Project Summary

Mark Long – Faculty of Applied Science, Engineering Physics, University of British Columbia
Supervisor: Prof. Roman Krems – Department of Chemistry, University of British Columbia

Generation of K-Forrelation Dataset for Benchmarking Hybrid Classical-Quantum Classifiers

1) MOTIVATION

The need for more suitable and relevant datasets to benchmark:

1) Hybrid classical-quantum classifiers
2) Algorithms to optimize quantum circuit design for classification tasks

A few research suggested datasets for quantum machine learning, but they are more suited for training models on quantum control and tomography (Perrier et al., 2021) or simply based on heuristic about difficulty in simulating entanglement (Schatzki et al., 2021)

A classification dataset based on the *k-fold Forrelation (k-Forrelation)* problem is interesting because:

- k-Forrelation is formally proved to *maximally separate the query complexity* between classical and quantum computation in the black-box model (Aaronson, 2014) – which makes it more directly relevant to quantum classifiers than generic datasets (*Note*: this research does <u>*not*</u> attempt to compare quantum against classical classifiers)
- k-Forrelation decision problem is *PromiseBQP-complete* and has been proved to be within the expressiveness of Variational Quantum Classifier and Quantum Support Vector Machine (Jager & Krems, 2022)
- k-Forrelation datasets can be generated with *different parameterizations* that allow evaluation of model performance and computational cost at scales

This research addresses the following challenges regarding the k-Forrelation dataset:

- The positive class is exponentially rare at larger problem sizes, making it prohibitively difficult to sample a balanced dataset
- Random sampling in the function space is incapable of generating examples with high positive thresholds, which requires the development of a novel sampling algorithm
- The k-Forrelation decision problem is theoretically formulated but the relative classification difficulty at different parameterizations have not been studied

The generated k-Forrelation datasets could also enable future research on performance criteria suitable for benchmarking quantum classifiers beyond accuracy (e.g., scalability), or to statistically confirm hypotheses of algorithm improvement.

2) CONTRIBUTIONS

- An algorithm to generate k-Forrelation datasets with high positive class threshold
- An analysis of the properties of k-Forrelation datasets in classification task
- Guidelines for the generation of k-Forrelation datasets for benchmarking
- Suggestions and demonstration for potential uses of k-Forrelation datasets *(in progress)*

3) METHODS

An algorithm to generate the k-Forrelation dataset with a high positive class threshold

- Express Φ (correlation value) in recursive "layer-wise" structure
- Generalize sampling strategy for 2-forrelation instances to k-Forrelation dataset through approximate Fourier Transform and random Gaussian reassignment (and smallest-hamming-distance equivalent, for policy space)
- Numerical demonstration of generalized Fourier Sampling in the general space and the policy space by analyzing distributions

A framework to understand the properties of k-Forrelation datasets

- Analytical derivation of the mean and variance of Φ distribution as functions of n and k (in the general and policy space)
- Numerical experiments confirming the derived distribution of Φ
- Suggest a theoretical framework to analyze the relative difficulty of k-Forrelation datasets of different values of n, k, and *threshold*
- Experimental validation of the framework:
    - Generate k-Forrelation datasets of various n and k using random sampling (use odd k since it is easier to sample the positive class with high threshold)
    - Train SVMs with an optimized basic classical kernel (e.g., RBF) on all generated datasets to confirm the predicted trends in *relative* difficulty w.r.t. n, k, and *threshold*

Guidelines for the generation of benchmark k-Forrelation datasets

- Suggest generation guidelines based on the framework and restrictions in values of k w.r.t. n
- Explain the choice of random sampling as the generative algorithm in the limit of small n

Suggestions (and demonstration) for potential applications of k-Forrelation datasets

- Literature review on how quantum classifier's performance is evaluated and how the constructive algorithms are designed
- Literature review on related areas where a benchmark dataset might be useful (e.g., dimensionality reduction algorithms for quantum communication)
- Detail use cases for the k-Forrelation datasets
- Numerically demonstrate a simple use case to evaluate constructive algorithms for quantum kernel:
    - Test the effect of dimensionality reduction on the learnability of k-Forrelation datasets
    - Benchmark Elham's algorithm with other proposed constructive algorithms on the datasets with reduced dimensionality
- Numerically demonstrate a constructive algorithm based on structural embedding (potentially a separate work)

4) RESULTS & DISCUSSION

4.1) An algorithm to generate k-Forrelation dataset with high positive threshold

(Could also be useful in enforcing k-Forrelation distributions of higher mean for benchmarking QuGANs)

4.1.1. Positive class sampling for 2-Forrelation (Aaronson, 2014)

To generate two Boolean functions that are highly likely to be forrelated:

- Generate a random vector $\boldsymbol{v} = R^{2^n}$ from a normal distribution
- Set $f_1 = \text{sign}(\boldsymbol{v})$ and $f_2 = \text{sign}(\widehat{\boldsymbol{v}})$, where $\widehat{\boldsymbol{v}}$ is the discrete Fourier transform of $\boldsymbol{v}$

4.1.2. Positive class sampling for k-Forrelation (this work)

Although 2-Forrelation is not a sub-problem of k-Forrelation, this work proposes a generalization of the sampling strategy for any arbitrary number of functions. This is done by leveraging the linearized form of Φ and backtracking the randomized vector $\boldsymbol{v}$. Recall that Φ can be written as follow:

$$\Phi(k,n) = \frac{1}{2^{(k+1)n/2}} \, sum[\overrightarrow{\Omega_k}] \quad \begin{cases} \overrightarrow{\Omega_2} = (A\vec{f_1}) \odot \vec{f_2} \\ \overrightarrow{\Omega_k} = (A\overrightarrow{\Omega_{k-1}}) \odot \vec{f_k} \end{cases}$$

The motivation is to view $\vec{\Omega}_{k-1}$ as approximately a function f_{k-1}. Then, following the 2-Forrelation sampling algorithm, f_{k-1} can be taken to possess the sign of some randomly generated vector $\boldsymbol{v}$. We thus generate a random vector from the normal distribution and assign to it the sign of f_{k-1}. Finally, we have $f_k = \text{sign}(\widehat{\boldsymbol{v}})$. The pseudo-code is shown below:

FOURIER GENERATOR (k):

if k = 1:
    choose f_1 randomly from the policy space
    generate random vector v from a normal distribution
    assign sign of f_1 to the absolute value $|v|$ to obtain v'
    return $[f_1]$, v'
else:
    previous functions, v'_{k-1} = **FOURIER GENERATOR (k-1)**
    $f_k = \text{sign}(\widehat{v'}_{k-1})$
    calculate $\vec{\Omega}_k$
    generate random vector v_k from a normal distribution
    assign sign of $\vec{\Omega}_k$ to the absolute value $|v_k|$ to obtain v'_k
    return [previous functions, f_k], v'_k
end if

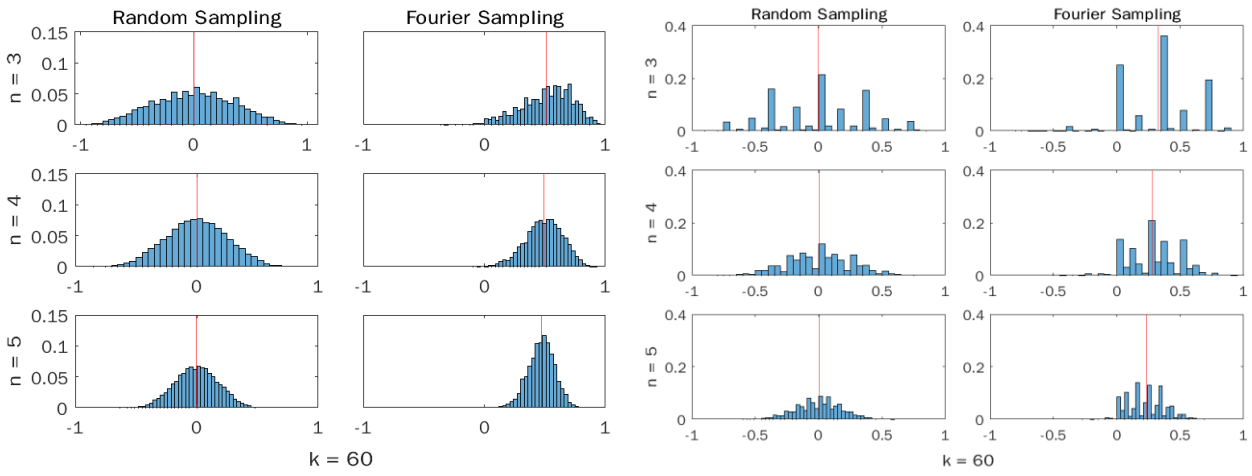

FIG. 9. Φ distributions for random sampling and Fourier sampling in (left) general space and (right) policy space, with n from 3 to 5 and $k = 60$

In the general space, Fourier sampling increases E[Φ] to around 0.5 (Figure 9, left panel). In the policy space, Fourier sampling increases E[Φ] to around 0.25, which will still approach 0 if n grows further (Figure 9, right panel). That said, for intermediate values of n, Fourier sampling allows significant improvement over random sampling in generating the positive class when *threshold* is far from 0.

4.2) Framework for predicting relative difficulty of k-Forrelation datasets

4.2.1. Derivation of mean and variance of Φ distributions

Recall the definition of Φ for a *k-Forrelation* instance with Boolean input size n

$$\Phi_{f_1,\ldots,f_k} := \frac{1}{2^{(k+1)n/2}} \sum_{x_1,\ldots,x_k \in \{0,1\}^n} f_1(x_1)(-1)^{x_1 \cdot x_2} f_2(x_2)(-1)^{x_2 \cdot x_3} \cdots (-1)^{x_{k-1} \cdot x_k} f_k(x_k)$$

$$f_1, \ldots, f_k : \{0,1\}^n \to \{-1,1\}$$

With f_1 to f_k randomly sampled from a domain, the value of Φ can be taken as a discrete random variable with expectation value $E_{f_1,\ldots,f_k \sim \mathbb{D}}[\Phi]$ and variance $Var_{f_1,\ldots,f_k \sim \mathbb{D}}[\Phi]$.

Consider the *general space* (every Boolean function returns +1 or -1 uniformly at random) – it can be proved that E[Φ] = 0 and Var[Φ] = $1/2^n$ using symmetry arguments, which implies that Φ distributions in the general space likely peak around 0 with standard deviation independent of k and decreasing exponentially in n. This can be confirmed by experimental results in Figure 1:

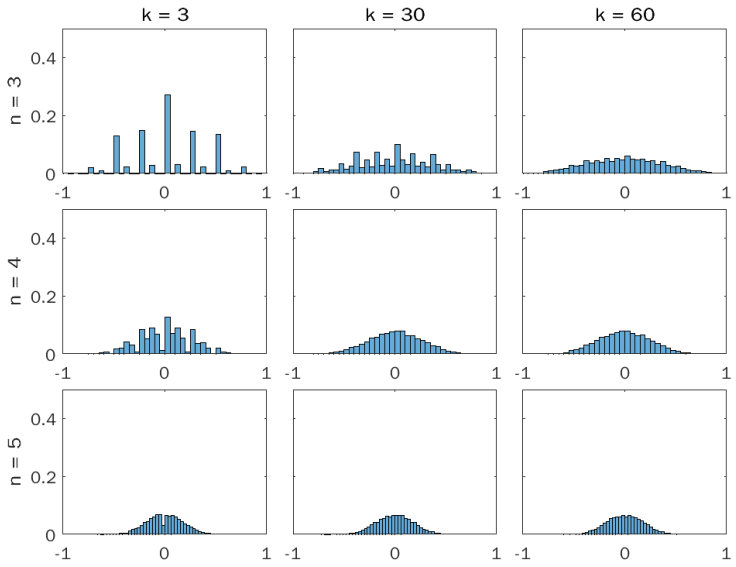

FIG. 1. Distributions of Φ for random sampling in *general space* with increasing values of n and k

Now, consider the sampling domain where each k-Forrelation instance must satisfy two constraints: (1) each Boolean function $f_i(x)$ in the instance is either constant $f_i(x) = +1$, or has the form $f_i(x) = (-1)^{C_i(x)}$ where $C_i(x)$ is the product of at most 3 bits in x; and (2) at least one Boolean function in the instance depends on exactly 3 bits in x. We herein refer to this sampling domain as the *policy space*. The Φ distributions in the policy space are shown in Figure 2:

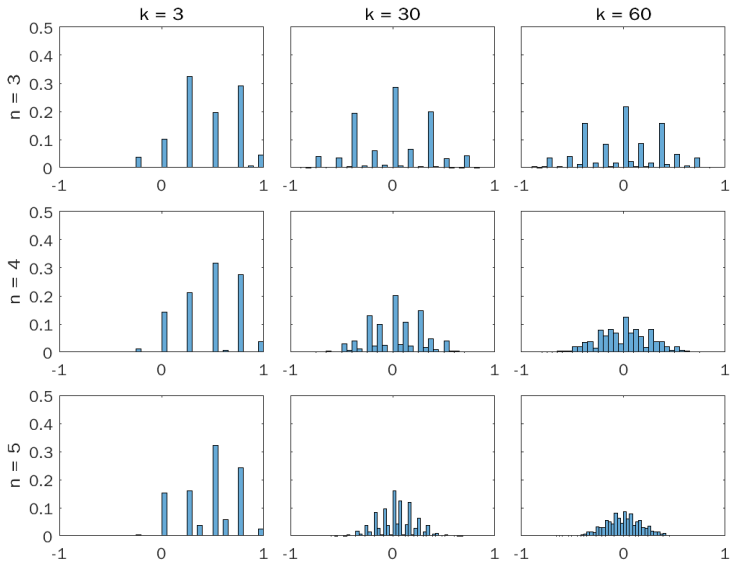

FIG. 2. Distributions of Φ for random sampling in *policy space* with increasing values of n and k

In contrast to the general space, $E[\Phi]$ in the policy space is non-zero for smaller k, but still approaches 0 as k increases. Just as before, $\mathrm{Var}[\Phi]$ is independent of k and decreasing exponentially in n. (Still working on the exact analytical forms of $E[\Phi]$ and $\mathrm{Var}[\Phi]$ in the policy space to confirm these observations)

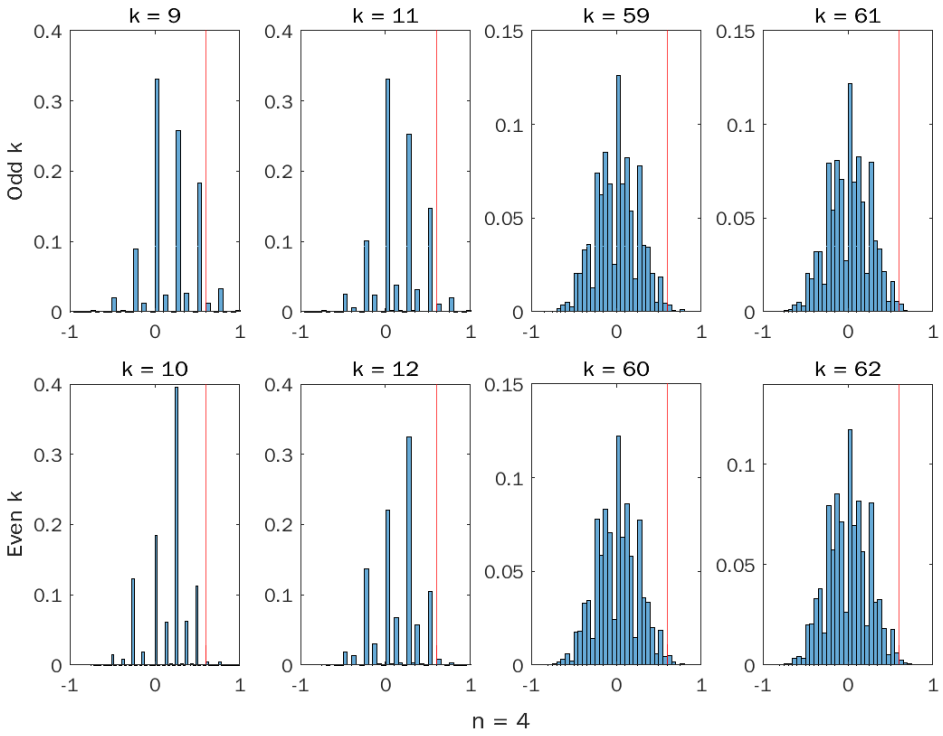

FIG. 3a. Distributions of Φ for random sampling in *policy space* with odd and even k. The red line marks a positive *threshold* of 0.6

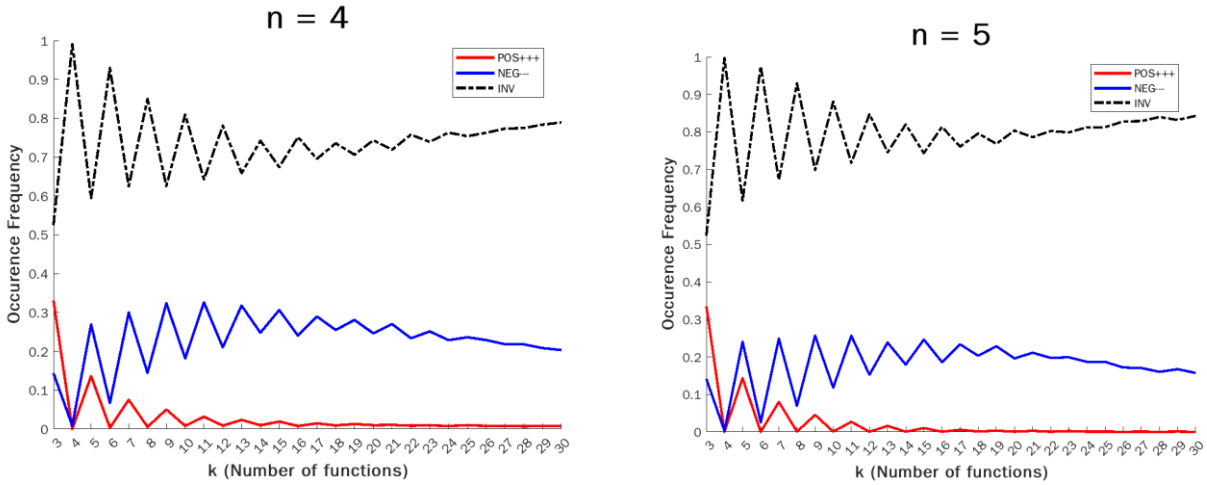

FIG. 3b. Class sampling frequencies in policy space with different values of n and increasing k. Positive threshold is set to 0.6

Φ distributions for odd and even k in policy space differ considerably when values of k are relatively small, but much less so when k is large (Figure 3a) (Explanation can be found by referring to the analytical form of $\mathrm{E}[\Phi](n,k)$ and $\mathrm{Var}[\Phi](n,k)$ in the policy space). There is nothing inherently special about k being even or odd, but the chosen *threshold* can make sampling for positive class in one case easier than the other. For example, Figure 3b shows that sampling with a positive threshold of 0.6 is easier with odd k. For the following discussions, we will sample the *k-Forrelation* datasets with odd k, without loss of generality.

4.2.2. Trends of relative difficulty with respect to n, k, and threshold

We want to study how the empirical difficulty of the k-Forrelation dataset changes with n, k, and (positive) *threshold*. The datasets used in the following experiments are generated with the following properties: Generative algorithm: Random Sampling (to observe the actual trends without imposing artifacts from the data sampling process); Size: 10,000 points (balanced) where train and test splits depend on the experiment; Values of n: from 3 to 6; Values of k: odd numbers from 9 to 21; Values of *threshold*: 0.2, 0.5, and 0.6. This makes a total of 84 distinct k-Forrelation datasets.

Classical SVM with an optimized RBF kernel is used to classify the datasets. It is a general choice of classifier since, at the moment, we are studying only the *relative* empirical difficulty of the datasets by comparing how well a classifier with fixed complexity can perform on each of them. A poor performance of the RBF kernel on a given dataset does not mean that dataset is hard for every arbitrary classical classifier, but it does imply greater difficulty relative to other datasets.

(i) Difficulty increases in k

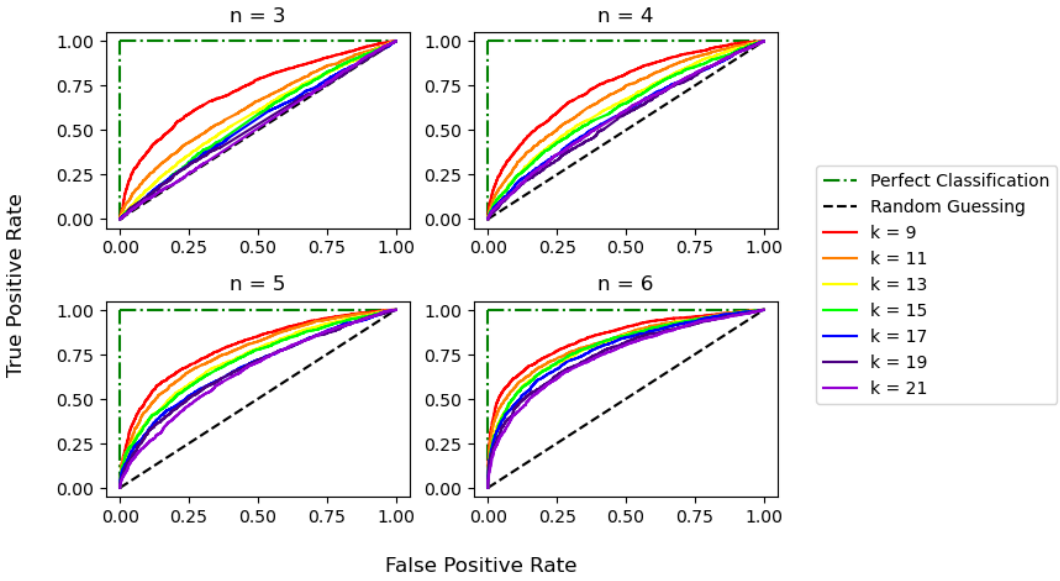

FIG. 4. Receiver Operator Characteristic (ROC) curves for optimized RBF kernels with increasing k. Datasets are randomly sampled from the policy space with *threshold* = 0.5. *Dimensionality is made equal across datasets of each n*

To observe how empirical difficulty changes with k, the ROC curves corresponding to odd k from 9 to 21 and n from 3 to 6 are plotted for SVM classifiers with RBF kernel (Figure 4). For each (n, k) dataset, the hyperparameters of the RBF kernel is first tuned with Bayesian Optimization using 5,000 training points and 3-fold validation. The model is then trained with 10,000 points (including those used for hyperparameters tuning) and tested on 5,000 new points. Dimensionality is made equal across datasets of different k values by adding buffer identity functions. ROC curve is commonly used to analyze the predictive power of a classifier where the closer a curve is to the top left, the better the model separates the two classes. Generally, the performance of classical SVM worsens with the increasing number of Boolean functions in a k-Forrelation dataset.

Pope (2021) distinguished the influence of *extrinsic* and *intrinsic dimensionality* on the learnability of a dataset. The number of samples needed to learn a well-conditioned decision boundary between two classes grow exponentially with the *intrinsic* dimensionality and is independent of the ambient dimension where

the dataset lives (*extrinsic* dimensionality). Since each Boolean function in a k-Forrelation instance represents one intrinsic dimension (whose value is the truth table of the Boolean function itself), increasing k quickly increases the intrinsic dimensionality of the dataset and raises its difficulty. This observation holds whether or not extrinsic dimensionality of the datasets are calibrated.

(ii) Difficulty decreases in n

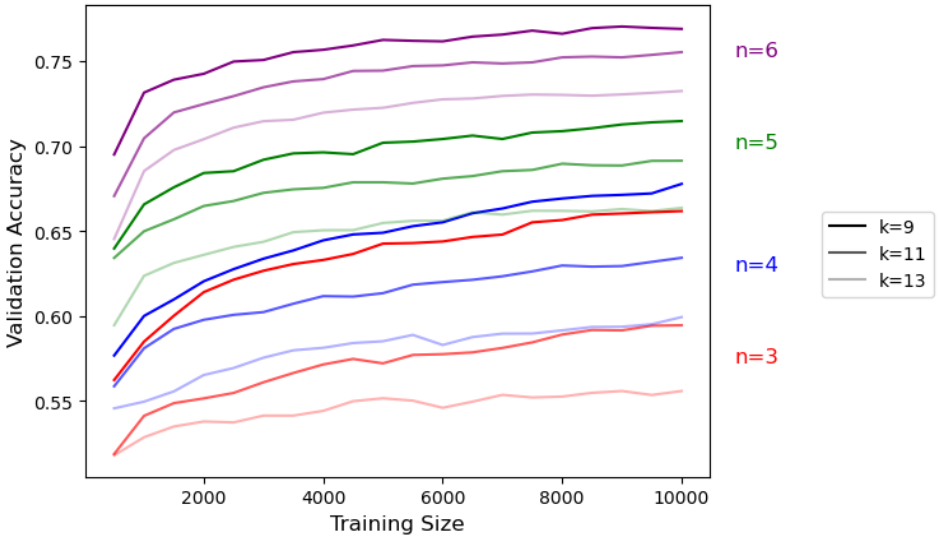

FIG. 5. Learning curves for optimized RBF kernel with k = 9-13 and n increasing from 3 to 6 Datasets are randomly sampled from the policy space with *threshold* = 0.5

Plotting the learning curves on datasets of fixed k and increasing n (Figure 5) suggests that empirical difficulty decreases with higher n. When k is fixed, increasing n does not add any additional intrinsic dimension but only increases the range of "values" (possible truth tables) each dimension can have. Since the number of intrinsic dimensions is the same across datasets with the same k, we expect no considerable increase in difficulty.

The observed decrease in difficulty is likely due to the exponentially shrinking subspace where the positive samples live. The size of the policy space (total number of possible k-Forrelation instances) depends on n and k as $S(n,k) = (1+n+\mathrm{C}_n^2+\mathrm{C}_n^3)^k - (1+n+\mathrm{C}_n^2)^k$. With a fixed k value, the size of the policy space increases polynomially in n. However, recall that the variance of the Φ distribution (and thus the occurrence of positive class) decreases exponentially in n ($\mathrm{Var}[\Phi] = 1/2^n$). Aaronson (2014) also remarked that positive samples tend to be close to each another in Hamming distance. Together, this means that the positive samples are clustered within an increasingly restrained subspace when n grows. Thus, a hyperplane that separates them from the negative class can more easily be found, leading to relatively easier datasets.

(iii) Framework for predicting relative classification difficulty

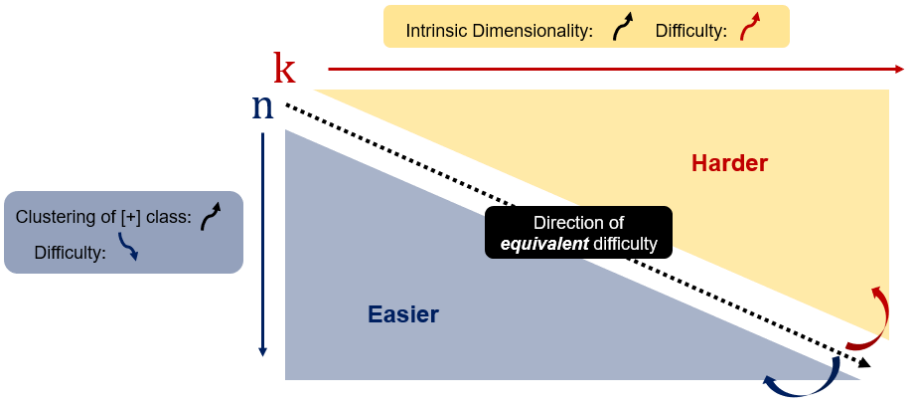

FIG. 6. A framework for predicting the relative difficulty of k-Forrelation datasets

Following earlier discussions, Figure 6 illustrates a suggested framework for predicting the relative classification difficulty of k-Forrelation datasets. This framework implies: (1) Simply increasing n will reduce the empirical difficulty of the dataset, even though it corresponds to an exponential increase in the required computational resource to calculate Φ exactly (one must distinguish the difficulty of exact calculation from that of probabilistic classification); (2) Going to the highest allowable k at any n results in the hardest dataset; (3) There exists a direction where simultaneous increase in both n and k results in negligible change in difficulty.

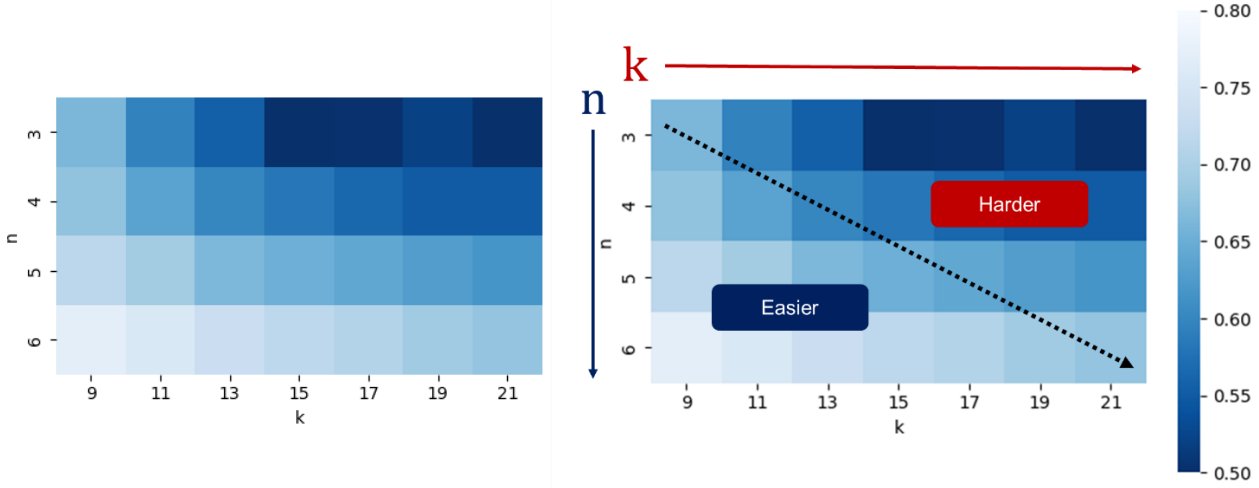

FIG. 7. Test accuracy of SVM with optimized RBF kernel on randomly sampled *k-Forrelation* datasets from the policy space with *threshold* = 0.5

Figure 7 shows the test accuracy of SVM classifiers trained with 10,000 samples from datasets of increasing n and k. The annotated right panel demonstrates agreement with the predictive framework where a direction of equivalent difficulty separates the relatively easier datasets (high n) from the harder ones (high k). Note that a practical dataset restricts how large k can be with respect to n (k=poly(n)) and thus an arbitrary large k cannot be chosen for smaller n.

(iv) Difficulty decreases with threshold

Since every Φ distribution has a variance that contracts exponentially in n, the effect of the chosen positive *threshold* on the difficulty of the datasets is of great interest for the generative algorithm. As n goes to infinity (and with k of comparable values), Φ distributions in the policy space peak about zero with diminishing deviation. This makes sampling the positive class exceptionally challenging when *threshold* is

set to a value far away from zero. Aaronson (2014) defined a positive threshold of 0.6 to maximize the separation in query complexity between a classical and quantum computer. This *threshold* could theoretically be lowered without affecting the computational complexity of k-Forrelation.

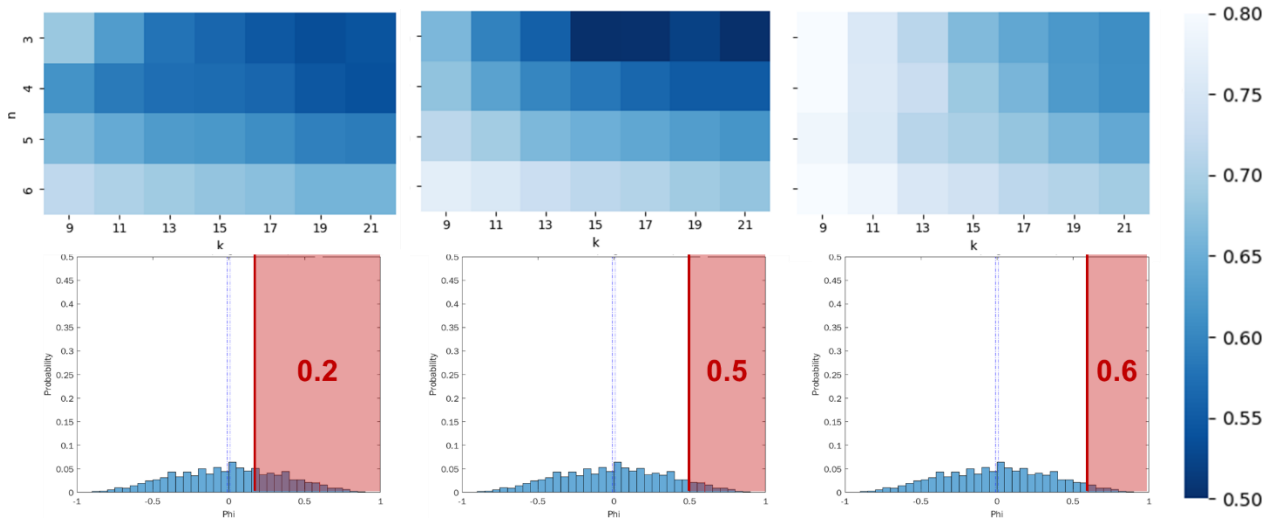

FIG. 7. Test accuracy of SVM with optimized RBF kernel on randomly sampled k-Forrelation datasets from the policy space with *threshold* = 0.2, 0.5, and 0.6

The impact of lowering the *threshold* on dataset difficulty (Figure 7) can be predicted by anticipating the change in clustering of the positive class. When n and k are fixed, lowering the *threshold* increases the occurrence of positive class and thus expands the subspace that they occupy. This consequently makes it harder to identify a hyperplane in the feature space that can separate the classes well and leads to more difficult datasets.

4.3) Guidelines for the generation of benchmark k-Forrelation datasets

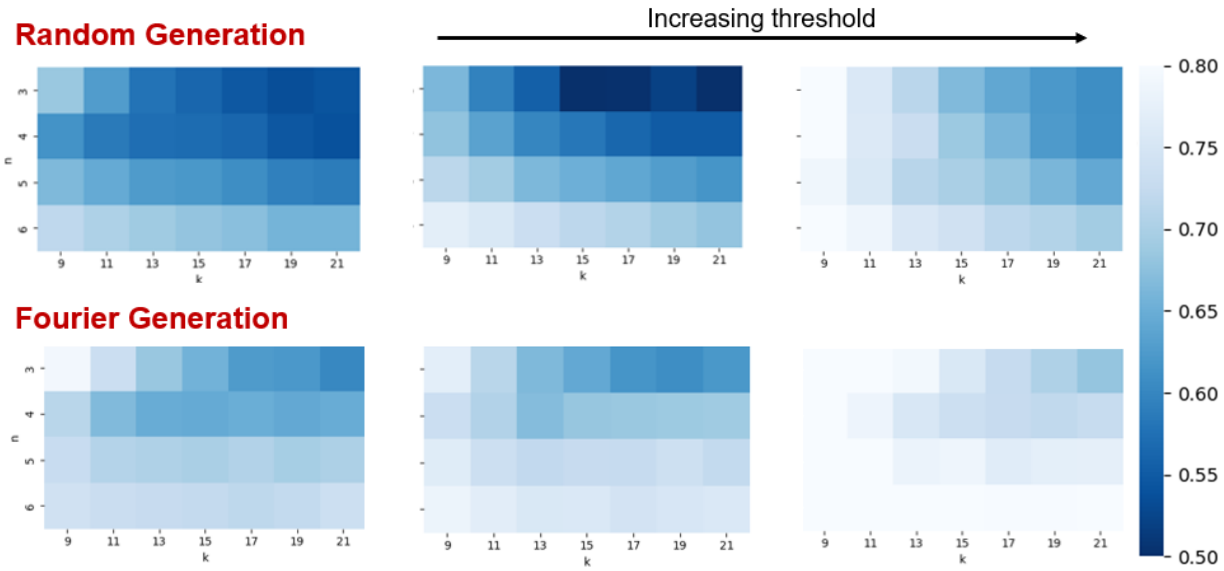

FIG. 8. Test accuracy of SVM with optimized RBF kernel on k-Forrelation datasets generated randomly and via discrete Fourier transform (algorithm description at the end)

Random sampling provides an upper bound on the classification difficulty of datasets generated by any algorithm. This is because an algorithm inherently imposes mathematical rules that limit which positive and negative samples are conceivable. For example, datasets generated from the discrete Fourier transform algorithm are generally easier to classify (Figure 8). The only challenge of random sampling is to sample a great number of positive instances when the positive *threshold* is high (say, 0.5). However, since it has been shown that a high *threshold* is not necessary (or desirable) to produce difficult datasets, random sampling is the most effective algorithm for our purpose.

Based on these conclusions, we suggest the generation of benchmark k-Forrelation datasets following the guidelines below:

- **Choose n:** Go as high as desirable, or computationally capable, depending on the purpose. If n_{algo} is the number of qubits being used in a quantum classifier, then it is suggested that $n_{algo} \geq n$
- **Choose k:** Go as high as possible according to the chosen n, with the restriction $k = \text{poly}(n)$. From our experiments, $k \approx n^2$ is a good heuristic
- **Choose positive threshold:** Near 0, but not overlapping with the negative class (defined to be instances with $|\Phi| \leq 0.01$) – e.g., around 0.02
- **Generative algorithm:** Random Sampling (for $n < 10$). Larger n would require Fourier Sampling, but the dataset is expected to be relatively easier than randomly sampled

4.4) Suggestion for the use of k-Forrelation datasets

K-Forrelation datasets are binary-labelled categorical datasets whose classification tasks are PromiseBQP-complete. By changing the values of n, k, and *threshold*, researchers can generate multiple datasets of varying difficulty. These datasets can be utilized in the following example applications:

- Benchmarking constructive algorithms for quantum classifiers
- Benchmarking dimensionality reduction methods in quantum communication (each k-Forrelation instance can be taken as storing its label as the target information. This information is jointly determined by elements of the vector that encodes the instance)
- Benchmarking architectures of quantum generative adversarial networks (QuGANs) whose task is to generate new valid k-Forrelation examples of a given class
- Evaluating quantum classifiers with *structural* feature map (instead of parameterized unitary)

5) NEXT STEPS

Results To Plan For:

1) **Goal:** Understand how dimensionality reduction on k-Forrelation datasets affects their learnability.
   Action: Train SVM with optimized RBF kernel on dimensionality-reduced k-Forrelation datasets. The dimension of the latent vector should reflect the approximated intrinsic dimensionality of the dataset (i.e., d(latent vector) $\approx k$)

2) **Goal:** Demonstrate benchmarking of constructive algorithms. (Need datasets to be compressible without significantly affecting learnability to make them compatible with existing constructive algorithms that involve parameterized unitaries and a small number of qubits)
   Action: Try Elham's algorithm and other constructive algorithms on the k-Forrelation datasets and show that they can be used as an informative comparison criterion

3) **Goal:** Demonstrate a constructive algorithm that involve structural feature embedding (inspiration from the k-Forrelation feature map), instead of parameterized unitary, for classification of categorical datasets *(likely another research on its own)*
Action: Think of an algorithm that search different "blueprints" (rules for structural embedding) and find the optimal one for a given dataset, and go from there. (already have one algorithm running but no significant results yet)

6) REFERENCES:

Aaronson, Scott, and Andris Ambainis. "Forrelation: A problem that optimally separates quantum from classical computing." *SIAM Journal on Computing 47.3* (2018): 982-1038. *arXiv preprint arXiv:1411.5729v1* (2014)

Perrier, Elija, Akram Youssry, and Chris Ferrie. "Qdataset: Quantum datasets for machine learning." *arXiv preprint arXiv:2108.06661* (2021)

Pope, Phillip, et al. "The intrinsic dimension of images and its impact on learning." *arXiv preprint arXiv:2104.08894* (2021).

Schatzki, Louis, et al. "Entangled datasets for quantum machine learning." *arXiv preprint arXiv:2109.03400* (2021)