PROTOTYPE-GUIDED OPTIMIZATION OF QUANTUM DATA EMBEDDINGS WITH LINEAR COST

Anonymous authors
Paper under double-blind review

ABSTRACT

In supervised quantum machine learning with classical data, classification performance critically depends on the geometry induced by quantum data embedding. Existing data-adaptive approaches rely on pairwise fidelity evaluations between samples, which scale as $O(B^2)$ for B samples and create a computational bottleneck in iterative embedding optimization. To address this issue, we propose a prototype-guided framework for efficient data-adaptive quantum embedding. Using class-prototype separation and sample-to-prototype compactness, we derive a measurable lower certificate for class-ensemble distinguishability and construct a prototype-based objective requiring only $O(B)$ fidelity evaluations. We integrate this objective into an end-to-end pipeline that combines task-specific quantum circuit architecture optimization with continuous refinement of the data-embedding parameters. Across five benchmark datasets with up to 20 qubits, our method retains competitive quantum kernel support vector machine (SVM) performance while reducing fidelity-evaluation cost, with particularly strong gains over budget-matched pairwise optimization on T-MNIST. Geometric analyses examine how the learned embeddings relate to kernel alignment and classification margins, revealing dataset-dependent benefits and limitations. We further derive a prototype-based classifier in the reproducing kernel Hilbert space that provides an explicit witness for the SVM objective and a low-cost proxy for model selection, and extend the framework to multiple prototypes for multimodal class structures.

1 INTRODUCTION

Quantum machine learning (QML) seeks to harness quantum information processing to enhance how data are represented, processed, and learned from (Biamonte et al., 2017; Schuld et al., 2015). When classical data are used in a quantum learning model, they must first be encoded into quantum states, making the choice of quantum data embedding a fundamental design decision (Schuld & Killoran, 2019; Hur et al., 2024). The embedding determines how relationships in the input space are represented in quantum state space and can therefore strongly influence the behavior of downstream learning models (Schuld & Killoran, 2019; Huang et al., 2021; Hur et al., 2024).

This role is particularly evident in supervised QML, where the objective is to learn a predictive relation between input data and their labels. In this setting, the geometry of the embedded states influences how effectively different classes can be distinguished (Lloyd et al., 2020; Hur et al., 2024; Hur & Park, 2025). This dependence appears across different quantum classification models. In quantum kernel methods, fidelities between embedded quantum states define the kernel used by a classical learner such as a support vector machine (SVM) (Schuld & Killoran, 2019; Havlíček et al., 2019), making the induced geometry directly relevant to kernel alignment and predictive performance (Huang et al., 2021; Kübler et al., 2021; Glick et al., 2024; Hubregtsen et al., 2022). In variational quantum classifiers, the embedding likewise affects expressive power (Schuld et al., 2021; Pérez-Salinas et al., 2020) and predictive performance (Hur et al., 2024; Heo & Park, 2026), trainability (Hur et al., 2024; Thanasilp et al., 2023), and generalization (Hur & Park, 2025; Caro et al., 2021; Banchi et al., 2021). Designing task-adaptive quantum embeddings is therefore a central problem in supervised QML.

The importance of quantum data embedding in supervised QML has motivated the development of various data-adaptive embedding approaches. Existing methods construct embeddings by learn-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ing classical transformations of the input values (Hur et al., 2024), optimizing trainable parameters within a predefined circuit ansatz (Lloyd et al., 2020; Hubregtsen et al., 2022), or extending the optimization to the circuit architecture and its parameterization (Heo & Park, 2026; Altares-López et al., 2021; Moretti et al., 2025; Nguyen & Chen, 2022; Lei et al., 2024). Despite their methodological differences, these approaches share the goal of inducing a quantum representation that is better aligned with the target learning task.

Many approaches characterize this task through the pairwise geometry of embedded samples. Whether assessed through supervised learning objectives, kernel–target alignment, or downstream quantum kernel SVM (QKSVM) performance, exact empirical evaluation commonly relies on pairwise state-overlap information (Hur et al., 2024; Lloyd et al., 2020; Hubregtsen et al., 2022; Altares-López et al., 2021; Moretti et al., 2025; Lei et al., 2024). Therefore, exact evaluation over B samples requires $O(B^2)$ pairwise quantum fidelity evaluations, and this cost causes computational bottleneck of the repeating optimization process. Subsampling and learned surrogate models can reduce this overhead, but rely on partial objective estimates, or on the performance of the surrogate model (Sahin et al., 2024; Lei et al., 2024).

To address this bottleneck, we propose a prototype-guided framework that replaces dense sample-pair comparisons with sample-to-prototype relations. Prototype separation and sample-to-prototype compactness provide a measurable lower certificate for class-ensemble distinguishability and motivate an objective requiring $O(B)$ fidelity evaluations for B samples with fixed number of prototypes. We further construct a prototype-based reproducing kernel Hilbert space (RKHS) (Paulsen & Raghupathi, 2016) classifier as an explicit witness for the soft-margin SVM objective (Shalev-Shwartz et al., 2007) and evaluate it as a low-cost proxy for model selection. The framework combines genetic algorithm (GA) (Katoch et al., 2020) architecture search with Neural Quantum Embedding (NQE) (Hur et al., 2024) refinement and extends to multiple prototypes for multimodal class structures. Across five benchmark datasets with up to 20 qubits, the method retains competitive QKSVM performance with lower fidelity-evaluations. And the advantage is observed on T-MNIST. Analyses of our theoretical lower bound for class-ensemble distinguishability, centered kernel-target alignment (CKTA) (Cortes et al., 2012), and sample-wise SVM margins clarify both the observed gains and the limitations of mean-based objective, while the multi-prototype extension improves classification on a multimodal benchmark.

2 Prototype-Guided Quantum Data Embedding

2.1 Quantum Data Embedding and Pairwise Optimization

Consider a binary classification problem with a labeled dataset $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where $\mathbf{x}_i \in \mathcal{X}$ denotes a classical input and $y_i \in \{+, -\}$ is its corresponding class label. Let $I_y = \{i : y_i = y\}$ denote the index set of class y. We denote m_y the number of samples belonging to class y, and $p_y = m_y/m$ be the corresponding empirical class prior. Given an n-qubit quantum data embedding circuit $U_\Phi(\mathbf{x})$, each input $\mathbf{x}_i$ is mapped to the pure quantum state $|\psi_\Phi(\mathbf{x}_i)\rangle = U_\Phi(\mathbf{x}_i)|0\rangle^{\otimes n}$, with associated density matrix $\rho_\Phi(\mathbf{x}_i) = |\psi_\Phi(\mathbf{x}_i)\rangle\langle\psi_\Phi(\mathbf{x}_i)|$.

After embedding, the samples in each class construct an empirical class-ensemble state, $\bar{\rho}_y = \frac{1}{m_y}\sum_{i:y_i=y}\rho_\Phi(\mathbf{x}_i)$. The trace norm of the prior-weighted difference,

$$\Delta_\Phi = \|p_+\bar{\rho}_+ - p_-\bar{\rho}_-\|_1 = 2D_{\mathrm{tr}}(p_+\bar{\rho}_+, p_-\bar{\rho}_-), \tag{1}$$

quantifies the distinguishability between the two class ensembles (Bae & Kwek, 2015) where $D_{\mathrm{tr}}(\cdot,\cdot)$ denotes trace distance between two quantum states. Δ_Φ has a direct operational meaning in binary quantum-state discrimination. In particular, it determines a lower bound on the empirical misclassification risk (Banchi et al., 2021; Hur et al., 2024), and upper bounds the mean classification margin associated with margin-based generalization (Hur & Park, 2025). Moreover, Δ_Φ cannot be increased by subsequent quantum processing because the trace norm is contractive under completely positive trace-preserving maps (Hur et al., 2024). This quantity is therefore an important embedding-level quantity that governs classification performance and generalization in QML.

Evaluating Δ_Φ directly is expensive because it requires spectral information about the mixed-state difference operator. This motivates the use of a Hilbert–Schmidt (HS) relaxation which expresses the class-ensemble difference in terms of within-class purities and cross-class overlap (Lloyd et al.,

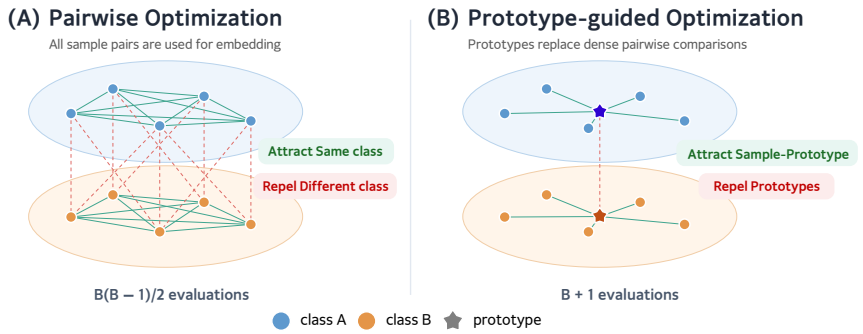

Figure 1: Comparison of pairwise and single-prototype-guided quantum embedding optimization. **(a)** Pairwise optimization promotes high fidelity for same-class sample pairs and low fidelity for cross-class pairs, requiring $B(B-1)/2$ evaluations. **(b)** The proposed method promotes high sample-to-class-prototype fidelity and low cross-class prototype fidelity, reducing the cost to $B+1$ evaluations.

2020; Hur et al., 2024). For pure-state embeddings, these quantities are reduced to pairwise fidelity kernel $k_\Phi(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Tr}[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)]$. Δ_Φ therefore admits the following lower bound.

$$\Delta_\Phi \geq \frac{1}{m}\left[\underbrace{\sum_{y\in\{+,-\}}\sum_{i,j\in I_y} k_\Phi(\mathbf{x}_i,\mathbf{x}_j)}_{\text{within-class purity terms}} - \underbrace{2\sum_{i\in I_+}\sum_{j\in I_-} k_\Phi(\mathbf{x}_i,\mathbf{x}_j)}_{\text{cross-class overlap term}}\right]^{1/2}. \tag{2}$$

The derivation is in Appendix A.1. This inequality identifies a class-discriminative geometry where same-class samples cluster while different-class samples remain well separated. The perspective provides a common geometric interpretation of existing data-adaptive quantum embedding methods. For multiclass datasets, we apply the analysis to each class pair, with m and the empirical priors defined within that pair, and report the mean pairwise class-ensemble distinguishability.

Even when the connection between this geometry and class-ensemble distinguishability is not made explicit, many data-adaptive embedding methods pursue closely related objectives. Lloyd et al. (2020) and Hur et al. (2024) explicitly connect this class-ensemble distinguishability to embedding optimization, by optimizing either its HS relaxation or a related fidelity-based objective. Other data-adaptive embedding methods do not make this connection explicit, but employ conceptually similar criteria based on quantum kernel–target alignment (Hubregtsen et al., 2022; Lei et al., 2024) or downstream QKSVM performance (Altares-López et al., 2021; Moretti et al., 2025).

For B samples, exact evaluation requires $\binom{B}{2} = \frac{B(B-1)}{2} \in O(B^2)$ distinct pairwise kernel evaluations. This cost is incurred repeatedly during embedding optimization, creating a computational bottleneck. Pair subsampling can reduce the number of evaluated kernel entries, but optimizes only a partial or stochastic estimate of the full pairwise geometry (Hur et al., 2024; Sahin et al., 2024). Learned surrogate models can further reduce direct quantum-kernel evaluations, but their effectiveness depends on the accuracy of the surrogate predictor (Lei et al., 2024). These limitations motivate an efficiently computable lower certificate for class-ensemble distinguishability.

2.2 Prototype-Based Ensemble Certificate

To construct such a certificate, we consider one prototype per class, $\{\mathbf{c}_+, \mathbf{c}_-\} \in \mathcal{X}$ with the application-dependent construction rule. The prototype quantum state of class y is defined as $\rho_y^c = \rho_\Phi(\mathbf{c}_y)$. The prototype fidelity therefore becomes $F_c = k_\Phi(\mathbf{c}_+, \mathbf{c}_-)$ and their trace distance is $D_c = D_{\mathrm{tr}}(\rho_+^c, \rho_-^c) = \sqrt{1-F_c}$ since we consider pure state embedding.

We define the mean within-class trace-distance radius to quantify within-class compactness around the prototypes after embedding. It measures the average trace distance between the embedded sam-

ples and their corresponding prototypes:

$$\bar{r}_y^{\mathrm{tr}} = \frac{1}{m_y} \sum_{i \in I_y} D_{\mathrm{tr}} \left(\rho_\Phi(\mathbf{x}_i), \rho_y^c \right) = \frac{1}{m_y} \sum_{i \in I_y} \sqrt{1 - k_\Phi(\mathbf{x}_i, \mathbf{c}_y)}. \tag{3}$$

The prototype fidelity F_c and the class-wise radii $\bar{r}_y^{\mathrm{tr}}$ respectively quantify inter-class prototype separation and within-class compactness. Together, they yield the following lower certificate for class-ensemble distinguishability Δ_Φ.

Proposition 1 (Prototype-based Ensemble Certificate)**.** *For arbitrary class prototypes* $\mathbf{c}_+$ *and* $\mathbf{c}_-$*, class-ensemble distinguishability* Δ_Φ *satisfies*

$$\|p_+ \bar{\rho}_+ - p_- \bar{\rho}_-\|_1 \geq \sqrt{1 - 4p_+p_-F_c} - 2\left(p_+ \bar{r}_+^{\mathrm{tr}} + p_- \bar{r}_-^{\mathrm{tr}}\right). \tag{4}$$

Proposition 1 shows that the distinguishability of the full class ensembles can be lower-bounded using quantities determined only by the prototypes and their associated samples. Specifically, the first term captures the separation between the two prototype states, while the second penalizes the dispersion of each class around its prototype. Thus, well-separated prototypes together with compact class-wise embeddings yield a larger lower certificate for Δ_Φ. The result follows Appendix A.2. For later use, we denote the right-hand side of Eq. 4 by the prior-weighted prototype certificate,

$$\mathcal{C}_{\mathrm{prior}} = \sqrt{1 - 4p_+p_-F_c} - 2\left(p_+ \bar{r}_+^{\mathrm{tr}} + p_- \bar{r}_-^{\mathrm{tr}}\right). \tag{5}$$

A single prototype may not adequately represent a class with multimodal structure. We thus associate each class $y \in \{+, -\}$ with a set of K_y prototypes, $\mathcal{P}_y = \{\mathbf{c}_{y,1}, \ldots, \mathbf{c}_{y,K_y}\}$. Each sample x_i is assigned to a prototype of its class according to a predefined rule, with assignment $a(i) \in \{1, \ldots, K_{y_i}\}$ fixed before optimization.

This assignment partitions each class into prototype-specific groups $I_{y,a} = \{i \in I_y : a(i) = a\}$, whose empirical proportions $\pi_{y,a} = |I_{y,a}|/m_y$ determine the contribution of each prototype state $\rho_{y,a}^c = \rho_\Phi(\mathbf{c}_{y,a})$. The resulting prototype mixture for class y is

$$\widetilde{\rho}_y^c = \sum_{a=1}^{K_y} \pi_{y,a} \rho_{y,a}^c = \frac{1}{m_y} \sum_{i \in I_y} \rho_{y,a(i)}^c. \tag{6}$$

Extending Eq. 3 to the multi-prototype setting, we define the mean trace distance between the samples of class y and their respective prototypes as

$$\bar{r}_{y,K_y}^{\mathrm{tr}} = \frac{1}{m_y} \sum_{i \in I_y} D_{\mathrm{tr}} \left(\rho_\Phi(\mathbf{x}_i), \rho_{y,a(i)}^c \right). \tag{7}$$

Using these definitions, we extend Proposition 1 to the multi-prototype setting.

Proposition 2 (Multi Prototype-based Ensemble Certificate)**.** *For the class-wise prototype sets* $\mathcal{P}_y = \{\mathbf{c}_{y,1}, \ldots, \mathbf{c}_{y,K_y}\}$*, the class-ensemble distinguishability* Δ_Φ *satisfies*

$$\|p_+ \bar{\rho}_+ - p_- \bar{\rho}_-\|_1 \geq \|p_+ \tilde{\rho}_+^c - p_- \tilde{\rho}_-^c\|_1 - 2 \sum_{y \in \{+,-\}} p_y \bar{r}_{y,K_y}^{\mathrm{tr}}. \tag{8}$$

The proof is provided in Appendix A.3.

The resulting bound provides a lower certificate for Δ_Φ based on the geometry of the prototype mixtures and the average distances between samples and their assigned prototypes. The prototype-mixture separation term can be viewed as a prototype-level counterpart of the pairwise geometry in Section 2.1. Its fidelity-based expansion favors large separation between prototypes from different classes and coherence among prototypes within the same class. The radius term further encourages each sample to remain close to its assigned prototype. For multiclass datasets, Propositions 1 and 2 are applied to each class pair, and averaging the resulting bounds yields a lower certificate for the mean pairwise class-ensemble distinguishability. These geometric conditions motivate the practical prototype-guided objective introduced next.

2.3 PROTOTYPE-GUIDED OPTIMIZATION

Motivated by Proposition 2, we construct a practical fidelity-based objective with three components: (1) sample-to-prototype compactness $\mathcal{L}_{\text{sample}}$, (2) within-class prototype coherence $\mathcal{L}_{\text{in}}$, and (3) cross-class prototype separation $\mathcal{L}_{\text{out}}$. Since directly evaluating the trace-distance quantities in the certificate is generally costly on quantum hardware, we follow prior quantum embedding approaches (Lloyd et al., 2020; Hur et al., 2024) and use fidelity as practical surrogates for the corresponding geometry.

Let $\mathcal{Y}$ be a general class set and $y, z \in \mathcal{Y}$ be index the classes. With these notations, the three components identified before are defined as

$$\mathcal{L}_{\text{sample}} = \sum_{y\in\mathcal{Y}} \sum_{i\in I_y} \left[1 - k_\Phi(\mathbf{x}_i, \mathbf{c}_{y,a(i)})\right], \tag{9}$$

$$\mathcal{L}_{\text{in}} = \sum_{y\in\mathcal{Y}} \sum_{a,a'=1}^{K_y} \left[1 - k_\Phi(\mathbf{c}_{y,a}, \mathbf{c}_{y,a'})\right], \tag{10}$$

$$\mathcal{L}_{\text{out}} = \sum_{y<z} \sum_{a=1}^{K_y} \sum_{b=1}^{K_z} k_\Phi(\mathbf{c}_{y,a}, \mathbf{c}_{z,b}). \tag{11}$$

Accordingly, decreasing $\mathcal{L}_{\text{in}}$ and $\mathcal{L}_{\text{out}}$ promotes the prototype-level geometry underlying the certificate, while $\mathcal{L}_{\text{sample}}$ provides a squared-distance surrogate for the sample-to-prototype radius penalty.

We collect the three components into the prototype-guided objective

$$\mathcal{L}_{\text{PG}}(\vartheta) = \left(\mathcal{L}_{\text{sample}}(\vartheta), \mathcal{L}_{\text{in}}(\vartheta), \mathcal{L}_{\text{out}}(\vartheta)\right), \tag{12}$$

where ϑ denotes the optimization variables that determine the quantum embedding. The formulation is agnostic to any optimization algorithm used to balance the three components. In our experiments, they are combined using a weighted sum.

Prototype assignments are determined once in the classical representation space before embedding optimization and kept fixed across candidate embeddings. After embedding, each sample is compared only with its assigned prototype so that $\mathcal{L}_{\text{sample}}$ requires B fidelity evaluations. For $\mathcal{L}_{\text{in}}$, the within-class prototype terms require $\sum_y \binom{K_y}{2}$ fidelity evaluations, while $\mathcal{L}_{\text{out}}$ requires all $\sum_{y<z} K_y K_z$ cross-class prototype pairs. Hence, the exact number of distinct fidelity evaluations is

$$N_{\text{fid}} = B + \sum_y \binom{K_y}{2} + \sum_{y<z} K_y K_z = B + \binom{K_{\text{tot}}}{2}, \quad K_{\text{tot}} = \sum_{y\in\mathcal{Y}} K_y. \tag{13}$$

Thus, the proposed objective requires $O(B + K_{\text{tot}}^2)$ fidelity evaluations. For K_{tot} fixed independently of B, this reduces to $O(B)$, compared with $O(B^2)$ for full pairwise optimization.

2.4 PROTOTYPE RKHS WITNESS FOR QUANTUM KERNEL CLASSIFICATION

Beyond improving the efficiency of embedding optimization, the prototype construction also provides a direct connection between the optimized embedding geometry and downstream classification. We make this connection explicit through the RKHS (Paulsen & Raghupathi, 2016) induced by the fidelity kernel, using the soft-margin SVM as the downstream classifier. Although the analysis below is specific to the SVM, the underlying prototype geometry is not restricted to kernel classifiers: for a fixed embedding, both kernel classifiers and variational quantum classifiers act as linear decision functions on the embedded quantum-state feature space (Schuld & Killoran, 2019).

We now return to the single-prototype setting and define the prototype-based RKHS classifier

$$h_{\text{proto}}(\mathbf{x}) = k_\Phi(\mathbf{x}, \mathbf{c}_+) - k_\Phi(\mathbf{x}, \mathbf{c}_-). \tag{14}$$

By the reproducing property, $h_{\text{proto}} \in \mathcal{H}_\Phi$, and its squared RKHS norm is $||h_{\text{proto}}||^2_{\mathcal{H}_\Phi} = 2(1 - F_c) = 2D_c^2$. For the embedding Φ, we consider the normalized soft-margin SVM objective (Shalev-Shwartz et al., 2007) with resepectly identifying the class labels $+$ and $-$ with $+1$ and -1,

$$\mathcal{J}^\star_{\text{SVM}}(\Phi) = \inf_{\substack{f\in\mathcal{H}_\Phi \\ b\in\mathbb{R}}} \left\{ \frac{\lambda_{\text{SVM}}}{2} \|f\|^2_{\mathcal{H}_\Phi} + \frac{1}{m} \sum_{i=1}^m \left[1 - y_i \left(f(\mathbf{x}_i) + b\right)\right]_+ \right\}. \tag{15}$$

Table 1: Held-out QKSVM performance (MCC/AUC, 10-seed means). The ZZfeaturemap applies ZZ rotations only between nearest-neighbor qubits on a rectangular lattice. Full pairs provides the $O(B^2)$ reference, while Pairwise-B is the budget-matched $O(B)$ pairwise baseline. The GA→NQE panel compares Pairwise-B and Prototype after NQE refinement of the GA-selected embeddings.

Stage	Method	IRIS	VERTEBRAL	T-CELL	GLASS	T-MNIST
		Test MCC / Test AUC				
Fixed	ZZFeatureMap	.6470 / .8842	.1981 / .6749	.4542 / .7686	.3553 / .8025	.0703 / .5707
GA	Full pairs $[O(B^2)]$	**.9554 / .9994**	.6152 / **.9234**	**.5548** / .8034	**.6057** / .8723	–
	Pairwise-B $[O(B)]$	.9200 / .9966	**.6403** / .9125	.5427 / .8141	.5907 / .8603	.6357 / .8883
	Prototype $[O(B)]$	.9387 / .9964	.5939 / .8851	.5502 / **.8283**	.5979 / **.8837**	**.7606 / .9437**
GA→NQE	Pairwise-B $[O(B)]$	.9419 / **.9976**	**.6471 / .9093**	**.5624** / .8007	.5826 / .8729	.6423 / .8962
	Prototype $[O(B)]$	**.9574** / .9975	.6460 / .8978	.5543 / **.8325**	**.5919 / .8765**	**.7787 / .9516**

where $\lambda_{\mathrm{SVM}} > 0$ denotes the regularization parameter. Using h_{proto} as an explicit RKHS witness, we establish the following upper bounds on its optimal value.

Proposition 3 (Prototype RKHS Witness for the SVM Objective)**.** *For a fixed quantum embedding* Φ*, the optimal value of the normalized soft-margin SVM objective satisfies*

$$\mathcal{J}^{\star}_{\mathrm{SVM}}(\Phi) \leq \inf_{\tau \geq 0} \left\{ \frac{\lambda_{\mathrm{SVM}}}{2} \tau^2 \|h_{\mathrm{proto}}\|^2_{\mathcal{H}_\Phi} + \frac{1}{m} \sum_{i=1}^{m} [1 - \tau y_i h_{\mathrm{proto}}(\mathbf{x}_i)]_+ \right\}. \tag{16}$$

The inequality provides a computable upper-bound surrogate for the optimal SVM objective by the prototype-based RKHS classifier h_{proto}. Therefore, an embedding that yields a well-performing prototype classifier also provides evidence that a good SVM classifier can be constructed. Computing this bound requires evaluating each sample against both class prototypes, resulting in $O(2B)$ fidelity evaluations. Therefore, it provides a cheaper indicator of the downstream linear classification quality. The extension to multiple prototypes is provided in Appendix B.

3 Results

In this section, we evaluate our prototype-guided quantum embedding method across five benchmark datasets—IRIS (Fisher, 1936), T-CELL (Daniels et al., 2022), VERTEBRAL (Barreto & Neto, 2005), GLASS (German, 1987), tailored-MNIST (Lecun et al., 1998)—spanning different dimensionalities and class structures. Dataset details are provided in Appendix C. The quantum embeddings use 4, 6, 8, 9, and 20 qubits for IRIS, VERTEBRAL, T-CELL, GLASS, and T-MNIST, respectively. Optimization is performed using only the training set, while model selection uses the validation set. The test set is reserved exclusively for final evaluation. For each learned embedding, we construct the quantum fidelity kernel and evaluate QKSVM (Hearst et al., 1998). The Matthews correlation coefficient (MCC) is used as the primary evaluation metric because it is applicable to multiclass settings and robust to class imbalance. The area under the receiver operating characteristic curve (ROC–AUC) is reported as a secondary metric for the same QKSVM model selected by its MCC. Unless otherwise stated, all experiments are conducted using noiseless statevector simulation in *Qiskit* (Javadi-Abhari et al., 2024).

3.1 Single Prototype Optimization Results

To evaluate the efficiency and predictive performance of the proposed objective, we first use a GA (Katoch et al., 2020) to search for a data-adaptive circuit architecture. The circuit architecture is optimized progressively across increasing depths, with the best circuit found at each depth used to initialize the search at the next depth. This layer-wise strategy reuses previously optimized circuit structure rather than restarting the architecture search independently at each depth. After the GA, We further apply NQE (Hur et al., 2024) to the selected circuit, keeping its architecture fixed while continuously refining the data embedding. A residual neural network (He et al., 2016) predicts additive corrections to the original feature values, with its parameters trained to minimize

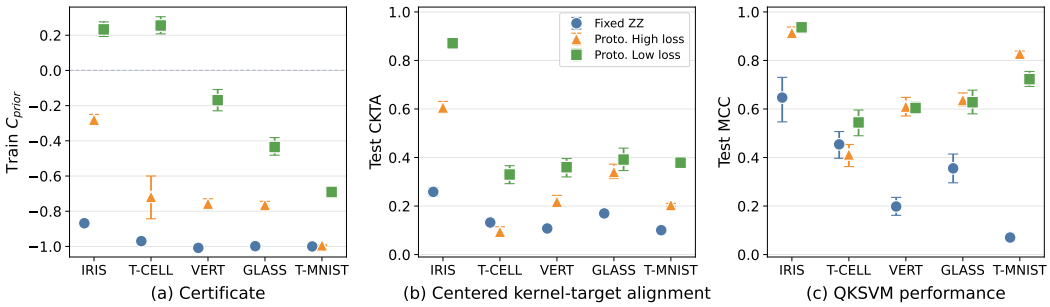

Figure 2: Prototype geometry and downstream behavior. VERT means VERTEBRAL. For each seed, ten unique GA circuits with the lowest and highest training prototype losses form the low- and high-loss groups, respectively. **(a)** Training C_{prior}, where the dashed line marks the threshold for a non-vacuous positive certificate. **(b)** Test pairwise-macro CKTA. **(c)** Test QKSVM MCC. Markers and error bars denote the mean and 95% seed-bootstrap CI.

the corresponding embedding objective through gradient-based optimization. Together, these stages yield an end-to-end optimization pipeline combining discrete architecture search with continuous embedding refinement. Detailed settings are provided in Appendix D.

We compare the proposed method with three quantum embedding baselines in Table 1. The fixed ZZ feature map (Havlíček et al., 2019) serves as a standard data-independent embedding, with its circuit depth selected according to validation MCC. Full pairwise optimization, described in Section 2.1, evaluates the fidelity objective over all training-sample pairs and serves as the $O(B^2)$ reference. We also include a budget-matched pairwise baseline (Pairwise-B) that randomly samples a subset of training pairs so that each objective evaluation uses the same number of fidelity evaluations as the proposed method. For a controlled comparison, all methods use the same genetic-algorithm configuration and quantum-circuit search space, differing only in their optimization objectives and fidelity-evaluation budgets. We omit full-pairwise NQE refinement due to the computational cost of differentiating through all $O(B^2)$ fidelities at each update in our experimental setting. The GA→NQE comparison therefore includes only the proposed and budget-matched objectives.

Table 1 summarizes the held-out QKSVM performance after GA search and subsequent NQE refinement. At the GA stage, the prototype-guided objective remains competitive with full-pairwise optimization while requiring only $O(B)$ fidelity evaluations, and consistently improves upon the fixed ZZ feature map. Under the same $O(B)$ fidelity budget, its advantage is particularly pronounced on the 20-qubit T-MNIST task, improving MCC from .6357 to .7606 over Pairwise-B. After NQE refinement, the prototype objective achieves higher MCC than Pairwise-B on IRIS, GLASS, and T-MNIST, with the largest gap again observed on T-MNIST (.7787 vs. .6423).

Further analysis of the final T-MNIST embeddings shows that the prototype-guided method yields greater pairwise class-ensemble distinguishability and larger RKHS-normalized QKSVM margins for most test samples than Pairwise-B, providing geometric evidence consistent with its performance advantage (Appendix F.1). Overall, these results demonstrate that prototype-guided optimization can substantially reduce fidelity-evaluation cost while retaining competitive QKSVM performance, although its relative benefit remains dataset dependent.

3.2 Empirical Behavior of the Certificate and RKHS Witness

To examine whether the geometry promoted by the prototype objective transfers beyond its training loss, we compare ZZ feature map with ten circuits having the lowest and highest training prototype losses during GA optimization. The groups are constructed using only the train objective. We evaluate C_{prior} on the train set, CKTA on the test set, and the test MCC of the corresponding QKSVM.

As shown in Figure 2(a)–(b), the low-loss circuits consistently yield the largest C_{prior} and CKTA across all datasets. They also achieve higher mean MCC than the ZZ feature map, with the clearest separation on IRIS and T-CELL, where C_{prior} becomes positive. On the remaining datasets, the certificate remains negative and therefore provides no improvement over the trivial zero lower bound.

Table 2: Held-out MCC/AUC after validation reranking of the same dataset-specific bank of 20 prototype-optimized circuits (10-seed means). Here, N and V denote the numbers of training and validation samples, respectively, and C denotes the number of classes, with one prototype per class. P0 selects using validation prototype loss, P1 using the exact prototype RKHS classifier in Proposition 3, and P2 using a full validation QKSVM. ΔMCC denotes P1–P0, and the leading fidelity cost of each selector is shown in the parentheses, the exact cost is in Appendix E.

	Linear-cost selectors			Full-QKSVM reference
Dataset	P0 (V)	P1 (CV)	Δ MCC	P2 (N^2)
IRIS	.9387/.9964	.9387/.9967	+.0000	.9387/.9976
VERTEBRAL	.5939/.8851	.6189/.8920	**+.0250**	.6201/.8946
T-CELL	.5502/.8283	.5502/.8283	+.0000	.5502/.8283
GLASS	.5979/.8837	.6073/.8800	**+.0094**	.6001/.8865
T-MNIST	.7606/.9437	.7557/.9427	−0.0049	.7657/.9446

Higher CKTA does not, however, translate monotonically into MCC, particularly on T-MNIST. While CKTA summarizes global Gram-matrix alignment, QKSVM performance also depends on margins near the decision boundary. Relative to the high-loss circuits, the low-loss circuits improve RKHS-normalized geometric test margins for most samples but worsen them for a majority of samples in the hardest difficulty decile (Appendix F.2). This pattern suggests that the mean-based objective can favor easier samples at the expense of difficult boundary samples, illustrating why improved average embedding geometry need not yield better QKSVM performance.

We next test whether the prototype construction can be used for validation candidate selection. The 20 prototype-optimized circuits in the bank of GA is reranked using the validation prototype loss (P0), the exact prototype RKHS classifier in Proposition 3 (P1) and a full validation QKSVM (P2).

Table 2 shows that P1 improves or preserves the mean test MCC relative to P0 on four of five datasets, and matches or exceeds P2 on three. These results do not identify the prototype classifier with the trained QKSVM. They rather show that it provides a classifier-aware RKHS witness that can serve as a useful proxy. When the additional validation cost is affordable, prototype classifier offers a more informative selector than the prototype loss without requiring the quadratic train kernel.

3.3 Multi Prototype Opimization Results

A single prototype may be insufficient when a class contains multiple distinct modes. We therefore evaluate the multi-prototype extension of Proposition 2 on GLASS. To alleviate its class sparsity, the original glass types are grouped into three broader classes: float-processed window glass, non-float-processed window glass, and non-window glass. These three classes contain two, one, and three original sublabels, respectively. We therefore use $K = (2, 1, 3)$, where each entry denotes the number of prototypes for the corresponding class in the order listed above. One prototype is constructed per subclass, and each sample remains assigned to its subclass prototype throughout optimization. Thus, the experiment directly tests whether representing distinct within-class modes by separate prototypes improves the geometry targeted by the multi-prototype objective.

As a result, the multi-prototype embedding increases the test MCC from 0.592 to 0.647, while the AUC remains similar (0.876 to 0.884). Figure 3(a) compares the resulting QKSVM with the single-prototype model. The out-of-fold (OOF) margin distribution also shifts toward larger values overall, although the multi-prototype model retains a somewhat longer negative tail. This indicates that the improvement is not explained by uniformly increasing every sample margin. Rather, the multi-prototype representation improves the margin geometry for a substantial portion of the samples while a smaller set of difficult boundary samples remains.

To examine the mechanism more directly, Figure 3(b) relates the sample-wise trace distance reduction to the assigned prototype, $r_{K=1} - r_{\text{multi}}$, to the corresponding OOF margin gain, $m_{\text{multi}} - m_{K=1}$. The two quantities exhibit a positive association ($\rho_s = 0.33$) that implies samples whose local prototype radius is reduced more tend to obtain larger classification margins. Although

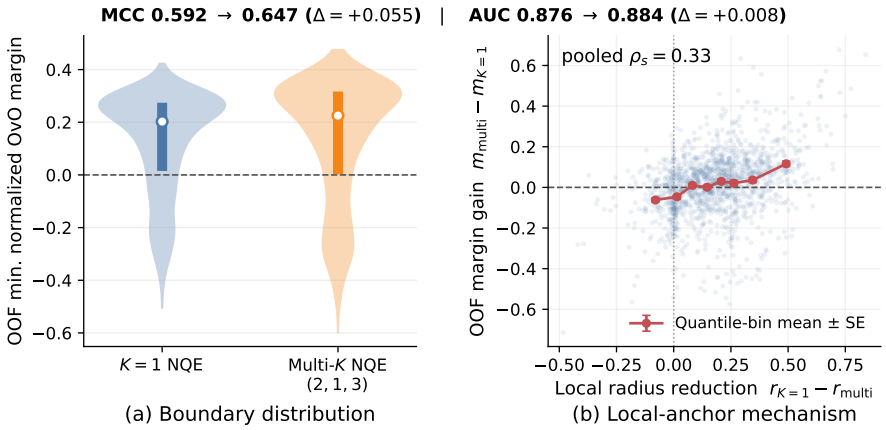

Figure 3: Multi-prototype optimization on GLASS. **(a)** Vilolin plots of OOF minimum normalized one-vs-one margin distributions for the single-prototype ($K = 1$) and multi-prototype ($K = (2, 1, 3)$) model, where the entries of K denote the class-specific numbers of prototypes. The white circle and the bar indicate the median and interquartile range (IQR) of each distribution, respectively. **(b)** OOF margin gain versus the reduction in sample-to-assigned-prototype trace-distance radius. Quantile-bin means $\pm$ standard errors are shown. ρ_s means the pooled Spearman correlation.

the association is moderate rather than deterministic, its direction is consistent across the quantile bins and supports the mechanism underlying Proposition 2. These results provide empirical evidence that multiple prototypes can improve local class coverage and translate the reduction in sample-to-prototype radius into a more favorable QKSVM decision geometry.

4 Summary and Discussion

We introduced a prototype-guided framework for efficient data-adaptive quantum data embedding optimization. By replacing dense sample-pair comparisons with prototype separation and sample-to-prototype compactness, the proposed method provides a measurable lower bound for class-ensemble distinguishability and an explicit RKHS witness for the soft-margin SVM objective. This reduces the fidelity-evaluation cost from $\mathcal{O}(B^2)$ to $\mathcal{O}(B)$ for a constant number of prototypes.

Across the benchmark tasks, the resulting embeddings remained competitive with full-pairwise optimization while consistently improving upon the fixed feature map, showing that effective data-adaptive quantum embeddings do not necessarily require exhaustive pairwise evaluations.

The empirical analyses clarify the scope of this compression. Lower prototype loss consistently produced more favorable training certificates and held-out kernel alignment, but these improvements did not translate monotonically into QKSVM performance. This reflects the fact that C_{prior} is a lower bound for embedding-level class distinguishability rather than a direct characterization of the final SVM decision boundary. The prototype RKHS classifier provides a closer bridge to downstream classification: using it for validation improved or preserved MCC-based circuit selection on four of five datasets while retaining linear evaluation cost. We therefore interpret the certificate as a geometric diagnostic and the RKHS construction as a computationally inexpensive witness of classifier quality, rather than as exact predictors of the trained QKSVM.

The multi-prototype experiment further shows that multiple representatives can better capture multimodal class structure, improving GLASS MCC while reducing local sample-to-prototype distances. However, the gains remain dataset-dependent, and difficult boundary points are not fully explained by prototype geometry alone. More generally, the proposed fidelity objective is a surrogate for the trace-distance certificate, and its effectiveness depends on prototype quality and class structure. The present multi-prototype study also relies on known subclass information, leaving automatic prototype construction, adaptive assignment, and evaluation under finite-shot noisy hardware as important directions for future work.

5 AI USE DISCLOSURE

In this work, we used generative AI tools to create or edit software code and make up draft parts of a research paper. We also used the AI tools to summarize or analyze existing literature and edit a research paper to improve readability.

6 ETHICS STATEMENT

This work focuses on the theoretical and methodological development of quantum data embeddings and their evaluation on existing benchmark datasets. We did not collect new personal data or conduct new experiments involving human participants. Dataset sources and preprocessing procedures are described in Appendix C. The biomedical benchmarks are used to evaluate classification performance, and the results do not establish clinical validity.

7 REPRODUCIBILITY STATEMENT

To support reproducibility, we provide derivations and proofs of the theoretical results in Appendices A and B. Appendix C describes the datasets, label construction, preprocessing, and data split. Appendix D details the genetic algorithm search and neural quantum embedding refinement. Appendix E specifies the validation-based circuit selection rules, including the prototype RKHS classifier, and their fidelity-evaluation costs. Appendix F describes the additional margin analyses. All quantum simulations use noiseless statevector simulation, as specified in the experimental setup. Implementation code for the GA-to-NQE pipeline and some figure-generation procedures is provided in the **Supplementary Material**.

REFERENCES

Sergio Altares-López, Angela Ribeiro, and Juan José García-Ripoll. Automatic design of quantum feature maps. *Quantum Science and Technology*, 6(4):045015, aug 2021. doi: 10.1088/2058-9565/ac1ab1. URL https://doi.org/10.1088/2058-9565/ac1ab1.

Joonwoo Bae and Leong-Chuan Kwek. Quantum state discrimination and its applications. *Journal of Physics A: Mathematical and Theoretical*, 48(8):083001, jan 2015. doi: 10.1088/1751-8113/48/8/083001. URL https://doi.org/10.1088/1751-8113/48/8/083001.

Leonardo Banchi, Jason Pereira, and Stefano Pirandola. Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum*, 2:040321, Nov 2021. doi: 10.1103/PRXQuantum.2.040321. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.040321.

Guilherme Barreto and Ajalmar Neto. Vertebral Column. UCI Machine Learning Repository, 2005. DOI: https://doi.org/10.24432/C5K89B.

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

Matthias C. Caro, Elies Gil-Fuster, Johannes Jakob Meyer, Jens Eisert, and Ryan Sweke. Encoding-dependent generalization bounds for parametrized quantum circuits. *Quantum*, 5:582, November 2021. ISSN 2521-327X. doi: 10.22331/q-2021-11-17-582. URL https://doi.org/10.22331/q-2021-11-17-582.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

Kyle G. Daniels, Shangying Wang, Milos S. Simic, Hersh K. Bhargava, Sara Capponi, Yurie Tonai, Wei Yu, Simone Bianco, and Wendell A. Lim. Decoding car t cell phenotype using combinatorial signaling motif libraries and machine learning. *Science*, 378(6625):1194–1200, 2022. doi: 10.1126/science.abq0225. URL https://www.science.org/doi/abs/10.1126/science.abq0225.

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

B. German. Glass Identification. UCI Machine Learning Repository, 1987. DOI: https://doi.org/10.24432/C5WW2P.

Jennifer R. Glick, Tanvi P. Gujarati, Antonio D. Córcoles, Youngseok Kim, Abhinav Kandala, Jay M. Gambetta, and Kristan Temme. Covariant quantum kernels for data with group structure. *Nature Physics*, 20(3):479–483, January 2024. ISSN 1745-2481. doi: 10.1038/s41567-023-02340-9. URL http://dx.doi.org/10.1038/s41567-023-02340-9.

Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998. doi: 10.1109/5254.708428.

Jaewoong Heo and Daniel K. Park. Generative quantum data embeddings for supervised learning. *arXiv preprint arXiv:2605.30866*, 2026.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.

Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan H. S. Derks, Paul K. Faehrmann, and Johannes Jakob Meyer. Training quantum embedding kernels on near-term quantum computers. *Phys. Rev. A*, 106:042431, Oct 2022. doi: 10.1103/PhysRevA.106.042431. URL https://link.aps.org/doi/10.1103/PhysRevA.106.042431.

Tak Hur and Daniel K. Park. Understanding generalization in quantum machine learning with margins. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26338–26360. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/hur25a.html.

Tak Hur, Israel F. Araujo, and Daniel K. Park. Neural quantum embedding: Pushing the limits of quantum supervised learning. *Phys. Rev. A*, 110:022411, Aug 2024. doi: 10.1103/PhysRevA.110.022411. URL https://link.aps.org/doi/10.1103/PhysRevA.110.022411.

Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D Nation, Lev S Bishop, Andrew W Cross, et al. Quantum computing with qiskit. *arXiv preprint arXiv:2405.08810*, 2024.

Sourabh Katoch, Sumit Chauhan, and Vijay Chahar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80:8091–8126, Oct 2020. doi: 10.1007/s11042-020-10139-6.

Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. *Advances in Neural Information Processing Systems*, 34:12661–12673, 2021.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Cong Lei, Yuxuan Du, Peng Mi, Jun Yu, and Tongliang Liu. Neural auto-designer for enhanced quantum kernels. In *International Conference on Learning Representations*, volume 2024, pp. 17817–17841, 2024.

Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.

Ryan Mandelbaum. Scaling for quantum advantage and beyond. IBM Quantum Computing Blog, November 2025. URL https://www.ibm.com/quantum/blog/qdc-2025.

R Moretti, A Giachero, V Radescu, and M Grossi. Enhanced feature encoding and classification on distributed quantum hardware. *Machine Learning: Science and Technology*, 6(1):015056, feb 2025. doi: 10.1088/2632-2153/adb4bc. URL https://doi.org/10.1088/2632-2153/adb4bc.

Nam Nguyen and Kwang-Cheng Chen. Quantum embedding search for quantum machine learning. *IEEE Access*, 10:41444–41456, 2022. doi: 10.1109/ACCESS.2022.3167398.

Vern I Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152. Cambridge university press, 2016.

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. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 02 2020. doi: 10.22331/q-2020-02-06-226.

M. Emre Sahin, Benjamin C. B. Symons, Pushpak Pati, Fayyaz Minhas, Declan Millar, Maria Gabrani, Stefano Mensa, and Jan Lukas Robertus. Efficient Parameter Optimisation for Quantum Kernel Alignment: A Sub-sampling Approach in Variational Training. *Quantum*, 8:1502, October 2024. ISSN 2521-327X. doi: 10.22331/q-2024-10-18-1502. URL https://doi.org/10.22331/q-2024-10-18-1502.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019. doi: 10.1103/PhysRevLett.122.040504. URL https://link.aps.org/doi/10.1103/PhysRevLett.122.040504.

Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015. doi: 10.1080/00107514.2014.964942. URL https://doi.org/10.1080/00107514.2014.964942.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A*, 103:032430, Mar 2021. doi: 10.1103/PhysRevA.103.032430. URL https://link.aps.org/doi/10.1103/PhysRevA.103.032430.

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pp. 807–814, 2007.

Supanut Thanasilp, Samson Wang, Nhat Anh Nghiem, Patrick Coles, and Marco Cerezo. Subtleties in the trainability of quantum machine learning models. *Quantum Machine Intelligence*, 5(1):21, 2023.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

APPENDIX

A PROOFS

A.1 DERIVATION OF THE HILBERT–SCHMIDT RELAXATION (EQUATION 2)

For a class ensemble distinguishability

$$\Delta_\Phi = \|p_+\bar{\rho}_+ - p_-\bar{\rho}_-\|_1 = 2D_{\mathrm{tr}}(p_+\bar{\rho}_+, p_-\bar{\rho}_-),$$

it admits the following lower bound

$$\Delta_\Phi \geq \frac{1}{m}\Bigg[\underbrace{\sum_{y\in\{+,-\}}\sum_{i,j\in I_y} k_\Phi(\mathbf{x}_i,\mathbf{x}_j)}_{\text{within-class purity terms}} - \underbrace{2\sum_{i\in I_+}\sum_{j\in I_-} k_\Phi(\mathbf{x}_i,\mathbf{x}_j)}_{\text{cross-class overlap term}}\Bigg]^{1/2}. \tag{17}$$

Proof. Let

$$A_\Phi = p_+\bar{\rho}_+ - p_-\bar{\rho}_-.$$

Since the Schatten norms satisfy $\|A\|_1 \geq \|A\|_2$ for any operator A, we have

$$\Delta_\Phi = \|A_\Phi\|_1 \geq \|A_\Phi\|_2 = \|A_\Phi\|_{\mathrm{HS}},$$

where

$$\|A_\Phi\|_{\mathrm{HS}} = \sqrt{\mathrm{Tr}(A_\Phi^\dagger A_\Phi)}.$$

Because A_Φ is Hermitian,

$$\|A_\Phi\|_{\mathrm{HS}}^2 = \mathrm{Tr}(A_\Phi^2).$$

Expanding the right-hand side gives

$$\mathrm{Tr}(A_\Phi^2) = p_+^2\,\mathrm{Tr}(\bar{\rho}_+^2) + p_-^2\,\mathrm{Tr}(\bar{\rho}_-^2) - 2p_+p_-\,\mathrm{Tr}(\bar{\rho}_+\bar{\rho}_-).$$

Using

$$\bar{\rho}_y = \frac{1}{m_y}\sum_{i\in I_y}\rho_\Phi(\mathbf{x}_i), \qquad p_y = \frac{m_y}{m},$$

the class-purity terms can be written as

$$\begin{aligned} p_y^2\,\mathrm{Tr}(\bar{\rho}_y^2) &= \frac{p_y^2}{m_y^2}\sum_{i,j\in I_y}\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right] \\ &= \frac{1}{m^2}\sum_{i,j\in I_y}\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right]. \end{aligned}$$

Likewise, the cross-class term satisfies

$$\begin{aligned} p_+p_-\,\mathrm{Tr}(\bar{\rho}_+\bar{\rho}_-) &= \frac{p_+p_-}{m_+m_-}\sum_{i\in I_+}\sum_{j\in I_-}\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right] \\ &= \frac{1}{m^2}\sum_{i\in I_+}\sum_{j\in I_-}\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right]. \end{aligned}$$

Combining these expressions yields

$$\Delta_\Phi \geq \frac{1}{m}\left[\sum_{y\in\{+,-\}}\sum_{i,j\in I_y}\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right] - 2\sum_{i\in I_+}\sum_{j\in I_-}\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right]\right]^{1/2}.$$

This derivation holds for arbitrary density operators $\rho_\Phi(\mathbf{x}_i)$, including mixed states. For the pure-state embeddings considered in this work,

$$\rho_\Phi(\mathbf{x}_i) = |\psi_\Phi(\mathbf{x}_i)\rangle\langle\psi_\Phi(\mathbf{x}_i)|,$$

and therefore

$$\mathrm{Tr}\left[\rho_\Phi(\mathbf{x}_i)\rho_\Phi(\mathbf{x}_j)\right] = \left|\langle\psi_\Phi(\mathbf{x}_i)\mid\psi_\Phi(\mathbf{x}_j)\rangle\right|^2 = k_\Phi(\mathbf{x}_i,\mathbf{x}_j),$$

which gives Eq. 17. □

The within-class sums include the diagonal terms $i = j$. For normalized pure-state embeddings, these terms satisfy $k_\Phi(\mathbf{x}_i,\mathbf{x}_i) = 1$ and require no quantum evaluation. Together with the symmetry of the fidelity kernel, this leaves $B(B-1)/2$ distinct off-diagonal fidelity evaluations for B samples.

A.2 PROPOSITION 1

For arbitrary class prototypes $\mathbf{c}_+$ and $\mathbf{c}_-$, class-ensemble distinguishability Δ_Φ satisfies

$$\|p_+\bar{\rho}_+ - p_-\bar{\rho}_-\|_1 \geq \sqrt{1-4p_+p_-F_c} - 2\left(p_+\bar{r}_+^{\mathrm{tr}} + p_-\bar{r}_-^{\mathrm{tr}}\right).$$

Proof. By the reverse triangle inequality,

$$\|p_+\bar{\rho}_+ - p_-\bar{\rho}_-\|_1 \geq \left\|p_+\rho_+^c - p_-\rho_-^c\right\|_1 - p_+\left\|\bar{\rho}_+ - \rho_+^c\right\|_1 - p_-\left\|\bar{\rho}_- - \rho_-^c\right\|_1.$$

The convexity of the trace norm gives

$$\left\|\bar{\rho}_y - \rho_y^c\right\|_1 \leq \frac{1}{m_y}\sum_{i\in I_y}\left\|\rho_\Phi(\mathbf{x}_i) - \rho_y^c\right\|_1 = 2\bar{r}_y^{\mathrm{tr}}.$$

Moreover, for the two pure prototype states,

$$\left\|p_+\rho_+^c - p_-\rho_-^c\right\|_1 = \sqrt{(p_+-p_-)^2 + 4p_+p_-(1-F_c)} = \sqrt{1-4p_+p_-F_c}.$$

□

A.3 PROPOSITION 2

For the class-wise prototype sets $\mathcal{P}_y = \{\mathbf{c}_{y,1}, \ldots, \mathbf{c}_{y,K_y}\}$, the class-ensemble distinguishability Δ_Φ satisfies

$$\|p_+\bar{\rho}_+ - p_-\bar{\rho}_-\|_1 \geq \|p_+\tilde{\rho}_+^c - p_-\tilde{\rho}_-^c\|_1 - 2\sum_{y\in\{+,-\}} p_y\bar{r}_{y,K_y}^{\mathrm{tr}}.$$

In the balanced case,

$$\left\|\frac{1}{2}\bar{\rho}_+ - \frac{1}{2}\bar{\rho}_-\right\|_1 \geq \|\frac{1}{2}\tilde{\rho}_+^c - \frac{1}{2}\tilde{\rho}_-^c\|_1 - (\bar{r}_{+,K_+}^{\mathrm{tr}} + \bar{r}_{-,K_-}^{\mathrm{tr}})$$

Proof. By reverse triangle inequality,

$$\|p_+\bar{\rho}_+ - p_-\bar{\rho}_-\|_1 \geq \|p_+\tilde{\rho}_+^c - p_-\tilde{\rho}_-^c\|_1 - p_+\|\bar{\rho}_+ - \tilde{\rho}_+^c\|_1 - p_-\|\bar{\rho}_- - \tilde{\rho}_-^c\|_1.$$

The induced-weight identity gives

$$\tilde{\rho}_y^c = \sum_{a=1}^{K_y}\pi_{y,a}\rho_{y,a}^c = \frac{1}{m_y}\sum_{a=1}^{K_y}|I_{y,a}|\rho_{y,a}^c = \frac{1}{m_y}\sum_{i\in I_y}\rho_{y,a(i)}^c.$$

Therefore,

$$\bar{\rho}_y - \tilde{\rho}_y^c = \frac{1}{m_y}\sum_{i\in I_y}\left(\rho_\Phi(\mathbf{x}_i) - \rho_{y,a(i)}^c\right).$$

Using convexity,

$$\begin{aligned}
\|\bar{\rho}_y - \tilde{\rho}_y^c\|_1 &= \left\|\frac{1}{m_y}\sum_{i\in I_y}\left(\rho_\Phi(\mathbf{x}_i) - \rho_{y,a(i)}^c\right)\right\|_1 \\
&\leq \frac{1}{m_y}\sum_{i\in I_y}\|\rho_\Phi(\mathbf{x}_i) - \rho_{y,a(i)}^c\|_1 \\
&= 2\bar{r}_{y,K_y}^{\mathrm{tr}}.
\end{aligned}$$

□

Connection to the prototype-guided objective. Applying the Hilbert–Schmidt relaxation from Appendix A.1 to the prototype-mixture separation gives

$$\|p_+\tilde{\rho}_+^c - p_-\tilde{\rho}_-^c\|_1 \geq \|p_+\tilde{\rho}_+^c - p_-\tilde{\rho}_-^c\|_{\mathrm{HS}},$$

where

$$\begin{aligned}\|p_+\tilde{\rho}_+^c - p_-\tilde{\rho}_-^c\|_{\mathrm{HS}}^2 \quad &= \sum_{y\in\{+,-\}} p_y^2 \sum_{a,a'=1}^{K_y} \pi_{y,a}\pi_{y,a'} k_\Phi(\mathbf{c}_{y,a}, \mathbf{c}_{y,a'}) \\ &\quad - 2p_+p_- \sum_{a=1}^{K_+}\sum_{b=1}^{K_-} \pi_{+,a}\pi_{-,b} k_\Phi(\mathbf{c}_{+,a}, \mathbf{c}_{-,b}).\end{aligned}$$

This expansion motivates high within-class prototype fidelity and low cross-class prototype fidelity. For pure-state embeddings, $1 - k_\Phi(\mathbf{x}_i, \mathbf{c}_{y,a(i)}) = D_{\mathrm{tr}}(\rho_\Phi(\mathbf{x}_i), \rho^c_{y,a(i)})^2$, which provides a squared-distance surrogate for the radius terms. Thus, L_{sample}, L_{in}, and L_{out} serve as unweighted geometric surrogates for the corresponding certificate terms.

B Multiple-Prototype RKHS Witness for the SVM Objective

We extend Proposition 3 to the multiple-prototype setting introduced in Proposition 2. For each class $y \in \{+,-\}$, define the prototype mean embedding in the RKHS $\mathcal{H}_\Phi$ as

$$\mu_y = \sum_{a=1}^{K_y} \pi_{y,a} k_\Phi(\cdot, \mathbf{c}_{y,a}).$$

The corresponding multiple-prototype RKHS classifier is

$$\begin{aligned}h_{\mathrm{multi}}(\mathbf{x}) &= \mu_+(\mathbf{x}) - \mu_-(\mathbf{x}) \\ &= \sum_{a=1}^{K_+} \pi_{+,a} k_\Phi(\mathbf{x}, \mathbf{c}_{+,a}) - \sum_{b=1}^{K_-} \pi_{-,b} k_\Phi(\mathbf{x}, \mathbf{c}_{-,b}).\end{aligned} \tag{18}$$

Thus, h_{multi} compares the similarity of a sample to the empirical prototype mixtures of the two classes, rather than to a single prototype from each class.

By the reproducing property, $\mu_+, \mu_- \in \mathcal{H}_\Phi$ and hence $h_{\mathrm{multi}} \in \mathcal{H}_\Phi$. Its squared RKHS norm is

$$\begin{aligned}\|h_{\mathrm{multi}}\|_{\mathcal{H}_\Phi}^2 &= \|\mu_+ - \mu_-\|_{\mathcal{H}_\Phi}^2 \\ &= \|\mu_+\|_{\mathcal{H}_\Phi}^2 + \|\mu_-\|_{\mathcal{H}_\Phi}^2 - 2\langle \mu_+, \mu_-\rangle_{\mathcal{H}_\Phi}.\end{aligned}$$

Applying the reproducing property to each term gives

$$\begin{aligned}\|h_{\mathrm{multi}}\|_{\mathcal{H}_\Phi}^2 = &\sum_{a,a'=1}^{K_+} \pi_{+,a}\pi_{+,a'} k_\Phi(\mathbf{c}_{+,a}, \mathbf{c}_{+,a'}) \\ &+ \sum_{b,b'=1}^{K_-} \pi_{-,b}\pi_{-,b'} k_\Phi(\mathbf{c}_{-,b}, \mathbf{c}_{-,b'}) \\ &- 2\sum_{a=1}^{K_+}\sum_{b=1}^{K_-} \pi_{+,a}\pi_{-,b} k_\Phi(\mathbf{c}_{+,a}, \mathbf{c}_{-,b}).\end{aligned} \tag{19}$$

This construction leads directly to the multiple-prototype analogue of Proposition 3.

Proposition 4 (Multiple-Prototype RKHS Witness)**.** *For a fixed quantum embedding Φ, let h_{multi} be defined by Eq. 18. Then the optimal normalized soft-margin SVM objective satisfies*

$$\mathcal{J}^\star_{\mathrm{SVM}}(\Phi) \leq \inf_{\tau\geq 0}\left\{ \frac{\lambda_{\mathrm{SVM}}}{2}\tau^2 \|h_{\mathrm{multi}}\|_{\mathcal{H}_\Phi}^2 + \frac{1}{m}\sum_{i=1}^m [1 - \tau y_i h_{\mathrm{multi}}(\mathbf{x}_i)]_+ \right\}. \tag{20}$$

Proof. By definition, the optimal normalized soft-margin SVM objective is

$$\mathcal{J}^{\star}_{\mathrm{SVM}}(\Phi) = \inf_{\substack{f\in\mathcal{H}_\Phi \\ b\in\mathbb{R}}} \left\{ \frac{\lambda_{\mathrm{SVM}}}{2}\|f\|^2_{\mathcal{H}_\Phi} + \frac{1}{m}\sum_{i=1}^{m}[1 - y_i(f(\mathbf{x}_i)+b)]_+ \right\}.$$

Since $h_{\mathrm{multi}} \in \mathcal{H}_\Phi$, for every $\tau \geq 0$ the scaled function

$$f_\tau = \tau h_{\mathrm{multi}}$$

also belongs to $\mathcal{H}_\Phi$. Therefore, $(f_\tau, 0)$ is an admissible candidate in the SVM optimization problem, and hence

$$\mathcal{J}^{\star}_{\mathrm{SVM}}(\Phi) \leq \frac{\lambda_{\mathrm{SVM}}}{2}\tau^2\|h_{\mathrm{multi}}\|^2_{\mathcal{H}_\Phi} + \frac{1}{m}\sum_{i=1}^{m}[1 - \tau y_i h_{\mathrm{multi}}(\mathbf{x}_i)]_+$$

for every $\tau \geq 0$. Taking the infimum over $\tau \geq 0$ yields Eq. 20. □

Evaluating $h_{\mathrm{multi}}(\mathbf{x}_i)$ requires the fidelity between x_i and every prototype from both classes. For a sample of size B, the sample-dependent cost is therefore

$$B(K_+ + K_-)$$

fidelity evaluations. The RKHS norm in Eq. 19 can be computed using

$$P_K = \binom{K_+}{2} + \binom{K_-}{2} + K_+K_-$$

distinct prototype-to-prototype fidelities, since the diagonal terms satisfy $k_\Phi(\mathbf{c}_{y,a}, \mathbf{c}_{y,a}) = 1$. Thus, the total number of distinct fidelity evaluations required for the multiple-prototype witness is

$$N^{\text{multi-witness}}_{\mathrm{fid}} = B(K_+ + K_-) + \binom{K_+}{2} + \binom{K_-}{2} + K_+K_- = O(BK_{\mathrm{tot}} + K^2_{\mathrm{tot}}).$$

It remains linear in B for a fixed number of prototypes.

Finally, when $K_+ = K_- = 1$, the prototype weights are unity and

$$\mu_+ = k_\Phi(\cdot, \mathbf{c}_+), \qquad \mu_- = k_\Phi(\cdot, \mathbf{c}_-),$$

so that

$$h_{\mathrm{multi}} = k_\Phi(\cdot, \mathbf{c}_+) - k_\Phi(\cdot, \mathbf{c}_-) = h_{\mathrm{proto}}.$$

Moreover,

$$\|h_{\mathrm{multi}}\|^2_{\mathcal{H}_\Phi} = 2 - 2k_\Phi(\mathbf{c}_+, \mathbf{c}_-) = 2(1 - F_c).$$

Hence Proposition 4 reduces exactly to Proposition 3.

C Dataset Details

This section summarizes the datasets, label construction, preprocessing, and Train/validation/Test partitions used in our experiments.

Common split and preprocessing protocol. Common split and preprocessing protocol. For IRIS, VERTEBRAL, and GLASS, we first construct stratified 70/30 outer-training/Test splits and reserve 20% of each outer-training partition for validation, resulting in approximately 56/14/30 Train/validation/Test proportions. T-CELL uses a fixed 172/74 outer split, with the 172-example training pool further divided into 137 Train and 35 validation examples. All competing methods use identical sample indices within each dataset and seed. However, T-MNIST uses a fixed 150/150/250 Train/validation/Test split.

IRIS We use all 150 examples and four continuous features from the UCI Iris dataset (Fisher, 1936), retaining the original three balanced species classes. A *StandardScaler* (Pedregosa et al., 2011) fitted on the outer-training pool transforms the four inputs, which are mapped directly to four qubits without dimensionality reduction.

Table 3: Summary of the datasets used in the experiments. The # Features column reports the input dimensionality, and $\rightarrow$ indicates dimensionality reduction prior to quantum embedding.

Dataset	Samples	#.Features	Qubits	Train / Val / Test
IRIS (Fisher, 1936)	150	4	4	84 / 21 / 45
VERTEBRAL (Barreto & Neto, 2005)	310	6	6	173 / 44 / 93
T-CELL (Daniels et al., 2022)	246	$60 \rightarrow 8$	8	137 / 35 / 74
GLASS (German, 1987)	213	9	9	119 / 30 / 64
T-MNIST (Lecun et al., 1998)	550	$784 \rightarrow 20$	20	150 / 150 / 250

VERTEBRAL We use the binary Vertebral Column 2C (Barreto & Neto, 2005) task containing 210 abnormal and 100 normal examples with six biomechanical features. After standardization, each coordinate is transformed as $\mathbf{x} \rightarrow (\pi/2)\tanh(\mathbf{x})$ and mapped one-to-one to six qubits.

T-CELL We use 246 examples from the combinatorial CAR-T signaling-motif dataset of Daniels et al. (2022). Four motif positions are represented as 15-category one-hot vectors, yielding a 60-dimensional input. We label an example positive when its `Nalm 6 Cytotoxicity` exceeds 0.62 and negative otherwise. The fixed 172-example training pool is stratified into 137 Train and 35 validation examples, with 74 additional examples used for Test. A CNN–LSTM encoder (Daniels et al., 2022) trained on the Train subset maps the inputs to eight angles via $\mathbf{z} = \pi\tanh(\mathbf{z}_{\text{raw}})$. The frozen encoder is applied to validation and Test, and the eight coordinates are mapped directly to eight qubits.

GLASS We use the nine continuous chemical features of the UCI Glass Identification dataset (German, 1987) after removing the identifier and one duplicated feature vector, resulting in 213 examples. The original labels are regrouped into three classes: float-processed window glass, non-float-processed window glass, and non-window glass. Features are standardized and transformed by $\mathbf{x} \rightarrow (\pi/2)\tanh(\mathbf{x})$ before being mapped to nine qubits.

T-MNIST We construct a balanced five-class subset of MNIST (Lecun et al., 1998) using digits 0–4. We select 30 Train, 30 validation, and 50 Test images per digit, giving fixed partitions of 150/150/250 examples, and scale pixel values to $[0, 1]$. A convolutional Wasserstein autoencoder (WAE) (Tolstikhin et al., 2018) trained on the 150 train images produces a 20-dimensional latent representation. The latents are standardized using statistics of trainset, transformed as $\mathbf{x} \mapsto (\pi/2)\tanh(\mathbf{x})$, and mapped one-to-one to 20 qubits. The learned encoder and transformation are applied unchanged to validation and Test.

D Experimental Details

D.1 Genetic-Algorithm Circuit Search

Stage-wise circuit construction. Let Q and d denote the number of qubits and the input dimension, respectively. We construct a data-embedding circuit with at most $L_{\max} = 5$ encoding layers,

$$U_L(\mathbf{x}) = B_L(\mathbf{x}) \cdots B_1(\mathbf{x}), \qquad B_\ell(\mathbf{x}) = E_{\text{lat}} V_\ell(\mathbf{x}).$$

Each one-qubit block V_ℓ contains two data-dependent rotations per qubit,

$$V_\ell(\mathbf{x}) = \bigotimes_{q=1}^{Q} \Big[R_{a_{\ell q 2}}(\tau_{\ell q 2}(\mathbf{x}))\, R_{a_{\ell q 1}}(\tau_{\ell q 1}(\mathbf{x})) \Big],$$

where the rotation axis is selected from

$$a_{\ell q s} \in \{X, Y, Z\},$$

and angle function is selected from

$$\mathcal{T}(\mathbf{x}) = \{\mathbf{x}_j : 1 \le j \le d\} \cup \Big\{ \frac{\mathbf{x}_j \mathbf{x}_k}{\pi} : 1 \le j < k \le d \Big\} \cup \{0, \pi\}, \qquad \tau_{\ell q s}(\mathbf{x}) \in \mathcal{T}(\mathbf{x}).$$

Table 4: Dataset-specific circuit layouts and GA hyperparameters. Q is the number of qubits, lattice specifies the rows and columns of the rectangular grid, and $|\mathcal{E}_{\mathrm{lat}}|$ counts its nearest-neighbor CZ edges per layer. Population and Epochs/stage denote the population size and maximum number of generations at each of the five sequential stages. E_{keep} is the number of tournament-selected parents protected from mutation. Patience is the number of consecutive generations without sufficient improvement in the stage's monitored criterion before early stopping.

Dataset	Q	Lattice	$\lvert\mathcal{E}_{\mathrm{lat}}\rvert$	Population	Epochs/stage	E_{keep}	Patience
IRIS	4	2×2	4	200	50	8	10
VERTEBRAL	6	2×3	7	300	100	5	15
T-CELL	8	4×2	10	300	100	5	15
GLASS	9	3×3	12	300	100	5	15
T-MNIST	20	4×5	31	300	100	5	15

Thus, each one-qubit gene specifies a rotation axis together with a single-feature, feature-product, or constant rotation angle. After crossover and mutation, a repair operation ensures that every input feature appears at least once among the rotation slots of each active layer.

The two-qubit block E_{lat} is fixed as

$$E_{\mathrm{lat}} = \prod_{(u,v)\in\mathcal{E}_{\mathrm{lat}}} \mathrm{CZ}_{uv},$$

where $\mathcal{E}_{\mathrm{lat}}$ contains the horizontal and vertical nearest-neighbor edges of an open rectangular lattice. This connectivity was motivated by the square-lattice architecture of the recent IBM Quantum Nighthawk processor (Mandelbaum, 2025). We therefore use a fixed nearest-neighbor lattice throughout the search and optimize only the data-encoding single-qubit rotations.

Sequential optimization and candidate-bank construction. At stage ℓ, the previously selected layers $B_1, \ldots, B_{\ell-1}$ form a frozen prefix, and the GA evolves only one newly initialized active layer B_l. Each individual is concatenated with this prefix and evaluated using the corresponding training objective. The population is updated by tournament selection with $k = 4$, structural crossover, and mutation with probability 0.15. Conditional on mutation, the active genome is re-randomized either at the whole-layer or single-qubit level. The best training-loss individual is appended to the frozen prefix at the end of each stage. Early stopping is applied independently when the incumbent fails to improve by at least 10^{-4} for the patience reported in Table 4.

We retain a global bank of the 20 unique circuits with the lowest observed training losses among candidates of depths 2–5. Candidate identity is defined by the complete circuit genome. Validation and test data are not used to evolve the GA population or construct the candidate bank. After the search, a prespecified validation rule selects the final circuit, and the selected circuit is fixed during the NQE fine-tuning stage.

Fidelity kernel and prototype construction. For a circuit genome G and its embedding unitary U_Φ, we use the fidelity kernel

$$k_\Phi(\mathbf{x}, \mathbf{x}') = \left|\langle 0|U_\Phi^\dagger(\mathbf{x})U_\Phi(\mathbf{x}')|0\rangle\right|^2.$$

Let N be the training size, C the number of classes, and n_y the number of training samples in class y, and $P = \binom{C}{2}$.

Because the input features are used as rotation-gate angles, each coordinate can be interpreted as a periodic angular variable. We therefore construct the $K = 1$ class prototype in the input angle space using the coordinate-wise circular mean rather than the arithmetic mean,

$$(\mathbf{c}_y)_j = \mathrm{atan2}\left(\frac{1}{n_y}\sum_{i:y_i=y}\sin\mathbf{x}_{ij}, \frac{1}{n_y}\sum_{i:y_i=y}\cos\mathbf{x}_{ij}\right).$$

Prototype objective. The single Prototype GA minimizes

$$\mathcal{L}_{\mathrm{P}}(\Phi) = \underbrace{\frac{1}{\binom{C}{2}} \sum_{a<b} k_\Phi(\mathbf{c}_a, \mathbf{c}_b)}_{\mathcal{L}_{\mathrm{cross}}} + \lambda_{\mathrm{P}} \underbrace{\frac{1}{C} \sum_{y} \frac{1}{n_y} \sum_{i:y_i=y} [1 - k_\Phi(\mathbf{x}_i, \mathbf{c}_y)]}_{\mathcal{L}_{\mathrm{sample}}},$$

The first term penalizes fidelity between distinct class prototypes, while the second promotes high fidelity between each sample and its own class prototype. The weight λ_{P} is selected using validation data at the dataset level and remains fixed during test evaluation. The values used for IRIS, VERTEBRAL, T-CELL, GLASS, and T-MNIST are 0.5, 0.7, 0.3, 0.5, and 0.5, respectively.

For N training samples and C classes, one evaluation of the prototype objective requires

$$Q_{\mathrm{P}} = N + \binom{C}{2}$$

logical fidelity evaluations.

Multi-prototype objective. For GLASS, we extend the single-prototype construction by retaining one prototype per original sublabel within each merged class, giving $(K_1, K_2, K_3) = (2, 1, 3)$. Prototypes are computed as coordinatewise circular means of their inner-Train samples. Each sample remains assigned to its original sublabel prototype, and $\pi_{y,a} = n_{y,a}/n_y$ denotes its fixed within-class mixture weight.

We extend $\mathcal{L}_{\mathrm{sample}}$ by replacing c_y with the assigned prototype $\mathbf{c}_{y,a(i)}$. For $\mathcal{L}_{\mathrm{cross}}$, each class-pair fidelity is replaced by the mixture-weighted fidelity $\sum_{a,b} \pi_{y,a}\pi_{z,b} k_\Phi(\mathbf{c}_{y,a}, \mathbf{c}_{z,b})$. We additionally penalize separation between prototypes belonging to the same class,

$$\mathcal{L}_{\mathrm{in}}(\Phi) = \frac{1}{C^2} \sum_{y} \sum_{a,a'=1}^{K_y} \pi_{y,a}\pi_{y,a'} \left[1 - k_\Phi(\mathbf{c}_{y,a}, \mathbf{c}_{y,a'})\right].$$

Unlike the single-prototype weighted sum, Multi-K GA jointly optimizes three components using an augmented Tchebycheff fitness. The implemented component vector is

$$\left(\ell_1, \ell_2, \ell_3\right) = \left(\mathcal{L}_{\mathrm{sample}}, \mathcal{L}_{\mathrm{in}}, \frac{C-1}{C}\mathcal{L}_{\mathrm{cross}}\right).$$

Here, the cross-class coefficient is $2/C^2$, rather than $1/\binom{C}{2}$ used in the single-prototype objective.

For each component, we define

$$z_j(\Phi) = \frac{\ell_j(\Phi) - q_{j,0.1}}{\max\{q_{j,0.9} - q_{j,0.1}, 10^{-3}\}},$$

where $q_{j,0.1}$ and $q_{j,0.9}$ are its empirical 10th and 90th percentiles in the initial population of the first GA stage. The final scalar fitness is

$$\mathcal{L}_{\mathrm{MK}}(\Phi) = \max_{j \in \mathcal{A}} z_j(\Phi) + \frac{0.05}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} z_j(\Phi),$$

where $\mathcal{A}$ contains components whose initial interquantile range exceeds 10^{-12}. Thus, GA primarily minimizes the largest normalized component, with a small additional penalty on their mean. All three component coefficients are one, and the normalization references remain fixed throughout the search. Each fitness evaluation requires $N + \binom{K_{\mathrm{tot}}}{2} = N + 15$ fidelity terms.

Full-pairwise objective. Let

$$\mathcal{P}_{aa} = \{(i,j) : i < j,\ y_i = y_j = a\}, \qquad \mathcal{P}_{ab} = \{(i,j) : y_i = a,\ y_j = b\}, \quad a < b.$$

The Full-pairwise GA (*Full*) uses the class-balanced loss

$$\mathcal{L}_{\mathrm{F}}(\Phi) = \frac{1}{2}\mathcal{L}_{\mathrm{same}}(\Phi) + \frac{1}{2}\mathcal{L}_{\mathrm{diff}}(\Phi),$$

where

$$\mathcal{L}_{\text{same}}(\Phi) = \frac{1}{C} \sum_{a} \frac{1}{|\mathcal{P}_{aa}|} \sum_{(i,j)\in\mathcal{P}_{aa}} \left[1 - k_\Phi(\mathbf{x}_i, \mathbf{x}_j)\right]^2,$$

and

$$\mathcal{L}_{\text{diff}}(\Phi) = \frac{1}{P} \sum_{a<b} \frac{1}{|\mathcal{P}_{ab}|} \sum_{(i,j)\in\mathcal{P}_{ab}} k_\Phi(\mathbf{x}_i, \mathbf{x}_j)^2.$$

Because every unordered training pair is evaluated, its logical fidelity cost per candidate is

$$Q_{\text{F}} = \binom{N}{2}.$$

Budget-matched objective. To separate the effect of prototype-based supervision from the number of fidelity evaluations, the budget-matched baseline (*Pairwise-B*) uses the same logical budget as the Prototype objective. At generation t, a deterministic seeded set $\mathcal{E}_t$ of

$$Q_{\text{U}} = N + \binom{C}{2} = Q_{\text{P}}$$

training pairs is sampled without replacement from the complete pair pool. The same sampled set is used for all candidate circuits in that generation and is refreshed at the next generation.

Writing $\mathcal{E}_t^{\text{same}}$ and $\mathcal{E}_t^{\text{diff}}$ for its same- and different-class subsets, the objective is

$$\mathcal{L}_{\text{U}}^{(t)}(\Phi) = \frac{1}{2|\mathcal{E}_t^{\text{same}}|} \sum_{(i,j)\in\mathcal{E}_t^{\text{same}}} \left[1 - k_\Phi(\mathbf{x}_i, \mathbf{x}_j)\right]^2 + \frac{1}{2|\mathcal{E}_t^{\text{diff}}|} \sum_{(i,j)\in\mathcal{E}_t^{\text{diff}}} k_\Phi(\mathbf{x}_i, \mathbf{x}_j)^2.$$

Prototype and *Pairwise-B* therefore use identical fidelity budgets while differing in their supervision structure. The *Prototype* objective therefore connects each sample to a persistent class anchor, whereas *Pairwise-B* observes a refreshed subset of sample pairs.

Prototype and *Full* use the generation-best raw training loss to update the stage incumbent. *Pairwise-B* uses an EMA of the generation-best sampled loss, with decay 0.9, and retains the raw-loss-best individual from the corresponding generation. Candidate-bank ranking always uses observed raw losses.

Validation-based circuit selection. After GA search, we select the circuit with the lowest validation loss from the frozen candidate bank. *Prototype* uses the class-balanced prototype objective, whereas the *Full* use their corresponding validation pairwise losses. For the *Pairwise-B*, the validation pair set is fixed across candidate circuits. The selected circuit is then kept fixed during NQE refinement. Test data are not used for circuit selection.

D.2 Neural Quantum Embedding Refinement

Fixed-circuit residual refinement. After the GA selects embedding circuit Φ, NQE keeps all quantum gates, feature assignments, and entangling edges fixed and learns only a classical residual map before the circuit. For every dataset and method, g_θ is the same shared simple D2–H32 network,

$$r_\theta(\mathbf{x}) = W_3\ \mathrm{ReLU}(W_2\ \mathrm{ReLU}(W_1\mathbf{x} + b_1) + b_2) + b_3,$$
$$g_\theta(\mathbf{x}) = \mathbf{x} + \widetilde{r}_\theta(\mathbf{x}).$$

Both hidden layers have width 32. The weight and bias of output layer (W_3, b_3) are initialized to zero, so $g_{\theta_0}(\mathbf{x}) = \mathbf{x}$ exactly and epoch zero reproduces the selected GA embedding. This identity initialization prevents a random input transformation from perturbing the GA-optimized embedding before refinement begins. A single mapper transforms the input, and the resulting features are shared across all encoding layers of the fixed circuit.

We compare two residual domains while keeping the neural network architecture fixed. The unbounded version uses $\widetilde{r}_\theta = r_\theta$, whereas the bounded version uses

$$\widetilde{r}_\theta(x) = \alpha \tanh\left(\frac{r_\theta(x)}{\alpha}\right), \qquad \alpha = \frac{\pi}{4},$$

where the hyperbolic tangent is applied coordinate-wise. Hence the latter enforces $|g_\theta(\mathbf{x})_j - \mathbf{x}_j| \leq \pi/4$ without changing the derivative of the residual branch at the identity initialization. This comparison examines how restricting the magnitude of the learned input corrections affects NQE fine-tuning and downstream classification performance.

Training objectives. During NQE refinement, we optimize only the residual network parameters θ, keeping the GA-selected circuit fixed. The prototypes remain fixed in input space, and both samples and prototypes are passed through the same residual mapper before fidelity evaluation. *Prototype* and *Pairwise-B* NQE retain their respective GA training objectives, evaluated on the mapped inputs.

Improving average prototype compactness does not necessarily improve the samples near the downstream decision boundary, as illustrated in Appendix F.2. This motivates examining whether continued compactness minimization is beneficial during NQE refinement. We therefore consider a Dual-Guard variant that removes the direct incentive for further compactness reduction once a prescribed class-wise improvement has been attained. Specifically, we replace the mean compactness penalty with a soft constraint penalty relative to the initial GA embedding.

Let

$$Q_y(\theta) = \frac{1}{n_y} \sum_{i:y_i=y} \left[1 - k_{\Phi,\theta}(\mathbf{x}_i, \mathbf{c}_y)\right], \qquad v_y(\theta) = \frac{Q_y(\theta)}{\max\{Q_y(\theta_0), 10^{-8}\}} - (1-\delta).$$

Here, θ_0 denotes the identity-initialized mapper, and $v_y(\theta) \leq 0$ indicates that the prescribed compactness target has been attained. The guarded objective is

$$\mathcal{L}_{\mathrm{G}}(\theta, \boldsymbol{\eta}) = \mathcal{L}_{\mathrm{cross}}(\Phi, \theta) + \frac{1}{C} \sum_y \left(\eta_y [v_y(\theta)]_+ + \frac{\rho}{2} [v_y(\theta)]_+^2 \right),$$

where $[u]_+ = \max\{u, 0\}$, $\delta = 0.05$, and $\rho = 0.1$. The multipliers are initialized as $\eta_y^{(0)} = \lambda_{\mathrm{P}} Q_y(\theta_0)$ and updated by projected dual ascent:

$$\eta_y^{(t+1)} = \mathrm{clip}_{[0,10]} \left(\eta_y^{(t)} + 0.05\, v_y(\theta_t) \right).$$

This soft penalty encourages, but does not guarantee, satisfaction of the class-wise targets. It reuses the same sample–prototype fidelities and therefore requires no additional quantum evaluations.

Optimization and hyperparameter grid. Each run performs 40 Adam updates with a constant learning rate, zero weight decay, and global gradient-norm clipping at 1.0. Prototype uses the full inner-training sample–prototype objective, whereas *Pairwise-B* uses a refreshed budget-matched pair set at each update. For Dual Guard, λ_{P} determines the initial dual multipliers rather than a fixed compactness weight. The residual mapper states are saved at

$$\mathcal{T} = \{0, 5, 10, 15, 20, 25, 30, 35, 40\}.$$

For the four tabular datasets, the validation search uses

$$\alpha \in \left\{ \frac{\pi}{4},\ \text{unbounded} \right\}, \quad \eta_\theta \in \{0.001, 0.003, 0.005, 0.01\}.$$

The D2–H32 architecture, optimizer, training horizon, and all remaining settings are held common across this grid. The bound and learning rate are frozen at the dataset–method level using validation data only; Test results do not enter this choice. Table 5 lists the frozen endpoints.

Checkpoint and QKSVM selection. NQE is trained only on the train partition. For Mean Prototype, the checkpoint in $\mathcal{T}$ with the smallest validation loss is selected using the frozen inner-Train prototypes. Dual Guard instead uses a feasibility-first rule. Checkpoints are ranked by the maximum positive class-wise validation violation, followed by the mean positive violation, cross-prototype fidelity, and the earlier epoch. Validation violations are measured relative to the epoch-zero validation compactness.

For *Pairwise-B*, the checkpoint with the smallest validation pairwise loss is selected. The same seeded validation-pair set is used across checkpoints for the tabular datasets, whereas T-MNIST uses all unordered validation pairs. These validation evaluations do not update the mapper.

Table 5: Frozen NQE hyperparameters used for the main comparison."Unb." denotes an unbounded residual. Checkpoints are selected using each method's validation objective.

Dataset	Prototype loss	λ_{P}	Prototype bound/LR	Pairwise-B bound/LR
IRIS	Mean	0.5	Unb./0.010	Unb./0.005
VERTEBRAL	Mean	0.7	Unb./0.010	Unb./0.010
T-CELL	Mean	0.3	Unb./0.003	Unb./0.010
GLASS	Dual Guard	0.5	$\pi/4$/0.010	Unb./0.010
T-MNIST	Mean	0.5	$\pi/4$/0.001	$\pi/4$/0.001

After the NQE refinement, the kernel on the complete outer-training set, comprising inner Train and validation, is constructed. For the single-prototype experiments and their *Pairwise-B* baselines, the QKSVM regularization parameter is selected by stratified five-fold cross-validation on this set, using MCC and the grid

$$\mathcal{C}_{\mathrm{SVM}} = \{0.01, 0.05, 0.1, 0.25, 0.5, 1, 2.5, 5, 10, 25, 50\}. \tag{21}$$

The QKSVM is then refitted on the complete outer-training set and evaluated on the held-out Test set. The mapper is not refitted on validation or Test data, and neither the checkpoint nor the SVM regularization parameter is reselected using Test performance.

Multi-prototype GLASS refinement. Multi-K NQE retains the GA-stage prototypes, sublabel assignments, and mixture weights with $(K_1, K_2, K_3) = (2, 1, 3)$. The mapper architecture and training schedule are unchanged. Mean NQE minimizes $\ell_1 + \ell_2 + \ell_3$ without Tchebycheff normalization, whereas Dual Guard replaces ℓ_1 with the class-wise guard, initialized with $\eta_y^{(0)} = Q_y(\theta_0)$. Checkpoint selection follows the corresponding Mean or Guard rule, with $\ell_2 + \ell_3$ as the Guard structural tie-breaker.

The selected bound/LR settings are unbounded/0.01 for Mean and $(\pi/4)/0.003$ for Dual Guard. QKSVM evaluation follows the same procedure and the comparison also uses independently validation-selected single- and multi-prototype endpoints, with no Test-based reselection.

E Validation-Based Circuit Selection

Common candidate bank. The selector comparison changes only the post-search choice of circuit. For each dataset and seed, we use the same bank of $M = 20$ unique prototype-optimized circuits as the main GA comparison. The bank retains the lowest training-loss candidates encountered across GA stages after genotype deduplication. Validation scores do not affect GA fitness or bank membership. We consider the single-prototype setting, with one fixed training prototype per class. The train and validation sizes are denoted by N and V, respectively, and the number of classes by C. Class prototypes $\mathbf{c}_y$ are computed once from train using the same coordinate-wise circular means as in GA search and remain fixed in input space for all candidates. The three selectors below rerank this identical bank.

P0: validation prototype loss. P0 selects the candidate with the smallest GA prototype objective evaluated on the validation set, keeping the training prototypes and λ_{P} fixed. Each candidate requires $V + \binom{C}{2}$ fidelity evaluations, one per validation sample–own-prototype pair and one per unordered prototype pair. Ties are resolved by the training-loss ordering of the bank and then by the deterministic candidate identifier.

P1: exact prototype RKHS classifier. P1 evaluates each validation sample against every class prototype and predicts

$$\widehat{y}_\Phi(x) = \operatorname*{arg\,max}_{y \in \{1,\dots,C\}} k_\Phi(x, c_y).$$

For two classes, this is the sign rule induced by $h_{ab}(x) = k_\Phi(x, c_a) - k_\Phi(x, c_b)$, the prototype RKHS witness. The multiclass implementation selects the largest class score directly.

Candidates are ranked by validation MCC of these predictions. To resolve MCC ties, we use the 10th percentile of the normalized prototype margins,

$$\gamma_i^{\mathrm{proto}}(\Phi) = \min_{b \neq y_i} \frac{k_\Phi(x_i, c_{y_i}) - k_\Phi(x_i, c_b)}{\sqrt{2\big[1 - k_\Phi(c_{y_i}, c_b)\big]}}.$$

The denominator is $\|h_{y_i b}\|_{\mathcal{H}_\Phi}$. The margin therefore measures the least favorable true-class comparison in RKHS units. When candidates have the same validation MCC, we prefer the one with a larger margin at the 10th percentile of the validation margin distribution, favoring better separation near the difficult end of the validation samples. If this value is also tied, we prefer the smaller validation prototype loss and then use the fixed candidate order. Candidates whose prototypes from different classes map to nearly identical quantum states ($\sqrt{1 - k_\Phi(c_a, c_b)} \leq 10^{-12}$) are given lower priority, because their normalized margins involve division by a near-zero RKHS norm.

P1 requires neither SVM fitting nor a regularization search. In particular, it uses the witness classifier's validation MCC, not a direct minimization of the SVM-objective upper bound in Proposition 3 or an optimization of its scaling parameter.

P2: validation QKSVM reference. For every candidate, P2 constructs the Gram matrix of train set and chooses the SVM regularization parameter by stratified five-fold MCC cross-validation on the train set. The QKSVM is then fitted on all train samples and scored on validation samples using the rectangular validation–train kernel whose size is $V \times N$. The candidate with the largest validation MCC is selected; ties prefer larger inner-Train CV MCC, smaller validation prototype loss. We set the SVM regularization grid in Eq. 21 and the CV folds are shared across candidates within each split. P2 is therefore a sample-trained QKSVM reference, not another prototype-classifier rule.

Validation cost. For each candidate circuit, the numbers of distinct fidelity evaluations required by the selectors are

$$Q_{\mathrm{P0}} = V + \binom{C}{2},$$
$$Q_{\mathrm{P1}} = CV + \binom{C}{2},$$
$$Q_{\mathrm{P2,\,SVM}} = \binom{N}{2} + NV.$$

The prototype-pair evaluations in P1 are used to normalize the margins for tie-breaking. If P0 evaluations are available, P1 requires only $(C-1)V$ additional sample–prototype fidelities. Self-fidelities equal one and need not be evaluated, and the same kernels are reused across SVM hyperparameters and CV folds. These counts describe logical fidelity evaluations rather than wall-clock runtime. For binary classification, P1 uses $2V$ sample–prototype evaluations and the corresponding multiclass cost is CV, in addition to the prototype-pair terms. P0 and P1 remain linear in V for fixed C, whereas P2 requires the dense training kernel.

After circuit selection, all three selectors use the same downstream QKSVM evaluation protocol described in Appendix D. The comparison therefore assesses the prototype classifier as an embedding-selection proxy, not as a replacement for the fitted QKSVM or a guarantee of its Test performance.

F T-MNIST Optimization Analysis

F.1 Why Prototype-Guided Optimization Performs well

We analyze the final T-MNIST embeddings reported in Table 1. For classes a and b, let

$$\Delta_{\Phi,ab} = \frac{1}{2}\left\|\rho_{\Phi,a} - \rho_{\Phi,b}\right\|_1, \qquad \rho_{\Phi,a} = \frac{1}{n_a}\sum_{i:y_i=a} \rho_\Phi(\mathbf{x}_i). \tag{22}$$

We evaluate this quantity exactly on the train set for all class pairs with the final selected embedding model after NQE.

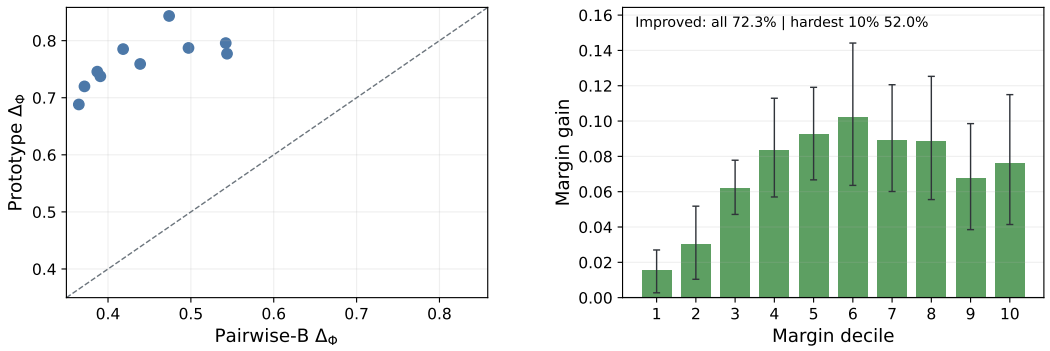

Figure 4: Geometric analysis for the T-MNIST NQE endpoints in Table 1. **(a)** Exact train class-pair ensemble distance $\Delta_{\Phi,ab}$. Each marker represents one of the ten class pairs and reports its average over ten seeds. **(b)** Mean Test margin gain, $\Delta\widetilde{m} = \widetilde{m}^{\text{Prototype}} - \widetilde{m}^{\text{Pairwise-B}}$, within deciles defined by the leave-one-seed-out *Pairwise-B* difficulty. Decile 1 contains the hardest 10% of Test samples. Error bars indicate 95% seed-bootstrap intervals.

To connect this Train-side geometry to the downstream classifier, we also measure the weakest one-vs-one (OvO) QKSVM margin of each Test sample. Because the SVM regularization parameter can differ between methods and seeds, we use the RKHS-normalized geometric margin

$$\widetilde{m}_i = \min_{k \neq y_i} \frac{d_{y_i,k}(\mathbf{x}_i)}{\|w_{y_i,k}\|_{\mathcal{H}}}, \tag{23}$$

where $d_{y_i,k}(\mathbf{x}_i) > 0$ favors the true class in the binary comparison with class k. For each seed, we estimate the difficulty of a test sample using its average *Pairwise-B* margin over the other nine seeds. We rank the samples by this independent reference and divide them into ten equal-sized groups, with decile 1 containing the hardest samples. Within each group, we compare the *Prototype* and *Pairwise-B* margins for the seed. Thus, the margin used to define sample difficulty does not come from the seed whose two methods are being compared.

Figure 4(a) shows that all ten class-pair markers lie above the diagonal. Thus, *Prototype* increases $\Delta_{\Phi,ab}$ for every class pair, rather than improving only the macro average. The average of exact class-ensemble distance increases from 0.443 to 0.764, accompanied by an increase in Test MCC from 0.642 to 0.779.

Figure 4(b) further shows how this improved class geometry is reflected in the decision boundaries. *Prototype* increases the geometric margin in 72.3% of the observations, and the mean margin gain is positive in all ten difficulty deciles. However, the gain is weakest in the hardest decile. Only 52% of its observations improve, and its mean gain is 0.0154 with a 95% interval of $[+0.0027, 0.0270]$. Therefore, *Prototype* improves class-ensemble separation and most test margins, but its advantage becomes less reliable near the most difficult decision boundaries. These findings provide a geometric account consistent with the *Prototype* advantage reported in Table 1.

F.2 Where Prototype-Guided Optimization Falls Short

We further examine why the favorable geometry produced by the prototype objective on T-MNIST does not translate monotonically into improved QKSVM performance in Figure 2. We use exactly the same circuit bank as in that analysis. For each of the ten GA seeds, the ten circuits with the lowest and highest inner-training prototype losses are retained.

For each circuit, we compute the RKHS-normalized geometric OvO margin $\widetilde{m}_i$ defined previously. Within each seed and loss group, we first average this margin over the ten circuits and then calculate the paired change for the same test sample,

$$\Delta\widetilde{m}_i = \widetilde{m}_i^{\text{Low}} - \widetilde{m}_i^{\text{High}}.$$

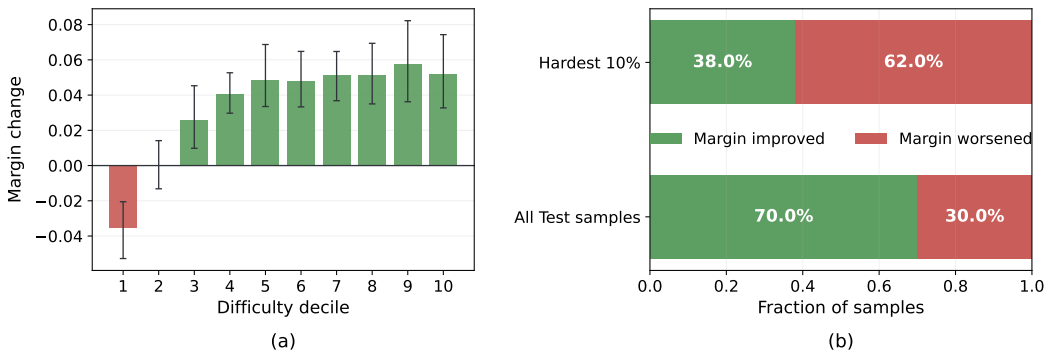

Figure 5: T-MNIST lower-margin-tail diagnostic using the all-depth circuit bank from Figure 2. **(a)** Mean paired change in RKHS-normalized geometric margin from High- to Low-loss circuits across Test-sample difficulty deciles. Decile 1 contains the hardest samples, as determined by the leave-one-seed-out High-loss reference. Error bars denote 95% seed-bootstrap intervals. **(b)** Fractions of paired seed–sample observations whose margin improves or worsens, evaluated over all Test samples and over the hardest decile. Circuit groups and QKSVM hyperparameters are fixed before Test evaluation.

A positive value therefore means that the Low-loss circuits improve the sample's weakest OvO boundary.

To avoid defining sample difficulty using the same noisy score being compared, we use a leave-one-seed-out reference. For seed s, each test sample is ranked by its mean High-loss margin over the other nine seeds and assigned to one of ten equally sized difficulty groups. Decile 1 contains the hardest 10% of samples. The Low-minus-High margin change for seed s is then evaluated within these independently determined groups.

Figure 5(b) shows that the discrepancy is concentrated in the lower-margin tail rather than distributed uniformly across the test set. Across all paired test observations, the Low-loss circuits improve the geometric margin in 70% of cases, increasing the mean margin from 0.0533 to 0.0873. Most difficulty deciles therefore benefit from the geometry favored by the prototype objective.

The behavior reverses, however, for the most difficult samples. Only 38% of the observations in the hardest decile improve, and their mean margin change is -0.0354. Consistently, the test lower-decile margin decreases from -0.0059 to -0.0614, while the fraction of samples with a nonpositive weakest margin increases from 12.0% to 19.2%. Thus, the increase in the average margin masks a deterioration near the most difficult OvO boundaries.

These results indicate an objective–downstream mismatch rather than a failure to optimize the prototype objective. Lower prototype loss improves the average prototype geometry and benefits most samples, but the mean-based objective does not explicitly protect the lower-margin tail that is particularly important to the QKSVM.