

000 001 QUAN-DORCET: TOURNAMENT-BASED 002 ONE-VS-ONE QUANTUM CLASSIFICATION FOR 003 ROBUST SINGLE-SHOT INFERENCE 004 005

006 **Anonymous authors**
007 Paper under double-blind review
008
009
010

011 ABSTRACT 012

013 Quantum machine learning (QML) promises powerful classification capabilities,
014 but suffers from fragile output encodings and high sampling demands—especially
015 in multiclass settings. Traditional schemes such as one-hot and binary encoding
016 either produce interpretable outputs too rarely or require many shots to achieve
017 reliable predictions. We propose a decision aggregation framework for quantum
018 multiclass classification based on round-robin tournament scoring. Each output
019 qubit represents a binary comparison between class pairs, and the final prediction is
020 determined by majority wins—yielding a Condorcet-style winner when one exists.
021 This structure improves both the resolvability and accuracy of single-shot predic-
022 tions, outperforming standard encodings under few-shot conditions. Our method
023 retains global entanglement while localizing decision tasks, enabling interpretable
024 inference that remains reliable under intrinsic quantum randomness, without sac-
025 rificing expressivity. Empirical results show that this approach achieves high
026 accuracy and interpretability with significantly fewer measurements, suggesting a
027 promising direction for future quantum classifiers.
028

029 1 INTRODUCTION 030

031 Quantum machine learning (QML) seeks to harness the unique properties of quantum systems—such
032 as superposition, entanglement, and interference—to perform learning tasks that may be intractable
033 for classical models. A central tool in quantum machine learning is the parameterized quantum
034 circuit (PQC), a variational quantum model that applies trainable quantum gates to optimize a task-
035 specific objective function (Cerezo et al., 2021a; Schuld and Petruccione, 2021). These circuits are
036 often trained using classical optimization techniques, and their outputs are typically interpreted via
037 expectation values or discretized measurements.

038 Despite the theoretical promise of QML, an inherent sampling bottleneck poses a fundamental
039 challenge for scalable quantum inference and will persist as a key consideration for future applications.
040 Quantum measurements collapse highly-expressive quantum states into binary outcomes, requiring
041 repeated executions of the circuit to draw samples—or shots—to estimate meaningful statistics
042 (Schuld and Petruccione, 2021). This limitation is particularly acute in multiclass classification,
043 where the structure of the output encoding plays a critical role. In one-hot encoding schemes, the
044 proportion of resolvable outputs—i.e., those that correspond to valid class predictions—vanishes
045 exponentially with the number of classes, making inference increasingly unreliable (Chen et al., 2024).
046 Binary encoding schemes, including standard and Gray codes, avoid this combinatorial collapse but
047 suffer from a different issue: individual bits are often noisy and weakly correlated with the true class,
048 leading to poor accuracy unless a large number of shots are used (LaRose and Coyle, 2020). These
049 problems are distinct but linked by a common theme: the difficulty of extracting reliable, discrete
050 decisions from quantum models under limited measurement budgets. Therefore, rather than focusing
051 on expectation values or aggregate statistics across many shots, we examine the quality of individual
052 measurement samples—what we call *single-shot inference* (not to be confused with few-shot regimes
053 referring to small training datasets).

To evaluate performance under these constraints, we introduce the metric of *shot resolvability*, defined
as the probability that a single measurement sample yields a valid and unambiguous class prediction.

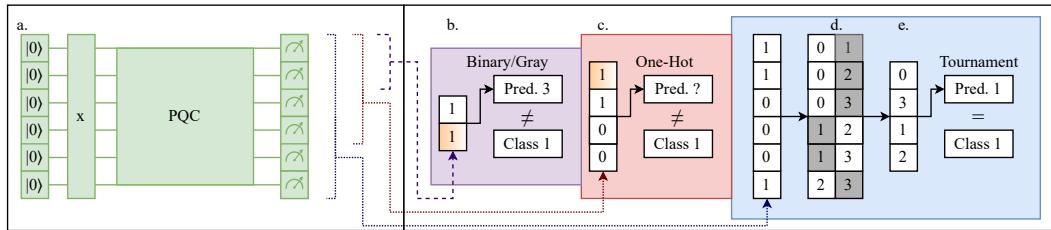


Figure 1: Contrived example illustrating common inference errors across different PQC output encoding strategies and the robustness of our tournament-based approach.

(a) *Non-contribution*: Standard PQC setup with angular-encoded inputs, a learnable circuit (we test six variants), and Pauli-Z basis measurements. (b) Binary (similarly Gray) encoding demonstrating misclassification—ie. predicting the wrong class. (c) One-hot encoding demonstrating a nonclassification—ie. predicting no class. (d) Our tournament mapping decomposes multiclass inference into pairwise quantum comparisons, where each output represents a vote between two classes (darker box indicates the chosen class). (e) Final class is determined by tallying votes across all comparisons. As the number of classes increases, the tournament structure introduces redundancy that helps mitigate both misclassification and nonclassification, improving single-shot inference reliability.

We address the challenge of shot resolvability by introducing a decision aggregation framework for quantum multiclass classification. Rather than relying on global output encodings, our method decomposes the classification task into a series of binary comparisons between class pairs. Each comparison is implemented as a binary quantum classifier operating on a shared entangled state. The outputs of these classifiers are aggregated using a round-robin tournament structure, where each class competes against every other, and the final prediction is determined by majority wins (David, 1959). This approach leverages the statistical robustness of binary decisions and the emergent structure of tournament theory, which ensures that as the number of classes grows, the likelihood of a unique Condorcet-style winner converges to unity while being bound below by 50% (Malinovsky and Moon, 2024).

Importantly, this framework does not discard the global coherence of the quantum model. All classifiers operate within the same entangled quantum state, allowing input information to propagate across the full Hilbert space. The aggregation mechanism simply localizes the decision task, enabling interpretable inference that remains reliable under intrinsic quantum randomness, without sacrificing expressivity. Empirical results show that this method significantly improves accuracy under few-shot regimes, with a particular emphasis on single-shot reliability, outperforming traditional encoding schemes in both sample efficiency and decision consistency.

Our results assume idealized, noiseless conditions to isolate algorithmic behavior from hardware-specific noise. Hardware applicability and noise resilience remain open challenges. Furthermore, our method introduces a trade-off between quadratic qubit scaling and exponential sampling cost. We discuss these limitations in Section 5.

This paper makes the following contributions:

- We introduce *shot resolvability* as a key metric for evaluating the reliability of single-shot predictions in quantum classifiers, providing a practical lens for assessing inference quality under limited measurement budgets.
- We propose a novel output encoding for black-box variational quantum classifiers (VQCs) based on round-robin tournament scoring, leveraging the statistical properties of Condorcet-style decision aggregation to improve both resolvability and accuracy.
- We develop a differentiable training procedure for our tournament-based encoding by embedding pairwise class comparisons into a continuous simplex structure, enabling end-to-end optimization via standard backpropagation.
- We present a comprehensive empirical evaluation across multiple circuit architectures and datasets, demonstrating that our method consistently outperforms standard global encoding schemes in single-shot regimes

108

2 RELATED WORK

109

110 2.1 QUANTUM CLASSIFICATION

111
112 Quantum machine learning (QML) has produced a wide range of classification models, including
113 quantum adaptations of support vector machines (Rebentrost et al., 2014), convolutional neural
114 networks (Cong et al., 2019; Bokhan et al., 2022a), and generative models (Benedetti et al., 2019).
115 Many of these rely on hybrid architectures, where a parameterized quantum circuit (PQC) is embedded
116 within a classical pipeline (Chalumuri et al., 2021; Stein et al., 2022; Shi et al., 2023; Liu et al., 2021).
117 While effective in simulation, hybrid models typically depend on expectation values or floating-point
118 outputs, which require extensive sampling.119 Recent work has explored direct multiclass classification using PQCs without hybridization (Zhou
120 et al., 2023; Hur et al., 2022; Shen et al., 2024), but these approaches often rely on thresholding or
121 maximum selection over expectation values, which again necessitate high shot counts. Moreover,
122 most prior methods use global output encodings such as one-hot or binary schemes, which suffer
123 from either low resolvability or poor robustness to bit-level noise (Chen et al., 2024; LaRose and
124 Coyle, 2020; Di Matteo et al., 2021). Some recent efforts have explored alternative encodings such
125 as amplitude-based or angle-based schemes (Schuld et al., 2020), but these typically require deeper
126 circuits or more complex post-selection.127

128 2.2 OUTPUT ENCODINGS IN QUANTUM CLASSIFICATION

129 Most prior work in quantum multiclass classification relies on global output encodings such as
130 one-hot, binary, or Gray code representations. One-hot encoding is conceptually simple and widely
131 used in both classical and quantum settings (Bokhan et al., 2022a; Dhara et al., 2024), but suffers
132 from exponentially vanishing resolvability as the number of classes increases, since only K out of 2^K
133 bitstrings correspond to valid outputs (Chen et al., 2024). Binary encoding is more qubit-efficient
134 (often fully valid), but is highly sensitive to bit-flip noise and Hamming-distance errors, which can
135 cause semantically large misclassifications from single-bit perturbations (LaRose and Coyle, 2020;
136 Ding et al., 2025). Gray code encoding mitigates some of this sensitivity by ensuring adjacent class
137 labels differ by only one bit, and has been used in quantum classification tasks (Di Matteo et al.,
138 2021; Bokhan et al., 2022a), but still lacks semantic structure and remains vulnerable to cumulative
139 noise in few-shot regimes.140 To improve single-shot reliability, we draw inspiration from tournament theory. Round-robin tour-
141 naments have long been studied as a framework for pairwise comparison and ranking (Zermelo,
142 1929; David, 1959). Recent results show that the probability of a unique Condorcet-style winner in a
143 random tournament converges to unity as the number of classes increases (Malinovsky and Moon,
144 2024). We leverage this structure to design a decision aggregation framework in which each output
145 qubit represents a binary comparison between class pairs, and the final prediction is determined by
146 majority wins.147

148 3 METHOD

149 We will first introduce the challenges with using existing output encodings for single-shot inference
150 in Section 3.1. Following this, we will present the theory which leads to the improvement in
151 inference of our tournament method over the baselines in Section 3.2. We will then introduce the
152 post-processing method used to differentiably train a PQC to output quality round-robin tournament
153 results in Section 3.3, as well as other training decisions. After the main contributions, we discuss the
154 circuit setup and variations in Section 3.4 and the computational tools used in Section 3.5. These
155 sections are included to increase reproducibility, however for the interested reader, we have expanded
156 upon the intricacies of the the circuit blocks we choose in Section A.3 and on quantum computation
157 in general in Section A.4 .158

159 3.1 OUTPUT ENCODING EFFECTS ON SHOT RESOLVABILITY

160 In quantum multiclass classification, the choice of output encoding plays a critical role in determining
161 both the resolvability of measurement outcomes and the accuracy of predictions under limited

162 sampling. We evaluate four encoding strategies—one-hot, binary, Gray code, and our proposed
 163 tournament-based encoding—with a focus on their behavior under few-shot and single-shot inference
 164 regimes.

165 *One-Hot* - One-hot encoding assigns each class to a unique qubit, with the correct class represented
 166 by a single qubit in the excited state (e.g., $|1\rangle$) and all others in the ground state ($|0\rangle$). This encoding
 167 is conceptually simple and widely used in classical multiclass classification. However, in quantum
 168 settings, it suffers from a severe validity bottleneck: only K out of 2^K possible bitstrings correspond
 169 to valid one-hot outputs, where K is the number of classes. Thus, the probability of obtaining a
 170 resolvable output from a random measurement decays exponentially as $P_{\text{valid}} = K/2^K$, making
 171 inference unreliable under few-shot conditions (Chen et al., 2024).

172 *Binary and Gray Code Encoding* - Both binary encodings map each class label to a binary rep-
 173 resentation across $\lceil \log_2 K \rceil$ qubits. These encodings are highly efficient in terms of qubit usage
 174 and have maximal resolvability: every bitstring corresponds to a class label, modulo padding for
 175 non-power-of-two class counts. However, robustness to sampling variability under few-shot inference
 176 is poor. Individual qubits contribute to multiple bits of the class label, and noise in any bit can lead to
 177 misclassification. Moreover, binary encoding is sensitive to Hamming distance errors, where small
 178 perturbations in the bitstring can result in large semantic shifts in class prediction (LaRose and Coyle,
 179 2020).

180 Gray code encoding modifies binary encoding such that consecutive class labels differ by only one
 181 bit. This reduces the impact of single-bit errors, improving robustness under low-shot conditions.
 182 However its accuracy gains under single-shot inference are modest and context-dependent. In our
 183 experiments, Gray code fails to unilaterally outperform standard binary encoding in single-shot
 184 accuracy, and still falls short of our tournament-based method.

186 Table 1: Comparison of output encoding strategies in terms of validity and accuracy under few-shot
 187 and many-shot regimes.

189 Encoding Method	190 Resolvability	191 Accuracy (Single-Shot)	192 Accuracy (Many-Shot)
190 One-Hot	191 Low ($\sim K/2^K$)	192 Low	193 High
190 Binary	191 High (Full coverage)	192 Moderate	193 Moderate–High
190 Gray Code	191 High (Full coverage)	192 Moderate–Low	193 Moderate–High
190 Tournament (Ours)	191 High ($\rightarrow 1$ as $K \rightarrow \infty$)	192 High	193 High

195 3.2 TOURNAMENT-BASED ENCODING (OURS)

197 Our proposed encoding frames multiclass classification as a round-robin tournament among class
 198 pairs (Moon et al., 1968). Each output bit represents a binary decision between two classes, and the
 199 final prediction is determined by majority wins (Copeland-style). This structure corresponds to an
 200 orientation of a complete directed graph over K vertices, with $K(K - 1)/2$ binary decisions. While
 201 our method does not require a Condorcet winner to produce a prediction, Condorcet theory provides
 202 the probabilistic guarantees that underpin tournament encoding and motivate the Quan-dorcer design.

204 Theoretical results from Malinovsky and Moon (2024) show that the probability of a unique winner
 205 in a random tournament converges to 1 as $K \rightarrow \infty$. This implies that even stochastic or partially
 206 incorrect binary decisions can yield a resolvable class prediction, meaning the convergence properties
 207 remain valid regardless of hardware noise or backend fidelity. In this work, we evaluate the method
 208 under idealized, noiseless conditions to isolate algorithmic behavior and demonstrate these properties
 209 empirically. As illustrated in Figure 2, even when cycles occur among some pairwise comparisons,
 210 the Copeland-style aggregation still produces a unique prediction. The example shows a $K = 4$
 211 tournament where three edges agree and the remaining form a cycle, demonstrating that cycles do not
 212 prevent resolvability under our framework.

213 Unlike one-hot or binary encodings, tournament-based encoding does not require global agreement
 214 across qubits. Each decision is localized, yet the model retains global coherence via shared entan-
 215 glement. This duality of local decision simplicity with global state expressivity is a key factor in
 the superior performance of the method. Empirically, our results in Section 4.2 and Section 4.3

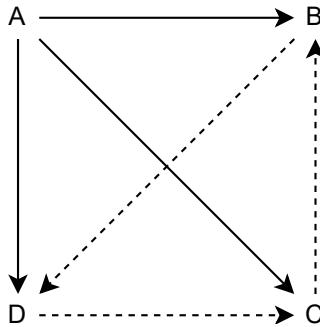


Figure 2: A round-robin tournament of four classes. Arrows indicate the direction of pairwise preference (from the class that wins the comparison to the class that loses). Note the cycle (dashed) among classes B, C, and D—B beats D, D beats C, and C beats B—where no Condorcet winner exists. However, class A defeats all others and is selected as the tournament winner by majority vote (Copeland-style). If A had lost to any one of the other three, the result would be a tie, rendering the tournament unresolvable under our framework.

show our method achieves high resolvability and accuracy even with a single shot, outperforming other encodings in low-shot regimes while matching their performance under high sampling and simulation.

3.3 TRAINING PROCEDURES

All training is conducted under noiseless simulation using the same PQC architecture described in Section 3.4. This standard practice in quantum machine learning isolates algorithmic behavior from hardware-specific noise and ensures fair comparison across encodings. Our focus is on how each output encoding is interpreted and optimized during training, giving each method the best opportunity to perform under its own assumptions.

All encodings require a continuous representation for gradient-based optimization. For binary encodings (binary and Gray), this is straightforward: the model outputs continuous values that can be trained using binary cross-entropy or distance-based losses against a known bitstring target. One-hot encoding, widely used in both classical and quantum classification (Bokhan et al., 2022a; Chen et al., 2024), corresponds to moving the center of mass of a probability simplex toward the correct vertex, and is typically trained using cross-entropy loss.

Our tournament-based encoding presents a unique challenge: it produces a vector of binary comparisons between class pairs, many of which are undefined for a given target class. Specifically, only the comparisons involving the true class c_k have a well-defined target; the rest are structurally ambiguous. This makes it inappropriate to apply a bitwise loss across all outputs. To address this, we introduce a novel continuous training method that leverages the geometric correspondence between round-robin tournaments and the edges of a regular simplex. By interpolating each binary comparison along the edge connecting its two associated class vertices, we construct class-specific mass points within the simplex. This allows us to compute distances to the true class vertex and apply a softmax-based loss, analogous to one-hot training.

Formally, the PQC outputs expectation values $\langle Z_i \rangle$ for each qubit i , corresponding to binary comparisons between class pairs (c_i, c_j) . These are passed through a nonlinear activation function $\phi : [-1, 1] \rightarrow [0, 1]$ (see Section A.1.3) to produce confidence scores $e_{ij} = \phi(\langle Z_{ij} \rangle)$. Each score is used to interpolate between the vertices v_i and v_j of a regular, zero-centered $(K - 1)$ -simplex:

$$\mathbf{p}_{\{i,j\}} = (1 - e_{\{i,j\}})\mathbf{v}_i + e_{\{i,j\}}\mathbf{v}_j.$$

For each class c_k , we compute the average of the interpolated points along its incident edges:

$$\mathbf{n}_k = \frac{1}{K - 1} \sum_{j \neq k} \mathbf{p}_{\{k,j\}}.$$

270 This yields a class-specific mass point \mathbf{n}_k within the simplex. We then compute the Euclidean
 271 distance between each class’s mass point and its corresponding vertex:
 272

$$273 \quad d_k = \|\mathbf{n}_k - \mathbf{v}_k\|,$$

274 and apply a softmax transformation to the inverted distances to produce class scores:
 275

$$276 \quad p_k = \frac{\exp(1 - d_k)}{\sum_{j=1}^K \exp(1 - d_j)}. \\ 277 \\ 278$$

279 These scores are used in a symmetric cross-entropy loss:
 280

$$281 \quad \mathcal{L} = \sum_{k=1}^K [y_k \log(p_k) + (1 - y_k) \log(1 - p_k)], \\ 282 \\ 283$$

284 where y_k is the one-hot target label for class k . This formulation retains the benefits of softmax
 285 normalization while preserving class-specific optimization manifolds. Unlike standard cross-entropy,
 286 which only penalizes incorrect predictions, symmetric cross-entropy encourages confident separation
 287 between correct and incorrect classes. This is particularly beneficial in our setting, where each class is
 288 defined by its incident binary comparisons. Our observations are consistent with prior work showing
 289 that symmetric cross-entropy improves class separation and robustness to sampling variability (Wang
 290 et al., 2019; Das and Chaudhuri, 2019; Huang et al., 2020).

291 One-hot training uses the same softmax symmetric cross-entropy directly on the activations of the
 292 expectation values $\langle Z_i \rangle$ from the PQC. For binary and Gray code encodings, we use the same
 293 symmetric cross-entropy formulation, omitting the softmax normalization step, as the targets are
 294 bitstrings rather than one-hot vectors.

295 Each PQC model is trained with a batch size of 32 for 6 epochs using the Adam optimizer with
 296 an exponential decay learning rate scheduler (Kingma and Ba, 2017), with a starting learning rate
 297 of 0.01, a decay rate of 0.9, and scheduler steps equal to one-tenth of the total training steps. This
 298 configuration was selected based on an ablation study in Section A.1.4 .

300 3.4 CIRCUIT DESIGN

301 To ensure consistency and comparability across encoding methods, all experiments use a shared
 302 PQC architecture. We adopt the dual-angle encoding scheme from Hur et al. (2022); Munikote
 303 (2024), which has demonstrated strong performance in prior work. Input features are encoded using
 304 $W = \binom{K}{2}$ qubits, where K is the number of classes. Each qubit receives two features—one via a
 305 Pauli-X rotation and one via a Pauli-Y rotation—yielding a total of $2W$ encoded features. Input
 306 data is scaled to the range $[-1, 1]$ to ensure unique embeddings, and dimensionality reduction is
 307 performed using a reproducible autoencoder with dropout (Bishop, 2006).

308 The main circuit topology is a 2-design qubit ring (Cerezo et al., 2021b), where each wire is
 309 connected to its two neighbors via alternating layers of computational blocks. These blocks consist
 310 of parameterized single-qubit rotations and two-qubit controlled operations. We evaluate six well-
 311 established block types: CNN7 and CNN8 (Sim et al., 2019; Hur et al., 2022), $SO(4)$ and $SU(4)$
 312 (Wei and Di, 2012; Vatan and Williams, 2004). We also test on a slightly different multi-qubit
 313 entangling setup known as Strongly Entangling Layers (Schuld et al., 2020). This setup applies
 314 parameterized $SU(2)$ rotations on each individual qubit, then applies a 2-qubit controlled gate to each
 315 consecutive pair of qubits. We test this setup with both CNOT and controlled-Z gates as the 2-qubit
 316 gates. Descriptions and diagrams of each block are provided in Section A.3 , and a schematic of the
 317 overall setup is shown in Figure 1. We use four layers of ring blocks or SEL layers in all experiments,
 318 though this depth can be adjusted to trade off expressivity and gate cost, as shown in Section A.1.1 .
 319 Importantly, our results are not tied to any specific circuit block or depth—our method operates as a
 320 post-processing framework and is compatible with a wide range of architectures.

321 Measurement strategies differ slightly between encoding methods: the one-hot framework mea-
 322 sures K qubits corresponding to class vertices, binary and Gray frameworks measure $\lceil \log_2 K \rceil$, and
 323 the tournament framework measures all W qubits. Measuring a subset of wires is standard practice in
 PQC training (Hur et al., 2022; Bokhan et al., 2022b; Zhou et al., 2023; Shen et al., 2024; Stein et al.,

324 2022), and has even been linked to improved gradient behavior and reduced barren plateau effects
 325 (Cerezo et al., 2021b; Leone et al., 2024; Cerezo et al., 2024). All measurements are performed in the
 326 Pauli-Z basis.

327

328

329

3.5 COMPUTATIONAL TOOLS

330

331 All experiments were performed using the Python packages JAX (Bradbury et al., 2018) and Penny-
 332 Lane (Bergholm et al., 2022). JAX is an auto-differentiation package that enables the computation of
 333 gradients for machine learning models and just-in-time compilation for highly parallel processes such
 334 as batched PQC operations. PennyLane is a superconducting quantum computing package for Python
 335 that interfaces with most modern superconducting quantum computer APIs and machine learning
 336 packages, including JAX, which enables rapid training and testing of the PQCs used in this work.
 337 The full spread of experiments was obtained from 100 kCPU-hours on two Intel Xeon Gold 6130s,
 with another 300 used for the ablations.

338

339 Due to our primary contributions being post-processing methods, our main results are computed under
 340 noiseless CPU simulation, though our ablation in Section A.1.2 shows that the relative inference
 341 performance between the tournament and one-hot methods changes very little when performed using
 342 (retired) IBM noise models. In practice, simulation remains feasible for small K (e.g. $K \leq 6$) and
 343 shallow circuits (≤ 4 layers), which we report in Section 4. These regimes reflect the intended scope
 344 of this study: evaluating encoding strategies under strict sampling constraints rather than optimizing
 for hardware execution. Code is provided on (Anonymous) GitHub.

345

346

347

4 RESULTS

348

349 To evaluate the performance of our tournament-based decision aggregation framework, we compare
 350 it against one-hot, binary, and Gray code. We train and test on five permutations of two datasets and
 351 six block circuits each, using the same random seeds for each permutation for direct comparison of
 352 the methods. Tests were done on both the MNIST Digits dataset (LeCun et al., 1998) and the MNIST
 353 Fashion dataset (Xiao et al., 2017), with five random subsets of K -classes. The same five random
 354 subsets were used for each permutation with additional classes chosen from the remaining digits
 355 as K increases. These chosen datasets are balanced and have clean labels to isolate encoding effects.
 356 Robustness to imbalance and overlapping classes is an open question and discussed in Section 5.

357

358

359

4.1 METRICS

360

361 We report results using metrics designed to reflect the reliability of discrete predictions under limited
 362 sampling. The resolvable accuracy A_R measures the proportion of resolvable measurement shots that
 363 yield the correct class label, capturing per-shot correctness. The resolvability ratio R quantifies the
 364 fraction of resolvable measurement shots, and is calculated by measuring until 100 resolvable shots
 365 are collected, and dividing 100 by the number of shots needed to be measured to acquire those 100.

366

367 To assess single-shot performance more directly, we also present the shot accuracy A_s , computed
 368 over a fixed number of shots per test sample (e.g., 100), including unresolvable predictions. This
 369 metric reflects the probability that a single shot yields a correct prediction, and serves as our primary
 370 measure of single-shot inference quality. Additionally, we define the effective accuracy $A_e = RA_R$.
 371 This metric captures the expected correctness of a randomly sampled shot. While their definition
 372 implies $A_s \approx A_e$, our results show empirically that this is not the case, and that there is a correlation
 373 between resolvability and accuracy.

374

375

376

4.2 RESOLVABILITY

377

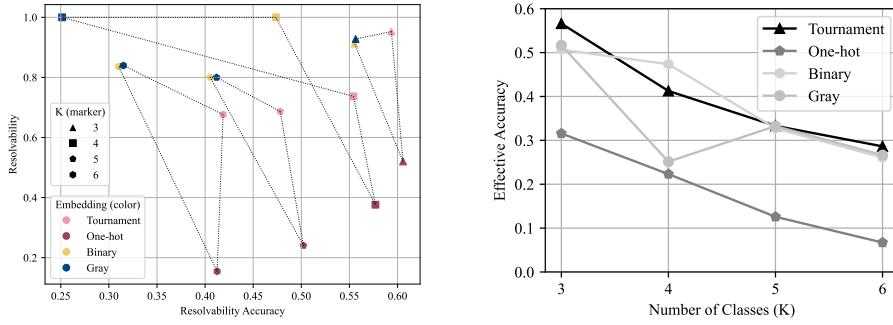
378 In Table 2, we report the resolvability ratio R and the resolvable-shot accuracy A_R . The relationship
 379 between the two is plotted on the left part of Figure 3. Tournament encoding consistently achieves
 380 high resolvability and strong per-shot correctness across all experiments. In contrast, binary and Gray
 381 encodings—despite producing resolvable outputs—exhibit lower accuracy due to their sensitivity to
 382 sampling fluctuations and lack of semantic structure in the output space.

378
379
380
381
382

Table 2: Comparison of our proposed tournament encoding against common encodings when looking for the ratio of resolvable outputs and their class accuracy. Tournament encoding demonstrates reliable performance over both metrics, while one hot has high accuracy with less resolvability, and the two binary-methods are highly resolvable but less accurate.

383
384
385
386
387
388
389
390
391

	R				A_R			
	Tournament	One-Hot	Binary	Gray	Tournament	One-Hot	Binary	Gray
Digits	3	93.16%	57.41%	92.04%	90.64%	58.27%	60.64%	54.50%
Digits	4	71.11%	38.70%	100.00%	100.00%	57.55%	60.25%	49.04%
Digits	5	66.01%	23.47%	78.43%	78.32%	47.36%	49.65%	39.44%
Digits	6	67.14%	15.54%	83.17%	83.69%	42.63%	41.44%	31.02%
Fashion	3	97.38%	46.62%	90.38%	94.92%	60.46%	60.51%	56.90%
Fashion	4	76.24%	36.55%	100.00%	100.00%	53.28%	55.16%	45.68%
Fashion	5	71.36%	24.63%	81.61%	81.65%	48.26%	50.78%	41.63%
Fashion	6	68.11%	15.33%	84.03%	84.22%	41.11%	41.06%	31.01%

392
393
394
395
396
397
398
399
400
401
402
403404
405
406
407
408
409Figure 3: *Left*: Plot showing the spread of resolvability and resolvable accuracy across all methods and K . Notice that the tournament method is always in the top-right corner, indicating solid reliability. *Right*: Plot showing trends in effective accuracy as K increases. While calculated from the left plot, clearly shows the consistency of the tournament method as compared to the erratic behavior of binary methods and the poor performance of one-hot.410
411
412
413
414
415
416

One-hot encoding suffers from low resolvability, particularly as the number of classes increases. While resolved shots tend to be accurate—indicating that the correct qubit is often excited—many incorrect qubits are also excited simultaneously. This reflects a fundamental limitation of one-hot encoding: it attempts to represent a confidence distribution over classes, but any nonzero confidence in an incorrect class can lead to misclassification. Inference requires extensive sampling to recover the dominant excitation.

417
418
419
420
421

These results confirm that tournament encoding uniquely balances resolvability and correctness, yielding interpretable and accurate predictions with minimal sampling. Importantly, this balance is maintained even as K increases, unlike binary encodings which exhibit sharp structural degradation. This stepwise breakdown is a direct consequence of the discrete nature of $\lceil \log_2 K \rceil$ and foreshadows the scaling limitations of binary methods.

422
423
424

4.3 SHOT QUALITY ANALYSIS

425
426
427
428
429

To further illustrate the trade-offs between encodings, we combine the resolvability metrics into effective accuracy $A_e = RA_R$ and compare it to the measured shot accuracy A_s . These metrics are similar when resolvability is high, but diverge as resolvability drops, since A_s includes unresolvable outputs. As shown in Table 3 and the right side of Figure 3, tournament encoding maintains strong performance across all class counts, while one-hot encoding degrades sharply.

430
431

Notably, binary encoding performs well at $K = 4$, but its accuracy drops significantly at $K = 5$, coinciding with a steep decline in resolvability. This highlights the importance of resolvability as a metric: binary encodings are structurally bound to degrade as K increases. Gray encoding

432 exhibits similar behavior, with accuracy falling below random guessing in some cases, suggesting
 433 that semantic distance between classes is not preserved under sampling variability.
 434

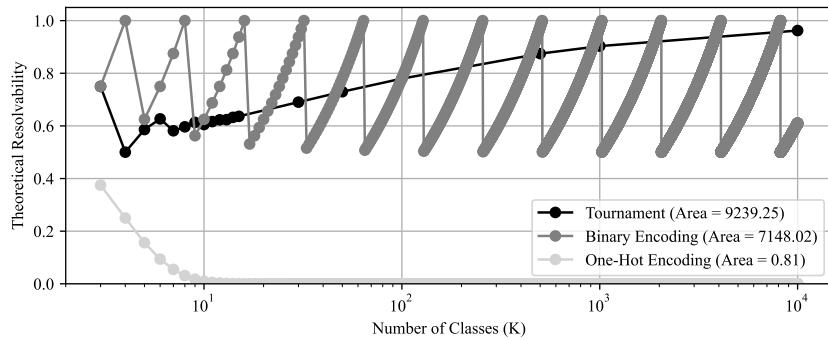
436 Table 3: Comparison of our proposed tournament encoding against common encodings when looking
 437 at the overall accuracy of discrete PQC accuracies. Shot accuracy and A_e are theoretically equal
 438 when resolvability is maximal, leading to a correlation for tournament, binary, and Gray encodings
 439 on the (far simpler) Digits dataset. One-hot degrades as resolvability goes down.

		A_s				$A_e = RA_R$			
		Tournament	One-Hot	Binary	Gray	Tournament	One-Hot	Binary	Gray
Digits	3	54.62%	38.82%	50.41%	49.93%	54.33%	34.89%	50.18%	49.33%
Digits	4	42.89%	27.41%	49.04%	25.05%	41.44%	24.03%	49.04%	25.05%
Digits	5	32.60%	14.86%	31.72%	32.01%	31.68%	12.26%	31.26%	31.44%
Digits	6	29.63%	8.21%	26.23%	26.75%	28.93%	6.81%	25.96%	26.47%
Fashion	3	59.03%	41.59%	52.27%	54.49%	58.90%	28.26%	51.15%	54.02%
Fashion	4	41.95%	25.02%	45.68%	25.22%	41.08%	20.60%	45.68%	25.22%
Fashion	5	35.73%	16.59%	34.68%	35.40%	34.93%	12.90%	34.20%	35.05%
Fashion	6	29.05%	8.28%	26.49%	27.13%	28.34%	6.59%	26.19%	26.79%

440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450 These trends reinforce the practical advantage of tournament encoding: it produces high-quality
 451 predictions without requiring filtering or extensive measurement. Unlike binary methods, tournament
 452 aggregation does not rely on a fixed bitstring structure and instead leverages pairwise comparisons,
 453 which scale more gracefully with K . This motivates a deeper analysis of scaling behavior, which we
 454 explore in the next section (Section 4.4) by examining the area-under-curve (AuC) of resolvability
 455 for each method.

456 4.4 SCALING BEHAVIOR AND STRUCTURAL LIMITS

457 While our experiments focus on relatively small class counts ($K \leq 6$), the structural implications
 458 of each encoding become increasingly important as K grows. Binary and Gray encodings exhibit a
 459 discrete-to-exponential mismatch: the number of valid class labels grows linearly with K , while the
 460 number of possible bitstrings grows exponentially with the number of qubits. This leads to a bounded
 461 oscillatory degradation in resolvability, particularly when $\log_2 K$ is not an integer. For example,
 462 binary encoding achieves full validity at $K = 4$ (using 2 qubits), but drops to 62.5% validity at
 463 $K = 5$ (using 3 qubits), as only 5 of the 8 possible bitstrings correspond to valid class labels. This
 464 structural fragility implies that binary encodings are inherently sensitive to class count and qubit
 465 budget. For non-power-of-two K , the fraction of valid bitstrings decreases, and the probability of
 466 generating an unresolvable output rises sharply. This behavior is not merely empirical—it is a direct
 467 consequence of the encoding scheme’s discrete nature. We present the theoretical lower bounds for
 468 the resolvability probability of random bitstrings in Figure 4.



486 Moreover, the aforementioned results by Malinovsky and Moon (2024) show that the likelihood of
 487 Condorcet-style aggregation converges to unity as $K \rightarrow \infty$. This convergence property is absent in
 488 binary encodings, for which there is an oscillatory bounded curve for all K . For one-hot, we use
 489 $K/2^K$, which decreases exponentially until vanishing.
 490
 491

492 5 LIMITATIONS

493
 494 Our proposed tournament-based encoding introduces a fundamental trade-off: quadratic qubit scaling
 495 with respect to the number of classes K . This requirement makes the approach impractical for
 496 large-scale problems until quantum hardware achieves significant improvements in qubit availability
 497 and fidelity. Consequently, all results in this paper are obtained under idealized conditions to isolate
 498 algorithmic behavior from hardware-specific noise. While this choice enables a clear evaluation of
 499 encoding strategies, robustness to real-device imperfections and resource constraints—both general
 500 and tournament-specific—remains an open challenge.
 501

502 These constraints position our work as a theoretical analysis of output encodings rather than a direct
 503 path to near-term hardware deployment. The guarantees we provide, such as the convergence of
 504 resolvability to unity as $K \rightarrow \infty$ (Malinovsky and Moon, 2024), are purely combinatorial and
 505 hold regardless of backend fidelity or noise. Our empirical evaluation under noiseless simulation
 506 demonstrates these properties in practice, but does not claim hardware readiness.
 507

508 In addition, our experiments assume balanced datasets with clean labels. Class imbalance and
 509 semantic overlap introduce structural challenges: imbalance may bias majority voting toward dominant
 510 classes, while overlapping decision boundaries can increase the likelihood of cycles, which our current
 511 framework discards as “unresolvable.” These effects are not unique to quantum classifiers—they also
 512 affect classical one-vs-one schemes—but their impact on resolvability and accuracy under tournament
 513 aggregation remains an open question. We highlight these limitations explicitly and view extensions
 514 such as weighted voting, cycle-aware heuristics, and adaptive tie-breaking as promising directions for
 515 future work.
 516

517 Future research should also explore strategies to mitigate quadratic scaling, such as hierarchical
 518 or sparse tournament structures, hybrid aggregation schemes, and alternative scoring mechanisms.
 519 Extensions inspired by classical tournament theory (e.g., Condorcet-cycle handling, Schulze methods)
 520 offer promising directions for improving both efficiency and resolvability. Assuming continued
 521 progress in quantum hardware and deeper theoretical development, a large-scale experimental study
 522 on real quantum processors would be a natural next step. Such work is essential before deploying
 523 Quan-dorcer-style models on high-dimensional datasets or production-level tasks.
 524

525 Finally, we acknowledge that a language model was used to refine the clarity and consistency of
 526 the manuscript. All conceptual contributions, experimental design, and theoretical insights remain
 527 entirely our own.
 528

529 6 CONCLUSIONS

530 In this paper, we take the first step toward improving the resolvability and accuracy of discrete outputs
 531 from multi-class PQC classifiers. Achieving this goal has broader implications for quantum machine
 532 learning, as reducing sampling requirements removes a significant obstacle to quantum computing.
 533 Our findings, supported by experiments, highlight a novel direction in quantum machine learning.
 534 We focus on designing models that yield resolvable and accurate discrete outputs more often by
 535 leveraging tournament solutions.

536 To achieve this, we propose a classical post-processing method for PQCs that maps the output
 537 space to a regular simplex, leading to the model learning a probabilistic directed graph over classes.
 538 Under deterministic inference, such models produce resolvable samples whenever there is a unique
 539 majority “winner,” rather than only when an exact bitstring is produced. This effect upper bounds
 540 sampling needs as the number of classes increases while still producing highly accurate single-shot
 541 measurements, as compared to stand one-hot or bitstring based methods.

540 REFERENCES
541

542 Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam,
543 and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow
544 quantum circuits. *npj Quantum Inf*, 5(1):45, May 2019. ISSN 2056-6387. doi:10.1038/s41534-
545 019-0157-8. URL <https://www.nature.com/articles/s41534-019-0157-8>.

546 Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith,
547 M. Sohaib Alam, Guillermo Alonso-Linaje, B. Akash Narayanan, Ali Asadi, Juan Miguel Arrazola,
548 Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R Bromley, Benjamin A. Cordier, Jack
549 Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony
550 Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isaacsson, David Ittah, Soran Jahangiri, Prateek Jain,
551 Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, Robert A. Lang, Christina Lee, Thomas
552 Loke, Angus Lowe, Keri McKiernan, Johannes Jakob Meyer, J. A. Montañez-Barrera, Romain
553 Moyard, Zeyue Niu, Lee James O’Riordan, Steven Oud, Ashish Panigrahi, Chae-Yeon Park,
554 Daniel Polatajko, Nicolás Quesada, Chase Roberts, Nahum Sá, Isidor Schoch, Borun Shi, Shuli
555 Shu, Sukin Sim, Arshpreet Singh, Ingrid Strandberg, Jay Soni, Antal Száva, Slimane Thabet,
556 Rodrigo A. Vargas-Hernández, Trevor Vincent, Nicola Vitucci, Maurice Weber, David Wierichs,
557 Roeland Wiersema, Moritz Willmann, Vincent Wong, Shaoming Zhang, and Nathan Killoran.
558 PennyLane: Automatic differentiation of hybrid quantum-classical computations, 2022. URL
559 <https://arxiv.org/abs/1811.04968>.

560 Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:
561 1122–1128, 2006.

562 Denis Bokhan, Alena S Mastiukova, Aleksey S Boev, Dmitrii N Trubnikov, and Aleksey K Fedorov.
563 Multiclass classification using quantum convolutional neural networks with hybrid quantum-
564 classical learning. *Frontiers in Physics*, 10:1069985, 2022a.

565 Denis Bokhan, Alena S. Mastiukova, Aleksey S. Boev, Dmitrii N. Trubnikov, and Aleksey K.
566 Fedorov. Multiclass classification using quantum convolutional neural networks with hy-
567 brid quantum-classical learning. *Front. Phys.*, 10:1069985, November 2022b. ISSN 2296-
568 424X. doi:10.3389/fphy.2022.1069985. URL <http://arxiv.org/abs/2203.15368>.
569 arXiv:2203.15368 [quant-ph].

570 James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclau-
571 rin, and Skye Wanderman-Milne. Jax: Composable transformations of python+numpy programs,
572 2018. URL <http://github.com/google/jax>. Version 0.2.20, Software.

573 M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fu-
574 jii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles.
575 Variational Quantum Algorithms. *Nat Rev Phys*, 3(9):625–644, August 2021a. ISSN 2522-
576 5820. doi:10.1038/s42254-021-00348-9. URL <http://arxiv.org/abs/2012.09265>.
577 arXiv:2012.09265 [quant-ph, stat].

578 M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent
579 barren plateaus in shallow parametrized quantum circuits. *Nat Commun*, 12(1):1791, March 2021b.
580 ISSN 2041-1723. doi:10.1038/s41467-021-21728-w. URL <https://www.nature.com/articles/s41467-021-21728-w>. Publisher: Nature Publishing Group.

581 M. Cerezo, Martin Larocca, Diego García-Martín, N. L. Diaz, Paolo Braccia, Enrico Fontana,
582 Manuel S. Rudolph, Pablo Bermejo, Aroosa Ijaz, Supanut Thanasilp, Eric R. Anschuetz, and Zoë
583 Holmes. Does provable absence of barren plateaus imply classical simulability? Or, why we
584 need to rethink variational quantum computing, March 2024. URL <http://arxiv.org/abs/2312.09121>. arXiv:2312.09121 [quant-ph, stat].

585 Avinash Chalumuri, Raghavendra Kune, and B. S. Manoj. A hybrid classical-quantum
586 approach for multi-class classification. *Quantum Inf Process*, 20(3):119, March 2021.
587 ISSN 1573-1332. doi:10.1007/s11128-021-03029-9. URL <https://doi.org/10.1007/s11128-021-03029-9>.

594 Bingren Chen, Hanqing Wu, Haomu Yuan, Lei Wu, and Xin Li. A novel quantum algorithm for
 595 converting between one-hot and binary encodings. *Quantum Information Processing*, 23(6):203,
 596 2024.

597 Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature
 598 Physics*, 15(12):1273–1278, August 2019. ISSN 1745-2481. doi:10.1038/s41567-019-0648-8.
 599 URL <http://dx.doi.org/10.1038/s41567-019-0648-8>.

600 Rudrajit Das and Subhasis Chaudhuri. On the separability of classes with the cross-entropy loss
 601 function. *arXiv preprint arXiv:1909.06930*, 2019.

602 H. A. David. Tournaments and paired comparisons. *Biometrika*, 46(1/2):139–149, 1959. ISSN
 603 00063444, 14643510. URL <http://www.jstor.org/stable/2332816>.

604 Bidisha Dhara, Monika Agrawal, and Sumantra Dutta Roy. Multi-class classification using quantum
 605 transfer learning. *Quantum Information Processing*, 23(2):34, 2024.

606 Olivia Di Matteo, Anna McCoy, Peter Gysbers, Takayuki Miyagi, R. M. Woloshyn, and Petr Navrátil.
 607 Improving hamiltonian encodings with the gray code. *Physical Review A*, 103(4), April 2021.
 608 ISSN 2469-9934. doi:10.1103/physreva.103.042405. URL <http://dx.doi.org/10.1103/PhysRevA.103.042405>.

609 P Kingma Diederik. Adam: A method for stochastic optimization. (*No Title*), 2014.

610 Chao Ding, Shi Wang, Yaonan Wang, and Weibo Gao. Quantum machine learning for multi-
 611 class classification beyond kernel methods. *Physical Review A*, 111(6), June 2025. ISSN
 612 2469-9934. doi:10.1103/physreva.111.062410. URL <http://dx.doi.org/10.1103/PhysRevA.111.062410>.

613 Michael Felsberg, Sinan Kalkan, and Norbert Krüger. Continuous dimensionality characterization
 614 of image structures. *Image and Vision Computing*, 27(6):628–636, 2009. ISSN 0262-8856.
 615 doi:<https://doi.org/10.1016/j.imavis.2008.06.018>. URL <https://www.sciencedirect.com/science/article/pii/S0262885608001479>.

616 Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the anal-
 617 ysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.
 618 doi:10.1080/01621459.1937.10503522. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1937.10503522>.

619 Alessandro Gambini, Giorgio Nicoletti, and Daniele Ritelli. A structural approach to gudermannian
 620 functions. *Results in Mathematics*, 79(1):10, 2024.

621 Mohit Goyal, Rajan Goyal, P Venkatappa Reddy, and Brejesh Lall. Activation functions. *Deep
 622 learning: Algorithms and applications*, pages 1–30, 2020.

623 Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola,
 624 Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet
 625 in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

626 Feifei Huang, Jie Li, and Xuelin Zhu. Balanced symmetric cross entropy for large scale imbalanced
 627 and noisy data, 2020. URL <https://arxiv.org/abs/2007.01618>.

628 Tak Hur, Leeseok Kim, and Daniel K. Park. Quantum convolutional neural network for clas-
 629 sical data classification. *Quantum Mach. Intell.*, 4(1):3, June 2022. ISSN 2524-4906, 2524-
 630 4914. doi:10.1007/s42484-021-00061-x. URL <http://arxiv.org/abs/2108.00661>.
 631 arXiv:2108.00661 [quant-ph].

632 Ali Javadi-Abhari, Matthew Treinish, Blake Johnson, and Robert Davis. Quantum computing with
 633 qiskit. *arXiv preprint arXiv:2406.12345*, 2024.

634 Peter D. Johnson, Jonathan Romero, Jonathan Olson, Yudong Cao, and Alán Aspuru-Guzik. Qvector:
 635 an algorithm for device-tailored quantum error correction, 2017. URL <https://arxiv.org/abs/1711.02249>.

648 Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL
 649 <https://arxiv.org/abs/1412.6980>.

650

651 Ryan LaRose and Brian Coyle. Robust data encodings for quantum classifiers. *Physical Review A*,
 652 102(3):032420, 2020.

653 Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to
 654 document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

655

656 Lorenzo Leone, Salvatore F.E. Oliviero, Lukasz Cincio, and M. Cerezo. On the practical usefulness
 657 of the hardware efficient ansatz. *Quantum*, 8:1395, July 2024. ISSN 2521-327X. doi:10.22331/q-
 658 2024-07-03-1395. URL <http://dx.doi.org/10.22331/q-2024-07-03-1395>.

659 Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv*
 660 *preprint arXiv:1910.07454*, 2019.

661

662 Junhua Liu, Kwan Hui Lim, Kristin L. Wood, Wei Huang, Chu Guo, and He-Liang Huang. Hybrid
 663 quantum-classical convolutional neural networks. *Sci. China Phys. Mech. Astron.*, 64(9):290311,
 664 August 2021. ISSN 1869-1927. doi:10.1007/s11433-021-1734-3. URL <https://doi.org/10.1007/s11433-021-1734-3>.

665

666 Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv*
 667 *preprint arXiv:1608.03983*, 2016.

668

669 Ian MacCormack, Conor Delaney, Alexey Galda, Nidhi Aggarwal, and Prineha Narang. Branch-
 670 ing quantum convolutional neural networks, 2020. URL <https://arxiv.org/abs/2012.14439>.

671

672 Yaakov Malinovsky and John W Moon. On round-robin tournaments with a unique maximum score,
 673 2024.

674

675 John W Moon et al. Topics on tournaments. *New York*, 1968.

676

677 Nidhi Munikote. Comparing quantum encoding techniques. *arXiv preprint arXiv:2410.09121*, 2024.

678

679 M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary*
 680 *Edition*. Cambridge University Press, 2010. ISBN 978-1-139-49548-6. URL <https://books.google.se/books?id=-s4DEy7o-a0C>.

681

682 Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data
 683 classification. *Physical review letters*, 113(13):130503, 2014.

684

685 Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical*
 686 *Statistics*, 22(3):400–407, 1951.

687

688 Matías Roodschild, Jorge Gotay Sardiñas, and Adrián Will. A new approach for the vanishing
 689 gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, 9(4):351–360, 2020.

690

691 M. Schuld and F. Petruccione. *Machine Learning with Quantum Computers*. Quantum Science
 692 and Technology. Springer International Publishing, 2021. ISBN 978-3-030-83098-4. URL
 693 <https://books.google.se/books?id=-N5IEAAAQBAJ>.

694

695 Maria Schuld, Alex Bocharov, Krysta Svore, and Nathan Wiebe. Circuit-centric quantum
 696 classifiers. *Phys. Rev. A*, 101(3):032308, March 2020. ISSN 2469-9926, 2469-
 697 9934. doi:10.1103/PhysRevA.101.032308. URL <http://arxiv.org/abs/1804.00633>.
 698 arXiv:1804.00633 [quant-ph].

699

700 Kevin Shen, Bernhard Jobst, Elvira Shishenina, and Frank Pollmann. Classification of the Fashion-
 701 MNIST Dataset on a Quantum Computer, March 2024. URL <http://arxiv.org/abs/2403.02405> [quant-ph].

702

703 Mingrui Shi, Haozhen Situ, and Cai Zhang. Hybrid Quantum Neural Network Structures for
 704 Image Multi-classification, August 2023. URL <http://arxiv.org/abs/2308.16005>.
 705 arXiv:2308.16005 [quant-ph].

702 Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and Entangling Capability of
 703 Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum*
 704 *Technologies*, 2(12):1900070, 2019. ISSN 2511-9044. doi:10.1002/quate.201900070. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/quate.201900070>. [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/quate.201900070](https://onlinelibrary.wiley.com/doi/pdf/10.1002/quate.201900070).

707 Samuel A. Stein, Betis Baheri, Daniel Chen, Ying Mao, Qiang Guan, Ang Li, Shuai Xu, and Caiwen
 708 Ding. QuClassi: A Hybrid Deep Neural Network Architecture based on Quantum State Fidelity,
 709 March 2022. URL <http://arxiv.org/abs/2103.11307>. arXiv:2103.11307 [quant-ph].

711 Farrokh Vatan and Colin Williams. Optimal quantum circuits for general two-qubit gates. *Physical*
 712 *Review A—Atomic, Molecular, and Optical Physics*, 69(3):032315, 2004.

713 Leni Ven and Johannes Lederer. Regularization and reparameterization avoid vanishing gradients in
 714 sigmoid-type networks, 2021. URL <https://arxiv.org/abs/2106.02260>.

716 Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross
 717 entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international*
 718 *conference on computer vision*, pages 322–330, 2019.

719 Hai-Rui Wei and Yao-Min Di. Decomposition of orthogonal matrix and synthesis of two-qubit
 720 and three-qubit orthogonal gates, March 2012. URL <http://arxiv.org/abs/1203.0722>.
 721 arXiv:1203.0722 [quant-ph].

723 Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking
 724 machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

725 Ernst Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrschein-
 726 lichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.

728 Jie Zhou, Dongfen Li, Yuqiao Tan, Xiaolong Yang, Yundan Zheng, and Xiaofang Liu. A multi-
 729 classification classifier based on variational quantum computation. *Quantum Inf Process*, 22
 730 (11):412, November 2023. ISSN 1573-1332. doi:10.1007/s11128-023-04151-6. URL <https://doi.org/10.1007/s11128-023-04151-6>.

733 A APPENDIX

735 A.1 ABLATIONS

737 A.1.1 CIRCUIT DEPTH

739 In order to test the assumption that the results of our post-processing method are unrelated to circuit
 740 depth, we ran an ablation over the number of circuit block layers. We focus on $K \in \{4, 5\}$ for the
 741 circuits CNN8, SU(4), and SEL-X, and test the same metrics with layers $L \in \{1, 3, 6\}$. The results
 742 provided in Table 4 show that, indeed, the discrete outputs of circuits trained with our tournament
 743 method remain more resolvable and similarly accurate even as circuit depth varies with unsurprising
 744 minor changes in overall accuracy between both methods.

745 Note that the improvement in performance of the tournament method over the one-hot method is
 746 even higher for shallower circuits which makes it an even more promising tool to deal with the noisy
 747 hardware currently available today, as deeper circuits allow more time for hardware noise to destroy
 748 quantum computations.

749 A.1.2 HARDWARE NOISE

751 Our contributions compare different post-processing methods of the outputs from identically initial-
 752 ized PQCs, and are thus independent of hardware conditions—in theory. To ensure that this is this
 753 case, we ran our inference suite on both datasets with all of the circuit blocks with $K \in \{3, 4\}$ using
 754 noise models provided by IBM QisKit (Javadi-Abhari et al., 2024). This inference was performed
 755 using the same learned parameters trained under noiseless-simulation to produce the results provided
 in Section 4.

756
 757 Table 4: Ablation showing the differing performance of circuits trained and tested with a number of
 758 2-ring layers $L \in \{1, 3, 4, 6\}$. The main results of the paper are attained using $L = 4$. Blocks SU(4),
 759 CNN8, and SEL-X, the MNIST Digits dataset, and $K \in \{4, 5\}$ were used for the ablation.

K	Block	L	Method	Resolvable-only			Constant		Simulation
				A_M	A_m	$M_r (\downarrow)$	A_M	A_m	
762 4	CNN8	1	Tournament	58.26	82.68	0.62	38.52	53.44	86.31
763 4	CNN8	1	One-hot	56.04	80.06	0.39	23.91	6.73	83.42
764 4	SU(4)	1	Tournament	59.86	84.71	0.65	40.73	61.31	87.15
765 4	SU(4)	1	One-hot	57.64	83.38	0.42	26.44	10.88	85.75
766 4	SEL-X	1	Tournament	32.45	38.25	0.54	19.46	16.58	47.49
767 4	SEL-X	1	One-hot	37.63	42.32	0.27	13.6	3.73	50.33
768 4	CNN8	3	Tournament	61.53	87.62	0.70	44.78	70.92	89.91
769 4	CNN8	3	One-hot	62.57	88.97	0.50	32.76	22.59	90.4
770 4	SU(4)	3	Tournament	63.43	89.26	0.75	48.52	81.78	90.65
771 4	SU(4)	3	One-hot	64.96	89.04	0.54	36.67	36.68	90.82
772 4	SEL-X	3	Tournament	44.28	70.79	0.53	24.5	12.45	76.49
773 4	SEL-X	3	One-hot	39.15	64.28	0.30	12.46	0.01	76.67
774 4	CNN8	4	Tournament	64.03	89.17	0.73	48.2	80.43	90.54
775 4	CNN8	4	One-hot	64.38	89.47	0.54	36.35	35.86	90.9
776 4	SU(4)	4	Tournament	63.51	89.21	0.74	48.38	81.04	90.91
777 4	SU(4)	4	One-hot	64.55	89.86	0.57	38.07	44.31	91.48
778 4	SEL-X	4	Tournament	46.86	74.9	0.55	26.94	17.6	81.25
779 4	SEL-X	4	One-hot	48.04	75.95	0.37	19.32	2.59	84.21
780 4	CNN8	6	Tournament	62.27	89.57	0.75	47.71	81.91	91.15
781 4	CNN8	6	One-hot	62.14	89.62	0.53	34.18	28.36	91.34
782 4	SU(4)	6	Tournament	61.03	89.64	0.75	46.37	82.74	91.02
783 4	SU(4)	6	One-hot	62.12	89.72	0.53	34.12	28.04	91.57
784 4	SEL-X	6	Tournament	48.85	81.44	0.59	29.51	26.75	87.13
785 4	SEL-X	6	One-hot	50.42	82.67	0.40	21.02	1.48	87.96
786 5	CNN8	1	Tournament	47.67	71.91	0.62	30.72	34.38	75.6
787 5	CNN8	1	One-hot	41.57	63.17	0.33	14.78	0.21	69.79
788 5	SU(4)	1	Tournament	48.47	74.03	0.63	31.46	38.67	77.0
789 5	SU(4)	1	One-hot	43.64	66.19	0.35	16.44	0.41	71.76
790 5	SEL-X	1	Tournament	26.5	32.99	0.58	15.88	7.87	36.28
791 5	SEL-X	1	One-hot	27.91	30.81	0.21	7.5	0.5	32.07
792 5	CNN8	3	Tournament	52.19	81.1	0.65	34.98	50.46	83.39
793 5	CNN8	3	One-hot	51.93	81.48	0.37	20.46	0.94	84.81
794 5	SU(4)	3	Tournament	54.35	82.01	0.67	37.52	58.74	84.78
795 5	SU(4)	3	One-hot	56.11	84.36	0.40	23.76	4.0	86.78
796 5	SEL-X	3	Tournament	25.7	41.61	0.58	15.11	1.38	47.94
797 5	SEL-X	3	One-hot	25.22	39.07	0.21	5.55	0.0	48.38
798 5	CNN8	4	Tournament	53.71	80.97	0.67	37.2	57.27	83.77
799 5	CNN8	4	One-hot	53.68	83.28	0.39	22.16	1.97	85.33
800 5	SU(4)	4	Tournament	55.75	83.55	0.67	38.39	57.95	85.05
801 5	SU(4)	4	One-hot	56.35	85.33	0.40	24.13	4.31	87.78
802 5	SEL-X	4	Tournament	29.23	49.57	0.58	17.21	2.25	55.47
803 5	SEL-X	4	One-hot	27.61	45.6	0.22	6.55	0.0	55.76
804 5	CNN8	6	Tournament	53.99	82.15	0.67	37.39	58.81	84.53
805 5	CNN8	6	One-hot	55.09	84.74	0.39	23.0	2.4	87.23
806 5	SU(4)	6	Tournament	56.66	84.74	0.70	40.48	66.49	86.01
807 5	SU(4)	6	One-hot	57.28	86.03	0.41	25.42	5.93	88.14
808 5	SEL-X	6	Tournament	30.71	47.26	0.58	18.29	7.3	57.62
809 5	SEL-X	6	One-hot	32.46	56.96	0.26	8.85	0.0	68.98

800 These noise models allow PennyLane to simulate the noise of real-world IBM hardware. Given the
 801 differing sizes of the circuits, we use the noise models of different IBM machines for different values
 802 of K —namely IBM Belem Version 2 for $K = 3$ and IBM Oslo for $K = 4$. The varying levels
 803 of noise for the varying hardware make comparing performance across different values of K less
 804 productive, however the results in Table 5 still allow for a fair comparison of the primary results
 805 between the tournament and one-hot post-processing methods.

806 Noise-model experiments are included to illustrate relative performance trends under non-ideal con-
 807 ditions, not as claims of hardware readiness. Backends were selected based on Qiskit documentation
 808 at the time; updating to current backends would not affect the theoretical guarantees presented in the
 809 main text.

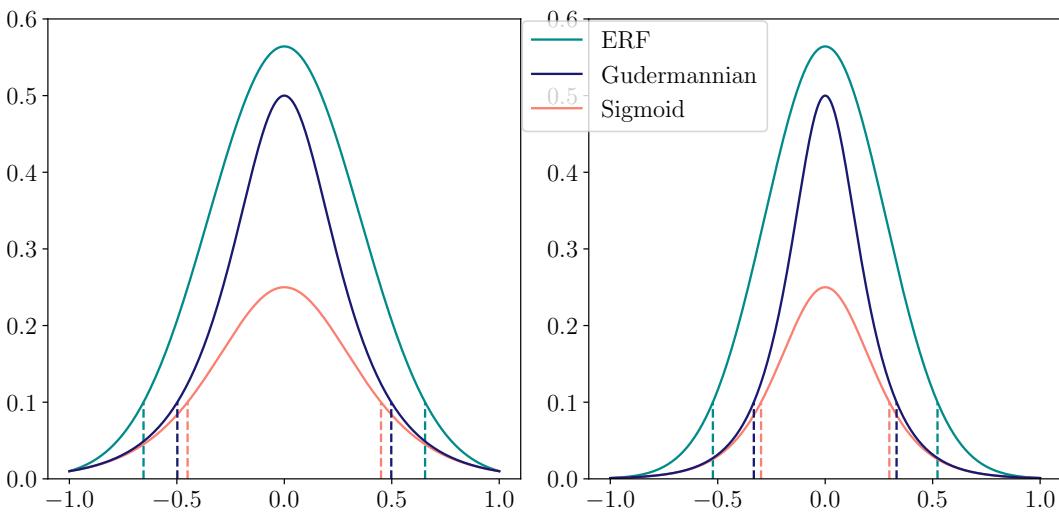
810 It can be clearly seen that hardware noise has a universally negative effect on the performance of even
 811 pretrained PQCs, however, the effectiveness of the tournament method over the one-hot method is
 812 still clearly visible. The relative performance between the two methods remains either identical or
 813 even improves for the tournament method. This is especially evident for the resolvable-only micro
 814 accuracy R_a , which remains better than guessing under the tournament method, but hovers near
 815 guessing level for the one-hot method. Given this improvement is attained using less samples—as
 816 evidenced by the superior resolvability ratio M_r —it is clear that the tournament method leads to
 817 much higher quality discrete output samples even under noisy conditions.

818
 819 **Table 5:** Ablation on inference performance of noiseless-trained models using all six block circuit
 820 variants on simulated noisy hardware.
 821

Dataset	K	Method	Resolvable-only			Constant		Simulation
			A_M	A_m	M_r	A_M	A_m	
Digits	3	Tournament	0.37 ± 0.04	0.43 ± 0.11	0.76 ± 0.01	0.28 ± 0.03	0.38 ± 0.10	0.69 ± 0.16
Digits	3	One-hot	0.33 ± 0.00	0.33 ± 0.01	0.38 ± 0.03	0.13 ± 0.01	0.00 ± 0.00	0.67 ± 0.20
Digits	4	Tournament	0.28 ± 0.02	0.34 ± 0.05	0.53 ± 0.02	0.15 ± 0.02	0.03 ± 0.06	0.76 ± 0.25
Digits	4	One-hot	0.25 ± 0.00	0.25 ± 0.01	0.27 ± 0.03	0.08 ± 0.01	0.00 ± 0.00	0.76 ± 0.26
Fashion	3	Tournament	0.39 ± 0.05	0.50 ± 0.11	0.77 ± 0.02	0.30 ± 0.04	0.46 ± 0.11	0.70 ± 0.16
Fashion	3	One-hot	0.33 ± 0.00	0.33 ± 0.01	0.37 ± 0.05	0.13 ± 0.02	0.00 ± 0.00	0.69 ± 0.20
Fashion	4	Tournament	0.26 ± 0.02	0.29 ± 0.05	0.52 ± 0.02	0.14 ± 0.01	0.02 ± 0.04	0.71 ± 0.24
Fashion	4	One-hot	0.25 ± 0.00	0.25 ± 0.01	0.30 ± 0.02	0.08 ± 0.01	0.00 ± 0.00	0.73 ± 0.21

A.1.3 ACTIVATION FUNCTIONS

834 For all four encodings, the expectation values from the PQC are activated using a sigmoid function,
 835 inspired by soft-thresholding (13) from Felsberg et al. (2009). This tempering reverses the mono-
 836 tonicity of the data and normalizes it, which both need to be done since the expectation value range
 837 for a quantum Pauli measurement is $[-1, 1]$, and expectation values of -1 and 1 are commonly used
 838 to represent a binary 1 and 0 , respectively (Nielsen and Chuang, 2010; Schuld and Petruccione, 2021).
 839 Activating expectation values this way enables us to reason about them as the probabilities that their
 840 qubits, when discretized through measurement, will output 1 as opposed to 0 .



858 **Figure 5:** Plots of the derivatives of the tempering functions after being scaled such that the chosen
 859 minimal gradient occurs at -1 and 1 . Dashed lines showing where each function has a gradient value
 860 of 0.1 , which illustrates the relative decrease in gradient scale between the three functions. Due to
 861 scaling the functions such that they have equivalent minimum values, the graphs are equal at their
 862 endpoints, but it can be seen that the logistic and Gudermannian functions have smaller gradients
 863 which also vanish sooner than in the error function (ERF).

864 The secondary goal in applying such a function is to ensure that the gradients returning to the circuit
 865 are minimal near expectation values of -1 and 1 , and maximal near 0 , since expectations near
 866 the extrema are more likely to discretize to either 1 or 0 , respectively, and expectation values of 0
 867 operate like coin-flips when discretized. Vanishing gradients from the sigmoid function have been
 868 a large enough problem in classical machine learning for them to be considered outdated (Ven and
 869 Lederer, 2021; Roodchild et al., 2020), but in this use case, it provides exactly the behavior we want.
 870 Originally, the logistic function was chosen due to the ease of calculating its gradient (Goyal et al.,
 871 2020), which, while efficient, may not lead to the optimal training behavior in quantum circuits.

872 There are many functions which have the required shape, with the biggest difference between them
 873 being their domains relative to their asymptotes as none reach diminished gradients in the domain
 874 $[-1, 1]$. Because of this, the inputs to the functions need to be scaled to make full use of this vanishing
 875 effect. This scaling can be such that the minimum gradient returning to the circuit is arbitrarily close
 876 to 0 , but the more this scaling is applied, the more of the input domain receives very little gradient,
 877 as shown in Fig A.1.3. In this study, we ablated over three sigmoid like functions - namely, the
 878 logistic function, the error function, and the Gudermannian function (Gambini et al., 2024) - and two
 879 minimum gradient levels for each - namely, 0.01 and 0.001 . To calculate the scaling, we simply find
 880 the input value to the first derivative of each function that gives the minimum value we set.

881
 882
 883 Table 6: Averaged Friedman rank over all relevant statistics on CNN7 with $K = 4$ using the edge
 884 method. Highest score was chosen as the optimal.

Function	ERF		Linear		Logistic		Gudermannian	
Min Grad	0.01	0.001	0.01	0.001	0.01	0.001	0.01	0.001
F-Score↑	5.0	4.375	2.875	2.875	4.725	3.875	4.25	3.25

890 To determine which sigmoid-like function to use for the main results, we performed our ablation
 891 process on the CNN7 block from Sim et al. (2019) using $K = 4$ on the MNIST Digits dataset, shown
 892 in Table 6. We compared ERF, Gudermannian, and the logistic function at minimal gradient values
 893 of 0.01 and 0.001 , as well as a linear monotonicity-reversing normalization.

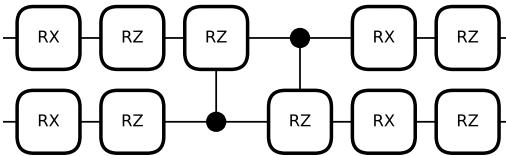
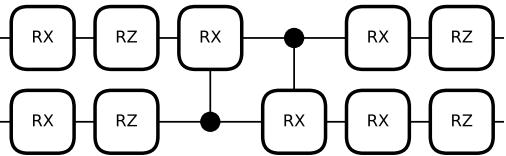
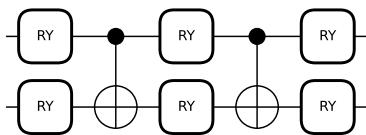
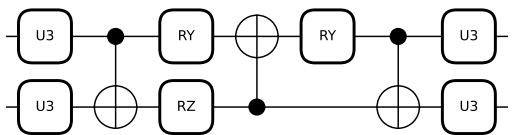
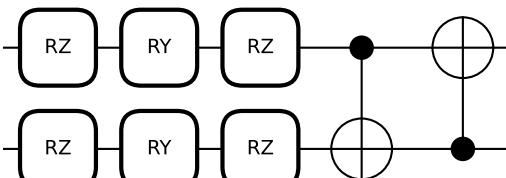
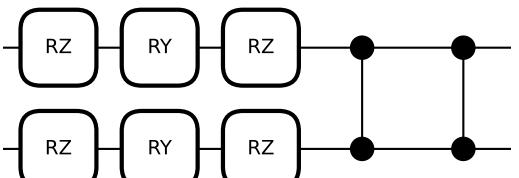
894 The scores presented are Friedman-scores computed over a range metrics: both sets of micro and
 895 macro accuracies, the threshold accuracy T and average distance between the top-two predictions,
 896 as well as the resolvability rate M_r . We use the Friedman-scores to decide on the best tempering
 897 method without focusing on a single metric.

900 A.1.4 OPTIMIZATION

902 To ascertain the best optimization strategy before running the full experimental suite, we ran an
 903 ablation across two optimizers, four learning rate schedulers, and three learning rates. The two
 904 optimizers are standard stochastic gradient descent (SGD), invented by Robbins and Monro (1951),
 905 and Adam, invented by Diederik (2014). The learning rate schedulers we tested were an exponential
 906 scheduler, as defined by Li and Arora (2019), a cosine scheduler, as defined by Loshchilov and Hutter
 907 (2016), a piecewise scheduler, as defined by Goyal et al. (2017), and no scheduler, also called a
 908 constant scheduler.

909 For the exponential scheduler, there were ten total transition steps over the full six epochs, with a
 910 decay rate of 0.9 . For the cosine scheduler, the number of steps was simply the number of training
 911 steps. For the piecewise scheduler, there were three transition steps with scale factors of 0.1 and 0.01 .

912 We first ran all the tests on $K = 3$ with the tournament method on the CNN7 block, shown in Table 7.
 913 To average over all the metrics, we look at the Friedman Rank (F-Rank) of each optimization strategy,
 914 which ranks the columns and averages the ranks over the rows (Friedman, 1937). Due to the tie
 915 between the piecewise scheduler and constant scheduler with the Adam optimizer, we opted to run a
 916 second set on the Adam optimizer with $K = 5$ instead. This test is shown in Table 8. As exponential
 917 decay with a learning rate of 0.01 ranked best for $K = 5$, and nearly as well as piecewise and constant
 for $K = 3$, this was chosen as the optimal setup.

918 A.2 FULL TESTING RESULTS
919920 Given the immense size of our full testing suite, there is no manageable way to include the full
921 tables in this print. The results from all tests run up to this date can be found in the repo linked in
922 Section 3.5.923
924
925
926 A.3 QML BLOCK DESCRIPTIONS
927928 Here we present information about the blocks used in the 2-qubit ring structure. In this section, we
929 will simply summarize the findings of the introductory works to justify their usage in this paper.
930931 The first four of the blocks were found in the work by Hur et al. (2022) and showed promising results
932 in all their experiments. In that paper, the reason each block was chosen was explained succinctly.
933 The CNN7 and CNN8 blocks were first introduced as 4-qubit error-correcting encoders by Johnson
934 et al. (2017). They showed the best expressibility in a study by Sim et al. (2019), leading to them
935 being chosen by Hur et al. (2022). Expressibility, in the context of QML, is a measure of the ability
936 of a circuit to produce a wide range of quantum states.937 The SO(4) block was shown by Wei and Di (2012) to be able to implement an arbitrary SO(4)
938 operation, and can be used to construct a fully entangled VQE. The SU(4) block was shown by Vatan
939 and Williams (2004) and MacCormack et al. (2020) to be able to implement any arbitrary 2-qubit
940 rotation.943 Figure 6: CNN7 Block from Sim et al. (2019),
944 as modified by Hur et al. (2022).
945947 Figure 7: CNN8 Block from Sim et al. (2019),
948 as modified by Hur et al. (2022).
949952 Figure 8: SO(4) Block from Wei and Di (2012),
953 as modified by Hur et al. (2022).
954956 Figure 9: SU(4) Block from Vatan and Williams
957 (2004), as modified by Hur et al. (2022).
958961 Figure 10: Strongly Entangling Layer Block
962 with CNOT im primitive (Sel-X) from Schuld
963 et al. (2020).
964966 Figure 11: Strongly Entangling Layer Block
967 with CZ im primitive (Sel-Z) from Schuld et al.
968 (2020).
969970 Strongly Entangling Layers is a popular multi qubit gate-operation that is available as a callable
971 function in the popular quantum computing package PennyLane (Bergholm et al., 2022). The setup

972 was invented in a paper by several of the authors responsible for the creation of PennyLane in Schuld
 973 et al. (2020), and has seen much use due to simplicity and expressibility.
 974

975 Note that for the Strongly Entangling Layers block all single-qubit operations are applied before the
 976 ring of two-qubit operations rather than in alternating full block rings like in CNN7, CNN8, SO(4)
 977 and SU(4), as visualized in Figure 1. We included this block in our analysis so as to demonstrate the
 978 efficacy of the tournament encoding independent of the 2-qubit ring structure.
 979

980 For even greater fairness, we include two versions using the two most common parameter-free
 981 2-qubit operations, namely the CNOT gate and CZ gate. For more information about the gate-
 982 operations performed in these blocks, we present an accelerated introduction to quantum computing
 983 in Section A.4.
 984

985 A.4 INTRODUCTION TO QUANTUM MACHINE LEARNING

986 In this section, we will give a low-level overview of the ideas from quantum computing needed to
 987 understand this work. This information is summarized from the works of Nielsen and Chuang (2010)
 988 and Schuld and Petruccione (2021) which cover it in much greater detail for the interested reader.
 989

990 *Qubits* - Qubits are the quantum equivalent to a bit in classical computing. The state of a qubit is
 991 represented as a two dimensional vector in a Hilbert space, with classical states 0 and 1 corresponding
 992 to the quantum states $|0\rangle$ and $|1\rangle$, where
 993

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ and } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1)$$

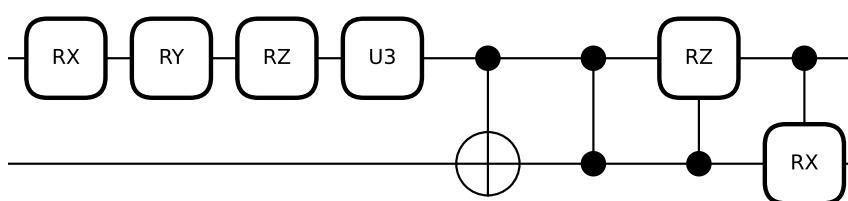
994 Unlike classical bits which are binary, the state of a qubit can be any length-1 vector in the two-
 995 dimensional complex vector space spanned by $|0\rangle$ and $|1\rangle$.
 996

997 *Gates* - Quantum gates are a quantum extension of classical reversible-logic gates.
 998

999 These transform states unitarily (complex angle-preserving), so correspond to complex rotations.
 1000 Simple examples include the Pauli-X, Pauli-Y, and Pauli-Z gates, written mathematically as σ_1 , σ_2
 1001 and σ_3 , respectively. Pauli-X, Pauli-Y, and Pauli-Z are also names for the cardinal axes within the
 1002 sphere of all possible states a single qubit can take, the so-called "Bloch sphere". The matrices which
 1003 represent these operations rotate a qubit π radians around the respective axis, and all of them can be
 1004 written at once as,
 1005

$$\sigma_j = \begin{pmatrix} \delta_{j3} & \delta_{j1} - i \delta_{j2} \\ \delta_{j1} + i \delta_{j2} & -\delta_{j3} \end{pmatrix}. \quad (2)$$

1006 Many other gates exist, including gates to elicit interactions between qubits and parameterized
 1007 versions of the Pauli Gates which allow rotations of arbitrary degree around their respective axes.
 1008 The subset used in this work can be found in Figure 12. A brief description of these is provided, but
 1009 more comprehensive details, as well as more gates, can be found in the works of Nielsen and Chuang
 1010 (2010) and Schuld and Petruccione (2021).
 1011



1012 Figure 12: Subset of quantum gates used in the circuits in this paper. From left to right: Pauli-X
 1013 rotation, Pauli-Y rotation, Pauli-Z rotation, 3-parameter unitary gate, CNOT, CZ, Controlled Pauli-Z
 1014 rotation, and controlled Pauli-X rotation.
 1015

1016 Of these gates, CNOT, CZ, and the controlled Pauli gates all apply their rotations conditionally
 1017 based on the state of the dotted wire. In the case of the controlled Pauli rotations, this rotation is
 1018 parameterized, where for CNOT and CZ, it is always a rotation of π radians about the Pauli-X and
 1019 Pauli-Z axes, respectively. The regular rotation gates are parameterized versions of their Pauli gates.
 1020

1026 The U3 gate is a special gate which applies a parameterized Pauli-Z, followed by a parameterized
 1027 Pauli-Y, and then another parameterized Pauli-Z, enabling any arbitrarily Euler rotation about the
 1028 Bloch Sphere.

1029 *Circuits* - The term "circuit" typically refers to a more complicated unitary operator built up from a
 1030 number of quantum gates that are composed sequentially. The term *wire* refers to single qubits as
 1031 they traverse the different operations within a circuit. The term "model" can often be interchanged
 1032 with "circuit," though perhaps self-evidently, only when the model can be represented as a circuit.
 1033

1034 *Measurements* - To extract information from a quantum circuit, a measurement of the qubits involved
 1035 must be performed. A measurement has an associated Hermitian operator (real-valued eigenvalues)
 1036 where the eigenvalues are the possible outcomes, and the squared length of the state projection onto
 1037 one of the eigenspaces determines the probability of the corresponding outcome. Due to convention,
 1038 the most common measurement in quantum computing is measurement in the "computational basis",
 1039 associated with the Hermitian Pauli-Z operator (Schuld and Petruccione, 2021; Nielsen and Chuang,
 1040 2010).

1041 A measurement always gives one of the eigenvalues of the Hermitian operator. For a Pauli-Z
 1042 measurement we obtain one of two discrete outputs, +1 or -1 (mapped to the bit values 0 or 1,
 1043 respectively). The *expectation value*, or the expected (average) output is then the weighted average of
 1044 the outcomes. For a Pauli-Z measurement, this would be

$$E(\sigma_3) = (+1)P(+1) + (-1)P(-1) \quad (3)$$

1045 Note that the range of this expression is $[-1, +1]$ because the eigenvalues of the σ_3 operator are +1
 1046 and -1 rather than the binary 0 and 1. If the two outcomes are equally probable, the expectation value
 1047 here is = 0 rather than = $1/2$, which becomes important when setting thresholds in the simulation
 1048 output.

1049 Such an expectation value can be calculated directly, though this is only possible in simulations. In an
 1050 actual machine, the outputs would be the discrete values +1 and -1, so to estimate the expectation
 1051 value when using a quantum computer one would need to count the outcomes and produce a point
 1052 sample from a series of measurements.

1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079

Table 7: Ablation with $K = 3$ over optimizers (Opt), learning rate schedulers (LRS), and learning rates (LR). Schedulers used include Exponential Decay *exp*, Cosine Decay *cos*, Piecewise Constant *step*, Constant *reg*, and Linear Decay *lin*.

Opt	LRS	LR	F-Rank (↑)	Valid			Constant		Simulation <i>T</i>
				A _M	A _m	S (↓)	A _M	A _m	
SGD	<i>exp</i>	0.01	14.875	54.15	73.73	116.36	47.02	72.7	78.33
SGD	<i>exp</i>	0.001	5.375	50.88	68.84	125.53	41.6	63.08	75.02
SGD	<i>exp</i>	0.0001	2.0	42.04	50.62	129.33	33.5	44.78	57.68
SGD	<i>reg</i>	0.01	17.0	54.16	73.77	116.32	47.04	72.77	78.39
SGD	<i>reg</i>	0.001	7.625	50.92	68.99	125.43	41.66	63.27	74.84
SGD	<i>reg</i>	0.0001	3.125	42.1	50.76	129.37	33.53	44.85	58.29
SGD	<i>step</i>	0.01	17.0	54.16	73.77	116.32	47.04	72.77	78.39
SGD	<i>step</i>	0.001	7.625	50.92	68.99	125.43	41.66	63.27	74.84
SGD	<i>step</i>	0.0001	3.125	42.1	50.76	129.37	33.53	44.85	58.29
SGD	<i>cos</i>	0.01	16.5	54.16	73.77	116.32	47.04	72.77	78.4
SGD	<i>cos</i>	0.001	6.875	50.92	68.98	125.43	41.66	63.23	74.85
SGD	<i>cos</i>	0.0001	2.375	42.09	50.74	129.36	33.53	44.84	58.28
SGD	<i>lin</i>	0.01	15.625	54.15	73.75	116.35	47.02	72.71	78.35
SGD	<i>lin</i>	0.001	6.125	50.89	68.84	125.52	41.61	63.11	75.03
SGD	<i>lin</i>	0.0001	2.5	42.05	50.65	129.34	33.5	44.79	57.72
Adam	<i>exp</i>	0.01	16.75	54.62	69.29	114.15	48.49	68.54	76.66
Adam	<i>exp</i>	0.001	19.875	55.49	75.16	114.41	48.92	74.25	78.26
Adam	<i>exp</i>	0.0001	9.5	51.7	70.33	123.64	42.71	65.42	75.14
Adam	<i>reg</i>	0.01	17.5	54.65	69.3	114.19	48.49	68.63	76.64
Adam	<i>reg</i>	0.001	21.125	55.49	75.07	114.35	48.94	74.29	78.27
Adam	<i>reg</i>	0.0001	12.125	51.76	70.52	123.53	42.78	65.6	75.17
Adam	<i>step</i>	0.01	17.5	54.65	69.3	114.19	48.49	68.63	76.64
Adam	<i>step</i>	0.001	21.125	55.49	75.07	114.35	48.94	74.29	78.27
Adam	<i>step</i>	0.0001	12.125	51.76	70.52	123.53	42.78	65.6	75.17
Adam	<i>cos</i>	0.01	17.375	54.65	69.29	114.19	48.49	68.64	76.65
Adam	<i>cos</i>	0.001	21.0	55.49	75.08	114.35	48.94	74.29	78.27
Adam	<i>cos</i>	0.0001	11.625	51.76	70.52	123.53	42.78	65.61	75.16
Adam	<i>lin</i>	0.01	17.75	54.63	69.3	114.15	48.5	68.59	76.66
Adam	<i>lin</i>	0.001	20.5	55.49	75.16	114.41	48.93	74.27	78.26
Adam	<i>lin</i>	0.0001	10.375	51.71	70.37	123.63	42.72	65.45	75.14

Table 8: Ablation with $K = 5$ using Adam over learning rates (LR), learning rate schedulers (LRS). Schedulers used include Exponential Decay *exp*, Cosine Decay *cos*, Piecewise Constant *step*, Constant *reg*, and Linear Decay *lin*.

Opt	LRS	LR	F-Rank (↑)	Valid			Constant		Simulation <i>T</i>
				A _M	A _m	S (↓)	A _M	A _m	
Adam	<i>exp</i>	0.01	11.0	49.25	80.43	151.77	33.16	47.5	83.99
Adam	<i>exp</i>	0.001	5.125	47.62	78.95	155.31	31.41	40.19	82.9
Adam	<i>exp</i>	0.0001	2.375	36.63	65.06	166.9	22.41	13.36	72.54
Adam	<i>cos</i>	0.01	9.625	49.23	80.36	151.79	33.14	47.49	83.92
Adam	<i>cos</i>	0.001	6.875	47.62	78.92	155.21	31.43	40.14	82.99
Adam	<i>cos</i>	0.0001	3.75	36.75	64.85	166.84	22.48	13.67	72.7
Adam	<i>step</i>	0.01	9.625	49.23	80.36	151.79	33.14	47.49	83.92
Adam	<i>step</i>	0.001	6.875	47.62	78.92	155.21	31.43	40.14	82.99
Adam	<i>step</i>	0.0001	3.75	36.75	64.85	166.84	22.48	13.67	72.7
Adam	<i>reg</i>	0.01	10.625	49.24	80.44	151.78	33.15	47.54	83.94
Adam	<i>reg</i>	0.001	6.75	47.62	78.96	155.22	31.42	40.33	82.98
Adam	<i>reg</i>	0.0001	3.5	36.75	65.11	166.87	22.48	13.61	72.71
Adam	<i>lin</i>	0.01	10.25	49.23	80.36	151.76	33.16	47.46	83.97
Adam	<i>lin</i>	0.001	5.25	47.6	78.78	155.27	31.41	40.32	82.9
Adam	<i>lin</i>	0.0001	2.875	36.65	64.97	166.9	22.42	13.44	72.57