Probabilistic Quantum SVM with Quadratic Unconstrained Binary Optimization

Anonymous submission

Abstract

Kernel Support Vector Machines (SVMs) are widely used for their effectiveness in non-linear classification and interpretability. However, classical kernel SVMs face scalability and computational challenges. Kernel Quantum SVMs (QSVMs) address this by reformulating training as a Quadratic Unconstrained Binary Optimization problem solvable by quantum devices. Yet, existing QSVMs suffer from hyperparameter sensitivity, limited robustness, weak multi-class support, and hardware constraints. We propose **Probabilistic Quantum SVM (PQ-SVM)**, a novel framework that aggregates suboptimal solutions sampled from quantum hardware and integrates them using a free energy minimization approach rooted in information thermodynamics. This enhances robustness and generalization while requiring no additional hardware cost. Experiments on benchmark datasets show that PQ-SVM outperforms prior QSVMs by **over 40%** in some settings and achieves accuracy comparable to or better than classical kernel SVMs under hyperparameter variation. On real quantum hardware, PQ-SVM also trains **1.2× to 2×** faster than classical SVMs, demonstrating both performance and practical feasibility.

1 Introduction

While deep learning methods dominate many large-scale tasks, SVMs remain highly competitive in domains where interpretability and theoretical guarantees are essential. SVMs are widely-used supervised learning algorithms for classification tasks (Abdiansah and Wardoyo 2015; Osuna and Girosi 1998). Training an SVM involves solving a convex optimization problem to find the optimal hyperplane that separates different classes, aiming to maximize the margin between classes while minimizing empirical risk. The kernel method is beneficial for SVM in classifying categories. However, classical kernel SVMs face computational bottlenecks with complexity as data size and feature dimensions grow (Wang, Huang, and Cheng 2016). These computational limitations motivate the search for scalable alternatives.

Quantum computing has emerged as a promising technology with potential to accelerate machine learning (ML), particularly in solving optimization problems such as Quadratic Unconstrained Binary Optimization (QUBO)-based training (Laydevant, Marković, and Grollier 2024; Niazi et al. 2024; Bondarenko and Feldmann 2020; Bittel and Kliesch 2021; He et al. 2025; Lloyd, Mohseni, and Rebentrost 2014; Moro and Prati 2023).

Quantum SVMs (QSVMs) with QUBO and kernel methods have been proposed to alleviate these bottlenecks. Quantum devices offer unique advantages in solving QUBO problems (Ichikawa et al. 2024; Willsch et al. 2022; Mohseni, McMahon, and Byrnes 2022; Inagaki et al. 2016). By transforming the kernel SVM training problem into a QUBO form, these quantum devices can efficiently search for the lowest energy state and the corresponding solution.

However, existing QSVMs face several fundamental challenges in hyperparameter, robustness, and deployment.

Hyperparameter Sensitivity and Robustness: QSVMs typically require a greater number of hyperparameters for encoding than classical SVMs—usually two additional parameters (Willsch et al. 2020). Moreover, existing QSVM has insufficient robustness of the model, as slight variations or noise in the data can significantly impact performance (Willsch et al. 2020; Bhatia and Phillipson 2021; Woun et al. 2025).

Qubit Limitations: The advantages of QSVMs often rely on solving problems involving large data. Current gate-based quantum devices typically provide fewer than 100 qubits, limiting QSVMs' ability to process large datasets (Ichikawa et al. 2024).

Insufficient Experiment: Moreover, experiments are still conducted using quantum annealers, with relatively few studies leveraging platforms such as coherent ising machine (CIM) (Mohseni, McMahon, and Byrnes 2022; Inagaki et al. 2016). Also, existing studies focus on binary classification problems, with solutions for multi-class classification being relatively underexplored.

To overcome these challenges, we propose a novel framework—Probabilistic Quantum Support Vector Machine (PQ-SVM) with kernel method—that fundamentally rethinks the QSVM formulation from a free energy perspective. PQ-SVM performs **post-processing on the sampled solutions**, i.e., probabilistic fusion, which makes the **Shannon entropy higher**. It can also be regarded as an entropy regularization. Since the energy of the fusion solution may be raised to suppress the advantage brought by the entropy increase, we further explain why **the gain of fusion is universal in the Helmholtz free energy perspective**.

This approach fundamentally improved robustness and parameter perturbations even better than traditional SVM, and delivers better generalization and robustness in practice.

The main contributions of this paper are as follows:

- **Probabilistic Framework for Robust classification:** We propose a probabilistic framework that integrates an ensemble of sampled low-energy solutions using Boltzmann-weighted averaging. From the perspective of physics, it is demonstrated that the probabilistic framework can make the classification more robust.
- **Fewer Hyperparameters, More Scalable:** By employing a batch-wise QUBO strategy and abandoning the encoding parameters B, K, PQ-SVM enables efficient classification with reduced qubit overhead—lowering qubit consumption by a factor of K compared to traditional QSVMs—thereby facilitating deployment on current quantum devices under strict resource constraints.
- **Extensive Validation:** We present comprehensive experimental evidence, demonstrating that PQ-SVM achieves superior to both classical SVMs and traditional QSVMs, across diverse datasets, kernel types, and quantum platforms—including CIM hardware—showcasing its robustness, practicality, and future scalability.

2 Related Work

This section reviews related advances in the evolution of quantum SVMs (or SVM-like). There are usually two implementations of Quantum SVMs.

Gate-Model Quantum Circuits Implementations: QSVMs using quantum circuits encode data, enabling efficient kernel computation and inner product evaluation (Rebentrost, Mohseni, and Lloyd 2014; Havlíček et al. 2019; Gentinetta et al. 2024; Jäger and Krems 2023; Li et al. 2024). Practical deployment of the method is limited by the number of available qubits and the fragility of quantum states (Ichikawa et al. 2024).

QUBO-based Implementations: The QUBO formalism enables efficient mapping of SVM training to quantum hardware such as Quantum Annealer (QA) and CIM (Date, Arthur, and Pusey-Nazzaro 2021a; Tang et al. 2024; Xu et al. 2023; Chang et al. 2024), by discretizing Lagrange multipliers and recasting the problem as binary energy minimization (Nimbe, Weyori, and Adekoya 2021; Acampora 2019; Daley et al. 2022). This approach has enabled practical quantum annealing experiments on platforms like D-Wave (Willsch et al. 2020; Li et al. 2024). However, further studies (Bhatia and Phillipson 2021) have shown that QA-trained QUBO-based QSVMs remain highly sensitive to noise and hyperparameters, leading to unstable generalization.

This paper will adopt the QUBO-based implementations, taking into account its advantages in terms of qubits and versatility.

3 Preliminary

This section provides the physics background for viewing machine learning as a canonical ensemble. Furthermore, we introduce how to progressively build QSVM with QUBO from classic kernel SVMs.

3.1 Canonical-Ensemble Perspective on ML

A machine learning task with a fixed number of parameters, parameter space, and hyperparameter can be viewed as a **canonical ensemble** (Carbone et al. 2023; Hu and Huang 2023). According to the knowledge of thermodynamics and statistical physics, the micro-states of the system obey **the Boltzmann distribution**. The Boltzmann distribution is widely used in machine learning for uncertainty modeling and robust optimization (Date, Arthur, and Pusey-Nazzaro 2021b; Swanson, Williams, and Jonas 2023; Kālis et al. 2023; Midgley et al. 2024; Laidlaw and Dragan 2022), including in models like Restricted Boltzmann Machines and Hopfield Networks.

Recent developments in *information thermodynamics* (Parrondo, Horowitz, and Sagawa 2015) have reviewed the Helmholtz free energy, $F = \langle E \rangle - TS$ with average energy $\langle E \rangle$, temperature T and entropy S, as a unified measure of stability and optimality by combining energy and entropy. The decrease of F is in the isothermal process of the canonical ensemble. Model training is to find the optimal point that simultaneously reduces the energy loss ⟨E⟩ and increases the generalization entropy S, i.e., the lowest free energy point (both Bayesian posterior/ Boltzmann machine training (Niazi et al. 2024; Moro and Prati 2023; Hu and Huang 2023)).

3.2 Quantum Devices for QUBO Optimization

Various quantum devices have been developed to tackle QUBO problems.

QAs: D-Wave's quantum annealers (Headquarters 2020) implement physical quantum tunneling to search for global minima of QUBO. They are among the earliest hardware capable of natively mapping optimization in machine learning to Hamiltonians, as demonstrated in (Rajak et al. 2023; Morita and Nishimori 2008).

CIMs: Coherent Ising Machines (Xu et al. 2023; Chi 2024) leverage photonic interference and quantum coherence to realize rapid energy minimization, such as those by QBoson (Zhong et al. 2020; Langford et al. 2011; Inagaki, Inaba, and Kawamura 2017).

Gate-Model Quantum Circuits: Gate-model devices, while qubit-limited (Ichikawa et al. 2024), support quantum circuit-based algorithms like Quantum Approximate Optimization Algorithm (QAOA) (Choi and Kim 2019; Zhou et al. 2023; Qian et al. 2024). Frameworks such as Qiskit (Cross 2018) can do QAOA.

3.3 The QUBO Formulation of QSVM

The classic kernel SVM training problem seeks the optimal Lagrange multipliers α_n by maximizing the following convex objective function:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) - \sum_{n=1}^{N}\alpha_n$$

$$\text{subject to} \quad C \geq \alpha_n \geq 0, \quad \sum_{n=1}^{N}\alpha_n y_n = 0 \tag{1}$$

Where:

- α_n are the Lagrange multipliers for each sample.
- C is the penalty parameter.
- $y_n, y_m \in \{\pm 1\}$ are the labels of the samples.
- $K(\mathbf{x}_n, \mathbf{x}_m)$ is the kernel function.
- N is the total number of training samples.

To adapt this optimization for QUBO, previous works (Willsch et al. 2020; Woun et al. 2025) give each α_n as an encoded sum of binary variables:

$$\alpha_n = \sum_{k=0}^{K-1} B^k \, a_{Kn+k} \tag{2}$$

Where $a_{Kn+k} \in \{0, 1\}$ are binary variables. B and K are the base and the number of binary bits for encoding.

Substitute the representation into the energy function:

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) \times \left(\sum_{k=0}^{K-1} a_{Kn+k} \right) \times \left(\sum_{l=0}^{K-1} a_{Km+l} \right) - \sum_{n=1}^{N} \left(\sum_{k=0}^{K-1} a_{Kn+k} \right) \tag{3}$$

To incorporate the SVM constraint $\sum_{n=1}^{N} \alpha_n y_n = 0$, we introduce a penalty term to the objective energy function:

$$E_{\text{total}} = E + \xi \left(\sum_{n=1}^{N} \left(\sum_{k=0}^{K-1} a_{Kn+k} \right) y_n \right)^2 \tag{4}$$

Where ξ is a positive penalty coefficient that controls the strictness of the constraint. By minimizing the energy, the binary solution a can be obtained.

The bias term computed from $\{\alpha_n\}$ via:

$$b = \frac{\sum_{n=1}^{N} \alpha_n (C - \alpha_n) \left(y_n - \sum_{m=1}^{N} \alpha_m y_m K(\mathbf{x}_m, \mathbf{x}_n) \right)}{\sum_{n=1}^{N} \alpha_n (C - \alpha_n)}, \tag{5}$$

where y_n is the label of the sample.

However, traditional QUBO-based QSVMs search for a single binary solution with the lowest energy. Such **a solution corresponds to zero Shannon entropy, leading to poor generalization and high hyperparameter sensitivity**. Moreover, the additional encoding hyperparameters also increase tuning complexity, qubit cost, and limit scalability.

4 Methodology

What we need to do is to get a probabilistic distribution of the variable solutions rather than merely minimizing the energy. Furthermore, to overcome the qubit limitations of current quantum devices, we employ a batch processing method by randomly sampling from the original dataset and performing multiple training. We also extend the binary classification to multi-class tasks.

4.1 Why is the Probabilistic Framework More Robust, from the Free Energy Perspective?

Just as in (Hu and Huang 2023) and as previously established, an ML system can be regarded as a canonical ensemble.

Directly deploying a single discrete QUBO solution often yields a model that deviates from the original continuous optimum and is highly sensitive (Bhatia and Phillipson 2021).

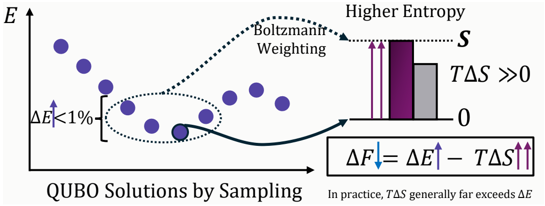
E
Higher Entropy
Boltzmann Weighting
S
$T\Delta S \gg 0$
0
ΔE<1%
$\Delta F\downarrow = \Delta E\uparrow - T\Delta S\uparrow\uparrow$
QUBO Solutions by Sampling
In practice, $T\Delta S$ generally far exceeds ΔE

Figure 1: Why Probabilistic Framework Causes F Down?

To mitigate this issue, we introduce a probabilistic framework as Figure 1: by fusing *many* low-energy states, we increase the Shannon entropy S while controlling the increase of the energy E. **Multiple sampling is the way for quantum devices** to find the optimal solution so **it does not cause additional time or call overhead**. Theoretically, by calculating ΔF, the difference in free energy, we can determine that the energy change ΔE is not significant, while the entropy increase ΔS is greater. This results in a reduction of the system's free energy, leading to a more robust SVM.

Let P_{eq} denote the canonical distribution (Boltzmann distribution) with arbitrary variable space (not necessarily binary), defined as the unique distribution that minimizes the Helmholtz free energy in the sense of the canonical ensemble. Correspondingly, the equilibrium free energy F_{eq} achieves the unique global minimum of the Helmholtz free energy, which is a constant in a given machine learning system.

Let $\Omega = \{\omega_1, ..., \omega_M\}$ denote the set of binary solutions returned by the sampler, where each state ω_i corresponds to a specific assignment of discrete variables a_j, and thus encodes a particular vector of Lagrange multipliers $\alpha(\omega_i)$. For every $\omega \in \Omega$ the QUBO energy is $E(\omega)$ as defined in Eq. 3. Each ω_i is an N-dimensional binary vector (i.e., N is the amount of data sampled per batch). We treat T as a constant of the ML system with $\beta = 1/(kT)$ and k is the Boltzmann constant.

To improve generalization beyond a single minimum-energy configuration, we construct a probabilistic solution framework. Instead of relying on a single deterministic output from quantum hardware, we collect a batch of low-energy samples $\{\omega_i\}_{i=1}^{M}$ and re-weight them according to their empirical Boltzmann probabilities. Specifically, we define two solution distributions:

- A **single-point distribution** centered at the lowest energy sample:
$$Q_{\text{single}}(\omega) = \delta_\omega, \tag{6}$$
where δ_ω places all mass on one solution returned by the hardware.

- A **mixture distribution** that captures uncertainty over multiple near-optimal solutions:

$$Q_{\text{mix}}(\omega_i) = \frac{e^{-\beta E(\omega_i)}}{\sum_{j=1}^{M} e^{-\beta E(\omega_j)}}, \tag{7}$$

where $E(\omega_i)$ is the QUBO energy of each sampled state.

By construction, Q_{mix} has strictly higher Shannon entropy than Q_{single}, since:

$$S(Q_{\text{single}}) = 0, S(Q_{\text{mix}}) = -\sum_{\omega} Q(\omega) \ln Q(\omega) > 0. \tag{8}$$

From an information-theoretic perspective, this entropy-based fusion enhances generalization.

To further ground this insight, we analyze the Helmholtz free energy of the distributions. For any distribution Q over the solution space Ω, the free energy is given by:

$$F(Q) = \langle E \rangle_Q - kTS(Q), \tag{9}$$

which can also be expressed using KL divergence:

$$F(Q) = F_{\text{eq}} + kT\, \mathrm{D}_{\mathrm{KL}}(Q \| P_{\text{eq}}), \tag{10}$$

where P_{eq} is the Boltzmann distribution that minimizes the free energy.

Applying this to Q_{single} and Q_{mix}, we compute the free energy difference:

$$\begin{aligned} \Delta F &= F(Q_{\text{mix}}) - F(Q_{\text{single}}) \\ &= kT \left[\mathrm{D}_{\mathrm{KL}}(Q_{\text{mix}} \| P_{\text{eq}}) - \mathrm{D}_{\mathrm{KL}}(Q_{\text{single}} \| P_{\text{eq}}) \right]. \end{aligned} \tag{11}$$

Since Q_{mix} more closely approximates P_{eq} than the degenerate Q_{single}, it follows that:

$$\mathrm{D}_{\mathrm{KL}}(Q_{\text{mix}} \| P_{\text{eq}}) < \mathrm{D}_{\mathrm{KL}}(Q_{\text{single}} \| P_{\text{eq}}) \implies \boxed{\Delta F < 0}. \tag{12}$$

This analysis reveals the core advantage of the probabilistic quantum SVM (PQ-SVM) framework: **the mixed-sample distribution achieves lower free energy and better approximates the true equilibrium, improving robustness and generalization over deterministic QUBO-based approaches**.

While increased entropy alone does not guarantee improved performance, our experimental results (see Experiment 5.1) show that the gain in entropy often offsets the small increase in average energy, resulting in a favorable free energy shift. Additional optimization details are discussed in Supplementary Sections 7–9.

4.2 Boltzmann Probabilistic Weighting

Previous QSVM approaches (Willsch et al. 2020; Woun et al. 2025) introduce auxiliary parameters B and K for encoding real-valued variables into binary representations, leading to a qubit requirement of $O(KN)$. In contrast, the proposed probabilistic framework eliminates the need for such auxiliary encodings. By directly using binary variables $a_n \in \{0, 1\}$, we reduce the qubit complexity to $O(N)$, achieving improved efficiency. Accordingly, we simplify the representation as $\alpha_n = a_n$.

The energy function in PQ-SVM is defined as:

$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m K(\mathbf{x}_n, \mathbf{x}_m)\, a_n a_m - \sum_{n=1}^{N} a_n, \tag{13}$$

where $K(\cdot, \cdot)$ is the kernel function.

To enforce the equality constraint $\sum_n \alpha_n y_n = 0$, we incorporate a penalty term into the total energy:

$$E_{\text{total}} = E + \xi \left(\sum_{n=1}^{N} a_n y_n \right)^2, \tag{14}$$

where ξ is a positive penalty coefficient.

Using the fused distribution Q_{mix}, the expected value of the Lagrange multiplier $\alpha_n^{'}$ is computed as:

$$\alpha_n^{'} = \mathbb{E}_{\omega \sim Q_{\text{mix}}} \left[\alpha_n(\omega) \right] = \sum_{i=1}^{M} Q_{\text{mix}}(\omega_i)\, \alpha_n(\omega_i). \tag{15}$$

The bias term b is then estimated based on $\alpha_n^{'}$ following the same procedure as described in Eq. 5.

4.3 Batch-wise Training under Qubit Budget

Current quantum hardware can accommodate only a limited number of qubits L_q. Given a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with class set $\mathcal{C} = \{1, \ldots, |\mathcal{C}|\}$, we therefore adopt a *batch-wise* protocol that keeps every individual quantum call below L_q. Following established baselines in (Hsu and Lin 2002; Rifkin and Klautau 2004), we adopt the following decomposition and select the label with the largest averaged margin. Details are provided in Supplementary Sections 4 and 10.

Binary tasks. For each class $c \in \mathcal{C}$ we have a binary dataset $\mathcal{D}^{(c)} = \{(\mathbf{x}_i, t_i^{(c)})\}$ with labels $t_i^{(c)} = +1$ if $y_i = c,\ -1$ otherwise.

Batch partition. Let $n_c = |\mathcal{D}^{(c)}|$ be the number of examples for class c after balancing. Given a user-defined batch size $N \le L_q$, we partition $\mathcal{D}^{(c)}$ into $n_b^{(c)} = \lceil n_c / N \rceil$ mini-datasets $\{\mathcal{D}_b^{(c)}\}_{b=1}^{n_b^{(c)}}$, each containing at most N samples, so that each optimization fits the qubit budget.

Per-batch sampling. Each mini-dataset $\mathcal{D}_b^{(c)}$ is mapped to a QUBO problem and solved on the chosen sampler, yielding a set of parameters $\left(\boldsymbol{\alpha}_b^{(c)},\, b_b^{(c)}\right)$.

Batch ensemble. For a test vector $\mathbf{x}$, the *batch-averaged decision score* of class c is

$$f_c(\mathbf{x}) \;=\; \frac{1}{n_b^{(c)}} \sum_{b=1}^{n_b^{(c)}} \Big[\sum_{i \in \mathcal{D}_b^{(c)}} \alpha_{b,i}^{(c)}\, t_i^{(c)}\, K(\mathbf{x}_i, \mathbf{x}) \;+\; b_b^{(c)} \Big]. \tag{16}$$

The final prediction is obtained by:

$$\hat{y}(\mathbf{x}) \;=\; \arg\max_{c \in \mathcal{C}} f_c(\mathbf{x}). \tag{17}$$

This method naturally extends the binary QSVM to multi-class **without additional qubit cost** with the scale as $\mathcal{O}(|\mathcal{C}|)$.

5 Experiments

We evaluate PQ-SVM and compare it with previous QSVMs implementations and classical SVMs.

Benchmark Datasets. We adopt six standard benchmark datasets from the `scikit-learn` library (Pedregosa et al. 2011; Buitinck et al. 2013). These datasets collectively cover low- to medium-dimensional settings, a range of sample sizes, and varying label complexities.

- **Banknote Authentication (Lohweg 2012)**: Binary classification with 1,372 samples and 4 real-valued features, representing statistical properties of banknote images.
- **Breast Cancer Wisconsin (Wolberg et al. 1993)**: Binary classification with 569 samples and 30 features, characterizing data from cell nuclei in digitized biopsies.
- **Iris (Fisher 1936)**: Dataset with 150 samples, 4 features, and 3 classes, describing measurements of iris flowers.
- **Wine (Aeberhard and Forina 1992)**: Dataset with 178 samples, 13 features, and 3 classes, containing attributes of wines from different cultivars.
- **Ecoli (Nakai 1996)**: Dataset with 336 samples, 7 features, and 8 classes, indicating protein localization sites.
- **Digits (Alpaydin and Kaynak 1998)**: Dataset with 1,797 grayscale images (0–9), each represented by a 64-dimensional vector of pixel values.
- **Epsilon (Chang and Lin 2011)**: A large-scale (400,000 training samples and 100,000 test samples), high-dimensional (2,000 features) binary dataset.

General Settings. All the experiments were conducted with an Xeon(R) Gold 6226R CPU operating at 2.90GHz and 64 GB of RAM. We conducted experiments by using a simulated quantum SDK to solve the QUBO problem (D-Wave for QA and Kaiwu for CIM) (Headquarters 2020; Xu et al. 2023; Cross 2018) and performed additional training on a CIM provided by Beijing Boson Quantum Technology Co., Ltd. Due to the computational limitations of simulating quantum circuits with Qiskit, its results are included only in Supplementary Section 1.

Default Hyperparameters. Unless otherwise noted, we use a grid of $\{C = 1, \gamma = 0.1,$ **batch size** $= 500\}$ when solving the QUBO via a simulated QA (i.e., D-Wave SDK). $\xi = 10^{-3}$ and $kT = 100$ are equivalent to scaling factors. These adjustments to hyperparameters ξ and kT do not affect the performance, which means **the impact on accuracy is less than 0.5%**, and we discuss the experiments on ξ and kT in Supplementary Section 2. Multi-class QSVM with probabilistic improvement is the result of applying an extension to the traditional QSVM.

5.1 Accuracy under Identical Hyperparameters

All datasets are split using stratified 5-fold cross-validation; results are averaged across folds and standard deviations are shown as error bars.

Figure 2 compares mean accuracy (± std dev) for four kernel functions. Our PQ-SVM not only consistently outperforms the unoptimized QSVM, but also matches or exceeds the classical SVM (Sklearn-SVC) under every kernel, and does so without any additional qubits or sampling overhead. By contrast, plain QSVM is highly unstable—its mean accuracy often collapses and variance grows large due to its sensitive encoding hyperparameters. Even after tuning the encoding parameters (B, K) as in (Willsch et al. 2020), QSVM delivers significant gains on only three of the seven datasets tested, with little or no improvement elsewhere. Moreover, classical SVM can underperform when its own hyperparameters are not exhaustively tuned—especially under the sigmoid kernel—whereas PQ-SVM maintains robust, high accuracy across all settings.

Remark on one-vs-one. We also did one-vs-one experiments on Digits with 67 runs. This is not a better choice compared to the same accuracy obtained by the original method with 10 runs. The logic applies to every pairwise class problem $\left(c_i, c_j\right)$. However the number of pairs grows quadratically, $N_{\text{pairs}} = \binom{|\mathcal{C}|}{2} = \frac{|\mathcal{C}|\left(|\mathcal{C}|-1\right)}{2}$.

Free energy decreases as entropy increases. On the Breast Cancer Wisconsin dataset, we compare the mean energy $\langle E\rangle$, system entropy S, and Helmholtz free energy F between individual and fused solution ensembles.

Before fusion, each individual solution has entropy $S = 0$, so the free energy simply equals its energy, **valued from 140.6 to 140.9**. The difference is **less than 1** so the probabilistic weight of each solution is approximately 1%. After fusing all $M = 100$ solutions, the mean energy $\langle E\rangle$ remains very close to $E_{\min}$, but the entropy rises sharply to $S \approx \ln M$, **producing an entropy term** $S \approx$ **460.5**. As a result, the free energy of the ensemble, $F = \langle E\rangle - S$, drops by more than 460 compared to any single solution. This dramatic free energy reduction, $\Delta F \approx S$, arises entirely from entropy gain. Thus, aggregating multiple sampled solutions does not heavily increase energy, but significantly boosts entropy.

This perspective lens clarifies why fusing several low-energy quantum samples—rather than taking the single “ground state”—gains a non-increasing free energy and therefore a more robust classifier.

5.2 Robustness under Varying Hyperparameters

Robustness to hyperparameters is a critical requirement for the practical deployment of machine learning models, especially in scenarios where exhaustive parameter tuning is infeasible. To this end, we comprehensively evaluate the parameter sensitivity and robustness of our PQ-SVM in comparison with standard classical SVM and baseline QSVM on the Wine dataset.

Figure 3 presents the accuracy landscapes of different models over a wide grid of regularization (C) and kernel scaling (γ) parameters on the Wine dataset. Keep ξ at the default value of 0.001 as it will not have a significant impact on performance. It is clear that PQ-SVM consistently maintains high accuracy $(\geq 97\%)$ across nearly the entire (C, γ) space. In stark contrast, both classical SVM and vanilla QSVM display significant drops in accuracy, with classical SVM frequently collapsing below 40% and QSVM exhibiting pro-

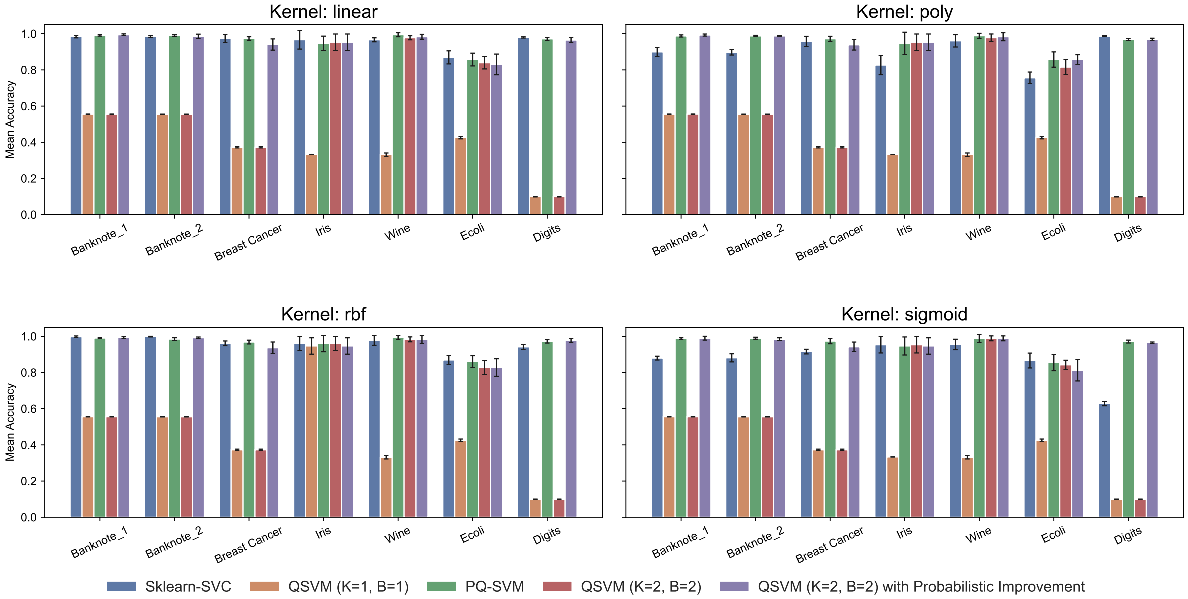

Figure 2: Performance comparison of five SVM variants on seven benchmark datasets and four kernel types.

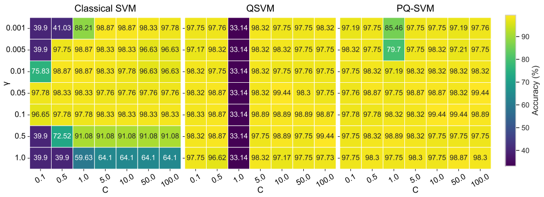

Figure 3: **Robustness under varying** $(C,\ \gamma)$**.**

nounced "dead zones" (regions with accuracy well below 85%). Notably, PQ-SVM does not require fine-tuning: it achieves near-optimal results regardless of parameter selection, demonstrating exceptional insensitivity to hyperparameters and thus extremely low tuning cost.

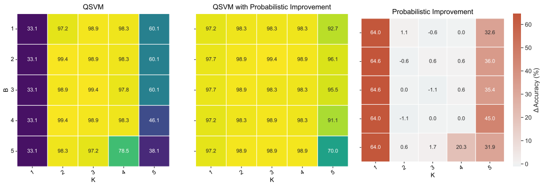

Figure 4: **Robustness under varying** $(B,\ K)$**.**

To further verify the applicability of the probabilistic method to the previous QSVM, Figure 4 visualizes the accuracy distribution on the (B, K) batch parameter space. The left and center heatmaps show the absolute accuracy (%) of QSVM and QSVM improved by our probabilistic method for different values of quantum-specific batch parameters B and K. The rightmost heatmap quantifies the improvement of our probabilistic method over QSVM (Δ accuracy, percentage points). The QSVM improved by our probabilistic method continues to deliver high accuracy ($\geq 95\%$) across almost all (B, K) settings, whereas the original QSVM suffers from catastrophic accuracy collapse, particularly for small values of B and K. The rightmost heatmap highlights the absolute improvement, showing that QSVM improved by our probabilistic method achieves up to $+60$ percentage points gain in the most challenging settings and never exhibits substantial performance loss compared to QSVM.

These results jointly demonstrate that our probabilistic method dramatically improves robustness and reliability over both classical and quantum-inspired SVMs. In practice, this means users can deploy our probabilistic method without meticulous hyperparameter search and expect consistently strong results, making it highly suitable for real-world, large-scale, and automated applications.

5.3 Evaluation on CIM Hardware

For the experiments performed on CIM, we select a new training subset consisting of 250 training and 100 test samples from Banknote 1 and 2 datasets, respectively, and train on CIM of Qboson Co. with 550 qubits. Iris dataset is fully applied. The parameters are still set to their defaults with the Radial Basis Function (RBF) kernel function.

Table 1 summarizes the results of simulation and test on CIM. Due to the reduction in sample size for the Banknote datasets, the performance of the classical SVM exhibits a certain degree of degradation.

The classical QSVM, which follows prior QA methods (Willsch et al. 2020; Woun et al. 2025), does not show a dramatic performance drop under the reduced dataset, but its baseline accuracy is already unsatisfactory. On CIM, PQ-SVM achieves best results, consistently demonstrating strong generalization across different data scales, with more than a 5% improvement over classical SVMs and over a 40%

Table 1: Performance comparison on three datasets. Boldface highlights either quantum device or best overall results. PQ-SVM on CIM achieves the highest or comparable accuracy, precision, and F1-score on all datasets, with significantly faster training.

Dataset	**Metric (%)**	**Sklearn-SVC (Classical SVM)**	**QSVM (Simulated QA)**	**PQ-SVM (Simulated QA)**	**PQ-SVM (Simulated CIM)**	**PQ-SVM (CIM)**
Banknote_1	Accuracy	90.71	54.28	94.29	96.00	**96.00**
	Precision	91.40	29.46	94.43	97.78	**97.78**
	Recall	90.71	54.28	94.29	93.62	**93.62**
	F1 Score	90.68	38.2	94.28	95.65	**95.65**
	Training Time	4.99 ms	1.19 s	1.23 s	13.94 s	**2.71 ms**
Banknote_2	Accuracy	89.29	55.74	92.86	95.00	**95.00**
	Precision	89.32	31.07	92.94	94.00	**94.00**
	Recall	89.29	55.74	92.86	91.89	**91.89**
	F1 Score	89.30	39.9	92.82	93.15	**93.15**
	Training Time	5.43 ms	1.15 s	1.19 s	15.47 s	**2.49 ms**
IRIS	Accuracy	98.33	33.33	95.00	98.33	**98.33**
	Precision	98.42	11.11	95.08	98.33	**98.42**
	Recall	98.33	33.33	95.00	98.00	**98.33**
	F1 Score	98.33	15.33	95.00	98.00	**98.33**
	Training Time	7.10 ms	2.84 s	2.94 s	17.74 s	**5.93 ms**

gain compared to classical QSVMs. For the Iris dataset, the data size remains unchanged. Both PQ-SVM on CIM hardware and classical SVM reach best accuracy, with PQ-SVM outperforming QSVM by more than 65% in accuracy. Furthermore, compared to the classical SVM, PQ-SVM on CIM hardware achieves a training speedup of approximately 2 times on Banknote dataset, and 1.2 times on the Iris dataset. This demonstrates that PQ-SVM is not only competitive in accuracy but also more efficient.

The strong consistency between the Simulated PQ-SVM performance and quantum device results further validates the reliability of the simulator and the effectiveness of simulation-based experiments.

5.4 Influence of Batch Size and Large Dataset

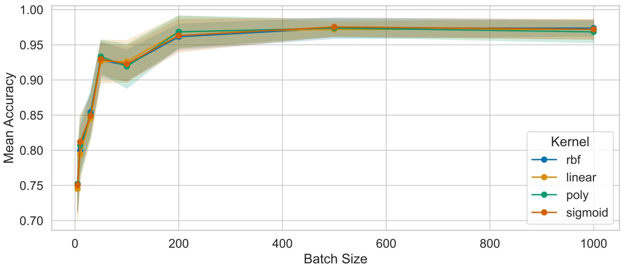

Figure 5: **Accuracy under varying batch size.**

To investigate the influence of batch size on the performance of PQ-SVM, we conduct a 5-fold cross-validation on the Breast Cancer dataset using different kernel functions. As shown in Figure 5, the batch size ranges from 5 to 1,000, and the mean accuracy along with its standard deviation is plotted for each kernel type.

We observe a clear upward trend in accuracy as the batch size increases, particularly in the small-batch regime ($\leq$ 100), where all kernels benefit substantially from larger batches. This suggests that QUBO-SVM benefits from larger optimization subspaces. Notably, the number of qubits-200, is already within the qubit capacity of quantum annealers and CIM (e.g., $\sim$500–1000 qubits) (Inagaki et al. 2016; Laydevant, Marković, and Grollier 2024; Niazi et al. 2024; Headquarters 2020), and since the qubit cost of PQ-SVM grows linearly with the sampled data size, this further confirms the practical feasibility and deployability of PQ-SVM on current quantum hardware.

Table 2: Accuracy (%) under large-scale dataset Epsilon

QSVM	LinearSVC	PQ-SVM	SVC
86.80	88.13	88.16	**88.62**

We evaluate a 40% subset of the Epsilon dataset, with batch size set to 1000. Linear kernel is used and then compared with LinearSVC. As summarized in Table 2, PQ-SVM achieves accuracy comparable to LinearSVC and kernel SVC under large-scale conditions. Details are provided in Supplementary Section 6.

6 Conclusion

We introduced a novel probabilistic framework. Experimental results on standard benchmark datasets demonstrate that PQ-SVM consistently outperforms previous QSVM, and maintains strong performance and robustness even without extensive hyperparameter tuning. Real-world evaluations using quantum hardware (i.e., CIM) further underscore the practical effectiveness and efficiency of PQ-SVM. Overall, experimental results confirm that adopting a probabilistic, entropy-aware perspective not only bridges existing quantum-classical performance gaps but also provides a robust and efficient foundation for future quantum ML.

References

Abdiansah, A.; and Wardoyo, R. 2015. Time complexity analysis of support vector machines (SVM) in LibSVM. *Int. J. Comput. Appl*, 128(3): 28–34.

Acampora, G. 2019. Quantum machine intelligence: Launching the first journal in the area of quantum artificial intelligence.

Aeberhard, S.; and Forina, M. 1992. Wine. DOI: https://doi.org/10.24432/C5PC7J.

Alpaydin, E.; and Kaynak, C. 1998. Optical Recognition of Handwritten Digits. DOI: https://doi.org/10.24432/C50P49.

Bhatia, H. S.; and Phillipson, F. 2021. Performance analysis of support vector machine implementations on the D-wave quantum annealer. In *International Conference on Computational Science*, 84–97. Springer.

Bittel, L.; and Kliesch, M. 2021. Training variational quantum algorithms is NP-hard. *Physical review letters*, 127(12): 120502.

Bondarenko, D.; and Feldmann, P. 2020. Quantum autoencoders to denoise quantum data. *Physical review letters*, 124(13): 130502.

Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; VanderPlas, J.; Joly, A.; Holt, B.; and Varoquaux, G. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122.

Carbone, D.; Hua, M.; Coste, S.; and Vanden-Eijnden, E. 2023. Efficient training of energy-based models using jarzynski equality. *Advances in Neural Information Processing Systems*, 36: 52583–52614.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3): 27:1–27:27. https://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chang, Y.-J.; Nien, C.-F.; Huang, K.-P.; Zhang, Y.-T.; Cho, C.-H.; and Chang, C.-R. 2024. Quantum Computing for Optimization With Ising Machine. *IEEE Nanotechnology Magazine*.

Chi, M. 2024. Application study of simulated annealing solver and CIM simulator based on QUBO model using kaiwu SDK. In *2024 International Conference on Electronics and Devices, Computational Science (ICEDCS)*, 902–907. IEEE.

Choi, J.; and Kim, J. 2019. A tutorial on quantum approximate optimization algorithm (QAOA): Fundamentals and applications. In *2019 international conference on information and communication technology convergence (ICTC)*, 138–142. IEEE.

Cross, A. 2018. The IBM Q experience and QISKit open-source quantum computing software. In *APS March meeting abstracts*, volume 2018, L58–003.

Daley, A. J.; Bloch, I.; Kokail, C.; Flannigan, S.; Pearson, N.; Troyer, M.; and Zoller, P. 2022. Practical quantum advantage in quantum simulation. *Nature*, 607(7920): 667–676.

Date, P.; Arthur, D.; and Pusey-Nazzaro, L. 2021a. QUBO formulations for training machine learning models. *Scientific reports*, 11(1): 10029.

Date, P.; Arthur, D.; and Pusey-Nazzaro, L. 2021b. QUBO formulations for training machine learning models. *Scientific reports*, 11(1): 10029.

Fisher, R. A. 1936. Iris. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56C76.

Gentinetta, G.; Thomsen, A.; Sutter, D.; and Woerner, S. 2024. The complexity of quantum support vector machines. *Quantum*, 8: 1225.

Havlíček, V.; Córcoles, A. D.; Temme, K.; Harrow, A. W.; Kandala, A.; Chow, J. M.; and Gambetta, J. M. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747): 209–212.

He, H.; Lin, X.; Chen, J.; and Xiao, Y. 2025. Q-Detection: A Quantum-Classical Hybrid Poisoning Attack Detection Method. *arXiv preprint arXiv:2507.06262*.

Headquarters, C. 2020. Technical Description of the D-Wave Quantum Processing Unit.

Hsu, C.-W.; and Lin, C.-J. 2002. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2): 415–425.

Hu, Z.; and Huang, H. 2023. Optimization and bayes: a trade-off for overparameterized neural networks. *Advances in Neural Information Processing Systems*, 36: 13492–13517.

Ichikawa, T.; Hakoshima, H.; Inui, K.; Ito, K.; Matsuda, R.; Mitarai, K.; Miyamoto, K.; Mizukami, W.; Mizuta, K.; Mori, T.; et al. 2024. Current numbers of qubits and their uses. *Nature Reviews Physics*, 1–3.

Inagaki, T.; Haribara, Y.; Igarashi, K.; Sonobe, T.; Tamate, S.; Honjo, T.; Marandi, A.; McMahon, P. L.; Umeki, T.; Enbutsu, K.; et al. 2016. A coherent Ising machine for 2000-node optimization problems. *Science*, 354(6312): 603–606.

Inagaki, T.; Inaba, K.; and Kawamura, R. 2017. Coherent Ising Machines—Optical Neural Networks Operating at the Quantum Limit. *npj Quantum Information*, 3: 23.

Jäger, J.; and Krems, R. V. 2023. Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines. *Nature Communications*, 14(1): 576.

Kālis, M.; Locāns, A.; Šikovs, R.; Naseri, H.; and Ambainis, A. 2023. A hybrid quantum-classical approach for inference on restricted Boltzmann machines. *Quantum Machine Intelligence*, 5(2): 44.

Laidlaw, C.; and Dragan, A. 2022. The Boltzmann Policy Distribution: Accounting for Systematic Suboptimality in Human Models. In *International Conference on Learning Representations*.

Langford, N. K.; Ramelow, S.; Prevedel, R.; Munro, W. J.; Milburn, G. J.; and Zeilinger, A. 2011. Efficient quantum computing using coherent photon conversion. *Nature*, 478(7369): 360–363.

Laydevant, J.; Marković, D.; and Grollier, J. 2024. Training an ising machine with equilibrium propagation. *Nature Communications*, 15(1): 3671.

Li, J.; Li, Y.; Song, J.; Zhang, J.; and Zhang, S. 2024. Quantum support vector machine for classifying noisy data. *IEEE Transactions on Computers*.

Lloyd, S.; Mohseni, M.; and Rebentrost, P. 2014. Quantum principal component analysis. *Nature physics*, 10(9): 631–633.

Lohweg, V. 2012. Banknote Authentication. DOI: https://doi.org/10.24432/C55P57.

Midgley, L.; Stimper, V.; Antorán, J.; Mathieu, E.; Schölkopf, B.; and Hernández-Lobato, J. M. 2024. SE (3) equivariant augmented coupling flows. *Advances in Neural Information Processing Systems*, 36.

Mohseni, N.; McMahon, P. L.; and Byrnes, T. 2022. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6): 363–379.

Morita, S.; and Nishimori, H. 2008. Mathematical foundation of quantum annealing. *Journal of Mathematical Physics*, 49(12).

Moro, L.; and Prati, E. 2023. Anomaly detection speed-up by quantum restricted Boltzmann machines. *Communications Physics*, 6(1): 269.

Nakai, K. 1996. Ecoli. DOI: https://doi.org/10.24432/C5388M.

Niazi, S.; Chowdhury, S.; Aadit, N. A.; Mohseni, M.; Qin, Y.; and Camsari, K. Y. 2024. Training deep Boltzmann networks with sparse Ising machines. *Nature Electronics*, 1–10.

Nimbe, P.; Weyori, B. A.; and Adekoya, A. F. 2021. Models in quantum computing: a systematic review. *Quantum Information Processing*, 20(2): 80.

Osuna, E.; and Girosi, F. 1998. Reducing the run-time complexity of support vector machines. In *International conference on pattern recognition (submitted)*. Citeseer.

Parrondo, J. M.; Horowitz, J. M.; and Sagawa, T. 2015. Thermodynamics of information. *Nature physics*, 11(2): 131–139.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Qian, Y.; Wang, X.; Du, Y.; Luo, Y.; and Tao, D. 2024. MG-Net: Learn to Customize QAOA with Circuit Depth Awareness. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Rajak, A.; Suzuki, S.; Dutta, A.; and Chakrabarti, B. K. 2023. Quantum annealing: An overview. *Philosophical Transactions of the Royal Society A*, 381(2241): 20210417.

Rebentrost, P.; Mohseni, M.; and Lloyd, S. 2014. Quantum support vector machine for big data classification. *Physical review letters*, 113(13): 130503.

Rifkin, R.; and Klautau, A. 2004. In Defense of One-vs-All Classification. *Journal of Machine Learning Research*, 5: 101–141. ISSN 1532-4435.

Swanson, K.; Williams, J. L.; and Jonas, E. M. 2023. Von mises mixture distributions for molecular conformation generation. In *International Conference on Machine Learning*, 33319–33342. PMLR.

Tang, L.; Yang, C.; Wen, K.; Wu, W.; and Guo, Y. 2024. Quantum computing for several AGV scheduling models. *Scientific Reports*, 14(1): 12205.

Wang, X.; Huang, F.; and Cheng, Y. 2016. Computational performance optimization of support vector machine based on support vectors. *Neurocomputing*, 211: 66–71.

Willsch, D.; Willsch, M.; De Raedt, H.; and Michielsen, K. 2020. Support vector machines on the D-Wave quantum annealer. *Computer physics communications*, 248: 107006.

Willsch, D.; Willsch, M.; Gonzalez Calaza, C. D.; Jin, F.; De Raedt, H.; Svensson, M.; and Michielsen, K. 2022. Benchmarking Advantage and D-Wave 2000Q quantum annealers with exact cover problems. *Quantum Information Processing*, 21(4): 141.

Wolberg, W.; Mangasarian, O.; Street, N.; and Street, W. 1993. Breast Cancer Wisconsin (Diagnostic). DOI: https://doi.org/10.24432/C5DW2B.

Woun, D. J.; Hamilton, K.; Coello Perez, E. A.; Chandra Shekhar, M.; Rios, F.; Gounley, J.; Suh, I.-S.; Humble, T.; Tourassi, G.; et al. 2025. Adiabatic quantum support vector machines. *Quantum Machine Intelligence*, 7(1): 1–14.

Xu, H.-Z.; Chen, J.-H.; Zhang, X.-C.; Lu, T.-E.; Gao, T.-Z.; Wen, K.; and Ma, Y. 2023. High-speed train timetable optimization based on space–time network model and quantum simulator. *Quantum Information Processing*, 22(11): 418.

Zhong, H.-S.; Wang, H.; Deng, Y.-H.; Chen, M.-C.; Peng, L.-C.; Luo, Y.-H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; et al. 2020. Quantum computational advantage using photons. *Science*, 370(6523): 1460–1463.

Zhou, Z.; Du, Y.; Tian, X.; and Tao, D. 2023. QAOA-in-QAOA: solving large-scale MaxCut problems on small quantum machines. *Physical Review Applied*, 19(2): 024027.

Reproduction Checklist

Methodology Description

- Includes a conceptual outline and/or pseudocode description of AI methods introduced. (*yes*)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results. (*yes*)
- Provides well marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper. (*yes*)

Theoretical Contributions

- Does this paper make theoretical contributions? (*yes*)
  If yes, please complete the list below.
  - All assumptions and restrictions are stated clearly and formally. (*yes*)
  - All novel claims are stated formally (e.g., in theorem statements). (*yes*)
  - Proofs of all novel claims are included. (*yes*)
  - Proof sketches or intuitions are given for complex and/or novel results. (*yes*)
  - Appropriate citations to theoretical tools used are given. (*yes*)
  - All theoretical claims are demonstrated empirically to hold. (*yes*)
  - All experimental code used to eliminate or disprove claims is included. (*yes*)

Datasets

- Does this paper rely on one or more datasets? (*yes*)
  If yes, please complete the list below.
  - A motivation is given for why the experiments are conducted on the selected datasets. (*yes*)
  - All novel datasets introduced in this paper are included in a data appendix. (*yes*)
  - All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (*yes*)
  - All datasets drawn from the existing literature are accompanied by appropriate citations. (*yes*)
  - All datasets drawn from the existing literature are publicly available. (*yes*)
  - All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (*yes*)

Computational Experiments

- Does this paper include computational experiments? (*yes*)
  If yes, please complete the list below.
  - This paper states the number and range of values tried per (hyper-)parameter during development, along with the criterion used for selecting the final parameter setting. (*yes*)
  - Any code required for pre-processing data is included in the appendix. (*yes*)
  - All source code required for conducting and analyzing the experiments is included in a code appendix. (*yes*)
  - All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (*yes*)
  - All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from. (*yes*)
  - If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (*yes*)
  - This paper specifies the computing infrastructure used for running experiments (hardware and software). (*yes*)
  - This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (*yes*)
  - This paper states the number of algorithm runs used to compute each reported result. (*yes*)
  - Analysis of experiments goes beyond single-dimensional summaries of performance to include measures of variation, confidence, or other distributional information. (*yes*)
  - The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (*yes*)
  - This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (*yes*)