Coherent Quantum Learning: Training via Coherent Evolution

Ignacio B. Acedo[1] Javier Gonzalez-Conde[1,2,4] Pablo Rodriguez-Grasa[2,3] Barry C. Sanders[5] Lirandë Pira[6]

[1]Quantum Mads, Calle Larrauri 1, Edificio A, piso 3, puerta 28, 48160 Derio, Spain
[2]Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain
[3]TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain
[4]EHU Quantum Center, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain
[5]Institute for Quantum Science and Technology, University of Calgary, Calgary, Alberta T3A 0E1, Canada
[6]Centre for Quantum Technologies, National University of Singapore, Singapore

Training is the mechanism by which learning systems extract structure from data [1, 2] and is typically formulated as an optimization process driven by gradients and iterative parameter updates [3]. In quantum machine learning [4, 5], this paradigm is usually retained: parameterized quantum circuits are evaluated at fixed parameter settings, gradients are estimated from measurement outcomes via techniques such as the parameter-shift rule [6–8], and optimization proceeds through a classical feedback loop [9, 10]. As a result, learning is effectively projected onto a classical process, and quantum coherence plays no active role in the optimization dynamics. This hybrid approach faces fundamental limitations: sampling fluctuations propagate into parameter updates, expressive models may exhibit barren plateaus where gradients vanish exponentially with system size [11, 12], and distinct parameter configurations are explored incoherently, forfeiting any advantage that quantum interference could provide in parameter space.

A natural question is whether classical training paradigms can be ported to quantum models. Recent work shows this is fundamentally obstructed: backpropagation-like gradient scaling is impossible without access to multiple copies of a quantum state, since measurement collapse prevents reuse of intermediate information [13, 14]. Even structured circuit classes that partially recover favorable gradient scaling [15] do not resolve the deeper issue. Other approaches have explored coherent parameter-updates and gradient-free methods for quantum optimization [16, 17]. These efforts suggest that quantum models may require training principles that are native to quantum mechanics, rather than adapted from classical algorithms. More generally, recent ideas aim to move beyond hybrid quantum–classical optimization by embedding learning directly into quantum dynamics [18–20]. A principled route in this direction is provided by *Quantum Hamiltonian Descent* [18], which recasts optimization as Schrödinger evolution over parameter space. Quantizing the dynamics induced by the Bregman–Lagrangian framework [21] via the path integral allows this method to replace a single classical trajectory with a wavefunction that naturally evolves toward low-loss configurations, exploiting superposition, interference, and tunneling to navigate non-convex landscapes. Recent results further indicate that such Hamiltonian-based approaches can offer improved complexity guarantees in certain regimes [22].

In this work, we introduce *Coherent Quantum Learning* (CQL), a framework that specializes these ideas to the setting where the objective function is generated by a parameterized quantum model acting on data. While quantum Hamiltonian descent is formulated as a general optimization framework, CQL addresses learning systems in which the optimization landscape arises from the interaction between model, dataset, and measurement process. In CQL, model parameters are promoted to quantum degrees of freedom and evolved under a Hamiltonian that encodes the cost function. Parameter configurations are explored in superposition and updated through unitary evolution, allowing interference in parameter space to shape learning. This eliminates the need for explicit gradient computation and classical feedback, replacing backpropagation-based optimization with coherent quantum evolution. Consequently, training corresponds to a sequence of data-dependent coherent transformations, allowing the optimization dynamics to be interpreted as a learning process rather than a search over a fixed function.

a. Constructing Coherent Quantum Learning: Model and Procedure. As a minimal illustration, consider a reduced version of the data re-uploading model [23] consisting of a single-parameter learning model for binary classification. Given input $x \in \mathbb{R}$ with label $y \in \{0, 1\}$, the model prepares the one-qubit state

$$|\psi(x,\theta)\rangle = R_y(\theta)\, U(x)|0\rangle = \alpha_0(x,\theta)|0\rangle + \alpha_1(x,\theta)|1\rangle, \tag{1}$$

where $U(x)$ is a data-encoding unitary (*e.g.* $R_x(x)$) and $R_y(\theta) = e^{-i\theta Y/2}$ is a trainable rotation. The loss $\mathcal{L}_y(x,\theta) = 1 - |\alpha_y(x,\theta)|^2$ penalizes incorrect classification outcomes. To enable coherent training, the parameter is discretized using m qubits, $\theta = 2\pi \sum_{j=1}^{m} \theta_j 2^{-j}$ with $\theta_j \in \{0, 1\}$, yielding a grid $\mathcal{G}_m$ of 2^m values in $[0, 2\pi)$, each represented by a computational basis state $|\theta\rangle = |\theta_1\theta_2 \ldots \theta_m\rangle$ on m ancillary qubits. By adding these ancillas, the model can be evaluated coherently over all parameter values simultaneously through a unitary $U_{x,\theta}$ that entangles model outputs with parameter configurations:

$$U_{x,\theta}\big(|0\rangle|0\rangle^{\otimes m}\big) = \sum_{\theta \in \mathcal{G}_m} \big[\alpha_0(x,\theta)|0\rangle|\theta\rangle + \alpha_1(x,\theta)|1\rangle|\theta\rangle\big]. \tag{2}$$

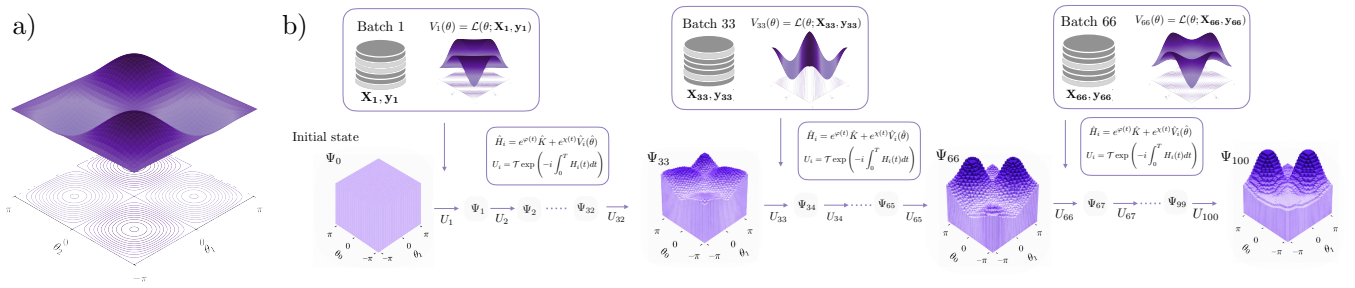


Figure 1. Numerical illustration of batch-based Coherent Quantum Learning (CQL). a) Loss landscape $\mathcal{L}(\theta; \mathbf{X}, \mathbf{y})$ evaluated over the full training dataset as a function of the two variational parameters θ_0, θ_1, with contours indicating the structure of the global objective. b) Evolution of the probability distribution of the parameter state $\hat{\theta}$ across sequential batched coherent training. Starting from a uniform superposition Ψ_0 over parameter space, the state is evolved under the CQL Hamiltonian, where the loss potential is constructed from individual training samples (batch size 1). The loss landscape associated with select batches is shown alongside the corresponding parameter distributions, and the final state Ψ_{100} after processing all 100 batches. As training progresses, the probability amplitude increasingly concentrates in regions of low loss, and after completing the full epoch, the final distribution anti-correlates with the global loss landscape shown in a), confirming that the coherent evolution has effectively learned an inverse representation of the objective function.

This operation entangles model outputs with parameter configurations on an $(m+1)$-qubit state and serves as the basic primitive enabling coherent exploration of the loss landscape. More than one data point can be evaluated simultaneously by adding one ancilla per datum, increasing the batch size; the cost function amplitude is then given by the bit-string of the corresponding labels.

The CQL dynamics are generated by the Hamiltonian

$$H_{\mathrm{CQL}}(t) = e^{\varphi(t)}\left(-\tfrac{1}{2}\hat{P}_\theta^2\right) + e^{\chi(t)}\, H_{\mathcal{L}}(x, \hat{\theta}), \tag{3}$$

where $e^{\varphi(t)}$, $e^{\chi(t)}$ are damping parameters, $\hat{P}_\theta^2$ is the discrete Laplacian in parameter space, and $H_{\mathcal{L}} = \sum_\theta \mathcal{L}(x,\theta)|\theta\rangle\langle\theta|$ is a diagonal Hamiltonian encoding the loss. The kinetic term promotes diffusion across parameter space, while the loss Hamiltonian biases the dynamics toward low-loss regions. In the absence of prior information, the parameter register is initialized in a uniform superposition, ensuring unbiased exploration of all configurations; if prior knowledge is available, a localized wavepacket centered around an informed initial guess can bias the evolution toward promising regions.

The construction of the loss Hamiltonian exploits the coherent model-evaluation unitary $U_{x,\theta}$. Using the amplitude-to-diagonal conversion of Ref. [24], query access to a controlled version of $U_{x,\theta}$ yields a block encoding of the diagonal loss operator with $\mathcal{O}(m)$ circuit depth and $\mathcal{O}(1)$ queries to the controlled model unitary. The time dependence of the CQL Hamiltonian poses an additional challenge, since simulating time-dependent Hamiltonians is not a native operation in the circuit model and typically introduces considerable overhead [25]. To address this, we employ the clock-dimension construction of Refs. [26, 27], which introduces an additional Hilbert space $\mathcal{H}_t$ with conjugate operators $\hat{X}_t$ and $\hat{P}_t$. This converts the non-autonomous evolution into a time-independent Hamiltonian on the enlarged space $\mathcal{H}_t \otimes \mathcal{H}_\theta$, where the scalar functions $e^{\varphi(t)}$ and $e^{\chi(t)}$ become diagonal operators in the clock coordinate. Once the full block encoding is assembled, using standard techniques including quantum signal processing [28] for polynomial transformations of operators, qubitization [29] implements the coherent training evolution. This formulation is naturally compatible with fault-tolerant architectures [30] where long coherent evolutions can be sustained.

Although one could in principle construct a batch containing the entire dataset, corresponding to coherent global training, this would require as many ancillas as data points and becomes impractical for large datasets. We therefore introduce an iterative batched training strategy, analogous to mini-batch methods in classical machine learning, whereby the dataset is partitioned into smaller subsets processed sequentially. The system evolves under the CQL Hamiltonian using the first batch, and the resulting parameter state is then used as the initial condition for a subsequent evolution encoding the next batch. This process is repeated until all batches have been processed, with the batch size introducing a trade-off between spatial resources (qubit count) and temporal resources (circuit depth). This batched regime fundamentally changes the role of the damping schedules. Whereas in global optimization one drives the wavefunction to concentrate strongly around minima, aggressive concentration in the batched setting would overfit to the current batch and erase information from previous ones. This phenomenon is directly analogous to stochastic gradient methods, where update magnitude is regulated through a learning rate. In CQL, the analogous control is the effective evolution time together with the damping schedules, which govern the balance between adaptation to the current batch and retention of previously acquired information.

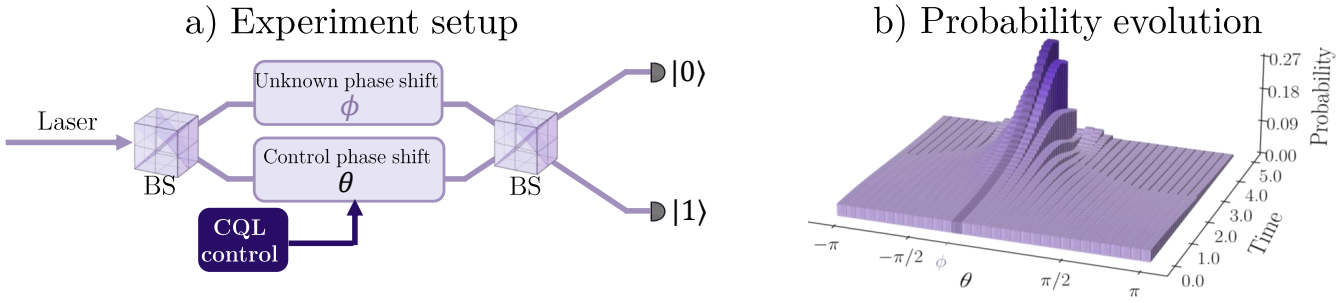


Figure 2. a): Schematic of the interferometric phase-estimation setup. A single photon propagates through a balanced Mach–Zehnder interferometer containing an unknown phase shift ϕ and a controllable phase shift θ, which is iteratively updated by the CQL protocol. b): Time evolution of the probability distribution over the discretized parameter register, $|\Psi(\theta, t)|^2$. Starting from a uniform superposition, the coherent dynamics progressively localize the probability amplitude around the global minimum of the loss landscape, yielding an accurate estimate of the unknown phase ϕ.

b. Quantum classification with a data re-uploading model. We first illustrate CQL on a binary classification task in which the objective is to distinguish points inside from outside a one-dimensional interval. The model is a single-qubit data re-uploading circuit with two layers of trainable rotations, and each parameter is discretized using five qubits over a grid of 2^5 points in $(-\pi, \pi)$. Figure 1 shows the evolution of the probability distribution over parameter space during one epoch of single-sample coherent training. Starting from a uniform superposition, the distribution progressively adapts to the structure of the cost function, developing pronounced peaks at parameter values associated with minima of the loss landscape. After processing all training samples, the final distribution anti-correlates with the global loss landscape: low-loss regions appear as high-probability components of the parameter state. Rather than outputting a single optimized parameter value, CQL generates a quantum representation of the loss landscape itself, where probability amplitudes encode the learned information about high-performing parameter regions directly in the final state.

c. Interferometric phase learning. As a second demonstration, we apply CQL to a physics-motivated parameter estimation task by learning an unknown phase shift in a balanced Mach–Zehnder interferometer, illustrated in Fig. 2a). A single photon is split into two paths containing an unknown phase ϕ and a controllable phase θ, respectively. The output interference depends only on the relative phase between the two arms, allowing the estimation problem to be formulated as the optimization of the interferometric loss $\mathcal{L}(\theta, \phi) = \sin^2[(\theta - \phi)/2]$, whose minimum is attained at $\theta = \phi$. The controllable phase is encoded in a five-qubit parameter register and is initialized in a uniform superposition. Figure 2b) shows the coherent evolution of the probability distribution throughout the optimization process. As the dynamics evolve, the probability amplitude progressively localizes around the global minimum of the interferometric loss landscape, accurately identifying the unknown phase. This result illustrates that CQL naturally extends beyond quantum machine learning, enabling coherent optimization of physical parameters without iterative classical optimization.

d. Summary and outlook. CQL defines a new general computational primitive for quantum learning in which superposition and interference become active resources that shape the training dynamics. The protocol applies to arbitrary parameterized quantum circuits, accommodates batched training, and is compatible with fault-tolerant quantum computing where long coherent evolutions can be leveraged for optimization. The computational complexity of CQL depends on the cost of Hamiltonian simulation and the construction of the loss Hamiltonian; the framework naturally inherits advances in block encoding and Hamiltonian simulation techniques. Several directions remain open, including rigorous complexity analyses, improved Hamiltonian constructions and discretization strategies, extensions to fault-tolerant quantum learning architectures [31, 32], integration with error mitigation techniques [33], and determining whether coherent learning can provide asymptotic advantages over hybrid optimization.

More broadly, this work suggests that quantum models may benefit from training principles native to quantum mechanics, where learning itself is treated as a physical process governed by quantum dynamics, rather than adaptations of classical algorithms that implicitly impose a classical structure on the optimization process.

Code availability. Code and data for the numerical results are available at `https://github.com/nacedob/Coherent-Quantum-Learning`.

[1] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4 (Springer, 2006).

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

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature **323**, 533 (1986).

[4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).

[5] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (Springer, 2018).

[6] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Physical Review A **98** (2018), ISSN 2469-9934, URL http://dx.doi.org/10.1103/PhysRevA.98.032309.

[7] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Physical Review A **99** (2019), ISSN 2469-9934, URL http://dx.doi.org/10.1103/PhysRevA.99.032331.

[8] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, Quantum **6**, 677 (2022), ISSN 2521-327X, URL http://dx.doi.org/10.22331/q-2022-03-30-677.

[9] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Physical Review A **101** (2020), ISSN 2469-9934, URL http://dx.doi.org/10.1103/PhysRevA.101.032308.

[10] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al., Nature Reviews Physics **3**, 625–644 (2021), ISSN 2522-5820, URL http://dx.doi.org/10.1038/s42254-021-00348-9.

[11] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nature Communications **9** (2018), ISSN 2041-1723, URL http://dx.doi.org/10.1038/s41467-018-07090-4.

[12] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, Nature Reviews Physics **7**, 174–189 (2025), ISSN 2522-5820, URL http://dx.doi.org/10.1038/s42254-025-00813-9.

[13] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Quantum Science and Technology **4**, 043001 (2019), ISSN 2058-9565, URL http://dx.doi.org/10.1088/2058-9565/ab4eb5.

[14] A. Abbas, R. King, H.-Y. Huang, W. J. Huggins, R. Movassagh, D. Gilboa, and J. McClean, in *Advances in Neural Information Processing Systems*, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Curran Associates, Inc., 2023), vol. 36, pp. 44792–44819, URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3caae2f725c8e2a55ecd600563d172-Paper-Conference.pdf.

[15] J. Bowles, D. Wierichs, and C.-Y. Park, Quantum **9**, 1873 (2025), ISSN 2521-327X, URL https://doi.org/10.22331/q-2025-10-02-1873.

[16] G. Verdon, J. Pye, and M. Broughton, *A universal training algorithm for quantum deep learning* (2018), 1806.09729, URL https://arxiv.org/abs/1806.09729.

[17] Q. Ye, S. Geng, Z. Han, W. Li, L. M. Duan, and D.-L. Deng, *Quantum automated learning with provable and explainable trainability* (2025), 2502.05264, URL https://arxiv.org/abs/2502.05264.

[18] J. Leng, E. Hickman, J. Li, and X. Wu, *Quantum Hamiltonian Descent* (2023), 2303.01471, URL https://arxiv.org/abs/2303.01471.

[19] J. Leng, Y. Zheng, Z. Jia, L. Fan, C. Zhao, Y. Peng, and X. Wu, *Quantum Hamiltonian Descent for Non-smooth Optimization* (2025), 2503.15878, URL https://arxiv.org/abs/2503.15878.

[20] Z. Chen, Y. Lu, H. Wang, Y. Liu, and T. Li, Communications in Mathematical Physics **406** (2025), ISSN 1432-0916, URL http://dx.doi.org/10.1007/s00220-025-05234-4.

[21] A. Wibisono, A. C. Wilson, and M. I. Jordan, Proceedings of the National Academy of Sciences **113** (2016), ISSN 1091-6490, URL http://dx.doi.org/10.1073/pnas.1614734113.

[22] A. B. Catli, S. Simon, and N. Wiebe, *Exponentially better bounds for quantum optimization via dynamical simulation* (2025), 2502.04285, URL https://arxiv.org/abs/2502.04285.

[23] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Quantum **4**, 226 (2020), ISSN 2521-327X, URL http://dx.doi.org/10.22331/q-2020-02-06-226.

[24] A. G. Rattew and P. Rebentrost, *Non-linear transformations of quantum amplitudes: Exponential improvement, generalization, and applications* (2023), 2309.09839, URL https://doi.org/10.48550/arXiv.2309.09839.

[25] J. Gonzalez-Conde, Z. Morrell, M. Vuffray, T. Albash, and C. Coffrin, Phys. Rev. A **111**, 062606 (2025), URL https://link.aps.org/doi/10.1103/PhysRevA.111.062606.

[26] Y. Cao, S. Jin, and N. Liu, Journal of Physics A: Mathematical and Theoretical **58**, 155304 (2025), URL https://dx.doi.org/10.1088/1751-8121/adb3fe.

[27] Y. Cao, S. Jin, and N. Liu, Phys. Rev. Res. **7**, 043186 (2025), URL https://link.aps.org/doi/10.1103/fkh5-b669.

[28] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (ACM, 2019), STOC '19, p. 193–204, URL http://dx.doi.org/10.1145/3313276.3316366.

[29] G. H. Low and I. L. Chuang, Quantum **3**, 163 (2019), ISSN 2521-327X, URL https://doi.org/10.22331/q-2019-07-12-163.

[30] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[31] P. Ivashkov, P.-W. Huang, K. Koor, L. Pira, and P. Rebentrost, *Qkan: Quantum kolmogorov-arnold networks* (2024), 2410.04435, URL https://arxiv.org/abs/2410.04435.

[32] A. G. Rattew, P.-W. Huang, N. Guo, L. Pira, and P. Rebentrost, *Accelerating inference for multilayer neural networks with quantum computers* (2025), 2510.07195, URL https://arxiv.org/abs/2510.07195.

[33] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, Reviews of Modern Physics **95** (2023), ISSN 1539-0756, URL http://dx.doi.org/10.1103/RevModPhys.95.045005.

Coherent Quantum Learning: Training via Coherent Evolution

Ignacio B. Acedo,[1,*] Javier Gonzalez-Conde,[1,2,3] Pablo Rodriguez-Grasa,[2,4] Barry C. Sanders,[5] and Lirandë Pira[6,†]

[1] *Quantum Mads, Calle Larrauri 1, Edificio A, piso 3, puerta 28, 48160 Derio, Spain*
[2] *Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain*
[3] *EHU Quantum Center, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain*
[4] *TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain*
[5] *Institute for Quantum Science and Technology, University of Calgary, Calgary, Alberta T3A 0E1, Canada*
[6] *Centre for Quantum Technologies, National University of Singapore, Singapore*

(Dated: July 8, 2026)

We introduce coherent quantum learning, a training framework for parameterized quantum models in which the model parameters are quantum degrees of freedom evolved under a Hamiltonian that encodes the loss function. Current quantum machine learning retains classical optimization: parameters are updated by a classical outer loop using gradient estimates from measurements, and quantum coherence has no role in the training dynamics, just as in any classical treatment of the same problem. In the quantum case, a parameter register initialized in superposition evolves unitarily, and probability amplitude concentrates near low-loss configurations through interference, without gradient computation or classical feedback. We give an explicit construction using block encodings and Hamiltonian simulation, applicable to arbitrary parameterized circuits. Numerical experiments on binary classification and interferometric phase estimation confirm that the evolved distribution peaks at the optimal parameters, matching gradient-based performance. The construction is suited to fault-tolerant hardware and extends to batched training via sequential Hamiltonian evolution.

Introduction.— Training is the process by which learning systems discover patterns and acquire structure from data [1, 2]. In classical machine learning, this is formulated as an optimization task, typically solved via discrete, iterative updates acting on parameter values [3–6]. From a physical point of view, learning can be interpreted as a dynamical process in which these updates trace out an evolving trajectory over time. Quantum computing introduces computational models in which information is encoded in quantum states and evolves according to quantum dynamics [7–9]. Quantum machine learning extends learning frameworks to quantum computational settings [10, 11]. Common approaches in quantum machine learning involve parameterized quantum systems trained to perform learning tasks, with information encoded in quantum states and processed through quantum operations and measurement. Most existing approaches to quantum learning, however, adopt a hybrid formulation in which parameter updates are delegated to a classical outer loop [12–15]. Quantum models are evaluated at specific parameter settings, and gradients are estimated from measurement outcomes via techniques such as the parameter-shift rule [16–18], with optimization proceeding through classical feedback. This approach faces fundamental scalability barriers: statistical fluctuations from sampling propagate into parameter updates, and expressive models may exhibit barren plateaus, where gradients vanish exponentially with system size [19, 20]. More fundamentally, distinct parameter configurations are explored incoherently, and any advantage arising from quantum interference and superposition in parameter space is potentially lost.

A natural question is whether classical training paradigms such as backpropagation can be ported to quantum models [21]. Recent work shows this is fundamentally obstructed: backpropagation-like gradient scaling is impossible without access to multiple copies of a quantum state, since measurement collapse prevents reuse of intermediate information [22, 23]. Even structured circuit classes that partially recover favorable gradient scaling [24] do not resolve the deeper issue. Recent work has also explored coherent parameter-updates and gradient-free approaches to quantum learning and optimization [25, 26]. These efforts suggest that quantum models may require training principles that are native to quantum mechanics, rather than adapted from classical algorithms.

More generally, recent ideas aim to move beyond hybrid quantum–classical optimization by embedding learning directly into quantum dynamics [27–29]. A principled route in this direction is provided by *Quantum Hamiltonian Descent* [27, 28], which recasts optimization as Schrödinger evolution over parameter space. Quantizing the dynamics induced by the Bregman–Lagrangian framework [30, 31] via the path integral allows this method to replace a single classical trajectory with a wavefunction that naturally evolves toward low-loss configurations, exploiting superposition, interference, and tunneling to navigate non-convex landscapes. Recent results further indicate that such Hamiltonian-based approaches can offer improved complexity guarantees in certain regimes [32]. This suggests that training quantum learning models can be realized as a coherent transformation, for example

* ibenito@quantum-mads.com
† lpira@nus.edu.sg

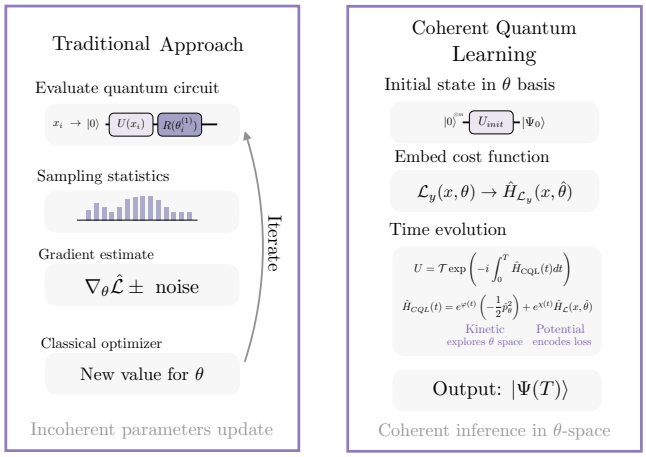


FIG. 1. Conceptual comparison of training in traditional approaches and Coherent Quantum Learning (CQL). The circuit illustrates one coherent training step, where an ancilla register controls discretized parameter shifts and encodes the loss as phase.

through Hamiltonian simulation [33–37], rather than as an externally driven classical routine.

In this Letter, we introduce *Coherent Quantum Learning (CQL)*, a new learning procedure that defines a general computational primitive for quantum learning. CQL applies to arbitrary parameterized circuit quantum models and does not rely on hybrid optimization loops or problem-specific structure. This comparison is depicted in Fig. 1. The construction of CQL is naturally compatible with fault-tolerant architectures, where long coherent evolutions can be sustained and leveraged for optimization. In this setting, training is realized through the simulation of a quantum dynamics acting on a parameter register, with the objective function encoded as an operator on a Hilbert space. This perspective unifies learning and quantum evolution within a single framework, in which quantum coherence functions as an active resource rather than a passive property.

Learning model.— As a minimal setting to illustrate the CQL protocol, we consider a reduced version of the data re-uploading model [38] consisting of a single-parameter learning model for binary classification. Given an input $x \in \mathbb{R}$ with label $y \in \{0, 1\}$, the model prepares the one-qubit state

$$|\psi(x,\theta)\rangle = R_y(\theta)\, U(x)\, |0\rangle = \alpha_0(x,\theta)\, |0\rangle + \alpha_1(x,\theta)\, |1\rangle\,, \tag{1}$$

where $U(x)$ is a data-encoding unitary (e.g., $R_x(x)$) and $R_y(\theta) = e^{-i\theta Y/2}$ is a trainable rotation. The cost function to be minimized is defined as

$$\mathcal{L}_y(x,\theta) = 1 - |\alpha_y(x,\theta)|^2, \tag{2}$$

where $\alpha_y = (1-y)\alpha_0 + y\alpha_1$, which penalizes incorrect classification outcomes.

To enable coherent training, we discretize the parameter θ using m bits, $\theta = 2\pi \sum_{j=1}^{m} \theta_j 2^{-j}$, $\theta_j \in \{0, 1\}$, which

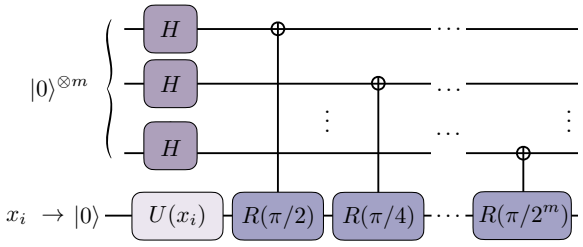


FIG. 2. Circuit for one coherent training step. An ancilla register controls discretized parameter shifts and encodes the loss as phase.

yields a grid $\mathcal{G}_m$ of 2^m possible values in $[0, 2\pi)$, each represented by a computational basis state $|\theta\rangle = |\theta_1\theta_2 \ldots \theta_m\rangle$ on m ancillary qubits. Therefore, one can define the corresponding discrete position and momentum operators in the θ basis denoted as $\hat{\theta}$, $\hat{P}_\theta$ respectively.

Crucially, by adding these m ancillary qubits, the model can be evaluated coherently over all parameter values of the grid by a unitary $U_{x,\theta}$ acting on the main qubit and the ancillas encoding the parameter register,

$$U_{x,\theta}\big(|0\rangle \otimes |0\rangle^{\otimes m}\big) = \sum_{\theta \in \mathcal{G}_m} \Big[\alpha_0(x,\theta)\, |0\rangle\, |\theta\rangle + \alpha_1(x,\theta)\, |1\rangle\, |\theta\rangle\Big]. \tag{3}$$

This operation entangles model outputs with parameter configurations on a $m+1$ qubit state, thus, the construction depicted in Fig. 2, serves as the basic primitive enabling coherent exploration of the loss landscape in the protocol introduced below.

Note that we can evaluate simultaneously more than one data point by adding one ancilla per data, thus increasing the size of the batch. In this case, the amplitude that defines the cost function, Eq. (2), is given by the bit-string of the corresponding labels, which can be mapped to the string $|0...0\rangle$ by applying X gates in those registers where the label is 1.

Coherent Quantum Learning (CQL) protocol.— We now introduce the protocol for training a single batch of the defined learning model coherently. For simplicity we will assume the batch is of size 1, where the extension to a larger size is straightforward. We build upon the time evolution introduced in [27],

$$\hat{H}_{CQL}(t) = e^{\varphi(t)} \left(-\frac{1}{2}\hat{P}_\theta^2\right) + e^{\chi(t)} \hat{H}_{\mathcal{L}}(x, \hat{\theta}), \tag{4}$$

where $e^{\varphi(t)}$, $e^{\chi(t)}$ are the damping parameters, $\hat{P}_\theta^2$ is the Laplacian operator in the θ basis and $\hat{H}_{\mathcal{L}}(x, \hat{\theta})$ is the diagonal Hamiltonian proportional to $\mathcal{L}_y(x, \theta)$ that encodes the objective function to minimize. In the absence of prior information, the initial condition is chosen as an equal superposition over the parameter space, ensuring an unbiased exploration of all configurations. If prior information is available (e.g., an informed initial guess θ_0), it can be incorporated by initializing the state as a

localized wavepacket (e.g., a Gaussian) centered around θ_0, thereby biasing the evolution toward promising regions of the landscape.

The first obstacle to simulating the dynamics efficiently on a gate-based quantum computer arises from the time dependency of the Hamiltonian. Simulating time-dependent Hamiltonians is not a native operation in the circuit model, and typically introduces considerable overhead [39]. To address this issue, we employ the technique outlined in Refs. [40, 41], which involves enlarging the Hilbert space, introducing a "clock dimension" [42]. This expansion transforms the time dependence of a given Hamiltonian into a position operator $\hat{X}_t$ in an additional Hilbert space. Consequently, albeit at the expense of enlarging the Hilbert space to $\mathcal{H}_t \otimes \mathcal{H}_\theta$, this approach enables us to work with a time-independent Hamiltonian

$$\hat{H}_{CQL} = \hat{P}_t \otimes \mathbb{I} + e^{\varphi(\hat{X}_t)} \otimes \left(-\frac{1}{2}\hat{P}_\theta^2\right) + e^{\chi(\hat{X}_t)} \otimes \hat{H}_{\mathcal{L}}(x, \hat{\theta}), \tag{5}$$

where we assume $\hat{X}_t$, $\hat{P}_t$ to be discretized by using n_t ancilla registers and $|t_0\rangle$, the initial state for the new register, to be a representation of a localized wave function (delta function) in the position basis at $x_t = 0$. In this way, the complexity of training our learning model is transferred to the simulation of the dynamics generated by Eq. (5). Throughout this Letter, we assume access to a block encoding of the Hamiltonian, and explicitly describe how to construct the novel components.

We firstly address how to construct the block encoding of $\hat{H}_{\mathcal{L}}(x, \hat{\theta}) = \sum_{\theta \in \mathcal{G}_m} \mathcal{L}(x, \theta) |\theta\rangle \langle\theta|$. The key ingredient in this construction is the unitary $U_{x,\theta}$ described in Eq. (3), which evaluates the learning model in every possible value of the grid for the trainable parameter θ. We show the explicit circuit implementation for $U_{x,\theta}$ in Fig. 2. Without loss of generality we can assume $y = 1$, which translates into $\mathcal{L}(x, \theta) = |\alpha_0(x, \theta)|^2$. Therefore, the next step is to achieve a block encoding of $\hat{H}_{\mathcal{L}}$ from query access to $U_{x,\theta}$. To this end, we can use Theorem 2 from Ref. [43] in order to prepare a $(1, m+4, 0)$ block encoding $U_{\sqrt{H}}$ of the diagonal operator $\sqrt{H} = \sum_{\theta \in \mathcal{G}_m} \alpha_0(x, \theta) |\theta\rangle \langle\theta|$ with $\mathcal{O}(m)$ circuit depth and $\mathcal{O}(1)$ queries to a controlled version of $U_{x,\theta}$. Note that the ancilla used for classifying x has been counted as an additional ancilla qubit in the block encoding. By using the same protocol but this time applied to $U_{x,\theta}^\dagger$ instead, we get $U_{\sqrt{H}^\dagger}$, a $(1, m+4, 0)$ block encoding of $\sqrt{H}^\dagger = \sum_{\theta \in \mathcal{G}_m} \alpha_0^*(x, \theta) |\theta\rangle \langle\theta|$. Using these two block encodings, we now build the block encoding of $\hat{H}_{\mathcal{L}}$, which actually can be written as a block encoding of $\sqrt{H}\sqrt{H}^\dagger$. To achieve the block encoding of a product from the individual block encodings of same the size [44], we can use a single ancillary qubit, see Supplemental Material. We thereby obtain a $(1, m+5, 0)$ block encoding of $\hat{H}_{\mathcal{L}}$ with $\mathcal{O}(m)$ circuit depth and $\mathcal{O}(1)$ queries to a controlled version of $U_{x,\theta}$.

Finally, in order to achieve the block encoding of $\hat{H}_{CQL}$, we need to obtain the block encoding of certain functions of $\hat{x}_t$, $\hat{P}_t$, $\hat{P}_\theta$. The discrete representation of these operators, broadly studied, is not unique, as it depends on the discretization, the numerical method employed, and the boundary conditions. Some explicit examples on how to construct the block encoding of these operators can be found in Ref. [45]. A systematic analysis of optimal discretizations and the control of operator norms to avoid unfavorable scaling is beyond the scope of this Letter.

Interferometer plus figure.

Next, one may apply the Quantum Singular Value Transformation (QSVT) [46] to realize a polynomial transformation to $\hat{x}_t$ approximating the functions $e^{\varphi(\hat{X}_t)}$, $e^{\chi(\hat{X}_t)}$, obtaining the corresponding block encodings. Alternatively, we can consider standard techniques for block-encoding arbitrary diagonal operators, e.g., via quantum read-only memory for classical function values of χ on the $\hat{X}_t$ register, or via the amplitude-to-diagonal conversion of Theorem 2 of Ref. [43] when the function values can be loaded as state amplitudes. After homogenizing all block encodings in the same Hilbert space to act in the same number of ancillas, the target Hamiltonian can be constructed as a linear combination of the individual block encodings.

Finally, since access to our Hamiltonian is provided via a block encoding, we use qubitization [34] to implement the time evolution of the initial state. In this way, we obtain a quantum state that encodes a probability distribution on the values of θ whose maximum corresponds to the optimal value for training our learning model.

Batched CQL.— So far, we have introduced general coherent training for a single batch. Although it is possible to construct a batch the size of the entire dataset, corresponding to coherent global training, this would require as many ancillas as data points. This is not a realistic scenario, as it becomes impractical for large datasets. Therefore, the need for iterative learning from different batches arises. We propose an iterative batched training strategy, analogous to mini-batch methods in classical machine learning, whereby the dataset is partitioned into smaller subsets that are processed sequentially.

Let us consider a training set divided into B batches. Starting from an initial parameter state, the system evolves under the Hamiltonian in Eq. (4) using the first batch of training samples. The resulting parameter state is then used as the initial condition for a subsequent evolution under an updated Hamiltonian that encodes the next batch into the data registers. This sequential process is repeated until all batches in the training set have been processed. Here, the choice of batch size introduces a fundamental trade-off between spatial resources, namely the qubit count, and temporal resources, namely the circuit depth. Figure 3 shows a schematic representation of batched coherent training and is discussed in more detail later.

The main challenge of batch training is for the quantum system to retain information from previous batches while, at the same time, learning from new samples. This requirement fundamentally alters the role of the damping

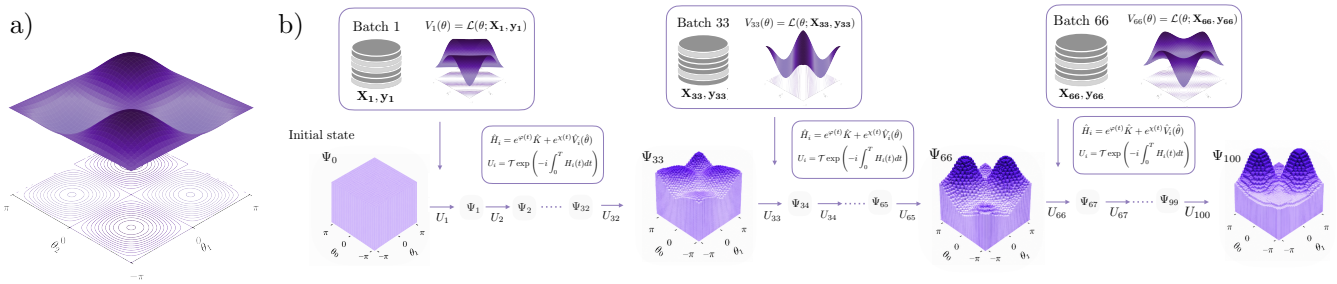


FIG. 3. Numerical illustration of batch-based Coherent Quantum Learning (CQL). (a) Loss landscape $\mathcal{L}(\theta; \mathbf{X}, \mathbf{y})$ evaluated over the full training dataset as a function of the two variational parameters θ_0, θ_1, with contours indicating the structure of the global objective. (b) Evolution of the probability distribution of the parameter state $\hat{\theta}$ across sequential batched coherent training. Starting from a uniform superposition Ψ_0 over parameter space, the state is evolved under the CQL Hamiltonian, where the loss potential is constructed from individual training samples (batch size 1). The loss landscape associated with select batches is shown alongside the corresponding parameter distributions, and the final state Ψ_{100} after processing all 100 batches. As training progresses, the probability amplitude increasingly concentrates in regions of low loss, and after completing the full epoch, the final distribution anti-correlates with the global loss landscape shown in (a), confirming that the coherent evolution has effectively learned an inverse representation of the objective function.

schedules compared to the quantum Hamiltonian descent framework [27]. In the conventional global optimization setting, the schedules $\varphi(t)$ and $\chi(t)$ are typically chosen such that $e^{\varphi(t)/\chi(t)} \to 0$ as $t \to \infty$, thereby concentrating the parameter-state distribution around the global minima of the loss landscape. In the batched setting, however, such aggressive concentration is generally undesirable, since it would overemphasize the information contained in the current batch and reduce robustness to subsequent batches while erasing information from previous batches.

This phenomenon is directly analogous to stochastic or mini-batch gradient methods, where the update magnitude is regulated through a learning rate hyperparameter. In the present framework, the analogous control parameter is the effective evolution time under the CQL Hamiltonian, together with the associated damping schedules. These quantities determine how strongly the parameter distribution adapts to each processed batch, and therefore govern the balance between local batch fitting and global generalization. Consequently, in the batched regime, the asymptotic divergence condition on the ratio $\varphi(t)/\chi(t)$ is no longer appropriate and need not be imposed.

Numerical demonstrations.— (a) **Quantum classification with a data re-uploading model**. To illustrate the proposed protocol, we consider a binary classification task in which the objective is to determine whether input points lie inside or outside a one-dimensional interval. As a quantum model, we employ a single-qubit data re-uploading circuit with two layers of trainable rotations, yielding a variational ansatz with two parameters. The corresponding loss landscape is shown in Fig. 3(a). Each parameter is then discretized using five qubits. Within this framework, we perform one training epoch using the coherent training protocol in the single-sample regime.

Fig. 3(b) presents the evolution of the probability distribution induced by the discretized variational parameter space. Starting from an initial uniform superposition, the probability distribution progressively adapts to the structure of the cost function, developing pronounced peaks at parameter values associated with minima of the loss landscape. As training proceeds, the distribution increasingly reflects the geometry of the underlying objective function, ultimately exhibiting a bimodal concentration around the dominant minima. These results indicate that the model effectively learns an inverse representation of the loss landscape encoded in the final probability amplitudes. Rather than producing a single optimized parameter value, the protocol generates a quantum representation of the landscape itself, where low-loss regions are encoded as high-probability components of the final state. Further details regarding the exact configuration of these numerical results are provided in the Supplementary Materials.

The computational complexity of CQL depends on the cost of Hamiltonian simulation and the construction of the loss Hamiltonian. While a complete complexity analysis is beyond the scope of this work, the framework naturally inherits advances in block encoding and Hamiltonian simulation. Determining whether coherent learning can provide asymptotic advantages over hybrid optimization remains an important open question.

(b) **Interferometric phase learning**. To demonstrate the applicability of CQL to physical systems and parameter estimation, we ground our framework in a concrete physics-motivated task: learning an unknown phase shift $\phi \in [-\pi, \pi)$ within a two-mode interferometer (such as a Mach-Zehnder setup). Unlike the classification example, this task does not arise from machine learning but from a physical inference problem. It illustrates that the same coherent learning protocol naturally applies to learning unknown Hamiltonian parameters.

The physical setup, illustrated in the upper-left panel of Fig. 4, consists of a balanced Mach–Zehnder interferometer [47, 48]. A single photon is split by a 50:50 beam splitter (BS) into two spatial modes, defining the com-

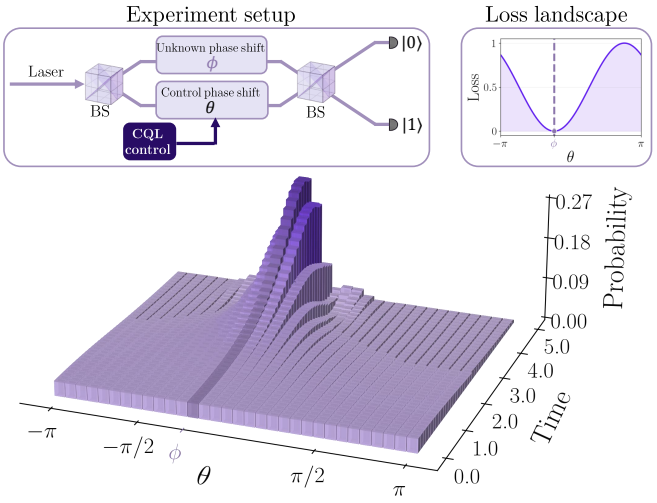


FIG. 4. Upper left: Schematic of the interferometric phase-estimation setup. A single photon propagates through a balanced Mach–Zehnder interferometer containing an unknown phase shift ϕ and a controllable phase shift θ, which is iteratively updated by the CQL protocol. Upper right: Corresponding loss landscape, $\mathcal{L}(\theta, \phi) = \sin^2[(\theta - \phi)/2]$, whose global minimum identifies the unknown phase. Main: Time evolution of the probability distribution over the discretized parameter register, $|\Psi(\theta, t)|^2$. Starting from a uniform superposition, the coherent dynamics progressively localize the probability amplitude around the global minimum of the loss landscape, yielding an accurate estimate of the unknown phase ϕ.

putational basis states $|0\rangle$ (lower arm) and $|1\rangle$ (upper arm). The upper arm contains an unknown phase shift $\phi \in [-\pi, \pi)$, while the lower arm contains a controllable phase shift θ, which is dynamically updated by the CQL protocol. The two paths are subsequently recombined at a second balanced beam splitter before projective measurement at the output ports.

The interference pattern depends solely on the relative phase between the two arms, yielding the measurement probabilities $p(0) = \sin^2\left(\frac{\phi-\theta}{2}\right)$ and $p(1) = \cos^2\left(\frac{\phi-\theta}{2}\right)$ (see Appendix S5 for the full algebraic derivation). Consequently, estimating the unknown phase ϕ reduces to identifying the control parameter θ that maximizes constructive interference. Within the CQL framework, this objective is encoded through the loss function $\mathcal{L}(\theta, \phi) = \sin^2\left(\frac{\theta-\phi}{2}\right)$, which attains its global minimum when $\theta = \phi$.

The parameter register representing the candidate values of θ is encoded using five qubits, corresponding to a discretized parameter space of 2^5 points. The register is initialized in a uniform superposition, representing a non-informative prior over the unknown phase. The subsequent evolution is simulated coherently, following the CQL dynamics introduced in this work. Here, we employ the time-dependent schedules $\phi(t) = 1/t$ and $\chi(t) = t$, which progressively suppress exploration while increasing the influence of the loss landscape. Details of the numerical implementation and simulation parameters are provided in Appendix S5.

Figure 4 illustrates the coherent evolution of the CQL parameter register throughout the optimization process. As the evolution progresses, the increasing influence of the loss potential and the simultaneous suppression of kinetic spreading drive the wavefunction toward the global minimum of the interferometric loss landscape. By the end of the evolution, the probability amplitude is strongly localized around the optimal parameter value, corresponding to the unknown phase shift ϕ. This demonstrates that the coherent dynamics successfully perform phase estimation without requiring repeated classical optimization steps.

Conclusion and outlook.— We have introduced Coherent Quantum Learning (CQL), a framework in which the training of quantum models is itself implemented as a coherent quantum process. Model parameters are promoted to quantum degrees of freedom and evolved under a Hamiltonian encoding the objective function, replacing iterative classical optimization with unitary dynamics in parameter space. In this formulation, superposition and interference become active computational resources that shape the learning dynamics, allowing parameter configurations to be explored collectively rather than sequentially.

Our construction provides a general and architecture-independent primitive for quantum learning. The protocol applies to arbitrary parameterized quantum circuits, naturally accommodates batched training strategies, and is compatible with fault-tolerant quantum computing architectures where long coherent evolutions can be sustained. Numerical simulations demonstrate that the resulting dynamics progressively concentrate probability amplitude around low-loss regions of the landscape, effectively encoding learned representations within the quantum state itself. Several directions remain open for future investigation, including complexity analyses, improved Hamiltonian constructions and discretization strategies, extensions to fault-tolerant quantum model architectures [49, 50], and integration with error mitigation techniques [51].

More broadly, this work points toward a reconsideration of how learning should be formulated in quantum systems. Much of the current effort in quantum machine learning seeks to reproduce classical training paradigms, such as gradient-based optimization, within quantum settings. However, these paradigms implicitly impose a classical structure on the learning process, potentially obscuring the role of coherence and interference. In contrast, the procedure introduced here treats learning itself as a physical process governed by quantum dynamics, suggesting that quantum models may benefit from training principles that are native to quantum mechanics rather than adapted from classical algorithms.

Code and data availability: The code and data for the numeric results are available at https://github.com/nacedob/Coherent-Quantum-Learning

Acknowledgments: We thank Xabier Gutierrez, Richard Kueng and José Ignacio Latorre for discussions. LP acknowledges support by the National Research Foundation, Singapore through the National Quantum Office, hosted in A*STAR, under its Centre for Quantum Technologies Funding Initiative (S24Q2d0009). LP also acknowledges support by A*STAR under its Young Investigator Research Grant (YIRG) M25N8c0131. PRG and JGC acknowledge support from HORIZON-CL4-2022-QUANTUM01-SGA project 101113946 OpenSuperQPlus100 of the EU Flagship on Quantum Technologies, the Spanish Ramón y Cajal Grant RYC-2020-030503-I, and the "Generación de Conocimiento" project Grant No. PID2021-125823NA-I00 funded by MICIU/AEI/10.13039/501100011033, by "ERDF Invest in your Future" and by FEDER EU. PGA acknowledges support from UPV/EHU Ph.D. Grant No. PIFG 22/25. We also acknowledge support from the Basque Government through Grants No. IT1470-22, the Elkartek project KUBIBIT - kuantikaren berrikuntzarako ibilbide teknologikoak (ELKARTEK25/79), and from the IKUR Strategy under the collaboration agreement between Ikerbasque Foundation and BCAM on behalf of the Department of Education of the Basque Government. This work has also been partially supported by the Ministry for Digital Transformation and the Civil Service of the Spanish Government through the QUANTUM ENIA project call – Quantum Spain project, and by the European Union through the Recovery, Transformation and Resilience Plan – NextGenerationEU within the framework of the Digital Spain 2026 Agenda.

[1] Christopher M Bishop and Nasser M Nasrabadi, *Pattern Recognition and Machine Learning*, Vol. 4 (Springer, 2006).
[2] Kevin P Murphy, *Machine learning: a probabilistic perspective* (MIT press, 2012).
[3] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning representations by back-propagating errors," Nature **323**, 533–536 (1986).
[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," Nature **521**, 436–444 (2015).
[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning* (MIT Press, 2016).
[6] Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan, "A variational perspective on accelerated methods in optimization," Proceedings of the National Academy of Sciences **113**, E7351–E7358 (2016), https://www.pnas.org/doi/pdf/10.1073/pnas.1614734113.
[7] Paul Benioff, "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines," Journal of Statistical Physics **22**, 563–591 (1980).
[8] Richard P. Feynman, "Simulating physics with computers," International Journal of Theoretical Physics **21**, 467–488 (1982).
[9] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
[10] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, "Quantum machine learning," Nature **549**, 195–202 (2017).
[11] Maria Schuld and Francesco Petruccione, *Supervised Learning with Quantum Computers* (Springer, 2018).
[12] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe, "Circuit-centric quantum classifiers," Physical Review A **101** (2020), 10.1103/physreva.101.032308.
[13] Jonas M. Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J. Coles, "An Adaptive Optimizer for Measurement-Frugal Variational Algorithms," Quantum **4**, 263 (2020).
[14] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert, "Stochastic gradient descent for hybrid quantum-classical optimization," Quantum **4**, 314 (2020).
[15] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles, "Variational quantum algorithms," Nature Reviews Physics **3**, 625–644 (2021).
[16] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," Physical Review A **98** (2018), 10.1103/physreva.98.032309.
[17] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran, "Evaluating analytic gradients on quantum hardware," Physical Review A **99** (2019), 10.1103/physreva.99.032331.
[18] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin, "General parameter-shift rules for quantum gradients," Quantum **6**, 677 (2022).
[19] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven, "Barren plateaus in quantum neural network training landscapes," Nature Communications **9** (2018), 10.1038/s41467-018-07090-4.
[20] Martín Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and M. Cerezo, "Barren plateaus in variational quantum computing," Nature Reviews Physics **7**, 174–189 (2025).
[21] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J. Osborne, Robert Salzmann, Daniel Scheiermann, and Ramona Wolf, "Training deep quantum neural networks," Nature Communications **11** (2020), 10.1038/s41467-020-14454-2.
[22] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini, "Parameterized quantum circuits as machine learning models," Quantum Science and Technology **4**, 043001 (2019).
[23] Amira Abbas, Robbie King, Hsin-Yuan Huang, William J. Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod McClean, "On quantum backpropagation, information reuse, and cheating measurement collapse," in *Advances in Neural Information Processing Systems*, Vol. 36, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt,

and S. Levine (Curran Associates, Inc., 2023) pp. 44792–44819.
[24] Joseph Bowles, David Wierichs, and Chae-Yeun Park, "Backpropagation scaling in parameterised quantum circuits," Quantum **9**, 1873 (2025).
[25] Guillaume Verdon, Jason Pye, and Michael Broughton, "A universal training algorithm for quantum deep learning," (2018), arXiv:1806.09729 [quant-ph].
[26] Qi Ye, Shuangyue Geng, Zizhao Han, Weikang Li, L. M. Duan, and Dong-Ling Deng, "Quantum automated learning with provable and explainable trainability," (2025), arXiv:2502.05264 [quant-ph].
[27] Jiaqi Leng, Ethan Hickman, Joseph Li, and Xiaodi Wu, "Quantum Hamiltonian Descent," (2023), arXiv:2303.01471 [quant-ph].
[28] Jiaqi Leng, Yufan Zheng, Zhiyuan Jia, Lei Fan, Chaoyue Zhao, Yuxiang Peng, and Xiaodi Wu, "Quantum Hamiltonian Descent for Non-smooth Optimization," (2025), arXiv:2503.15878 [math.OC].
[29] Zherui Chen, Yuchen Lu, Hao Wang, Yizhou Liu, and Tongyang Li, "Quantum langevin dynamics for optimization," Communications in Mathematical Physics **406** (2025), 10.1007/s00220-025-05234-4.
[30] Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan, "A variational perspective on accelerated methods in optimization," Proceedings of the National Academy of Sciences **113** (2016), 10.1073/pnas.1614734113.
[31] Michael Betancourt, Michael I. Jordan, and Ashia C. Wilson, "On symplectic optimization," (2018), arXiv:1802.03653 [stat.CO].
[32] Ahmet Burak Catli, Sophia Simon, and Nathan Wiebe, "Exponentially better bounds for quantum optimization via dynamical simulation," (2025), arXiv:2502.04285 [quant-ph].
[33] Guang Hao Low and Isaac L. Chuang, "Optimal Hamiltonian Simulation by Quantum Signal Processing," Physical Review Letters **118** (2017), 10.1103/physrevlett.118.010501.
[34] Guang Hao Low and Isaac L. Chuang, "Hamiltonian Simulation by Qubitization," Quantum **3**, 163 (2019).
[35] Jan Lukas Bosse, Andrew M. Childs, Charles Derby, Filippo Maria Gambetta, Ashley Montanaro, and Raul A. Santos, "Efficient and practical Hamiltonian simulation from time-dependent product formulas," Nature Communications **16**, 2673 (2025).
[36] Pei Zeng, Jinzhao Sun, Liang Jiang, and Qi Zhao, "Simple and High-Precision Hamiltonian Simulation by Compensating Trotter Error with Linear Combination of Unitary Operations," PRX Quantum **6**, 010359 (2025).
[37] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma, "Simulating Hamiltonian Dynamics with a Truncated Taylor Series," Phys. Rev. Lett. **114**, 090502 (2015).
[38] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre, "Data re-uploading for a universal quantum classifier," Quantum **4**, 226 (2020).
[39] Javier Gonzalez-Conde, Zachary Morrell, Marc Vuffray, Tameem Albash, and Carleton Coffrin, "Cost of emulating a small quantum annealing problem in the circuit model," Phys. Rev. A **111**, 062606 (2025).
[40] Yu Cao, Shi Jin, and Nana Liu, "Quantum simulation for time-dependent Hamiltonians—with applications to non-autonomous ordinary and partial differential equations," Journal of Physics A: Mathematical and Theoretical **58**, 155304 (2025).
[41] Yu Cao, Shi Jin, and Nana Liu, "Unifying framework for quantum simulation algorithms for time-dependent Hamiltonian dynamics," Phys. Rev. Res. **7**, 043186 (2025).
[42] Don N. Page and William K. Wootters, "Evolution without evolution: Dynamics described by stationary observables," Phys. Rev. D **27**, 2885–2892 (1983).
[43] Arthur G. Rattew and Patrick Rebentrost, "Non-linear transformations of quantum amplitudes: Exponential improvement, generalization, and applications," (2023), arXiv:2309.09839 [quant-ph].
[44] Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, and Fernando G. S. L. Brandão, *Quantum Algorithms: A Survey of Applications and End-to-end Complexities* (Cambridge University Press, 2025).
[45] Nikita Guseynov, Xiajie Huang, and Nana Liu, "Gate construction of block-encoding for Hamiltonians needed for simulating partial differential equations," Phys. Rev. Res. **7**, 033100 (2025).
[46] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe, "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19 (ACM, 2019) p. 193–204.
[47] Ludwig Zehnder, "Ein neuer interferenzrefraktor," Zeitschrift für Instrumentenkunde **11**, 275–285 (1891).
[48] Ludwig Mach, "Ueber einen interferenzrefraktor," Zeitschrift für Instrumentenkunde **12**, 89–93 (1892).
[49] Petr Ivashkov, Po-Wei Huang, Kelvin Koor, Lirandë Pira, and Patrick Rebentrost, "Qkan: Quantum kolmogorov-arnold networks," (2024), arXiv:2410.04435 [quant-ph].
[50] Arthur G. Rattew, Po-Wei Huang, Naixu Guo, Lirandë Pira, and Patrick Rebentrost, "Accelerating inference for multilayer neural networks with quantum computers," (2025), arXiv:2510.07195 [quant-ph].
[51] Zhenyu Cai, Ryan Babbush, Simon C. Benjamin, Suguru Endo, William J. Huggins, Ying Li, Jarrod R. McClean, and Thomas E. O'Brien, "Quantum error mitigation," Reviews of Modern Physics **95** (2023), 10.1103/revmodphys.95.045005.

Supplemental Material

for

Coherent Quantum Learning: Training through Coherent Evolution

CONTENTS

References 6

S1. Hamiltonian Construction 8
S1.1. Functions of the Position Operator 9
S1.2. Momentum Operator 9

S2. Cost Hamiltonian 10
S2.1. Extension to several simultaneously evaluated data points 11
S2.2. Block-encoding different cost functions 12
S2.3. Homogenizing Block Encodings 14

S3. Batched CQL Algorithm 14

S4. Details on Quantum Classification with a Data Re-Uploading Model 15

S5. Details on Interferometric Phase Learning 16
S5.1. Mach–Zehnder Interferometer State Evolution 16
S5.2. Numerical Implementation and Simulation Framework 17

Appendix S1: Hamiltonian Construction

Our CQL Hamiltonian is time dependent. To overcome the explicit time dependence of the CQL Hamiltonian, we use the clock-dimension construction of Refs. [40, 41]. The basic idea is to replace a non-autonomous Schrödinger equation by an autonomous one in a larger Hilbert space. Let the original parameter-space dynamics be generated by a time-dependent Hamiltonian, see Eq. (4). The corresponding evolution is formally given by the time-ordered exponential

$$U(T,0) = \mathcal{T} \exp\left(-i \int_0^T \hat{H}_{\mathrm{CQL}}(t)\, dt \right), \tag{S1}$$

which is difficult to implement directly because the Hamiltonian changes during the evolution. The clock construction removes this explicit time dependence by introducing an additional Hilbert space $\mathcal{H}_t$, with conjugate operators $\hat{X}_t$ and $\hat{P}_t$. The operator $\hat{X}_t$ stores the clock coordinate, while $\hat{P}_t = -i\partial_{x_t}$ generates translations along this coordinate. In the enlarged Hilbert space $\mathcal{H}_t \otimes \mathcal{H}_\theta$, the time-dependent scalar functions $e^{\varphi(t)}$ and $e^{\chi(t)}$ are promoted to diagonal operators $e^{\varphi(\hat{X}_t)}$ and $e^{\chi(\hat{X}_t)}$. This gives the time-independent Hamiltonian given by Eq. (5). The role of the first term, $\hat{P}_t \otimes \mathbb{I}$, is to translate the clock wavepacket through the clock register. If the clock register is initialized in a localized state $|t_0\rangle$, concentrated near $x_t = 0$, then the translation generated by $\hat{P}_t$ makes the system sample the values of $e^{\varphi(x_t)}$ and $e^{\chi(x_t)}$ along the clock coordinate during the evolution. In this way, the original time dependence is encoded as position dependence in an auxiliary register. After discretizing the clock with n_t qubits, the continuous operators $\hat{X}_t$ and $\hat{P}_t$ are replaced by finite-dimensional approximations acting on a grid of 2^{n_t} clock points.

This construction transfers the problem of implementing a time-dependent evolution to the problem of simulating the time-independent Hamiltonian in Eq. (5). The price paid is the enlargement of the Hilbert space and the need to block-encode the new operators appearing in the clock Hamiltonian. In particular, to construct the full Hamiltonian one needs block-encodings of $\hat{P}_t$, $\hat{P}_\theta^2$, the diagonal functions $e^{\varphi(\hat{X}_t)}$ and $e^{\chi(\hat{X}_t)}$, and the loss Hamiltonian $\hat{H}_{\mathcal{L}}(x, \hat{\theta})$. Once these block-encodings are available, the full Hamiltonian can be assembled using standard block-encoding arithmetic: tensor products for terms acting on different registers, multiplication for operator products, and linear-combination-of-unitaries techniques for the final sum.

S1.1. Functions of the Position Operator

For our purposes, we are interested in block-encodings of diagonal functions of the position operator, $\hat{X}$, such as $e^{\varphi(\hat{X})}$ and $e^{\chi(\hat{X})}$. Since $\hat{X}$ is diagonal on a uniform grid, any function $f(\hat{X})$ is also diagonal:

$$f(\hat{X}) = \mathrm{diag}\big(f(x_0), f(x_1), \ldots, f(x_{N-1})\big). \tag{S2}$$

One systematic way to construct such a block-encoding is to start from a block-encoding of the normalized position operator $\hat{X}/\alpha_X$, with spectrum contained in $[-1, 1]$, and then use QSVT to implement a polynomial approximation P_d to the desired function. More precisely, after mapping the spatial interval $[x_{\min}, x_{\max}]$ to $[-1, 1]$, one chooses a polynomial P_d such that

$$\max_{x\in[x_{\min},x_{\max}]} \left| P_d(x) - \frac{f(x)}{\alpha_f} \right| \le \varepsilon, \qquad f(x) = e^{\varphi(x)} \text{ or } f(x) = e^{\chi(x)}, \tag{S3}$$

where $\alpha_f \ge \max_{x\in[x_{\min},x_{\max}]} |f(x)|$ is the block-encoding normalization. Applying QSVT to the block-encoding of $\hat{X}/\alpha_X$ then gives a block-encoding of $P_d(\hat{X}) \approx \frac{f(\hat{X})}{\alpha_f}$. The query complexity is linear in the polynomial degree $N_{\text{queries}} = O(d)$ uses of the block-encoding of $\hat{X}/\alpha_X$ and its inverse, together with $O(d)$ single-qubit phase rotations, plus the gate cost of implementing the underlying block-encoding of $\hat{X}$. Therefore, the efficiency of this route is controlled by the approximation degree d. If f is analytic on and around the relevant interval, Chebyshev or minimax polynomial approximations can achieve $d = O(\log(1/\varepsilon))$ up to constants depending on the size of the analytic continuation region and the variation of f. By contrast, for functions with only finite smoothness, the degree may scale polynomially in $1/\varepsilon$. Thus, QSVT gives an asymptotically efficient block-encoding when the functions $e^{\varphi(x)}$ and $e^{\chi(x)}$ are sufficiently regular, but the cost can increase significantly for rapidly varying or nonsmooth coefficients.

There are also alternatives that exploit the fact that $f(\hat{X})$ is diagonal. One possibility is to construct a direct LCU block-encoding using a Walsh–Hadamard or Pauli-Z expansion of the sampled function on the grid:

$$f(\hat{X}) = \sum_{S\subseteq\{1,\ldots,n\}} c_S Z_S, \qquad Z_S = \bigotimes_{j=1}^{n} Z_j^{\mathbf{1}_{j\in S}}. \tag{S4}$$

For the linear position operator, only $O(n)$ such Pauli-Z strings are needed, which leads to a very efficient LCU construction as in Ref. [45]. For a general function $f(x)$, however, the Walsh expansion may contain many nonzero coefficients, potentially up to 2^n, unless f has additional structure or the expansion can be truncated efficiently. In that case, the LCU cost is controlled by the number of retained Walsh coefficients and by their ℓ_1-norm,

$$\alpha_{\text{LCU}} = \sum_S |c_S|. \tag{S5}$$

Another direct approach is to compute the grid value $f(x_j)$ reversibly and load it into an ancilla amplitude. Namely, one implements an arithmetic oracle of the form

$$|j\rangle|0\rangle \longmapsto |j\rangle \left(\frac{f(x_j)}{\alpha_f}|0\rangle + \sqrt{1 - \left|\frac{f(x_j)}{\alpha_f}\right|^2}\,|1\rangle \right), \tag{S6}$$

which directly block-encodes the diagonal matrix $f(\hat{X})/\alpha_f$. The cost of this construction is governed by the reversible arithmetic required to evaluate $f(x_j)$ to the desired precision, rather than by a QSVT polynomial degree. This can be preferable when f has an efficient arithmetic description, for example when $\varphi(x)$ or $\chi(x)$ are simple polynomials, rational functions, or elementary functions. Therefore, QSVT is a natural and general route when one already has a good block-encoding of $\hat{X}$ and a low-degree polynomial approximation to f, whereas direct Walsh/LCU or arithmetic amplitude-loading constructions may be more efficient for diagonal functions with exploitable structure.

S1.2. Momentum Operator

The choice for representing the momentum operator $\hat{P}$ on a discrete space is not unique and depends on the numerical scheme employed and the boundary conditions of interest, each leading to a different matrix representation

with distinct accuracy, sparsity, and stability properties. For instance, finite-difference discretizations produce sparse matrices whose entries depend explicitly on the boundary treatment, whereas Fourier spectral methods produce a diagonal representation in momentum space that is generally dense in position space. Consequently, a block encoding of the momentum operator should always be understood as a block encoding of a specific discretized version of the continuous operator, rather than of a unique finite-dimensional object.

A common drawback of these discretizations is the scaling of the norm of the momentum operator. In the absence of additional regularity assumptions, the derivative operator is unbounded. Consequently, refining the grid increases the largest momentum mode that can be represented and, therefore, increases the normalization factor required in a block-encoding. This directly affects the cost of Hamiltonian simulation. In typical discretizations, the norm of the discretized momentum operator scales linearly with the number of grid points, that is, as $O(N)$. Since $N = 2^n$ for an n-qubit spatial register, this corresponds to an exponential scaling in the number of qubits. An explicit methodology for achieving a block encoding of the momentum operator can be found in Ref. [45].

However, the unfavorable scaling is not intrinsic to all implementations, but rather to treating the derivative operator as a generic matrix whose block-encoding normalization is set by its worst-case operator norm. Since the derivative is unbounded, this worst-case norm necessarily grows as the grid is refined. To avoid paying this cost, one must either exploit additional structure of the derivative operator or impose regularity assumptions on the states being simulated.

To mitigate this scaling, we need to explore specific structures of the derivative matrices. The idea relies in the fact that the large norm of the discretized derivative is caused by high-frequency grid modes, restricting the dynamics to a subspace $|p| \leq K$ replaces the worst-case bound $\|P\| = O(N)$ by an effective bound $\|P\|_{\text{eff}} = O(K)$. For sufficiently smooth or analytic solutions, K can be much smaller than the total number of grid points N, so the Hamiltonian-simulation cost is controlled by the physically occupied bandwidth rather than by the formal ultraviolet cutoff of the discretization.

In order to illustrate how to achieve a block encoding of this operator (without mitigating the scaling), we can consider a periodic grid. Let S denote the cyclic shift

$$S|j\rangle = |j + 1 \pmod N\rangle. \tag{S7}$$

The central-difference momentum operator can be written as

$$\hat{P} = -i\frac{S - S^\dagger}{2\Delta x} = \frac{1}{2\Delta x}\left(-iS + iS^\dagger\right). \tag{S8}$$

Since $-iS$ and $iS^\dagger$ are unitary, this gives an immediate LCU block-encoding. With $\text{PREP}|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\text{SELECT} = |0\rangle\langle 0| \otimes (-iS) + |1\rangle\langle 1| \otimes (iS^\dagger)$, one obtains

$$(\langle +| \otimes I)\text{SELECT}(|+\rangle \otimes I) = \Delta x\, \hat{P}. \tag{S9}$$

Thus this construction gives a $\left(\frac{1}{\Delta x}, 1, 0\right)$ block-encoding of $\hat{P}$. Note that the rescaling factor $\frac{1}{\Delta x}$ will lead to a simulation time sacling like $t \sim \Delta x$ (exponential with number of qubits).

A block-encoding of $\hat{P}^2$ can be obtained in two natural ways. One may either compose two block-encodings of $\hat{P}$, using the standard block-encoding multiplication rule, or construct a block-encoding of $\hat{P}^2$ directly by exploiting the explicit finite-difference stencil of the second-derivative operator.

Appendix S2: Cost Hamiltonian

We now describe the construction of the loss Hamiltonian. The objective is to block-encode

$$\hat{H}_{\mathcal{L}}(x, \hat{\theta}) = \sum_{\theta \in \mathcal{G}_m} \mathcal{L}(x, \theta)\, |\theta\rangle\, \langle\theta|\,, \tag{S1}$$

which is diagonal in the parameter register. The key primitive is the coherent model-evaluation unitary $U_{x,\theta}$, which evaluates the quantum model simultaneously over all grid values of the trainable parameter. Acting on the model qubit and the m-qubit parameter register, it prepares

$$U_{x,\theta}\left(|0\rangle \otimes |0\rangle^{\otimes m}\right) = \sum_{\theta \in \mathcal{G}_m} \left[\alpha_0(x, \theta)\, |0\rangle\, |\theta\rangle + \alpha_1(x, \theta)\, |1\rangle\, |\theta\rangle\right]. \tag{S2}$$

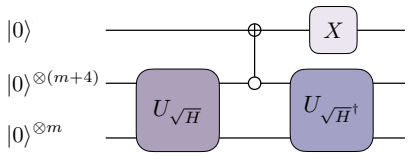


FIG. S1. Circuits that implements the block encoding of H from the block encodings of $\sqrt{H}$ and $\sqrt{H}^\dagger$. It requires access to the controlled block encoding of the roots of H, one more ancilla, an (m+4)-controlled generalized Toffoli gate and one X gate.

Thus the amplitudes of the model output are coherently correlated with the parameter values θ. For binary classification, and without loss of generality taking $y = 1$, the loss is

$$\mathcal{L}(x,\theta) = 1 - |\alpha_1(x,\theta)|^2 = |\alpha_0(x,\theta)|^2. \tag{S3}$$

Therefore, it is enough to construct a diagonal operator whose entries are $|\alpha_0(x,\theta)|^2$.

To do this, we first construct a block-encoding of the "square-root" diagonal operator

$$\sqrt{H} = \sum_{\theta\in\mathcal{G}_m} \alpha_0(x,\theta)\,|\theta\rangle\,\langle\theta|\,. \tag{S4}$$

Using the amplitude-to-diagonal conversion of Ref. [43], query access to a controlled version of $U_{x,\theta}$ gives a $(1, m+4, 0)$ block-encoding $U_{\sqrt{H}}$ of $\sqrt{H}$, with circuit depth $\mathcal{O}(m)$ and $\mathcal{O}(1)$ queries to the controlled model evaluation unitary. The ancilla used to store the classification outcome is included in the ancilla count. Applying the same construction to $U^\dagger_{x,\theta}$ gives a block-encoding $U_{\sqrt{H}^\dagger}$ of

$$\sqrt{H}^\dagger = \sum_{\theta\in\mathcal{G}_m} \alpha_0^*(x,\theta)\,|\theta\rangle\,\langle\theta|\,. \tag{S5}$$

Since the loss Hamiltonian is

$$\hat{H}_{\mathcal{L}}(x,\hat{\theta}) = \sqrt{H}\,\sqrt{H}^\dagger = \sum_{\theta\in\mathcal{G}_m} |\alpha_0(x,\theta)|^2\,|\theta\rangle\,\langle\theta|\,, \tag{S6}$$

we obtain a block-encoding of $\hat{H}_{\mathcal{L}}$ by multiplying the two block-encodings, see Fig. S1. Using the standard block-encoding product rule, with the two encodings acting on compatible ancilla spaces, this multiplication requires one additional ancilla qubit. Therefore, the resulting block-encoding of the loss Hamiltonian has parameters $(1, m+5, 0)$ with overall circuit depth $\mathcal{O}(m)$ and $\mathcal{O}(1)$ queries to a controlled version of $U_{x,\theta}$.

S2.1. Extension to several simultaneously evaluated data points

The construction above can be extended to a batch of data points evaluated coherently in the same circuit. For clarity, consider first a batch of two samples

$$\mathcal{B} = \{(x_1, y_1), (x_2, y_2)\}. \tag{S7}$$

Instead of using a single model-output qubit, we introduce two output qubits, one for each data point. The coherent model-evaluation unitary now acts as

$$U_{\mathcal{B},\theta}\left(|0\rangle|0\rangle\otimes|0\rangle^{\otimes m}\right) = \sum_{\theta\in\mathcal{G}_m} |\alpha(x_1,\theta)\rangle\otimes|\alpha(x_2,\theta)\rangle\otimes|\theta\rangle, \tag{S8}$$

where

$$|\alpha(x_b,\theta)\rangle = \alpha_0(x_b,\theta)|0\rangle + \alpha_1(x_b,\theta)|1\rangle, \qquad b = 1,2. \tag{S9}$$

Equivalently,

$$U_{\mathcal{B},\theta}\left(|00\rangle\otimes|0\rangle^{\otimes m}\right) = \sum_{\theta\in\mathcal{G}_m}\sum_{z_1,z_2\in\{0,1\}} \alpha_{z_1}(x_1,\theta)\alpha_{z_2}(x_2,\theta)|z_1 z_2\rangle|\theta\rangle. \tag{S10}$$

If the labels are y_1, y_2, the amplitude associated with classifying both examples correctly is

$$\beta_{\mathcal{B}}(\theta) = \alpha_{y_1}(x_1, \theta)\alpha_{y_2}(x_2, \theta). \tag{S11}$$

By applying X gates to those output qubits whose label is 1, the correct label string $|y_1 y_2\rangle$ can be mapped to $|00\rangle$. Thus, after this relabelling, the amplitude of the state $|00\rangle$ is precisely $\beta_{\mathcal{B}}(\theta)$.

Using the same amplitude-to-diagonal conversion as in the single-sample case, query access to $U_{\mathcal{B},\theta}$ gives a block-encoding of the diagonal operator

$$\sqrt{H}_{\mathcal{B}} = \sum_{\theta \in \mathcal{G}_m} \beta_{\mathcal{B}}(\theta)|\theta\rangle\langle\theta|. \tag{S12}$$

Multiplying this block-encoding by its adjoint gives

$$H_{\mathcal{B}}^{\text{corr}} = \sqrt{H}_{\mathcal{B}}\sqrt{H}_{\mathcal{B}}^{\dagger} = \sum_{\theta \in \mathcal{G}_m} |\beta_{\mathcal{B}}(\theta)|^2|\theta\rangle\langle\theta|. \tag{S13}$$

Since

$$|\beta_{\mathcal{B}}(\theta)|^2 = |\alpha_{y_1}(x_1, \theta)|^2|\alpha_{y_2}(x_2, \theta)|^2, \tag{S14}$$

this operator encodes the probability that both samples in the batch are classified correctly:

$$H_{\mathcal{B}}^{\text{corr}} = \sum_{\theta \in \mathcal{G}_m} p_{y_1}(x_1, \theta)p_{y_2}(x_2, \theta)|\theta\rangle\langle\theta|. \tag{S15}$$

Therefore, a natural joint batch loss is

$$\hat{H}_{\mathcal{L}}^{\text{joint}} = I - H_{\mathcal{B}}^{\text{corr}} = \sum_{\theta \in \mathcal{G}_m} \left[1 - p_{y_1}(x_1, \theta)p_{y_2}(x_2, \theta)\right] |\theta\rangle\langle\theta|. \tag{S16}$$

This construction is a direct generalization of the single-sample loss. However, it should be noted that it corresponds to a product-type batch objective, not to the usual empirical mean loss.

S2.2. Block-encoding different cost functions

The construction of the loss Hamiltonian can be generalized to a broad class of cost functions by observing that, in the CQL framework, the objective function always appears as a diagonal operator on the parameter register,

$$\hat{H}_{\mathcal{L}} = \sum_{\theta \in \mathcal{G}_m} \mathcal{L}(\theta)|\theta\rangle\langle\theta|. \tag{S17}$$

Thus, constructing a block-encoding of a cost function amounts to constructing a block-encoding of a diagonal matrix whose entries are the loss values over the discretized parameter grid.

The basic primitive is the coherent model-evaluation unitary

$$U_{x,\theta}\big(|0\rangle \otimes |0\rangle^{\otimes m}\big) = \sum_{\theta \in \mathcal{G}_m} \left[\alpha_0(x, \theta)|0\rangle|\theta\rangle + \alpha_1(x, \theta)|1\rangle|\theta\rangle\right], \tag{S18}$$

which evaluates the model simultaneously for all parameter values θ. For a binary classification task with label y, the relevant probability is

$$p_y(x, \theta) = |\alpha_y(x, \theta)|^2. \tag{S19}$$

Therefore, many common losses can be written as a scalar function of this probability,

$$\mathcal{L}_y(x, \theta) = F_y(p_y(x, \theta)). \tag{S20}$$

The block-encoding strategy then consists of two steps: first construct a block-encoding of the probability operator

$$\hat{P}_y = \sum_{\theta \in \mathcal{G}_m} p_y(x,\theta)|\theta\rangle\langle\theta|, \tag{S21}$$

and then apply a polynomial or arithmetic transformation that maps $p_y \mapsto F_y(p_y)$.

For example, using amplitude-to-diagonal conversion, query access to $U_{x,\theta}$ can be used to construct a block-encoding of

$$\sqrt{P}_y = \sum_{\theta \in \mathcal{G}_m} \alpha_y(x,\theta)|\theta\rangle\langle\theta|. \tag{S22}$$

Multiplying this block-encoding by its adjoint gives

$$\sqrt{P}_y \sqrt{P}_y^\dagger = \sum_{\theta \in \mathcal{G}_m} |\alpha_y(x,\theta)|^2 |\theta\rangle\langle\theta| = \hat{P}_y. \tag{S23}$$

Once $\hat{P}_y$ is block-encoded, different losses can be obtained by applying a function F_y to this diagonal operator:

$$\hat{H}_{\mathcal{L}} = F_y(\hat{P}_y) = \sum_{\theta \in \mathcal{G}_m} F_y(p_y(x,\theta))|\theta\rangle\langle\theta|. \tag{S24}$$

A first important family consists of polynomial losses. If F_y is a polynomial of degree d,

$$F_y(p) = \sum_{k=0}^{d} c_k p^k, \tag{S25}$$

then $F_y(\hat{P}_y)$ can be block-encoded using QSVT or by combining powers of the block-encoding of $\hat{P}_y$. In this case, the query complexity scales linearly with the polynomial degree:

$$N_{\text{queries}} = \mathcal{O}(d) \tag{S26}$$

uses of the block-encoding of $\hat{P}_y$ and its inverse, up to the cost of preparing the required QSVT phase sequence or LCU coefficients. This immediately covers losses such as

$$\mathcal{L}_y(x,\theta) = 1 - p_y(x,\theta), \tag{S27}$$

which has degree one, and mean-square-type losses such as

$$\mathcal{L}_y(x,\theta) = \left(1 - p_y(x,\theta)\right)^2, \tag{S28}$$

which have degree two.

More general smooth losses can be handled by polynomial approximation. Suppose F_y is bounded on the relevant interval $p \in [0,1]$, and choose a polynomial P_d such that

$$\max_{p\in[0,1]} \left| P_d(p) - \frac{F_y(p)}{\alpha_F} \right| \leq \varepsilon, \tag{S29}$$

where

$$\alpha_F \geq \max_{p\in[0,1]} |F_y(p)| \tag{S30}$$

is the block-encoding normalization. Applying QSVT to the block-encoding of $\hat{P}_y$ gives

$$P_d(\hat{P}_y) \approx \frac{F_y(\hat{P}_y)}{\alpha_F}. \tag{S31}$$

The cost is

$$N_{\text{queries}} = \mathcal{O}(d). \tag{S32}$$

If F_y is analytic on and around the interval of interest, the degree can often scale as

$$d = \mathcal{O}(\log(1/\varepsilon)), \tag{S33}$$

up to constants depending on the analytic domain. If F_y is nonsmooth or has singularities, the polynomial degree may scale polynomially in $1/\varepsilon$. For instance, a cross-entropy-type loss

$$\mathcal{L}_y(x,\theta) = -\log p_y(x,\theta) \tag{S34}$$

requires restricting the domain to

$$p_y(x,\theta) \geq \delta > 0 \tag{S35}$$

or regularizing it as

$$\mathcal{L}_y(x,\theta) = -\log(p_y(x,\theta) + \delta)\,, \tag{S36}$$

because the logarithm is singular at $p = 0$. The approximation degree then depends not only on ε, but also on the cutoff parameter δ.

Alternatively, the loss might be expressed as the expectation value of an observable O on the model evaluation qubit register. Suppose

$$\mathcal{L}(x,\theta) = \langle\psi(x,\theta)|O|\psi(x,\theta)\rangle, \tag{S37}$$

where O is a Hermitian observable with an efficient block-encoding or LCU decomposition. Since

$$|\psi(x,\theta)\rangle$$

is prepared coherently by the model-evaluation unitary, one can use a Hadamard-test-like or amplitude-to-diagonal construction to encode the diagonal operator

$$\hat{H}_{\mathcal{L}} = \sum_{\theta\in\mathcal{G}_m} \langle\psi(x,\theta)|O|\psi(x,\theta)\rangle|\theta\rangle\langle\theta|. \tag{S38}$$

If the final cost is a nonlinear function of this expectation value, the same QSVT or polynomial-approximation strategy can then be applied.

S2.3. Homogenizing Block Encodings

Suppose U_A is an $(\alpha_A, a, \varepsilon_A)$-block-encoding of A and U_B is an $(\alpha_B, b, \varepsilon_B)$-block-encoding of B. Let $m = \max(a,b)$ and pad the smaller block-encoding with dummy ancillas so that both use m ancilla qubits. Then choose $\alpha \geq \max\{\alpha_A, \alpha_B\}$. For $X \in \{A, B\}$, define $\gamma_X = \frac{\alpha_X}{\alpha}$. Adding one extra ancilla and a single-qubit unitary R_X satisfying $\langle 0|R_X|0\rangle = \gamma_X$, the unitary $\overline{U}_X = R_X \otimes \widetilde{U}_X$ satisfies

$$(\langle 0^{m+1}| \otimes I)\overline{U}_X(|0^{m+1}\rangle \otimes I) = \frac{X}{\alpha}. \tag{S39}$$

Thus, both block-encodings can be homogenized to have the same number of ancillas and the same normalization α. Once homogeneized we can sum the three terms using standard LCU.

Appendix S3: Batched CQL Algorithm

This appendix provides the core physical and mathematical details for the batched Coherent Quantum Learning CQL framework. To maintain a direct and transparent presentation, we omit implementation-specific hardware details, such as the auxiliary clock register dimension and block-encoding circuits, focusing instead on the continuous state evolution which applies uniformly across all batches. The complete operational implementation of this framework is structured as follows:

Algorithm 1: Batched Coherent Quantum Learning (CQL)

Input : Dataset $\mathcal{D}$ split into B batches $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_B\}$;
Total training epochs E;
Evolution time per batch τ.
Output : Optimized parameter state $|\Psi_{\text{final}}\rangle$.

`// 1. Setup Initial Search Space`
if *no prior information available* **then**
  Initialize parameter state $|\Psi\rangle$ as a flat, uniform superposition across all possible parameter options θ;
else
  Initialize parameter state $|\Psi\rangle$ as a localized Gaussian wavepacket centered at an informed initial guess θ_0;
;
`// 2. Main Training Loops`
for *epoch* $e = 1$ **to** E **do**
  Shuffle dataset batches;
  for *batch* $b = 1$ **to** B **do**
    Read current data samples from batch $\mathcal{B}_b$;
    Map the batch's cost function into the physical system as a loss landscape operator $\hat{H}_{\mathcal{L}}(\mathcal{B}_b)$;
    Combine Kinetic and Loss forces into the total batch Hamiltonian $\hat{H}^{(b)}_{\text{CQL}}(t)$;
    `// Let the physics naturally update the parameters`
    Evolve the quantum state continuously for duration τ:
    $|\Psi\rangle \leftarrow \text{Evolve}\left(|\Psi\rangle, \hat{H}^{(b)}_{\text{CQL}}, \tau\right)$;
    Note: The resulting state $|\Psi\rangle$ is passed directly as the starting point for the next batch (No classical measurement intermediary!);

$|\Psi_{\text{final}}\rangle \leftarrow |\Psi\rangle$;
return $|\Psi_{final}\rangle$

The training dataset $\mathcal{D}$ is partitioned into B mini-batches $\{\mathcal{B}_1, \ldots, \mathcal{B}_B\}$. The continuous model parameters are mapped onto an m-qubit discrete coordinate grid $\mathcal{G}_m$ containing 2^m states. For each batch $\mathcal{B}_b$, we define a diagonal loss Hamiltonian representing the potential energy landscape of that specific batch:

$$\hat{H}_{\mathcal{L}}(\mathcal{B}_b) = \sum_{\theta \in \mathcal{G}_m} \mathcal{L}(\mathcal{B}_b, \theta) \, |\theta\rangle \langle\theta| \tag{S1}$$

where $\mathcal{L}(\mathcal{B}_b, \theta)$ is the batch loss evaluated at coordinate θ.

The training process is driven by a time-dependent physical system combining kinetic exploration $\hat{K}$ and batch-specific potential attraction $\hat{H}_{\mathcal{L}}(\mathcal{B}_b)$. For any batch b, the system evolves under the Hamiltonian:

$$\hat{H}^{(b)}_{\text{CQL}}(t) = e^{\varphi(t)} \hat{K} + e^{\chi(t)} \hat{H}_{\mathcal{L}}(\mathcal{B}_b) \tag{S2}$$

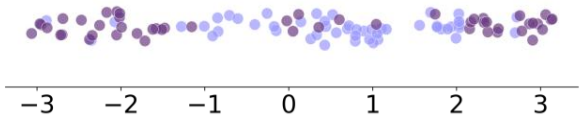
The quantum state vector is propagated sequentially from one batch to the next. The final state of batch $b-1$ serves directly as the initial state for batch b:

$$|\Psi_b\rangle = \mathcal{T} \exp\left(-i \int_0^{\tau} \hat{H}^{(b)}_{\text{CQL}}(t) \, dt\right) |\Psi_{b-1}\rangle \tag{S3}$$

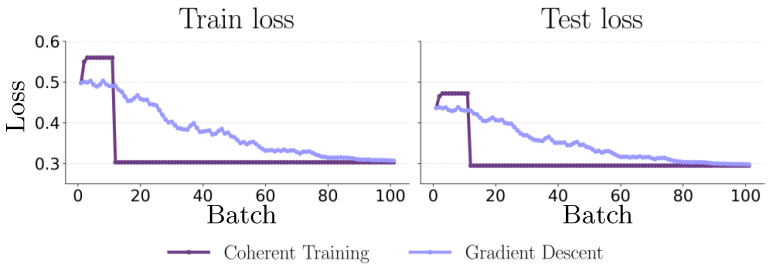
Because the evolution time τ per batch is finite and the damping ratio $e^{\varphi(t)}/e^{\chi(t)}$ does not collapse to zero within a single batch, the wavefunction retains a non-zero kinetic dispersion. This physical mechanism introduces an implicit regularization, analogous to a classical learning rate, preventing the state from freezing into a single batch's local minimum, thereby avoiding catastrophic forgetting across the global training sequence.

Appendix S4: Details on Quantum Classification with a Data Re-Uploading Model

In this section, we provide additional details on the numerical results presented in Fig. 3. We consider a single-qubit data re-uploading model with two layers, resulting in two trainable parameters. The parameter space is discretized



(a) Dataset employed in the numerical simulations. The task is a one-dimensional binary classification problem in which samples lying inside the interval $\left(-\frac{2\pi}{3}, \frac{\pi}{3}\right)$ are assigned to one class, while the remaining samples belong to the other class. Noise is added to the data. The training and test sets contain 100 and 500 samples, respectively.



(b) Figure comparing the train and test loss evolution during training for a traditional hybrid gradient descent method (lavender) and the proposed CQL method (purple).

FIG. S2. Numerical simulations and training performance for the proposed CQL method.

using five qubits per parameter, yielding a grid of 2^5 points over the interval $(-\pi, \pi)$, corresponding to a resolution of $\frac{2\pi}{2^5} \simeq 0.20$ radians. The dataset, shown in Fig. S2(a), consists of a one-dimensional binary classification task in which the objective is to determine whether a data point lies inside the interval $\left(-\frac{\pi}{3}, \frac{2\pi}{3}\right)$ or outside it. To avoid trivial separability, additive noise is introduced by perturbing the input data, thereby broadening the class boundary. The training and test sets contain 100 and 500 samples, respectively; the larger test set is used to obtain more reliable performance estimates, as evaluation is computationally inexpensive compared to training.

The learning dynamics are governed by the damping functions

$$\varphi(t) = \frac{1}{t^3}, \qquad \chi(t) = t^3,$$

which control the effective learning rate and convergence behavior. The total evolution time is set to 10^{-2}, s, chosen heuristically to balance learning effectiveness and robustness to outliers. The initial state of $\hat{\theta}$ is taken as a uniform superposition over all discretized parameter values, reflecting the absence of prior information. For comparison, the same model is also trained using a classical optimizer, namely stochastic gradient descent with learning rate 0.1 and random parameter initialization.

Figure S2(b) shows the evolution of both training and test losses during sequential single-sample updates induced by the CQL procedure. A conventional hybrid optimization scheme based on stochastic gradient descent, using the same batch size of one, is included for comparison. The loss rapidly converges—within approximately 15 updates—to a regime where the dominant minima are correctly identified, and the most probable sampled parameters correspond to near-optimal values. Beyond this point, further improvements in the loss are marginal. In contrast, the probability distribution over parameters exhibits a richer dynamical behavior, continuing to evolve after loss convergence. In particular, it progressively refines its structure and captures finer features of the landscape, thereby providing a more informative representation of the learning process than the loss alone.

Appendix S5: Details on Interferometric Phase Learning

S5.1. Mach–Zehnder Interferometer State Evolution

To rigorously model the evolution of a single photon injected into the horizontal input port from the left, we map the lower and upper spatial paths onto the standard single-qubit state space $\mathbb{C}^2$ where $|0\rangle = \begin{pmatrix}1 & 0\end{pmatrix}^T$ and $|1\rangle = \begin{pmatrix}0 & 1\end{pmatrix}^T$.

A symmetric 50:50 beam splitter (BS) is characterized by the unitary operation:

$$BS = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix},$$

where the off-diagonal imaginary component i captures the relative 90° phase shift acquired upon reflection. The dual-path phase shifting stage is represented by the diagonal unitary operation $P = \mathrm{diag}(e^{i\theta}, e^{i\phi})$.Beginning in the initial lower-path state $|\psi_i\rangle = |0\rangle$, the complete unitary evolution through the Mach–Zehnder interferometer sequence yields the final output state vector before measurement:

$$|\psi_f\rangle = BS\ P\ BS|0\rangle \& = \frac{1}{2} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{i\phi} \end{pmatrix} \begin{pmatrix} 1 \\ i \end{pmatrix} \& = \frac{1}{2} \begin{pmatrix} e^{i\theta} - e^{i\phi} \\ i(e^{i\theta} + e^{i\phi}) \end{pmatrix}.$$

By factoring out the common phase factor $ie^{i\frac{\phi+\theta}{2}}$ and exploiting Euler's trigonometric identities, the state vector can be compactly rewritten as a superposition of the computational basis kets:

$$|\psi_f\rangle = ie^{i\frac{\phi+\theta}{2}} \left[-\sin\left(\frac{\phi-\theta}{2}\right)|0\rangle + \cos\left(\frac{\phi-\theta}{2}\right)|1\rangle \right].$$

The projection probabilities $p(0) = |\langle 0|\psi_f\rangle|^2$ and $p(1) = |\langle 1|\psi_f\rangle|^2$ follow immediately via Born's rule, yielding the standard interference formulas utilized in the main text.

S5.2. Numerical Implementation and Simulation Framework

The numerical evaluation of the path-encoded phase estimation framework under CQL is implemented by mapping the continuous parameter space $\theta \in [-\pi, \pi]$ onto a discrete computational grid. For an n-qubit representation ($n = 5$), the coordinate space is uniformly partitioned into $2^n = 32$ discrete points, while the temporal domain over a total duration $T = 5.0$ is sampled via $N = 1000$ discrete time slices, defining a temporal resolution of $\delta t = 5 \times 10^{-3}$. The target state vector corresponding to the unknown phase shift ϕ is initialized within the simulator using the equivalent two-level projection $|\psi_{\text{target}}\rangle = \begin{pmatrix} \cos(\phi/2) & -i\sin(\phi/2) \end{pmatrix}^T$, which governs the physical loss landscape $\mathcal{L}(\theta) = 1 - |\langle\psi_{\text{target}}|\psi(\theta)\rangle|^2$. For the time-dependent schedules, we employ $\phi(t) = 1/t$ for the kinetic term and $\chi(t) = t$ for the potential contribution, which satisfy the requirements of the general QHD and CQL protocols.