

Beyond optimization: Harnessing quantum annealer dynamics for machine learning

Akitada Sakurai^{1,*}, Aoi Hayashi¹, Tadayoshi Matsumori², Daisuke Kaji³, Tadashi Kadowaki^{4,3}, and Kae Nemoto^{1,†}

¹*Okinawa Institute of Science and Technology Graduate University, Onna-son, Okinawa 904-0495, Japan*

²*Toyota Central R&D Labs., Inc., Yokomichi 41-1, Nagakute, Aichi, Japan*

³*DENSO Corporation, 1-8-15, Kounan, Minato-ku, Tokyo 108-0075, Japan*

⁴*Global Research and Development Center for Business by Quantum-AI Technology (G-QuAT), National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan*



(Received 23 December 2025; accepted 10 July 2026; published 10 August 2026)

Quantum annealing is typically regarded as a tool for combinatorial optimization, but its coherent dynamics also offer potential for machine learning. We present a model that encodes classical data into an Ising Hamiltonian, evolves it on a quantum annealer, and uses the resulting probability distributions as feature maps for classification. Experiments on the quantum annealer machine with the Digits dataset, together with simulations on MNIST, demonstrate that short annealing times yield higher classification accuracy, while longer times reduce accuracy but lower sampling costs. We introduce the participation ratio as a measure of the effective support of the output probability distribution and show its strong correlation with generalization.

DOI: [10.1103/revc-blqy](https://doi.org/10.1103/revc-blqy)

The quantum annealer (QA) was originally conceived as a heuristic framework for tackling combinatorial optimization problems by exploiting quantum fluctuations [1–7]. The basic principle is to embed an optimization problem into an Ising Hamiltonian, initialize the system in a simple ground state, and then slowly evolve the Hamiltonian so that the system remains in or near the ground state. Over the last decade, this approach has motivated the development of large-scale QA hardware, most prominently the D-Wave systems, which currently feature thousands of programmable qubits [8,9]. However, the same coherent processes that enable ground-state search also constitute a controllable quantum dynamical resource.

Since QA can be regarded as a specialized form of adiabatic quantum computation [2–7], its scope has steadily broadened [10]. Recent studies have investigated its role in simulating nonequilibrium quantum dynamics [11,12], probing quantum phase transitions, and supporting machine learning (ML) tasks [13,14]. In particular, annealers have been applied as Boltzmann samplers [15–18], as approximate solvers for combinatorial formulations of ML problems [19,20]. These developments reflect a shift in perspective: QAs are increasingly recognized not only as heuristic optimizers but also as programmable dynamical systems capable of processing information in ways not easily reproduced by classical algorithms.

In this article, we build on this perspective by proposing an ML framework that directly exploits the coherent dynamics of a QA as a source of expressive quantum features [21–30], in which N_f features can be prepared by only $\log_2(N_f)$ qubits. Instead of restricting the device to optimization or sampling roles, such as the optimization for feature extraction [31] in ML, we encode classical input data into the Ising Hamiltonian of the annealer, allow the system to evolve under the QA dynamics, and use the resulting output probability distributions as feature vectors for classification. In contrast to standard random-feature constructions, where the feature map is fixed analytically, here it is generated by controllable quantum dynamics, whose structure (e.g., localization and correlations) can be tuned via the annealing time. Unlike prior QA-based approaches that use sampling or optimization, we exploit the coherent dynamics as a tunable feature map. This viewpoint highlights the annealer as a quantum feature generator, expanding its role beyond optimization and sampling.

The significance of this approach is threefold. First, we demonstrate its feasibility on real hardware by performing 8×8 -handwritten digit classification [32,33] using the D-Wave Advantage System 7.1 [34,35] (hereinafter referred to as QA machine), confirming that meaningful features can be extracted under realistic experimental conditions. Second, we extend the method to the MNIST dataset [36], revealing a systematic dependence of model performance on the annealing time: Short-time dynamics yield superior generalization, while longer times degrade accuracy but reduce the number of required measurements (shots). Third, we introduce the participation ratio (PR) [37] and its variants as quantitative indicators of the model's effective size and show that they strongly correlate with classification accuracy, providing a principled diagnostic for model capacity and generalization. Although the feature space of the N -qubit QA is formally 2^N dimensional, its active size is controlled by the

*Contact author: akitada.sakurai@oist.jp

†Contact author: kae.nemoto@oist.jp

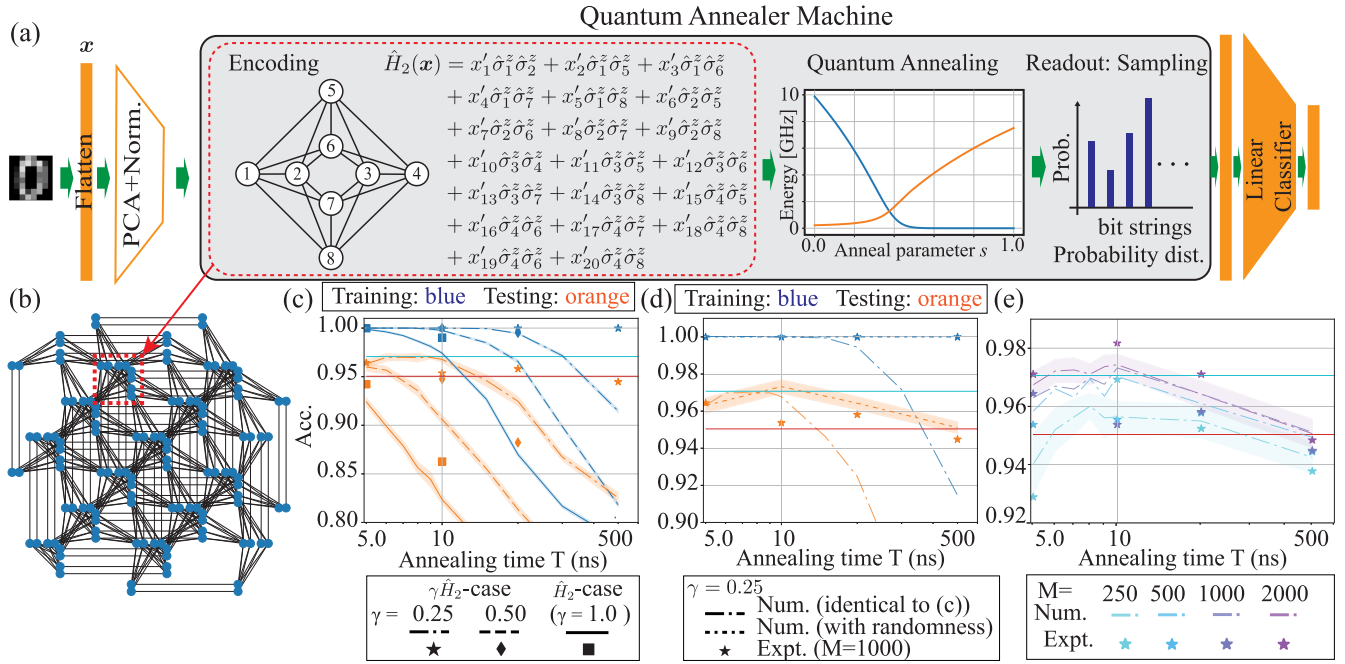


FIG. 1. Overview of the QA-based machine learning model and experimental results. (a) Schematic of our QML model implemented on a QA, e.g., the D-Wave Advantage System 7.1. Here, the PCA-based dimensionality compression is expressed as a linear transformation using the matrix W_{PCA} , where $x' = W_{\text{PCA}}x / \max |W_{\text{PCA}}x|$, and each element of this 20-dimensional vector x' is assigned to the coupling constants of the final Hamiltonian $\hat{H}_2$ of the QA. (b). Structure of the Pegasus graph and a representative subgraph used. (c) Classification accuracy as a function of annealing time for different values of the hyperparameter γ . We collected 1000 shots for each image and performed the learning procedure 10 times. The solid, dashed, and dash-dotted curves correspond to $\hat{H}_2$ and $\gamma \hat{H}_2$ cases with $\gamma = 0.25$ and 0.5 , respectively. Square, diamond, and star markers denote experimentally obtained accuracy rates for $\gamma = 0.25, 0.5$. (d) Accuracy rates for $\gamma = 0.25$ in the presence of Hamiltonian randomness. For each image, the output probability distribution was averaged over 100 random realizations of the Hamiltonian. Using these averaged probabilities, we simulated measurements with 1000 shots. The blue and orange dash-dotted lines show the averaged training and testing accuracy, respectively, each averaged over ten independent measurement simulations. The colored areas indicate the corresponding standard deviations. (e) Testing accuracy vs number of shots for simulation and experiment. Testing accuracy rates obtained from both numerical simulations and experiments are plotted as a function of the number of measurement shots M . Star-shaped points denote the experimentally measured testing accuracy. Dash-dotted curves correspond to the accuracy obtained from numerical simulations including Hamiltonian randomness. The curves represent averages, while the shaded areas indicate the associated standard deviations, shown here explicitly only for $M = 250$ and $M = 2000$. In panels (c)–(e), the blue and red solid lines show the training and testing performances of the linear model (without QA).

PR via the annealing time and remains substantially smaller in practice.

We begin by exploring the feasibility of our quantum machine learning (QML) model implemented on the QA machine. The QA performs the time evolution of the transverse-field Ising Hamiltonian $\hat{H}_{\text{Ising}}(s) = -\frac{A(s)}{2}\hat{H}_1 + \frac{B(s)}{2}\hat{H}_2$, where $\hat{H}_1 = \sum_{l=1}^N \sigma_l^x$ is the initial transverse magnetic field Hamiltonian and $\hat{H}_2 = \sum_{l>m}^N J_{lm} \hat{\sigma}_l^z \hat{\sigma}_m^z + \sum_{l=1}^N h_l \hat{\sigma}_l^z$ the final problem Hamiltonian. The coefficients J_{lm} and h_l represent the two-qubit coupling strengths and local longitudinal magnetic fields, respectively. The annealing schedule is defined by the functions $A(s)$ and $B(s)$. To achieve quantum annealing within a short, coherence-preserving timescale, we used the fast-anneal mode provided by D-Wave [35,38]. This restricts the use of the longitudinal magnetic field (h_l) and allows only the two-body interaction terms (J_{lm}). The coupling constants are constrained to the range $J_{lm} \in [-1, 1]$. In our numerical calculations, the functional forms of $A(s)$ and $B(s)$ in fast-anneal mode are based on data from Ref. [35].

The time evolution of quantum annealing encodes the input parameters x into the system's state, forming the basis of our feature-mapping procedure. By measuring the final state $|\phi(T)\rangle$ with the annealing time T in the computational basis, we obtain the output probability distribution $\mathbf{P}(x)$ corresponding to each bit string:

$$\mathbf{P}(x) = (|\langle 00 \cdots 0 | \phi \rangle|^2, \dots, |\langle 11 \cdots 1 | \phi \rangle|^2)^\top. \quad (1)$$

This distribution is reconstructed by performing multiple measurements (shots), and the resulting probability vector is treated as a classical feature representation of the input x . To ensure comparability across samples, we standardize the obtained output distributions based on the training data [32], applying the same transformation to the test data. This vector is the classical feature distribution $\mathbf{u}(x)$ of the input x to be input to the linear classifier. More details on the workflow of our QA-based ML are provided in Appendix C.

For the experimental demonstration, we implemented the QML model on the QA machine, shown schematically in Fig. 1(a). This device features 5554 qubits connected in a

topology known as the Pegasus graph. We use a subgraph of eight qubits [see Fig. 1(a)]. As a benchmark dataset, we employed the Digits dataset provided by `scikit-learn`, consisting of 8×8 grayscale images of handwritten digits. There are 1797 images categorized into 10 classes (0–9). Of these, 1347 images are used for training and 450 for testing. We apply principal component analysis (PCA) to compress each image representation into a 20-dimensional feature vector. Since PCA is a linear transformation, it does not improve the performance of a linear classifier on its own [39]. Each component of this reduced vector is then encoded into the coupling constants of $\hat{H}_2$, of the QA; see Fig. 1(a) and Appendix A.

In the experiment, the annealing time T plays a crucial role in determining the system's performance [see Fig. 1(c)]. Due to experimental constraints, the minimum accessible annealing time is approximately 5 ns, corresponding to the shortest duration available under the fast-anneal protocol. Although the training accuracy reaches relatively high values, the testing accuracy within this annealing-time regime remains below that achieved by a classical linear classifier. Increasing the annealing time generally decreases accuracy.

These observations suggest that exploring a shorter annealing-time regime, $T < 5$ ns, is essential for improved performance [35]. To effectively emulate shorter annealing times within the hardware's operational limits, we rescale the energy scale of the Hamiltonian using a parameter $\gamma \in [0, 1]$, as $\hat{H}_2 \rightarrow \gamma \hat{H}_2$. This parameter γ can be experimentally controlled by adjusting the normalization of the encoded data as $\mathbf{x}' \rightarrow \gamma \mathbf{x}'$. For $\gamma < 1$, the overall energy scale of the annealer Hamiltonian is reduced, effectively simulating a faster annealing process.

In Fig. 1(c), we plot the numerically obtained accuracy rates as a function of the annealing time for various values of γ within the short-time regime ($T \leq 5$ ns). The results indicate that the curves for $\gamma = 0.25, 0.5$ exhibit similar profiles to that of $\gamma = 1.0$, but shifted along the time axis. Importantly, the testing accuracy exceeds that of the classical linear classifier within a certain annealing-time range, demonstrating the potential advantage of appropriately scaled quantum annealing.

The experimental results of the image classification task performed on the D-Wave Advantage System are presented in Fig. 1(c). As shown in Fig. 1(d), the testing accuracy surpasses that of the classical linear classifier, demonstrating the capability of the QA to generate effective feature representations even under realistic noise conditions, which is specified later.

Although the hardware noise inevitably affects the measurements, the comparison between experimental and numerical results reveals that the experimentally obtained accuracy is consistently higher than the simulated one. This behavior suggests an effective extension of the annealing timescale in the experiment compared to the idealized numerical model. To quantitatively examine this effect, we analyze the deviation between the experimental probability distribution $P(\mathbf{x})$ and that obtained from numerical simulation. Let $P^{\text{exp}}(T^{\text{exp}})$ denote the experimentally measured probability distribution with the annealing time T^{exp} for a fixed image, and $P^{\text{sim}}(T)$ denote the corresponding numerically simulated distribution

for variable T . We define the squared error $L(T) = \sum_{i=1}^{2^N} (P_i^{\text{exp}}(T^{\text{exp}}) - P_i^{\text{sim}}(T))^2$, which quantifies the difference as a function of T . More details are discussed in Appendix B.

By averaging $L(T)$ over 100 training images, we find that the value of T^* minimizing the error is systematically shorter than the actual annealing time T^{exp} used in the experiments. This indicates that the experimentally obtained probability distributions resemble those of shorter-time numerical evolutions, which explains the observed shift in the accuracy shown in Fig. 1(c).

Regardless of the specific origin of noise in the experiment, its presence does not degrade the classification accuracy; on the contrary, it appears to enhance the performance of the QML model. To investigate this noise-assisted mechanism, we introduce controlled randomness into $\hat{H}_2$. In particular, we add random perturbations uniformly sampled from the range $[-0.1, 0.1]$ to the coupling strengths J_{lm} and introduce random longitudinal fields with amplitudes drawn from the same interval, even though explicit longitudinal-field control is unavailable in the fast-anneal protocol. Figure 1(d) shows the resulting accuracy rates for $\gamma = 0.25$ when such randomness is included. The behavior of both training and testing accuracies closely matches the experimental results, suggesting that the experimentally observed performance can be partially attributed to similar stochastic effects. Although this injected randomness does not constitute an exact model of the hardware noise, the agreement indicates that our QML algorithm is intrinsically robust to moderate random fluctuations in the physical Hamiltonian.

Next, we assess the effect of the number of measurement shots on the quality of the probability distribution read out from the QA. Since the distribution does not need to be reconstructed, the goal of this analysis is to determine how few shots are sufficient to achieve reliable classification. Figure 1(e) shows the experimentally obtained testing accuracy for different numbers of shots, along with numerical simulations including the Hamiltonian randomness. As indicated by the figure, 0.5×10^3 shots per image are sufficient in the experiment to achieve testing accuracy exceeding that of a classical linear classifier. This number of shots is significantly lower than what previous models have required [39,40], highlighting one potential advantage offered by the QA.

Finally, we assess the feasibility of this QML model for larger, more practical datasets. To evaluate this numerically, we extend the model to classify the MNIST dataset, which consists of 70 000 28×28 -pixel handwritten digit images. Following standard practices, 60 000 images are used for training and 10 000 for testing.

For generality, we consider a simplified QA model given by $\hat{H}_{\text{Ising}}(t) = (1 - t/T)\hat{H}_1 + (t/T)\hat{H}_2$, which is commonly employed in quantum annealing theoretical and numerical studies. The initial Hamiltonian is $\hat{H}_1 = -\hbar \sum_{i=1}^N \sigma_i^x$, and the final Hamiltonian is taken as a nearest-neighbor Ising model: $\hat{H}_2 = \hbar \sum_{i=1}^{N-1} J_i \sigma_i^z \sigma_{i+1}^z + \hbar \sum_{i=1}^N h_i \sigma_i^z$. Unlike the fast-anneal setting of the D-Wave device, here we assume that longitudinal magnetic fields can be applied freely. As with the Digits dataset, dimensionality reduction using PCA and normalization is performed on the MNIST images. For this $\hat{H}_2$, we encode the input $\mathbf{x}'$ as follows: $\hat{H}_2(\mathbf{x}') = \hbar \sum_{i=1}^{N-1} x'_i \sigma_i^z \sigma_{i+1}^z +$

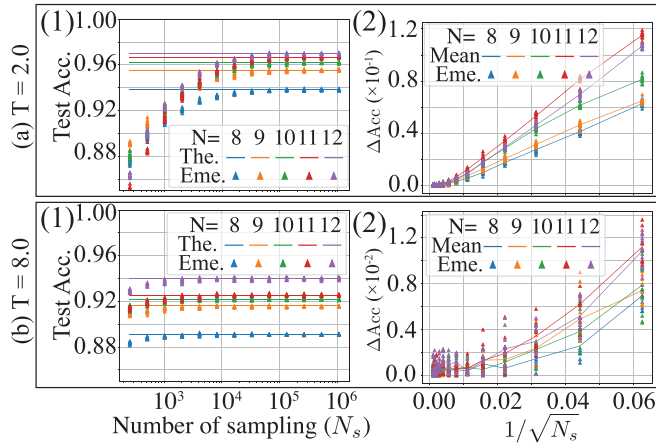


FIG. 2. Classification performance under finite-shot conditions and associated error. Classification results are shown for two annealing times: (a) $T = 2.0$ and (b) $T = 8.0$. For each case, panel (1) presents the test accuracy using the full theoretical distribution (solid lines), while panel (2) shows the absolute difference between performance based on theoretical vs estimated distributions for various numbers of qubits N . In panel (2), the vertical axis corresponds to the performance error, and the horizontal axis is $1/\sqrt{N_s}$, where N_s denotes the number of measurement shots.

$\hbar \sum_{l=1}^N x'_{N+l-1} \sigma_l^z$. This mapping preserves the structure of the classical input in the Hamiltonian couplings and fields, allowing the QA to process the higher-dimensional MNIST data.

To evaluate the performance of our model on the MNIST dataset, we analyzed the classification accuracy for different numbers of qubits ($N = 8 \sim 12$) and annealing times. High classification accuracy is observed in regions corresponding to relatively short annealing times, while performance gradually degrades as the annealing time increases. The highest performance occurs at $T = 2.0$ with the exact distribution, and this annealing time is long enough to transform the input state. By increasing the number of qubits, a higher peak classification accuracy can be achieved, as illustrated in Figs. 2(a1) and 2(b1). Figures 2(a2) and 2(b2) show the absolute error between the performance using theoretical and reconstructed distributions (vertical axis) versus $1/\sqrt{N_s}$ (horizontal axis), where N_s denotes the number of measurement shots. These results indicate that longer annealing times ($T = 8.0$) enable faster convergence to the theoretical performance with fewer shots. This behavior arises from the increased localization in the output probability distribution, which can be quantitatively confirmed through the analysis of various participation ratios.

The primary cost factors for the quantum computation are the execution time of the QA (annealing time) and the number of shots required to reconstruct the output probability distribution. The annealing time initially contributes to the learning performance, and after the learning performance is peaked, the longer annealing time the less number of shots required. These observations can be mediated by the averaged participation ratio (APR) over the training dataset ($\mathbb{D}$), $APR(\mathbb{D}) = \langle 1/\sum_{l=1}^{2^N} |\langle l|\phi \rangle|^4 \rangle_{\mathbb{D}}$, where $|l\rangle$ denotes the computational basis (or projection basis) and $|\phi\rangle$ represents

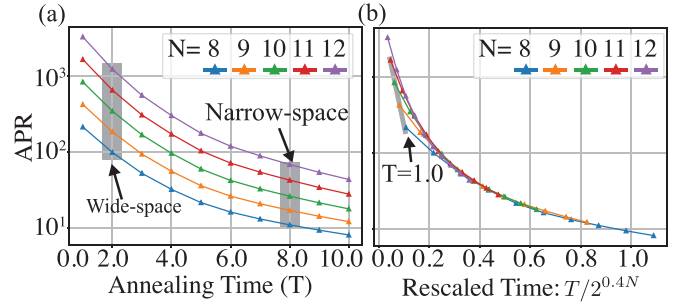


FIG. 3. Relationship between APR and annealing time for the MNIST dataset. (a) Direct relationship between APR and annealing time. The positions corresponding to $T = 2.0$ and $T = 8.0$ are highlighted in gray. (b) Relationship between APR and annealing time rescaled by the number of qubits. The result for $T = 1.0$, highlighted in gray, clearly deviates from the rest of the data, indicating that insufficient quantum dynamics occur at such short annealing times.

the quantum state after annealing. A smaller APR means a more localized state in the computational basis; thus, the APR strongly influences both the learning performance and the number of shots required for reliable sampling [39].

Figure 3(a) illustrates the connection between annealing time and APR. For a fixed annealing time, the APR increases exponentially with the number of qubits and is evenly spaced on a logarithmic scale. With a constant APR, the number of required shots may also grow exponentially as the number of qubits increases. Interestingly, as the annealing time increases, the gap between APR values for different numbers of qubits tends to shrink, indicating that the exponential growth rate of the APR may slow down at longer annealing times. The APR per Hilbert space size for $T = 2.0$ shows the evidence of the sufficient evaluation time for the initial state to be transformed significantly different. Representative output probability distributions for different annealing times are shown in Appendix D, illustrating the transition from more delocalized to more localized regimes.

Figure 3(b) presents APR as a function of the rescaled annealing time per qubit. The figure shows that the relationship between APR and rescaled annealing time ($T/2^{0.4N}$) converges toward a universal curve. This suggests that the annealing time required to reach a target APR may increase exponentially with the number of qubits. To achieve a probability distribution with the same magnitude of APR and maintain a comparable number of required shots across different qubit counts, the compression time must increase exponentially with qubit number.

To conclude, the present implementation has a computational cost that grows exponentially with the number of qubits when the full output distribution is used explicitly. At the same time, the required number of shots is controlled by the localization of the distribution, as quantified by the APR, and does not directly follow the full Hilbert-space dimension. Although further work is needed to clarify scalable implementations and the underlying mechanisms of QML with QAs, our experimental demonstration establishes the feasibility of using quantum annealing dynamics as a tunable feature map for classification.

ACKNOWLEDGMENTS

We thank W. J. Munro for the useful discussions. This work was partly funded by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) JPMXS0118069605, the JSPS KAKENHI Grant No. 21H04880, COI-NEXT JPMJPF2221, and JSPS KAKENHI under Grant No. 25K21306.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available because of legal restrictions preventing unrestricted public distribution. The data are available from the authors upon reasonable request.

APPENDIX A: ENCODING INTO THE FINAL ISING HAMILTONIAN $\hat{H}_2$

In the demonstration of our quantum-annealing-based machine learning with a real device, we used the D-Wave Advantage System 7.1 [34,35] (hereinafter referred to as the QA machine) and employed the Digits dataset provided by `scikit-learn`, consisting of 8×8 grayscale images of handwritten digits [32,33]. This device features 5554 qubits connected in a topology known as the Pegasus graph. We use a subgraph of eight qubits. To perform our calculation in the coherence-preserving time, we used the fast-anneal mode provided by D-Wave [35,38]. In this mode, we can only access the two-body interaction terms, and thus the total number of accessible terms is 20.

Each image is converted into a 64-dimensional vector $\mathbf{x}$, and it is encoded into the final Hamiltonian $\hat{H}_2$ by assigning the coupling constants to the vector with the dimensional reduction using the PCA. The PCA-based dimensionality compression is expressed as a linear transformation using the 20×64 matrix W_{PCA} . Since the coupling constant should be within the range -1 to 1 , they are normalized so that the maximum absolute value is 1. Thus, $\mathbf{x}'$ is given by

$$\mathbf{x}' = \frac{W_{\text{PCA}} \mathbf{x}}{\max |W_{\text{PCA}} \mathbf{x}|}. \quad (\text{A1})$$

Assign each element of this 20-dimensional vector $\mathbf{x}'$ to the coupling constants of $\hat{H}_2$ in the QA machine:

$$\begin{aligned} \hat{H}_2 = & x'_1 \hat{\sigma}_1^z \hat{\sigma}_2^z + x'_2 \hat{\sigma}_1^z \hat{\sigma}_5^z + x'_3 \hat{\sigma}_1^z \hat{\sigma}_6^z + x'_4 \hat{\sigma}_1^z \hat{\sigma}_7^z + x'_5 \hat{\sigma}_1^z \hat{\sigma}_8^z \\ & + x'_6 \hat{\sigma}_2^z \hat{\sigma}_5^z + x'_7 \hat{\sigma}_2^z \hat{\sigma}_6^z + x'_8 \hat{\sigma}_2^z \hat{\sigma}_7^z + x'_9 \hat{\sigma}_2^z \hat{\sigma}_8^z + x'_{10} \hat{\sigma}_3^z \hat{\sigma}_4^z \\ & + x'_{11} \hat{\sigma}_3^z \hat{\sigma}_5^z + x'_{12} \hat{\sigma}_3^z \hat{\sigma}_6^z + x'_{13} \hat{\sigma}_3^z \hat{\sigma}_7^z + x'_{14} \hat{\sigma}_3^z \hat{\sigma}_8^z + x'_{15} \hat{\sigma}_4^z \hat{\sigma}_5^z \\ & + x'_{16} \hat{\sigma}_4^z \hat{\sigma}_6^z + x'_{17} \hat{\sigma}_4^z \hat{\sigma}_7^z + x'_{18} \hat{\sigma}_4^z \hat{\sigma}_8^z + x'_{19} \hat{\sigma}_5^z \hat{\sigma}_6^z + x'_{20} \hat{\sigma}_4^z \hat{\sigma}_8^z. \end{aligned} \quad (\text{A2})$$

Because the encoded couplings can take both positive and negative values depending on the input, the resulting problem Hamiltonian is generally frustrated.

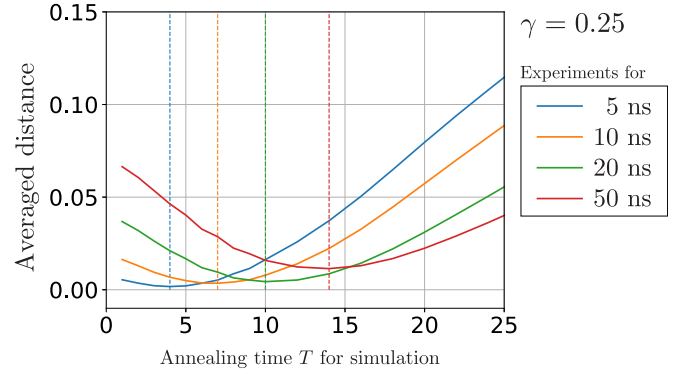


FIG. 4. The squared error $L(T)$ of Eq. (B1) between the probability distributions of the experiment and the numerical simulation as a function of the annealing time T for the simulation. The blue, orange, green, and red curves represent $L(T)$ with $T^{\text{exp.}} = 5, 10, 20$, and 50 ns and $\gamma = 0.25$. The dashed vertical lines indicate the simulation annealing time T^* for which the error $L(T^*)$ is minimum.

APPENDIX B: REPRODUCTION OF THE OBSERVED EXTENSION OF THE EFFECTIVE ANNEALING TIME DUE TO RANDOM TERMS WITHIN THE HAMILTONIAN IN EXPERIMENTS

Despite the noise in the experiment, the comparison between the experiment and the numerical calculation shows that the experimental accuracy is significantly higher. This looks as though the timescale has been extended. To quantitatively assess it, we analyze the deviation of the experiment's probability distribution from that of the numerical simulation. Let $P^{\text{exp.}}(T^{\text{exp.}})$ denote the probability distribution experimentally obtained for the annealing time $T^{\text{exp.}}$ with a fixed image. For the same image, we numerically calculate the probability distribution $P^{\text{sim.}}(T)$ for the annealing time T . We then consider the squared error

$$L(T) = \sum_{i=1}^{2^N} (P_i^{\text{exp.}}(T^{\text{exp.}}) - P_i^{\text{sim.}}(T))^2 \quad (\text{B1})$$

as a function of the annealing time T for the numerical simulation. By analyzing the squared error averaged over 100 training images, the annealing time T^* with the minimum error is shorter than the actual annealing time $T^{\text{exp.}}$ adopted in the experiments, as shown in Fig. 4. Conversely, it may be necessary to set the experimental annealing time longer in order to empirically obtain the same probability distribution as that for the annealing time T in numerical calculations.

APPENDIX C: BASIC WORKFLOW IN THE QA-BASED MACHINE LEARNING MODEL

This Appendix explains the basic workflow of our QA-based machine learning model for the MNIST dataset, as discussed in the article. In the MNIST data case, we use the Ising Hamiltonian given by

$$\hat{H}_2 = \hbar \sum_{l=1}^{N-1} J_l \sigma_l^z \sigma_{l+1}^z + \hbar \sum_{l=1}^N h_l \sigma_l^z, \quad (\text{C1})$$

where N is the total number of qubits.

For each image vector $\mathbf{x}$, the final Hamiltonian $\hat{H}_2$ is encoded by assigning J_l and h_l to the image vector. We perform PCA to reduce dimensionality. The PCA-based dimensionality compression is expressed as a linear transformation using the $(2N - 1) \times d$ matrix W_{PCA} , where d is the total number of an image's pixels. Furthermore, in order to keep the coupling coefficients within the valid range $J \in [-1, 1]$, they are normalized so that the maximum absolute value is 1:

$$\mathbf{x}' = \frac{W_{\text{PCA}}\mathbf{x}}{\max |W_{\text{PCA}}\mathbf{x}|}. \quad (\text{C2})$$

The normalized vector $\mathbf{x}'$ feeds into the quantum annealer and is encoded into the Hamiltonian $\hat{H}_2$:

$$\hat{H}_2(\mathbf{x}) = \hbar \sum_{l=1}^{N-1} x'_l \sigma_l^z \sigma_{l+1}^z + \hbar \sum_{l=1}^N x'_{N+l-1} \sigma_l^z. \quad (\text{C3})$$

Because the encoded couplings can take both positive and negative values depending on the input, the resulting problem Hamiltonian is generally frustrated. The state after the quantum annealing with annealing time T is given by

$$|\phi(T)\rangle = \mathcal{T} \exp\left(-\frac{i}{\hbar} \int_0^T \hat{H}(t, \mathbf{x}) dt\right) |\phi(0)\rangle, \quad (\text{C4})$$

where $\mathcal{T}$ denotes the time-ordered product and $|\phi(0)\rangle$ is the initial state: $|\phi(0)\rangle = \prod_{l=1}^N (|0\rangle_l + |1\rangle_l) / \sqrt{2}$, where $|0\rangle_l$ and $|1\rangle_l$ are eigenstates of $\hat{\sigma}_l^z$ for each qubit. Here, $\hat{H}(t, \mathbf{x})$ is the annealing Hamiltonian such as

$$\hat{H}(t, \mathbf{x}) = \left(1 - \frac{t}{T}\right) \hat{H}_1 + \left(\frac{t}{T}\right) \hat{H}_2(\mathbf{x}).$$

By measuring the final state $|\phi(T)\rangle$ in the computational basis, we obtain the output probability distribution $P(\mathbf{x})$ corresponding to each bit string:

$$P(\mathbf{x}) = \begin{pmatrix} |\langle 00 \cdots 0 | \phi(T) \rangle|^2 \\ |\langle 00 \cdots 1 | \phi(T) \rangle|^2 \\ \vdots \\ |\langle 11 \cdots 1 | \phi(T) \rangle|^2 \end{pmatrix}. \quad (\text{C5})$$

This distribution is estimated by multiple measurements (shots), and the resulting probability vector is treated as a classical feature vector. In machine learning, the entire process from the input $\mathbf{x}$ to the corresponding probability distribution $P(\mathbf{x})$ defines the high-dimensional nonlinear projection map. In other words, the QA process with computational measurement provides $N_f = 2^N$ features. Because of its high dimensionality and nonlinearity, the classification problem can be addressed with a linear model, as shown below. Although the feature (high-dimensional) space is formally 2^N dimensional, its active features are regulated by the PR through the annealing time and generally remain much smaller in practice.

Now, the probability distribution feeds into the linear model part. Because the task is classification, the linear model is a linear classifier. To standardize the optimization process in the linear model, we standardize the resulting output distribution using the training data (via the StandardScaler in `scikit-learn` [32]). The same transformation is applied to the test data. The standardized feature vector is denoted by $\mathbf{u}(\mathbf{x})$. Our linear model (classifier) is a simple multiclass perceptron [41], whose output is defined as

$$f(\mathbf{x}; \mathbf{W}, \mathbf{b}) = \text{Softmax}(\mathbf{W} \cdot \mathbf{u}(\mathbf{x}) + \mathbf{b}), \quad (\text{C6})$$

where $\mathbf{W}$ is a $(c \times 2^N)$ -dimensional weight matrix and $\mathbf{b}$ is a c -dimensional bias vector, where c is the total number of classes of the given task. A cross-entropy loss function is used for learning:

$$l(f(\mathbf{x}; \mathbf{W}, \mathbf{b}), \mathbf{t}) = - \sum_{i=1}^c t_i \log(f_i(\mathbf{x}; \mathbf{W}, \mathbf{b})), \quad (\text{C7})$$

where $(\mathbf{x}, \mathbf{t})$ denotes each training example and its target label [40,42]. Optimization is performed using a combination of the AdaGrad and minibatch methods to ensure stable convergence [43].

Due to the structure of our linear model, the weight matrix ($\mathbf{W}$) may require about $O(N_f = 2^N)$ memory storage. However, as seen in the main text, in regions where the annealing time is long, the PR of the state is sufficiently small, and in finite sampling, some computational bases have non-zero values. In other words, the number of active neurons in the linear model is approximately $(O(\text{PR}))$. From this, it may be possible that the weight matrix of the linear model can also be sparse, and the actual memory capacity can be somewhat limited. Additionally, the computational cost of the linear model is reduced in regions with long annealing times because the number of active neurons is effectively small. In other words, in quantum annealing-based random feature models, it is possible to concentrate information on specific computational bases by adjusting the annealing time, thereby reducing the computational cost from $(O(N_f))$ to approximately $(O(\text{PR}))$.

APPENDIX D: PROBABILITY DISTRIBUTION

We present the probability distribution at different annealing times ($T = 0.0, 2.0$, and 8.0) in Fig. 5 using the MNIST dataset. We randomly selected an input from the training data for each class and calculated its probability distribution with ten qubits. This illustrates the transition from more delocalized to more localized regimes. Since the final Hamiltonian $\hat{H}_2$ depends on the image, the localized part—namely, the ground state and lower excited states—also depends on it. As a result, the localized distribution captures and reflects the image's class information, as clearly shown in Fig. 5.

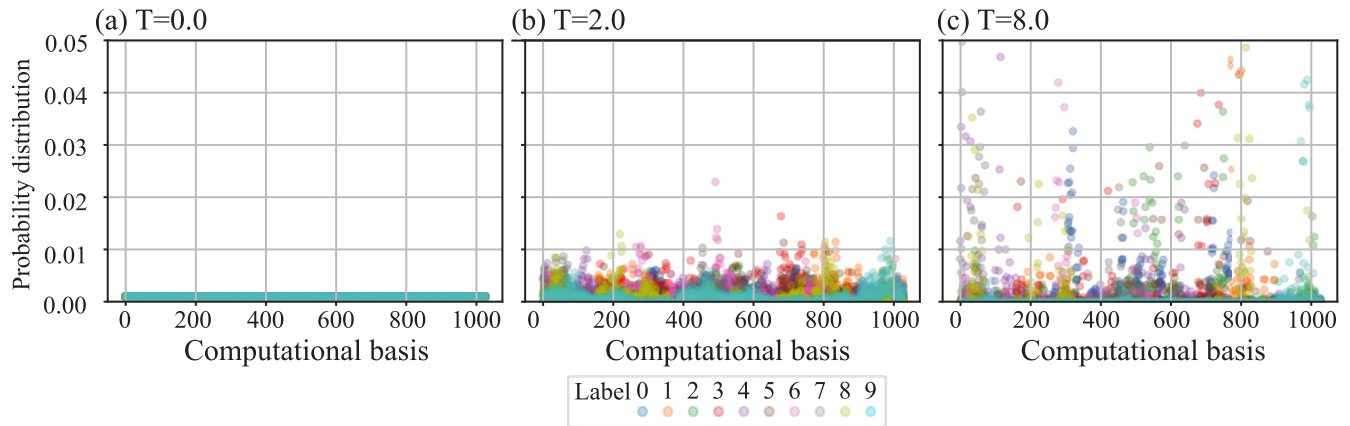


FIG. 5. Probability distribution for $T = 0.0, 2.0$, and 8.0 . In the calculation, we use a ten-qubit system and randomly select one image per class from the training dataset.

- [1] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, *Phys. Rev. E* **58**, 5355 (1998).
- [2] A. Das, B. K. Chakrabarti, and R. B. Stinchcombe, Quantum annealing in a kinetically constrained system, *Phys. Rev. E* **72**, 026701 (2005).
- [3] G. E. Santoro and E. Tosatti, Optimization using quantum mechanics: Quantum annealing through adiabatic evolution, *J. Phys. A: Math. Gen.* **39**, R393 (2006).
- [4] A. Das and B. K. Chakrabarti, *Colloquium: Quantum annealing and analog quantum computation*, *Rev. Mod. Phys.* **80**, 1061 (2008).
- [5] S. Morita and H. Nishimori, Mathematical foundation of quantum annealing, *J. Math. Phys.* **49**, 125210 (2008).
- [6] T. Albash and D. A. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [7] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, *Rep. Prog. Phys.* **83**, 054401 (2020).
- [8] C. C. McGeoch, *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*, Synthesis Lectures on Quantum Computing (SLQC) (Springer Cham, Switzerland, 2014).
- [9] D-Wave Systems Inc., D-Wave Leap Quantum Cloud Service, 2025, <https://docs.dwavesys.com/docs/latest/index.html>, accessed: 25 December 2025.
- [10] J. Wurtz, P. Lopes, N. Gemelke, A. Keesling, and S. Wang, Industry applications of neutral-atom quantum computing solving independent set problems, *arXiv:2205.08500*.
- [11] A. D. King *et al.*, Quantum critical dynamics in a 5,000-qubit programmable spin glass, *Nature (London)* **617**, 61 (2023).
- [12] A. D. King *et al.*, Beyond-classical computation in quantum simulation, *Science* **388**, 199 (2025).
- [13] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, Training a large scale classifier with the quantum adiabatic algorithm, *arXiv:0912.0779*.
- [14] A. Mott, J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu, Solving a Higgs optimization problem with quantum annealing for machine learning, *Nature (London)* **550**, 375 (2017).
- [15] D. Crawford, A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, Reinforcement learning using quantum Boltzmann machines, *Quantum Inf. Comput.* **18**, 51 (2018).
- [16] J. Liu, F. M. Spedalieri, K.-T. Yao, T. E. Potok, C. Schuman, S. Young, R. Patton, G. S. Rose, and G. Chamka, Adiabatic quantum computation applied to deep learning networks, *Entropy* **20**, 380 (2018).
- [17] T. Sato, M. Ohzeki, and K. Tanaka, Assessment of image generation by quantum annealer, *Sci. Rep.* **11**, 13523 (2021).
- [18] R. Shibukawa, R. Tamura, and K. Tsuda, Boltzmann sampling with quantum annealers via fast Stein correction, *Phys. Rev. Res.* **6**, 043050 (2024).
- [19] R. Y. Li, R. Di Felice, R. Rohs, and D. A. Lidar, Quantum annealing versus classical machine learning applied to a simplified computational biology problem, *npj Quantum Inf.* **4**, 14 (2018).
- [20] H. Salloum, K. Sabbagh, V. Savchuk, R. Lukin, O. Orabi, M. Isangulov, and M. Mazzara, Performance of quantum annealing machine learning classification models on ADMET datasets, *IEEE Access* **13**, 16263 (2025).
- [21] K. Fujii and K. Nakajima, Harnessing disordered-ensemble quantum dynamics for machine learning, *Phys. Rev. Appl.* **8**, 024030 (2017).
- [22] K. Fujii, H. Kobayashi, T. Morimae, H. Nishimura, S. Tamate, and S. Tani, Impossibility of classically simulating one-clean-qubit model with multiplicative error, *Phys. Rev. Lett.* **120**, 200502 (2018).
- [23] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, *Phys. Rev. A* **98**, 032309 (2018).
- [24] K. Fujii and K. Nakajima, Quantum reservoir computing: A reservoir approach toward quantum machine learning on near-term quantum devices, in *Reservoir Computing: Theory, Physical Implementations, and Applications*, edited by K. Nakajima and I. Fischer (Springer, Singapore, 2021), pp. 423–450.
- [25] K. Nakajima, K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, Boosting computational power through spatial multiplexing in quantum reservoir computing, *Phys. Rev. Appl.* **11**, 034021 (2019).

- [26] Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto, Natural quantum reservoir computing for temporal information processing, *Sci. Rep.* **12**, 1353 (2022).
- [27] R. A. Bravo, K. Najafi, X. Gao, and S. F. Yelin, Quantum reservoir computing using arrays of Rydberg atoms, *PRX Quantum* **3**, 030325 (2022).
- [28] P. Pfeffer, F. Heyder, and J. Schumacher, Hybrid quantum-classical reservoir computing of thermal convection flow, *Phys. Rev. Res.* **4**, 033176 (2022).
- [29] P. Mujal, R. Martínez-Peña, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Time-series quantum reservoir computing with weak and projective measurements, *npj Quantum Inf.* **9**, 16 (2023).
- [30] A. Hayashi, A. Sakurai, S. Nishio, W. J. Munro, and K. Nemoto, Impact of the form of weighted networks on the quantum extreme reservoir computation, *Phys. Rev. A* **108**, 042609 (2023).
- [31] M. Noori, S. S. Vedaie, I. Singh, D. Crawford, J. S. Oberoi, B. C. Sanders, and E. Zahedinejad, Analog-quantum feature mapping for machine-learning applications, *Phys. Rev. Appl.* **14**, 034034 (2020).
- [32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [33] E. Alpaydin and C. Kaynak, Optical recognition of handwritten digits, UCI Machine Learning Repository, 1998, <https://doi.org/10.24432/C50P49>
- [34] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, Next-generation topology of D-Wave quantum processors, [arXiv:2003.00133](https://arxiv.org/abs/2003.00133).
- [35] D-Wave Systems Inc., D-Wave documentation, 2025, https://docs.dwavequantum.com/en/latest/quantum_research/solver_properties_specific.html.
- [36] L. Deng, The MNIST database of handwritten digit images for machine learning research [Best of the Web], *IEEE Signal Process. Mag.* **29**, 141 (2012).
- [37] J. T. Edwards and D. J. Thouless, Numerical studies of localization in disordered systems, *J. Phys. C* **5**, 807 (1972).
- [38] V. Mehta, H. De Raedt, K. Michielsen, and F. Jin, Understanding the physics of D-Wave annealers: From Schrödinger to Lindblad to Markovian dynamics, *Phys. Rev. A* **112**, 032616 (2025).
- [39] A. Sakurai, A. Hayashi, W. J. Munro, and K. Nemoto, Simple Hamiltonian dynamics as a powerful resource for image classification, *Phys. Rev. A* **111**, 052432 (2025).
- [40] A. Sakurai, A. Hayashi, W. J. Munro, and K. Nemoto, Quantum optical reservoir computing powered by boson sampling, *Opt. Quantum* **3**, 238 (2025).
- [41] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. (Pearson, London, UK, 2010).
- [42] A. Sakurai, M. P. Estarellas, W. J. Munro, and K. Nemoto, Quantum extreme reservoir computation utilizing scale-free networks, *Phys. Rev. Appl.* **17**, 064044 (2022).
- [43] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.* **12**, 2121 (2011).