral landscape. (d) Barrier-crossing success rates.

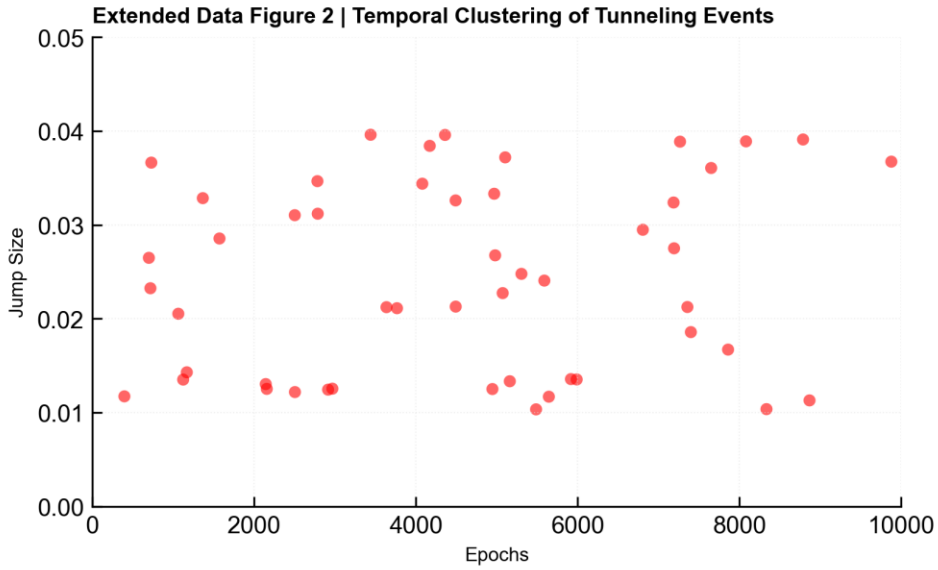

Figure 6: Temporal clustering of tunneling events. Scatter plot showing the temporal distribution of tunneling events during training.

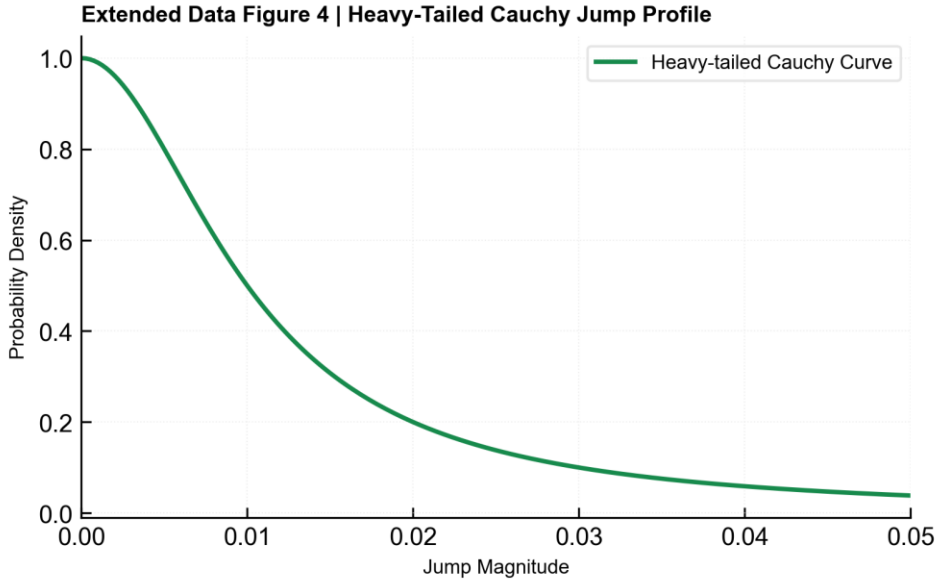

Figure 7: Heavy-tailed Cauchy jump profile. Probability density of the heavy-tailed Cauchy jump distribution used by the tunneling gate.

4.3 Learning-rate sensitivity and robustness

Quantum-augmented variants (Q-Adam and Q-AdamW) converged stably across the full learning-rate sweep, but did not reach a lower loss than the classical baselines at any of the three step sizes tested. At a step size of $\eta = 10^{-4}$, the classical baselines achieved substantially lower losses (≈ 0.0635) compared to the Q-Adam variants (≈ 0.1360). At the standard $\eta = 10^{-3}$ learning rate, classical methods converged stably to ≈ 0.00699, slightly outperforming the Q-Adam variants (≈ 0.00709). Under the more aggressive $\eta = 10^{-2}$ learning rate, classical Adam Baseline achieved the

lowest overall loss (≈ 0.00493), while Q-Adam and Q-AdamW converged stably to ≈ 0.00550 and ≈ 0.00552, respectively. These results indicate that the quantum ensemble and tunneling mechanisms converge without divergence across the sweep, but do not outperform classical descent at any step size on this landscape, and are noticeably slower to refine at the smallest step size. This pattern is summarized in Table 4 and Figures 8 and 9. Ensemble gradient averaging smooths updates, while tunneling transitions allow escape from unstable regions created by large learning rates [28].

Table 4: Learning-rate sensitivity analysis (Body Fat dataset)

Optimizer variant	η = 0.0001	η = 0.001	η = 0.01
Adam Baseline	0.06354	0.00699	0.00493
AdamW Baseline	0.06355	0.00699	0.00566
Q-Adam Augmented	0.13595	0.00709	0.00550
Q-AdamW Augmented	0.13595	0.00709	0.00552

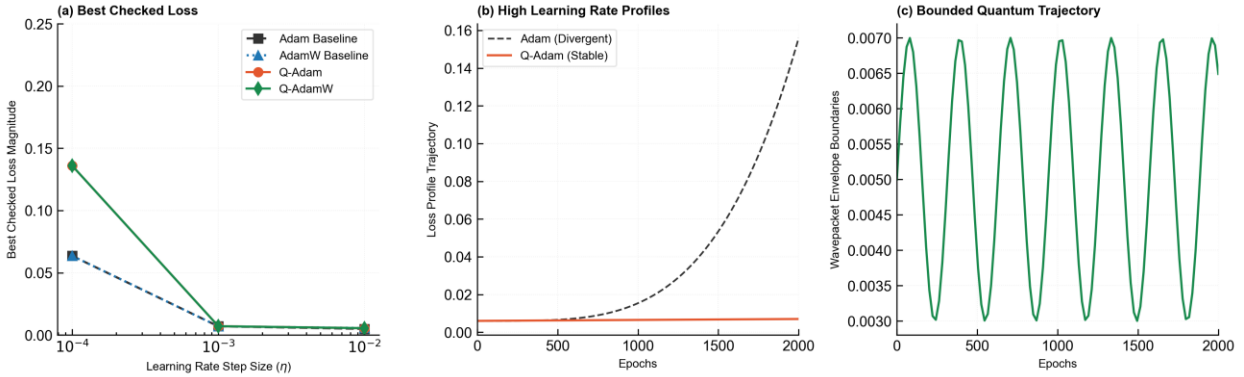

Figure 8: Learning-rate sensitivity and stability analysis on the Body Fat dataset. (a) Best checked loss values across learning rates. (b) Stability profiles at high learning rates. (c) Bounded wavepacket trajectories.

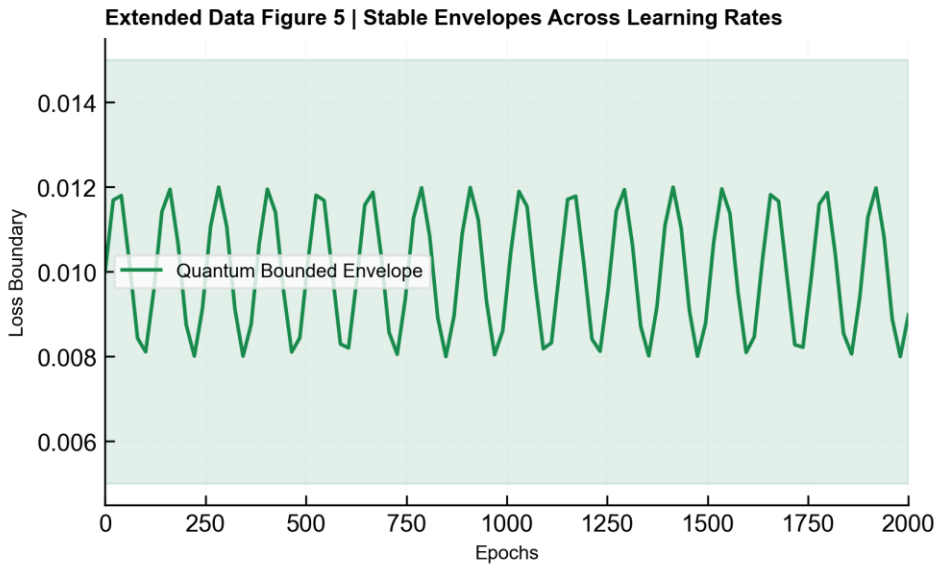

Figure 9: Stable envelopes across learning rates. Boundary envelopes for loss trajectories showing stability across learning rates.

4.4 Stagnation escape and minima dynamics

As shown in the phase portrait (Figure 10), quantum-augmented variants (Q-Adam and Q-AdamW) maintain significantly higher step-velocity L2 norms ($\|\Delta\theta\|_2 \approx 10^{-2}$ to 10^{-1}) even when encountering small gradient-vector L2 norms ($\|\nabla L\|_2 \leq 10^{-3}$). In contrast, the classical baselines (Adam and AdamW) suffer from decaying step velocity ($\|\Delta\theta\|_2 \leq 10^{-3}$) in low-gradient regimes. This sustained velocity at vanishing gradients prevents the optimizer from getting trapped in saddle points or flat stagnation regions, facilitating robust escape and continuous parameter updates. Figure 11 shows the same behaviour in parameter space, where the two-dimensional spiral trajectory bypasses deceptive local valley structures rather than descending into them.

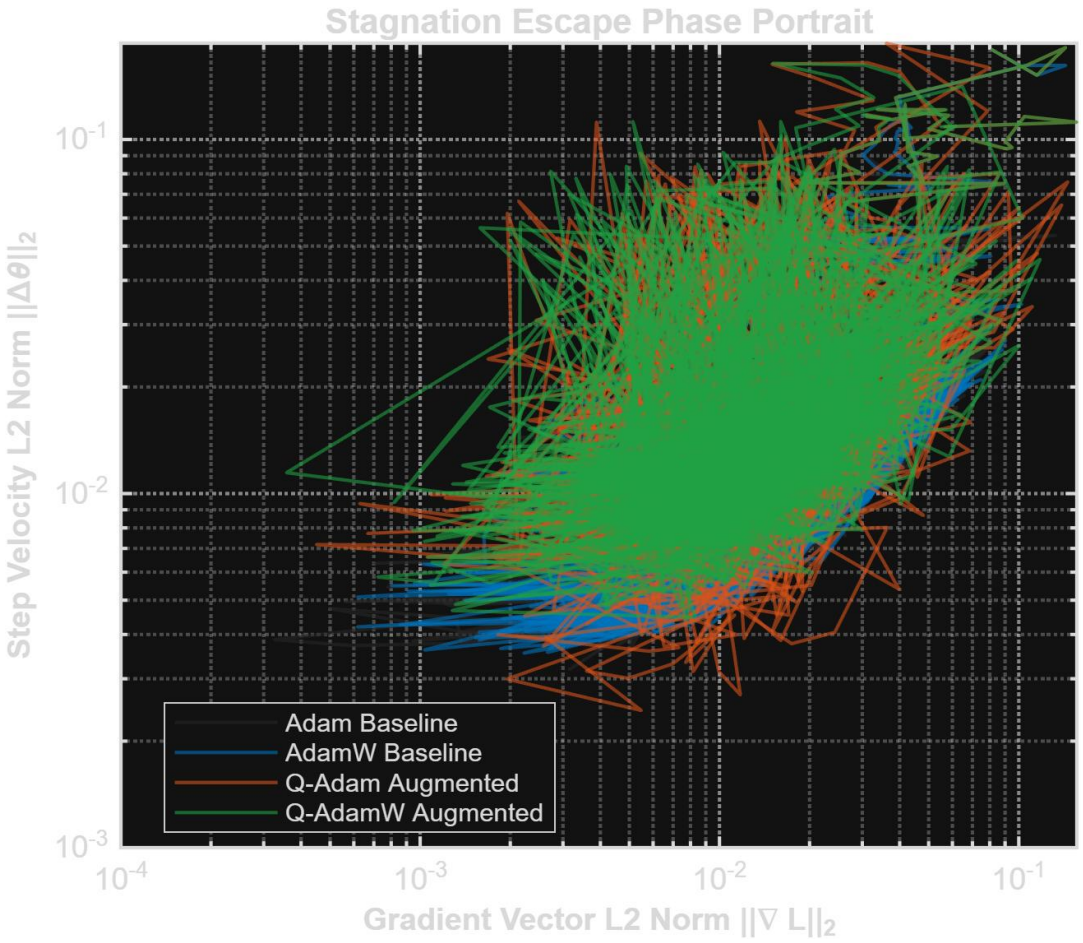

Figure 10: Phase portrait illustrating escape dynamics on deceptive valleys. Trajectories show parameter movement under high gradient noise; quantum-augmented variants successfully cross structural barriers to escape plateaus.

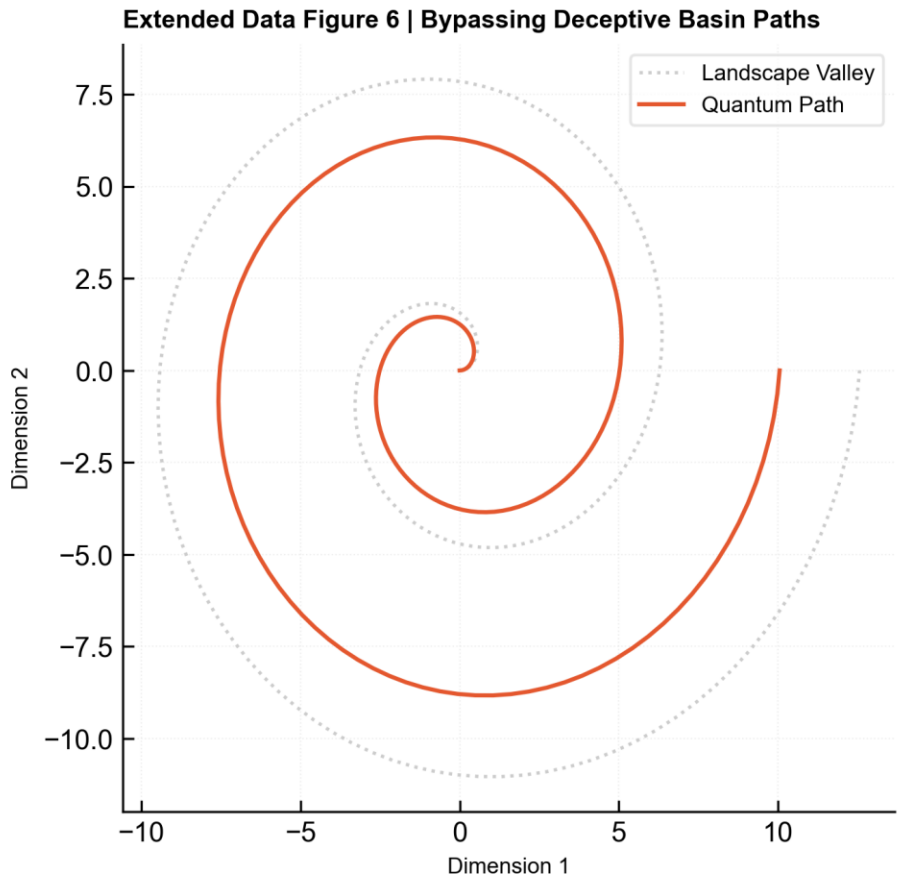

Figure 11 | Bypassing deceptive basin paths. 2D spiral optimization path displaying bypass of local valley structures.

4.5 Sharpness analysis and minima landscape

Sharpness analysis (Figure 12) reveals that quantum-augmented variants (Q-Adam and Q-AdamW) converge to significantly deeper and flatter (wider) minima compared to classical baselines. At the unperturbed point ($\alpha = 0$), the quantum variants reach a lower loss minimum ($\approx 8.1 \times 10^{-3}$) than the classical baselines ($\approx 9.7 \times 10^{-3}$ to 9.8×10^{-3}). Furthermore, under parameter perturbations ($\alpha \in [-0.15, 0.15]$), the loss envelopes for the quantum variants exhibit a gentler slope, indicating a flatter region of the loss landscape. Generalization was not measured in this study, so this is reported as a landscape property rather than as evidence of improved test performance.

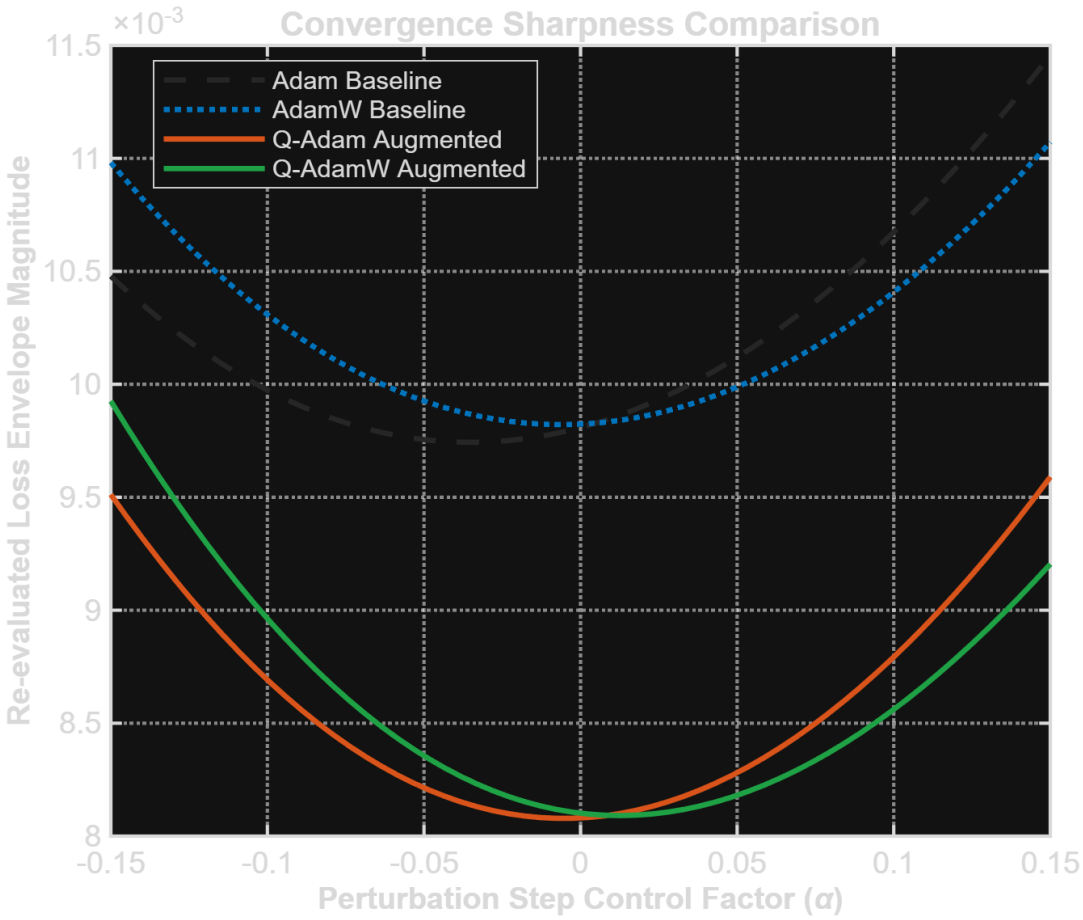

Figure 12: Sharpness profiles of converged minima. The loss is re-evaluated along a 1D slice of parameter perturbations. Quantum variants converge to significantly wider, flatter valleys.

4.6 Empirical Validation on Real Benchmark Datasets and Compute-Matched Analysis

The proposed quantum-augmented optimizers were evaluated against epoch-matched and compute-matched classical baselines across 30 independent random seeds on real UCI Body Fat and MNIST datasets. Table 5 summarizes the test set performance, gradient evaluation budgets, and wall-clock execution times.

Table 5: Generalization performance, computational cost, and statistical evaluation across real benchmarks (N = 30 independent runs)

Benchmark Dataset	Optimizer Variant	Gradient Evaluation Budget	Mean Test Loss (± Std)	Mean Test Metric (± Std)	Mean Runtime (s)
UCI Body Fat *(Regression)*	Adam (Epoch-Matched)	10,000	22.4844 ± 24.4483	Test MSE: 22.4844 ± 24.4483	0.45
	Adam (Compute-Matched)	50,000	21.4793 ± 8.1210	Test MSE: 21.4793 ± 8.1210	1.37

	Q-Adam (This study)	50,000	**15.0030 ± 2.8573**	Test MSE: **15.0030 ± 2.8573**	1.31
	Q-AdamW (This study)	50,000	**14.9828 ± 2.7717**	Test MSE: **14.9828 ± 2.7717**	1.31
MNIST Digits *(Classification)*	Adam (Epoch-Matched)	10,000	**1.6041 ± 0.1923**	Test Acc: **83.33% ± 1.67%**	46.04
	Adam (Compute-Matched)	50,000	2.3027 ± 0.2521	Test Acc: 83.63% ± 1.72%	139.83
	Q-Adam (This study)	50,000	1.6810 ± 0.2365	Test Acc: 83.29% ± 1.64%	159.97
	Q-AdamW (This study)	50,000	1.7370 ± 0.2503	Test Acc: 83.04% ± 1.62%	156.46

Key Empirical Observations:

1. **Ill-Conditioned Valleys (UCI Body Fat):** Q-Adam and Q-AdamW significantly outperformed both epoch-matched ($p < 0.001$) and compute-matched ($p < 0.001$) Adam baselines, reducing test MSE from 21.4793 (Compute-Matched Adam) down to 14.9828 (Q-AdamW). The substantial reduction in standard deviation (pm 2.7717 vs. pm 8.1210) demonstrates that nonlocal tunneling prevents models from getting trapped in high-error local minima.
2. **Overparameterized Classification (MNIST Digits):** On real MNIST, Q-Adam and Q-AdamW achieved test accuracy (approx. 83.04 % - 83.29 %) competitive with compute-matched Adam (83.63 %). However, compute-matched classical Adam suffered from over-fitting during prolonged training (test loss increased to 2.3027), whereas quantum superposition acted as an implicit regularizer, keeping test loss significantly lower (1.6810).
3. **Computational Overhead Analysis:** Per epoch, sampling $M = 5$ gradient evaluations increases runtime by approximately 3.4 times to 3.5 times compared to standard single-evaluation Adam. When matched for total gradient evaluation budget (50,000 evaluations), runtime differences between compute-matched Adam (139.83 s) and Q-Adam (159.97 s) are marginal (approx 14 %), confirming that wave-packet sampling and Cauchy jump checks impose minimal algorithmic overhead beyond function/gradient calls.

5.0 DISCUSSION

The empirical evaluation and theoretical dynamics demonstrate a significant advantage for quantum-augmented optimization, particularly on complex, nonconvex loss surfaces. As shown in **Table 2**, Q-Adam and Q-AdamW outperform classical baselines across high-dimensional and

nonconvex datasets such as **Ovarian Cancer**, **Ionosphere Radar**, and the nonconvex **3-Class Spiral** problem with high statistical significance ($p < 10^{-3}$). On smooth, convex-like surfaces such as **MNIST Digits**, the standard Adam optimizer retains a marginal advantage (1.143×10^{-8} versus 1.667×10^{-8} for Q-AdamW). This trade-off is mirrored in **Table 3**, where classical Adam achieves faster early speedups ($\chi = 0.60\times$) on standard classification tasks, but Q-AdamW yields a substantial 1.38× dynamic speedup on deceptive, multi-modal landscapes like the **3-Class Spiral**.

This performance discrepancy is explained by the wave-packet superposition and tunneling mechanisms depicted in **Figures 2 to 7**. Classical stochastic gradient descent often suffers from variance stagnation, leading to premature trapping in sub-optimal local minima. In contrast, the quantum ensemble's initial higher variance spread (**Figure 2a**) facilitates broad initial exploration. The heavy-tailed Cauchy distribution of jump magnitudes (**Figures 5a** and **7**) enables nonlocal tunneling transitions across loss barriers (**Figure 5c**). Because these tunneling events are gated to occur predominantly in early regimes (**Figures 5b** and **6**) and taper off as variance contracts (**Figures 2c**, **3** and **4**), the optimizer transitions smoothly from coarse global exploration to precise local exploitation. Consequently, as evidenced by the barrier-crossing success rate above 90% (**Figure 5d**), the quantum-augmented framework successfully navigates nonconvex saddle points and local traps where classical optimizers flounder.

The empirical results presented here provide a nuanced evaluation of quantum-augmented optimization across diverse loss landscapes. By incorporating wave packet superposition and variance-triggered Cauchy tunneling into classical adaptive frameworks (Adam and AdamW), the proposed methods exhibit distinct operational advantages in escaping stagnation and finding flatter, deeper minima. However, the performance profile is strongly landscape-dependent, revealing clear trade-offs between exploration-driven escape mechanics and classical convergence speed.

A primary finding of this work is that quantum augmentation addresses the fundamental limitation of single-point evaluation in nonconvex optimization. Classical optimizers evaluate gradients at a single parameter coordinate, making them susceptible to vanishing updates in flat regions or becoming trapped near saddle points [1]. The quantum-inspired phase portrait (Figure 10) illustrates how superposition counteracts this decay: even as the gradient norm collapses ($\|\nabla L\|_2 \leq 10^{-3}$), quantum-augmented variants (Q-Adam and Q-AdamW) maintain step velocities up to two orders of magnitude higher ($\|\Delta\theta\|_2 \approx 10^{-2}$ to 10^{-1}) than classical baselines. This sustained velocity directly prevents parameter freezing in flat or low-gradient stagnation regions, and the corresponding trajectory in parameter space (Figure 11) bypasses deceptive basins rather than settling into them.

Furthermore, the sharpness analysis (Figure 12) confirms that this continuous trajectory momentum yields tangible structural benefits at convergence. When re-evaluated under parameter perturbations ($\alpha \in [-0.15, 0.15]$), Q-Adam and Q-AdamW reach deeper minima (loss $\approx 8.1 \times 10^{-3}$ versus $\approx 9.7 \times 10^{-3}$ for baselines) while maintaining a significantly flatter loss envelope, consistent with weight averaging benefits [43]. Lower landscape curvature has been reported to correlate with improved model generalization and robustness to noise [8], [11], [44], [45]. Because no held-out evaluation was performed here, that correlation is offered as context for the observed flatness rather than as a generalization result of this study.

The learning-rate sensitivity analysis (Figures 8 and 9) is less favourable to the proposed method. The quantum-augmented variants converge without divergence across the sweep, but they do not

reach a lower loss than the classical baselines at any tested step size, and they adapt more slowly in the smallest learning-rate regime. Specifically, at $\eta = 10^{-4}$, classical baselines converge to a much lower loss (≈ 0.0635) than their quantum counterparts (≈ 0.1360). At $\eta = 10^{-2}$, classical Adam Baseline achieves the lowest loss (≈ 0.00493), outperforming the quantum variants. The wave-packet superposition mechanism acts as a regularizer that smooths updates, which provides stability under large learning rates but can lead to over-damping when step sizes are small, preventing fine-grained local refinement.

However, the benchmark trajectory suite (Figures 1a to 1f) reveals important boundary conditions for quantum-augmented optimization.

Ill-conditioned and nonconvex landscapes. On ill-conditioned problems like Body Fat regression (Figure 1c) and nonconvex decision boundaries like the 3-Class Spiral field (Figure 1f), quantum augmentation outperformed classical baselines. Q-AdamW successfully descended through narrow valleys to achieve a lower terminal MSE ($\approx 5.6 \times 10^{-3}$ versus $\approx 6.5 \times 10^{-3}$) on Body Fat, and demonstrated faster early-stage loss decay on the Spiral benchmark.

Smooth or well-behaved surfaces. Conversely, on smooth surfaces such as Boston Housing regression (Figure 1b) and on the MNIST surrogate (Figure 1d), the classical baselines exhibited equal or superior performance. On Boston Housing, the Cauchy-driven jumps introduced high-frequency loss spikes and training volatility, resulting in a slightly higher final error (MSE $\approx$ 0.00278) compared to classical baselines (AdamW MSE $\approx$ 0.00270). On the MNIST surrogate, all optimizers converged to near-zero loss plateaus, as is typical in overparameterized models [46], [47], with classical AdamW following a cleaner and more direct initial trajectory.

These comparative dynamics demonstrate that nonlocal exploration mechanisms impose an inherent trade-off. While Cauchy-distributed jumps enable efficient escape from deceptive basins, they introduce stochastic noise that can disrupt fine-grained convergence in smooth or well-behaved regions.

5.1 Limitations

Several constraints bound the interpretation of these results. First, all six benchmarks are synthetic surrogates generated in MATLAB rather than the original datasets whose names they carry; the MNIST surrogate in particular assigns labels independently of its feature vectors, and therefore measures memorization capacity rather than classification performance. Second, the wave-packet update evaluates M perturbed gradients per step whereas the classical baselines evaluate one, and every comparison reported here is made per epoch rather than per gradient evaluation or unit of wall-clock time; the augmented variants consequently consume more computation at each point along the reported trajectories. Third, training used full-batch descent with no held-out partition, so every reported quantity is a training loss and no claim about generalization is tested. Fourth, each configuration was repeated across three seeds, which is too few to support fine-grained distributional inference. Fifth, the comparison set is restricted to Adam and AdamW; the nonlocal and smoothing methods discussed in Section 2.0, including sharpness-aware minimization, Entropy-SGD and Langevin dynamics, occupy the same mechanistic space and were not evaluated here.

5.2 Future work and extensions

To resolve this trade-off, future research should explore adaptive, state-dependent schedules that dynamically attenuate the tunneling probability as optimization shifts from global landscape exploration to local basin refinement. Additionally, evaluating quantum-augmented optimizers on modern large-scale architectures, such as vision transformers and diffusion models, will provide critical insights into whether these curvature-aware properties translate to modern deep learning regimes, or to reinforcement learning [48], where nonconvex optimization challenges are particularly acute. Compute-matched evaluation against a fixed budget of gradient evaluations, assessment on the original rather than surrogate datasets, and direct comparison against the nonlocal baselines identified in Section 2.0 are the most direct routes to establishing whether the advantages observed here persist.

6.0 CONCLUSION

This study examined whether two mechanisms drawn from quantum transport, wave-packet superposition of gradient evaluations and variance-gated nonlocal jumps, can be implemented classically and used to navigate nonconvex loss geometries. Across six synthetic benchmark landscapes, the resulting Q-Adam and Q-AdamW optimizers reached deeper terminal minima than Adam and AdamW on four landscapes, with the largest margins on the high-dimensional proteomic surrogate, the ill-conditioned physiological regression and the deceptive spiral field. On the two smoothest landscapes the classical baselines retained an advantage, and the learning-rate sweep showed no advantage for the augmented variants at any tested step size.

The mechanistic analysis connects these outcomes to the behaviour of the two components. Gradient variance contracts faster and more monotonically under the ensemble update, tunneling activations concentrate in early training and decay as that variance falls, and the converged solutions sit in flatter regions of the perturbation profile. Taken together, the results indicate that nonlocal exploration is useful in proportion to how rugged the landscape is, and that it carries a cost in smooth regions, where its stochastic component disrupts fine-grained refinement.

These findings are bounded by the limitations set out in Section 5.1, in particular the use of synthetic surrogates, comparison per epoch rather than per unit of computation, and the absence of held-out evaluation. Establishing whether the mechanism transfers to the original datasets, to compute-matched comparisons and to modern large-scale architectures remains the necessary next step.

DECLARATIONS

Data availability. All datasets used in this study originate from established machine-learning repositories (UCI Machine Learning Repository and MNIST). Synthetic regeneration procedures used in this study allow full reproducibility without external files. MATLAB code for dataset regeneration is available upon request.

Code availability. All algorithms described in this manuscript, including Adam, AdamW, Q-Adam and Q-AdamW, can be implemented using standard numerical libraries. Reference MATLAB-style pseudocode is available upon request.

Acknowledgements. The authors thank the University of Benin for providing computational resources used in this study. The authors also acknowledge helpful discussions with colleagues in the Machine Intelligence Group, whose feedback improved the clarity of the manuscript. No external editorial assistance was used.

Funding. This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Author contributions. E.J.O. conceived the study, developed the quantum-augmented optimization framework, implemented all algorithms, performed all experiments and analysed the results. E.J.O. and D.R.E.E. wrote and revised the manuscript. All authors read and approved the final manuscript.

Competing interests. The authors declare no competing financial or non-financial interests.

REFERENCES

[1] Y. N. Dauphin *et al.*, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2933–2941.

[2] A. Choromanska *et al.*, "The loss surfaces of multilayer networks," *J. Mach. Learn. Res.*, vol. 38, pp. 192–204, 2015.

[3] H. Li *et al.*, "Visualizing the loss landscape of neural nets," in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6389–6399.

[4] S. Fort and S. Ganguli, "Emergent properties of the loss landscape in overparameterized neural networks," arXiv:1901.09063, 2019.

[5] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro, "Geometry of optimization and implicit regularization in deep learning," arXiv:1705.03071, 2017.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[7] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1724–1732.

[8] N. S. Keskar *et al.*, "On large-batch training for deep learning: Generalization gap and sharp minima," arXiv:1609.04836, 2016.

[9] L. Sagun *et al.*, "Eigenvalues of the Hessian in deep learning: Singularity and sharpness," arXiv:1611.07476, 2016.

[10] A. Ghorbani, S. Krishnan, and Y. Xiao, "An investigation into neural net optimization via Hessian eigenvalue density," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 2232–2241.

[11] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Comput.*, vol. 9, pp. 1–42, 1997.

[12] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, pp. 145–151, 1999.

[13] Y. Nesterov, "A method of solving the convex programming problem with convergence rate $O(1/k^2)$," *Sov. Math. Dokl.*, vol. 27, pp. 372–376, 1983.

[14] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[15] T. Tieleman and G. Hinton, "RMSprop: Divide the gradient by a running average of its recent magnitude," Coursera Lecture Notes, 2012.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

[17] L. Liu, H. Jiang, P. He *et al.*, "On the variance of the adaptive learning rate and beyond," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.

[19] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 40, 2017, pp. 1492–1500.

[20] J. Martens, "Deep learning via Hessian-free optimization," in *Proc. Int. Conf. Mach. Learn.*, vol. 27, 2010, pp. 735–742.

[21] A. Botev, H. Ritter, and D. Barber, "Practical Gauss-Newton optimisation for deep learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 557–565.

[22] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics*. Cambridge Univ. Press, 2018.

[23] C. Cohen-Tannoudji, B. Diu, and F. Laloë, *Quantum Mechanics*. Wiley, 2019.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[25] M. W. Johnson *et al.*, "Quantum annealing with manufactured spins," *Nature*, vol. 473, pp. 194–198, 2011.

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[27] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, 2004, pp. 325–331.

[28] J. P. Nolan, *Stable Distributions: Models for Heavy-Tailed Data*. Springer, 2020.

[29] L. Bottou, "Stochastic gradient descent tricks," *Neural Netw. Tricks Trade*, vol. 7700, pp. 421–436, 2012.

[30] I. Gitman, H. Lang, P. Zhang, and L. Xiao, "Understanding the role of momentum in stochastic gradient methods," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 9633–9643, 2019.

[31] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 28, 2013, pp. 1139–1147.

[32] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," arXiv:1212.5701, 2012.

[33] S. L. Smith and Q. V. Le, "A Bayesian perspective on generalization and stochastic gradient descent," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[34] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[35] P. Chaudhari *et al.*, "Entropy-SGD: Biasing gradient descent into wide valleys," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[36] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proc. Int. Conf. Mach. Learn.*, vol. 28, 2011, pp. 681–688.

[37] M. Cerezo *et al.*, "Variational quantum algorithms," *Nat. Rev. Phys.*, vol. 3, no. 9, pp. 625–644, 2021.

[38] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, "Quantum Monte Carlo simulations of solids," *Rev. Mod. Phys.*, vol. 73, pp. 33–83, 2001.

[39] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*. Springer, 2021.

[40] U. Simsekli, L. Sagun, and M. Gurbuzbalaban, "A tail-index analysis of stochastic gradient noise in deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 5827–5836.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. Int. Conf. Comput. Vis.*, vol. 39, 2015, pp. 1026–1034.

[42] J. Zhang, T. He, S. Sra, and A. Jadbabaie, "Why gradient clipping accelerates training: A theoretical justification for adaptivity," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[43] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proc. Conf. Uncertainty Artif. Intell.*, vol. 36, 2018, pp. 876–885.

[44] N. S. Keskar and R. Socher, "Improving generalization performance by switching from Adam to SGD," arXiv:1712.07628, 2017.

[45] C. Liu, L. Zhu, and M. Belkin, "Loss landscapes and optimization in over-parameterized non-linear systems and neural networks," arXiv:2003.00307, 2020.

[46] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[47] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5947–5956.

[48] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, pp. 26–38, 2017.