Physics Letters A 525 (2024) 129895

Contents lists available at ScienceDirect

Physics Letters A

journal homepage: www.elsevier.com/locate/pla

Letter

# Quantum inspired kernel matrices: Exploring symmetry in machine learning

Sebastian Raubitzek [a,*], Sebastian Schrittwieser [b], Alexander Schatten [a], Kevin Mallinger [a,b]

[a] SBA Research gGmbH, Floragasse 7/5.OG, Vienna 1040, Austria
[b] University of Vienna, Faculty of Computer Science, Christian Doppler Laboratory for Assurance and Transparency in Software Protection, Kolingasse 14-16, Vienna 1090, Austria

ARTICLE INFO

Communicated by M.G.A. Paris

*Keywords:*
Lie group
Machine learning
Quantum computing
Quantum machine learning
Support vector machine
Classification
Feature map

ABSTRACT

Quantum machine learning, an emerging field at the intersection of quantum computing and classical machine learning, has shown great promise in enhancing computational capabilities beyond classical bounds. A key element in this area of research is the utilization of Quantum Kernel Estimators, traditionally grounded in the symmetries of SU(2) groups associated with qubits. This study extends the conceptual framework of Quantum Kernel Estimators to incorporate a broader spectrum of symmetry groups. By harnessing the various structures of Lie groups, we develop novel quantum-inspired feature maps that offer more flexible and potentially powerful ways to encode and compress classical data into quantum states. We present a comprehensive theoretical introduction for this approach, followed by a methodology that integrates the developed feature maps into quantum-inspired kernel classifiers. Our results, derived from a series of computational experiments across various datasets, demonstrate the efficacy of this approach in comparison to traditional quantum and classical machine learning models. The findings not only underline the versatility of Lie-group theory in potentially enhancing quantum machine learning algorithms but also open new avenues for exploring complex symmetries in quantum information processing. This research bridges a gap between the study of symmetries and machine learning, paving the way for more sophisticated quantum algorithms capable of tackling complex, high-dimensional data in ways previously unattainable.

## 1. Introduction

Despite the considerable theoretical potential of quantum technologies, their practical implementation within quantum computers and circuits remains at a developmental stage, particularly when compared to the ongoing advancements and capabilities of classical computing systems. However, the potential improvements through quantum information processing technologies such as Quantum Computers, Quantum Key Distribution (QKD), and, as is the focus of this article, Quantum Machine Learning (QML), are widely recognized and extensively researched [1–3].

QML is an area of research that combines the principles of quantum computing with machine learning concepts, offering a potential way to enhance the computational speeds and capabilities of machine learning approaches by mapping data-based learning approaches to quantum architectures. Central to the advancement of QML, and the focus of this article, are Quantum Kernel Estimators (QKEs), which blend quantum information processing and the classical concept of kernel estimators [4]. We build on the foundational work of Schuld et al., which has been instrumental in demonstrating the application of quantum computing principles in the machine learning landscape [4–6]. Additionally, leveraging developments within IBM's Qiskit platform [7], our study delves deeper into the potential capabilities of quantum information processing. Past research has shown that the capabilities of QML approaches are limited compared to state-of-the-art classical machine learning approaches [8], emphasizing the need to refine the conceptual framework of QML.

In examining the current state of QKEs, we find that these are confined by SU(2)-esque group symmetries, due to the corresponding limitations of quantum hardware. Our aim is to show that these information processing infrastructures can be extended to arbitrary Lie groups in a straightforward manner. By extending the theoretical and practical

* Corresponding author.
*E-mail addresses:* sraubitzek@sba-research.org (S. Raubitzek), sebastian.schrittwieser@univie.ac.at (S. Schrittwieser), alexander.schatten@sba-research.org (A. Schatten), kevin.mallinger@sba-research.org (K. Mallinger).

https://doi.org/10.1016/j.physleta.2024.129895
Received 5 July 2024; Received in revised form 3 September 2024; Accepted 10 September 2024
Available online 12 September 2024
0375-9601/

framework of QKEs to include arbitrary Lie groups, we align our work with advances in particle physics, where symmetries beyond SU(2) are commonly used [9].

*Contributions*

- Extending Quantum Kernel Estimators (QKEs) to include arbitrary Lie groups.
- Evaluating the performance of extended Lie group symmetries in QKEs against basic SU(2)-based QKEs and classical machine learning classifiers.
- Providing a comprehensive framework and code repository for reproducibility and further research in the field.

We present our ideas and work in the following way: First, in Section 2, we discuss related work to our approach. Then, in Section 3, we introduce Quantum Kernel Estimators and the corresponding feature maps. We then show where SU(2) symmetries are used in basic QML feature maps and provide an extension of these to arbitrary Lie groups in Section 3.2. We then test these extended quantum feature maps for a range of standard classification datasets against basic quantum feature maps and a standard classical machine learning approach as baselines. Next, we describe our experimental setup and the corresponding datasets in Section 4, whereas we present and discuss the results in Section 5. A discussion on our findings is provided in Section 6 and we conclude our findings in Section 7. The complete code for our approach and our experiments can be found in a corresponding GitHub repository for reproducibility.

## 2. Related work

The potential improvements through quantum information processing technologies such as Quantum Computers, Quantum Key Distribution (QKD), and Quantum Machine Learning (QML) are widely recognized and extensively researched [1–3]. Schuld et al. have been instrumental in demonstrating the application of quantum computing principles in the machine learning landscape [4–6]. Past research has shown that the capabilities of QML approaches are limited compared to state-of-the-art classical machine learning approaches [8].

Recent work by Glick et al. [10] has demonstrated the efficacy of quantum kernels for data with inherent group structures, suggesting ways to deconstruct symmetries into SU(2) algebras. This aligns with our aim to extend quantum information processing structures beyond SU(2). Our study builds on these foundations, evaluating the performance of extended Lie group symmetries in QKEs against basic SU(2)-based QKEs and classical machine learning classifiers.

The study by Heredge et al. (2024), [11] explores the novel application of non-unitary operations in quantum machine learning through the Linear Combination of Unitaries (LCU) method. The authors introduce quantum-native implementations of Residual Networks (ResNet) and average pooling layers, which traditionally face challenges due to the unitary nature of quantum operations. Their approach potentially mitigates the issue of barren plateaus in quantum variational circuits, a significant obstacle in deep quantum neural networks, by probabilistically implementing quantum ResNets. Additionally, the paper proposes a method for projecting quantum-encoded data onto irreducible representation subspaces, offering a flexible framework to enforce symmetry in quantum data encoding.

In the work by West et al. [12], the authors propose a novel framework for quantum data analysis that focuses on the principles of symmetry and invariant theory within the quantum realm. They introduce a class of quantum neural networks that are inherently equivariant to the actions of symmetry groups, particularly emphasizing the role of Lie groups in encoding symmetries into quantum circuits by focusing on expressive sub-Hilbertspaces.

## 3. Methodology

Quantum Machine Learning (QML) merges the potential of quantum computing with the data-driven approaches of machine learning. Here, we focus on Quantum Kernel Estimators (QKE) [4], exploring how their inherent mechanisms can be expanded to work with various group symmetries beyond the properties of standard qubits alone.

QKEs operate on two pivotal steps: feature encoding and kernel computation. Feature encoding transforms classical data into quantum states through a quantum feature map $\Phi$, mapping input data $x$ into a Hilbert space $\mathcal{H}$. This process can be represented as:

$$|\psi(x)\rangle = U_\Phi(x)\,|0\rangle\,, \tag{1}$$

where $U_\Phi(x)$ is a unitary operation dependent on $x$, encoding the data into a quantum state $|\psi(x)\rangle$ starting from the initial state $|0\rangle$.

The corresponding kernel computation in QKEs is then defined as the inner product between these quantum states:

$$K(x,x') = |\langle\psi(x)|\psi(x')\rangle|^2, \tag{2}$$

where $|\psi(x)\rangle$ and $|\psi(x')\rangle$ are the quantum states corresponding to data points $x$ and $x'$, respectively. This quantum kernel function is then straightforwardly used as a precomputed kernel matrix in a support vector framework, i.e., Support Vector Machines (SVM) [13], for tasks like classification and regression.

The methodology of QKE builds on a quantum feature map's ability to encode data into quantum states and the quantum system's capacity to compute complex kernel functions efficiently. These feature maps, especially those of the Pauli class, are based on SU(2) symmetry properties, meaning they are matrix transformations that follow certain rules to project arbitrary data onto a qubit. However, we aim to show that this concept can be expanded to arbitrary Lie groups, provided one can produce the corresponding algebra of a Lie group. To demonstrate this, we will first discuss quantum feature maps, including one used in IBM's Qiskit [14], show where the mechanics of Lie groups are utilized, and expand this approach to several other Lie groups.

### 3.1. Feature maps

Feature maps are necessary in QML, serving to make classical data quantum, i.e. project classical data onto quantum states. Among various and custom quantum feature maps, the Z and ZZ feature maps are standard choices implemented in IBM'S Qiskit, [7]. These feature maps use the properties of Pauli matrices to generate rotations in a complex two-dimensional space to encode classical data into the quantum realm. The basic idea here is that similar to a standard rotation matrix, parameterized using an angle $\theta \in [0, 2\pi]$, one expands this methodology to complex rotations which are parameterized using the Pauli-matrices, the generators of SU(2). This gives rise to the following feature maps:

#### 3.1.1. The Z feature map

The Z feature map employs the Pauli-Z operator to encode classical data into quantum states. For a given data point $x$, it applies a phase rotation to each qubit, proportional to the corresponding feature value in $x$. Mathematically, this operation is described by:

$$U_Z(x) = \exp\left(i\sum_j x_j Z_j\right), \tag{3}$$

where $Z_j$ is the Pauli-Z-matrix acting on the $j$-th qubit, and $x_j$ is the $j$-th component of $x$. This results in a rotation around the Z-axis of the Bloch sphere, effectively encoding the data within the phase of the quantum state. Throughout the remainder of this article, we denote all results associated with the Z feature map with $Q_Z$.

### 3.2. Exploring symmetry in feature maps

The Pauli matrices, integral to the Z and other feature maps, are essential in quantum computing, reflecting the SU(2) Lie group's algebraic structure. Their application through the exponential map generates SU(2) elements, indicative of rotations in three-dimensional space, and exemplifies the connection between quantum feature maps and Lie group theory.

These feature maps, i.e., unitary transformations, especially those of the Pauli class, are based on SU(2) symmetry properties, meaning that there are matrix transformations that follow certain rules to project arbitrary data on a qubit. The behavior of these Pauli-class feature maps is governed by the Pauli matrices, which are three $2 \times 2$ complex matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4}$$

This connection shows that the Z map leverages the SU(2) Lie group structure via Pauli matrices, facilitating the encoding of data into quantum states and integrating SU(2)'s structure into Quantum Machine Learning (QML). However, the Pauli matrices are not exactly generators of SU(2), such that they need to be slightly changed. The Lie algebra $\mathfrak{su}(2)$ is then spanned by the Pauli matrices multiplied by $\frac{1}{2}i$:

$$\mathfrak{su}(2) = \left\{ \frac{1}{2} i\sigma_x, \frac{1}{2} i\sigma_y, \frac{1}{2} i\sigma_z \right\} \tag{5}$$

Building upon this, we introduce a methodology to employ arbitrary Lie groups for data encoding, extending beyond the conventional SU(2) based feature maps. Lie groups, a blend of algebra and geometry named after Sophus Lie, encapsulate continuous symmetries as mathematical entities that marry group properties with manifold smoothness. These groups articulate rotational and translational symmetries within mathematical and physical realms, offering a potent framework for describing symmetries [9,15].

Central to Lie groups is the concept of generators, the elements of the corresponding Lie algebra representing "infinitesimal" symmetries. Through the exponential map, denoted as exp, these generators facilitate the construction of Lie group elements from algebra elements, bridging Lie algebra and Lie groups and enabling the expression of group elements as exponentials of algebra components. This framework allows for the expansive use of Lie groups in our approach, providing a mechanism for encoding data into Hilbert spaces.

Thus, we can build feature maps that harness the structure and symmetries of various Lie groups beyond SU(2). For instance, for an arbitrary Lie group with generators $T_i$, we can define a generalized feature map as:

$$U(\vec{x}) = \exp\left( i \sum_j x_j T_j \right), \tag{6}$$

where $x_j \in [0, \pi]$ are normalized components of the classical data, and $T_j$ are the generators of the Lie algebra associated with the group, $U$ is the resulting group element. This approach allows for encoding classical data into Hilbert space by arbitrary Lie groups.

For our encodings, we always identify a Lie group $G$ that is sufficiently large, specifically one within the families of SO($n$), SL($n$), SU($n$), GL($n$), U($n$), O($n$), T($n$) which has an adequate number of generators, i.e., more or equal to the number of features, to parameterize each feature on the Lie group. We use our normalized feature vector $\vec{x} = (x_0, x_1, \dots, x_m)$ to parameterize the generators, thereby obtaining the corresponding group element $U_G(\vec{x})$:

$$U_G(\vec{x}) = \exp\left( i \sum_j x_j T_{G,j} \right), \tag{7}$$

where $x_j$ are again the individual components of the normalized feature vector, $T_{G,j}$ are the generators of the selected symmetry group G, and $U_G(\vec{x})$ is a $n \times n$ matrix representing the group element, with $n$ being the dimension of the resulting group element. This is illustrated in Fig. 1.

Should the number of generators exceed the number of features, we set the parameters for the excess generators to zero. This encoding transforms our data samples or vectors into a new feature space represented by:

$$|\psi(\vec{x})\rangle = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}^{\mathrm{T}} \cdot U_G(\vec{x}) \quad , \tag{8}$$

*Symmetry Groups*

In our experiments, we consider various Lie groups in dimension $n$ to encode multidimensional data. The in-depth details are discussed in Appendix A.

### 3.3. Employed symmetry group families

- **Orthogonal Group,** $O(n)$
- **Special Orthogonal Group,** $SO(n)$
- **Special Linear Group,** $SL(n, \mathbb{R})$
- **General Linear Group,** $GL(n, \mathbb{R})$
- **Unitary Group,** $U(n)$
- **Special Unitary Group,** $SU(n)$

## 4. Experimental setup

The primary aim of this research is to demonstrate the practical viability and effectiveness of our proposed methodology, thereby serving as a proof of concept for the underlying ideas. We test our previously discussed ideas on a diverse collection of publicly accessible datasets, all of them classification tasks. These datasets mostly consist of numerical values; however, to incorporate non-numerical data seamlessly, we adopted a strategy of mapping each unique non-numerical value to a distinct numerical representation within the $[0, \pi]$ interval. This conversion ensures uniform data treatment across all experiments.

Further, all features are normalized to be in the range $[0, \pi]$.

Furthermore, we partition each dataset using an 80/20 split for training and testing, respectively.

Our methodology is compared against two baselines to benchmark its performance. The first comparison is with standard feature maps, e.g., the z-feature map utilized in quantum machine learning platforms like Qiskit. The second baseline is the CatBoost classifier/regressor, renowned for its efficacy across various tasks even with default settings [17,8,16,18], which is precisely how we are using it. These comparisons are important for contextualizing our contributions among other machine-learning approaches, thus highlighting the potential advantages and disadvantages.

The effectiveness of our algorithms is shown using accuracy, precision, recall, and the F1 score as primary metrics. Following the conventions and implementations established by scikit-learn [14], our selection of metrics ensures adherence to the best practices for evaluating classification tasks.

### 4.1. Datasets

We employed the following four binary classification task data sets to verify our ideas:

#### 4.1.1. Climate model simulation crashes dataset

**Climate Model Simulation Crashes Dataset** (OpenML: climate-model-simulation-crashes, ID: 1467): This dataset, hosted on OpenML and originally from the UCI repository, contains simulated data related to climate model crashes. It consists of a total of 540 instances with 18 numerical attributes. The dataset is used to predict whether a climate model simulation will crash (binary outcome: 0 for no crash, 1 for crash). The attributes include various environmental and operational

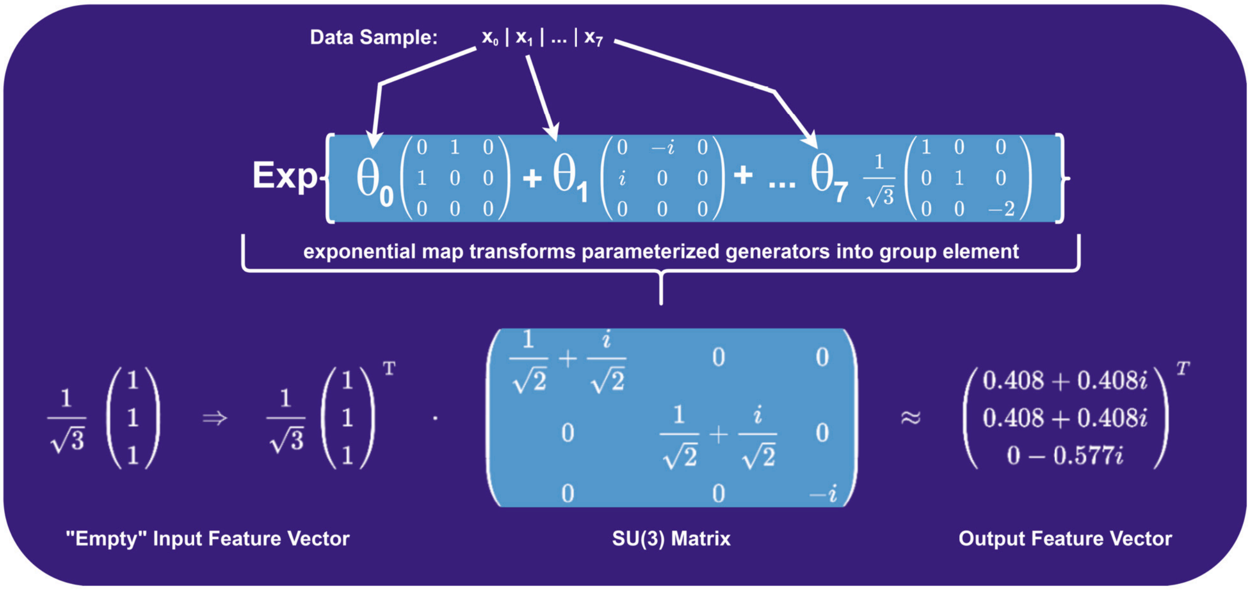


**Fig. 1.** Illustration of our data encoding strategy using the SU(3) group. A data sample with 8 features parameterizes the Gell-Mann matrices, which are then transformed into a group element via the exponential map. This group element is applied to a normalized "empty" input vector, yielding a complex three-component vector that embeds the information of the data sample. Note that the imaginary unit $i$ is part of *Exp* in this depiction.

conditions under which the climate model runs, such as temperature, pressure, and humidity.

#### 4.1.2. Indian Liver Patient Dataset (ILPD)

**Indian Liver Patient Dataset (ILPD)** (OpenML: ilpd, ID: 1480): Compiled by Bendi Venkata Ramana, M. Surendra Prasad Babu, and N. B. Venkateswarlu, this dataset focuses on diagnosing liver disease in patients from the north-east region of Andhra Pradesh, India. It includes records of 583 patients, with 416 liver patient records and 167 non-liver patient records. The dataset has 441 male and 142 female patient records and contains 11 attributes, including age, gender, and various liver function tests (like Total Bilirubin, Direct Bilirubin, Alkaline Phosphatase, Alanine Aminotransferase, Aspartate Aminotransferase, Total Proteins, Albumin), and Albumin and Globulin Ratio. The class label indicates whether the patient has liver disease (1) or not (2).

#### 4.1.3. Volcanoes on Venus - E5

**Volcanoes on Venus - E5** (OpenML: volcanoes-e5, ID: 1527): This dataset, hosted on OpenML, originates from radar images of Venus taken by the Magellan spacecraft. It focuses on the classification of volcanic structures. The dataset includes 152 instances with 4 numerical attributes that describe characteristics of the radar images, such as diameter, depth, and other morphological features. The target variable is binary, indicating whether the structure is a volcano (1) or not (0). This dataset is commonly used in image analysis and planetary science research to study the surface features of Venus.

#### 4.1.4. Thoracic surgery data

**Thoracic Surgery Data** (OpenML: thoracic-surgery, ID: 4329): This dataset was collected retrospectively at Wroclaw Thoracic Surgery Centre for patients who underwent major lung resections for primary lung cancer between 2007 and 2011. The Centre is associated with the Department of Thoracic Surgery of the Medical University of Wroclaw and Lower-Silesian Centre for Pulmonary Diseases, Poland. The research database is part of the National Lung Cancer Registry, administered by the Institute of Tuberculosis and Pulmonary Diseases in Warsaw, Poland. It comprises 470 instances with 17 attributes, including preoperative and operative characteristics such as age, diagnosis, forced vital capacity, and performance status. The binary class label (Risk1Yr) indicates the one-year survival status of the patient post-surgery (1 for survived, 0 for not survived).

### 4.2. Lie group symmetries and data

Lastly, we need to discuss connections between Lie group symmetries and data. The work by [12] is exemplary in taking advantage of symmetries in data. The researchers built a quantum neural network approach specifically designed to take advantage of the symmetries present in the data it was trained on. This means they ensured that their encoding commuted for different layers of their approach, thus respecting rotational symmetry by reducing the Hilbert space to a space that respects rotational symmetry. Furthermore, they selected qubits that could well express the carried information.

However, for the data presented in Section 4.1, we cannot identify certain symmetries based on its composition, as these datasets are not, in particular, e.g., image data with a particular symmetry present. Apart from datasets with rather obvious symmetries, one does not generally know if a dataset succumbs to certain symmetries, especially not for categorical data. Thus, it is necessary to test quantum machine learning approaches also on datasets with no apparent symmetries, as the previously discussed datasets are standard datasets chosen for availability and variety among numbers of features, and being made up of categorical and numerical features. These datasets serve as evidence that quantum learning approaches can perform reasonably even in non-professionally guided scenarios. However, what about datasets where obvious symmetries are incorporated in the presented dataset, also in the case of non-image but categorical and/or generally numerical data? Here, the approach by Raubitzek et al. [8] presents such a discussion for SU(2) symmetry, where the authors construct a dataset based on SU(2) symmetry, i.e., generating a classification dataset where certain parts of the SU(2) hypersphere are identified with one of two classes, i.e., constructing a binary classification with a feature space built from the symmetry. The researchers observe that no quantum advantage is given for these rather generally constructed datasets; sophisticated modern boost classifiers perform overall best. However, it needs to be mentioned that QML approaches performed (subjectively) reasonably, i.e., they did not underperform abysmally but provided, given a certain optimized set of hyperparameters, performances that could be used in a real-life classification task, though not being the best overall performances, especially in terms of speed.

Given the prior discussion on symmetry in data, we extend the approach from the work of Raubitzek et al. [8] to our current method. Specifically, we extend the SU(2)-based data approach to arbitrary symmetries and analyze if the proposed multi-Lie-group-encompassing QKE approach can yield reasonable performance in datasets where a respective part of the regarded Hilbert space is selected and identified with a certain class. Thus, we construct our dataset in the following way according to our prior introduction of the approach in Section 3.

Given a set of generators of an arbitrary Lie Group $T_{\mathrm{G,j}}$ and the corresponding group element:

$$U_{\mathrm{G}}\left(\vec{x}\right) = \exp\left(i\sum_{j} x_j T_{\mathrm{G},j}\right), \tag{9}$$

we know that this construction via the exponential map spans parts of our Lie manifold. We then choose our feature vector such that we assign a very particular part of our Lie manifold to a certain class, either 1 or 0, thus creating a binary classification task. Therefore, all we need to do is create a dataset consisting of a particular number of features $m$, ensuring that we have two regions of the Lie manifold to identify classes with.

We thus use a feature vector $\vec{x} = \left(x_0, x_1, x_2\right)$, where we select—via parametrization of this feature vector—two regions of the Lie manifold as:

$$\text{Class} = \begin{cases} 0 & \text{if } 0 < x_0 \leq \frac{\pi}{3}, \\ 1 & \text{if } \frac{2\pi}{3} < x_0 < \pi, \end{cases}$$

where we choose $x_0$ to be a random number within the boundaries of Classes 0 and 1.

The other components $x_1, x_2, \ldots$ are then chosen to be random numbers $0 < x_2 < \pi$ and $x_1$ is set to zero at all times.

It is important to note that these boundaries for feature-components are chosen arbitrarily, and different choices could be made depending on the specific application or depending on the study of the desired symmetry properties, e.g., researching the zeroth generator compared to the first one in terms of expressivity by setting one random and the other fixed.

Thus, we create a dataset consisting of 2 classes, 3 features, and generate 100, 250, 500, and 1000 samples overall, i.e., different amounts of data. Using this dataset with the discussed feature maps creates unique kernel matrices depending on the symmetry groups employed.

## 5. Results

This section consists of two parts; first, we discuss our exemplary real-life data set approaches, and then we discuss the approaches using synthetic data.

Our evaluation comprises several experiments documented in Tables 1, 2, 3, and 4, which detail the accuracy, precision, recall, and F1-score, respectively. These experiments leverage Lie group-based kernel matrices within a Support Vector Machine classifier framework. As noted by Shalev-Shwartz and Ben-David [19] and echoing the “No Free Lunch” Theorem [20], our findings affirm that no single algorithm universally outperforms across various datasets. This underscores the importance of tailoring assumptions in model selection, as a model’s performance significantly depends on data compatibility and methodological appropriateness.

In particular, the data reveal the need for assumptions regarding data symmetries for effective feature mapping onto manifolds, thereby distinguishing between classes. Our results illustrate that while non-quantum sophisticated boost classifiers like CatBoost generally dominate, the Z-feature map and other tested feature maps occasionally surpass the baseline, aligning with findings by Raubitzek et al. [8]. This indicates that different symmetry groups yield varying levels of performance depending on the problem at hand, reaffirming the principle that no single approach is universally applicable.

**Specific Observations:**

- For the *climate-model-simulation-crashes* dataset, the Special Linear Group (SL) and General Linear Group (GL), along with CatBoost, provide the highest accuracy. Precision, recall, and F1-score metrics indicate varying top performers, with the SL group showing particularly strong results according to the F1-score.
- The *haberman* dataset sees CatBoost outperforming all other approaches in all metrics, with SL and the Z-feature map also delivering competitive results.
- In the *Indian Liver Patient (ilpd)* dataset, while CatBoost leads in accuracy, the Z-feature map shows the best precision and F1-score.
- For the *thoracic-surgery* dataset, CatBoost again leads in accuracy, with the SL group and Z-feature map excelling in precision and recall/F1-score respectively.

**Key Findings:**

- All approaches perform reasonably well across different datasets.
- The SL group exhibits the strongest performance among our Lie group-based approaches.
- CatBoost confirms its effectiveness with robust performance across all metrics without the need for hyperparameter tuning, underscoring its utility in delivering high-quality results with minimal configuration.

### 5.1. Kernel matrices

In addition to our performance comparison, we also depicted the constructed kernel matrices using correlation plots, as done in similar work [21–24]. This approach provides another layer of interpretation for our experiments. Different groups compress data differently, leading to varying encoding dimensions of the data under study.

Observing the plots in Fig. 2, we notice that our Lie group-based feature maps produce kernel matrices for the SL($n$) case that are denser and exhibit less variation than the standard Z feature map. Although the variation in absolute values is higher for the SL($n$) case, the relative variation is less, as indicated by the many dark blue areas. We conclude that this is because the Lie group approaches compress the data, whereas the Z feature map expands the feature space. Furthermore, the standard Z feature map reveals a strongly visible line of correlation, which is barely noticeable in the SL($n$)-based kernel matrix. To illustrate these differences, we have plotted the best-performing kernel matrix, i.e., the SL($n$)-group-based one.

Additionally, we plotted in Fig. 2 a kernel matrix for the same problem that also has high accuracy and looks similar to the standard Z feature map, i.e., a very visible line of correlation in the center. Overall, we conclude that the best-performing kernel matrices of our approach can be fundamentally different from the standard feature map approaches both because of compression and because the encoding happens along many degrees of freedom, whereas the standard Z feature map manipulates only one degree of freedom, rotations on the Z-axis, and expands the features onto a complex 2-dimensional space using the respective Pauli generator.

### 5.2. Results synthetic data

Our second line of experiments details how the different Lie groups for encoding data work on a simple dataset where we know the result for sure, i.e., the dataset described in Section 4.2. Our results are collected in Tables 5, 6, 7, 8. What these results should show is that certain

**Table 1**
Accuracy of various approaches across different datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| climate-model-simulation-crashes | 0.8457 | 0.3519 | **0.9136** | 0.9074 | **0.9136** | 0.9074 | 0.3519 | **0.9136** |
| haberman | 0.7283 | 0.4783 | 0.7391 | 0.4130 | 0.7283 | 0.4130 | 0.4783 | **0.7500** |
| ilpd | 0.7200 | 0.4400 | 0.7143 | 0.5029 | 0.6629 | 0.5029 | 0.4400 | **0.7314** |
| thoracic-surgery | 0.7943 | 0.7305 | 0.8085 | 0.7447 | 0.7801 | 0.7518 | 0.7305 | **0.8227** |

**Table 2**
Precision of various approaches across different datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| climate-model-simulation-crashes | 0.5503 | 0.5309 | **0.7120** | 0.4565 | 0.4568 | 0.4565 | 0.5309 | 0.4568 |
| haberman | 0.6498 | 0.4877 | 0.6780 | 0.4675 | 0.6513 | 0.4675 | 0.4877 | **0.6896** |
| ilpd | 0.6451 | 0.4595 | 0.3634 | 0.5311 | 0.5269 | 0.5311 | 0.4595 | **0.6478** |
| thoracic-surgery | 0.5278 | 0.4882 | **0.5608** | 0.4980 | 0.4104 | 0.4486 | 0.4882 | 0.4143 |

**Table 3**
Recall of various approaches across different datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| climate-model-simulation-crashes | 0.5598 | **0.5806** | 0.5647 | 0.4966 | 0.5000 | 0.4966 | **0.5806** | 0.5000 |
| haberman | 0.6008 | 0.4848 | 0.5851 | 0.4627 | 0.5659 | 0.4627 | 0.4848 | **0.6276** |
| ilpd | **0.6470** | 0.4489 | 0.4883 | 0.5390 | 0.5204 | 0.5390 | 0.4489 | 0.6279 |
| thoracic-surgery | **0.5118** | 0.4899 | 0.5203 | 0.4984 | 0.4701 | 0.4696 | 0.4899 | 0.4957 |

**Table 4**
F1-score of various approaches across different datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| climate-model-simulation-crashes | 0.5541 | 0.3238 | **0.5882** | 0.4757 | 0.4774 | 0.4757 | 0.3238 | 0.4774 |
| haberman | 0.6077 | 0.4575 | 0.5856 | 0.4086 | 0.5590 | 0.4086 | 0.4575 | **0.6391** |
| ilpd | **0.6460** | 0.4194 | 0.4167 | 0.4878 | 0.5177 | 0.4878 | 0.4194 | 0.6349 |
| thoracic-surgery | **0.5024** | 0.4883 | 0.5107 | 0.4964 | 0.4382 | 0.4556 | 0.4883 | 0.4514 |

symmetries should work perfectly for the discussed data, as the information on the correct class is not obscured by a compression of the exponential map, thus generating expressive kernel matrices. In contrast, other symmetries should perform less well, depicting the information less accurately, compressing the information too much, and allowing the random or zero part of the feature to dominate the composition of the qubit/vector. And this is exactly what we see: the presented tables show that the Pauli-Z feature map always provides 100% accuracy, just like our CatBoost classifier. This is because the Pauli-Z matrix feature map projects each feature onto a single qubit, thus allowing the kernel classifier to easily single out the information about the correct class. CatBoost, on the other hand, is just an easy-to-use sophisticated boost classifier which, given the evidence of its successful applications across various problems, should also be capable of singling out the feature that contains the information about the correct class.

The situation looks different, however, for the different quantum-inspired feature maps based on varying Lie groups. Here, we see that the orthogonal and special orthogonal groups show almost perfect scores across all experiments and all metrics, except for the case with 1000 data points where the score is diminished. We interpret this diminished score for all metrics as a result of the dataset being composed of two random components, which might confuse the kernel estimator at a certain number of data points and thus slightly shift the boundary vector margins within the built model, generating a less than optimal model. We conclude, however, that the orthogonal and special orthogonal groups $O(n)$ and $SO(n)$ allow for good separation in this case. The $SU(2)$ group, like the unitary group, does not allow for good separation in this case. The special linear group and the general linear group also allow for reasonable scores across all metrics and different amounts of data. What is also visible here is that the special orthogonal $SO(n)$ and orthogonal $O(n)$ groups allow for the same expressivity in terms of good scores for the encoding of the data, as do the special linear $SL(n)$ and general linear $GL(n)$ groups, and the same for the unitary $U(n)$ and special unitary groups $SU(n)$. We can thus group these matrices into three families, and we conclude, given this evidence, that the special unitary and unitary groups do not allow for further compressing the information onto qubits, whereas the others do to varying degrees. This is supported by the fact that Pauli-Z encoding also does not compress the data onto a qubit but rather encodes one feature at a time onto one qubit.

## 6. Discussion

We performed a range of experiments to demonstrate how one can extend the computation of kernel matrices by employing arbitrary symmetry groups, i.e., Lie groups. Here, we built on a framework from Quantum Machine Learning where feature maps are used to encode data onto qubits. We extended this idea by generalizing the encoding process, which, in the standard case, is based on $SU(2)$ symmetry and uses rotation to encode data onto a qubit. We extended this approach by not only using $SU(2)$ symmetries but also a range of different Lie groups, i.e., $SO(n)$, $SL(n)$, $SU(n)$, $GL(n)$, $U(n)$, $O(n)$. The standard Pauli-Z matrix approach uses the exponential map and one of the generators of $SU(2)$; we use the exponential map and more than one generator per qubit. Our approach differs from the standard approach in terms of compression, i.e., we allow more information to be encoded onto one qubit/vector.

We tested this approach using standard machine learning classification data with mixed numerical and categorical features (Section 4.1) and a very simple dataset to observe the differences among different Lie groups in terms of encoding simplified data points (Section 4.2). Our results for the real-life datasets show that extending the quantum kernel estimator framework can provide reasonable results for certain datasets compared to our baseline, a CatBoost classifier. By "reason-

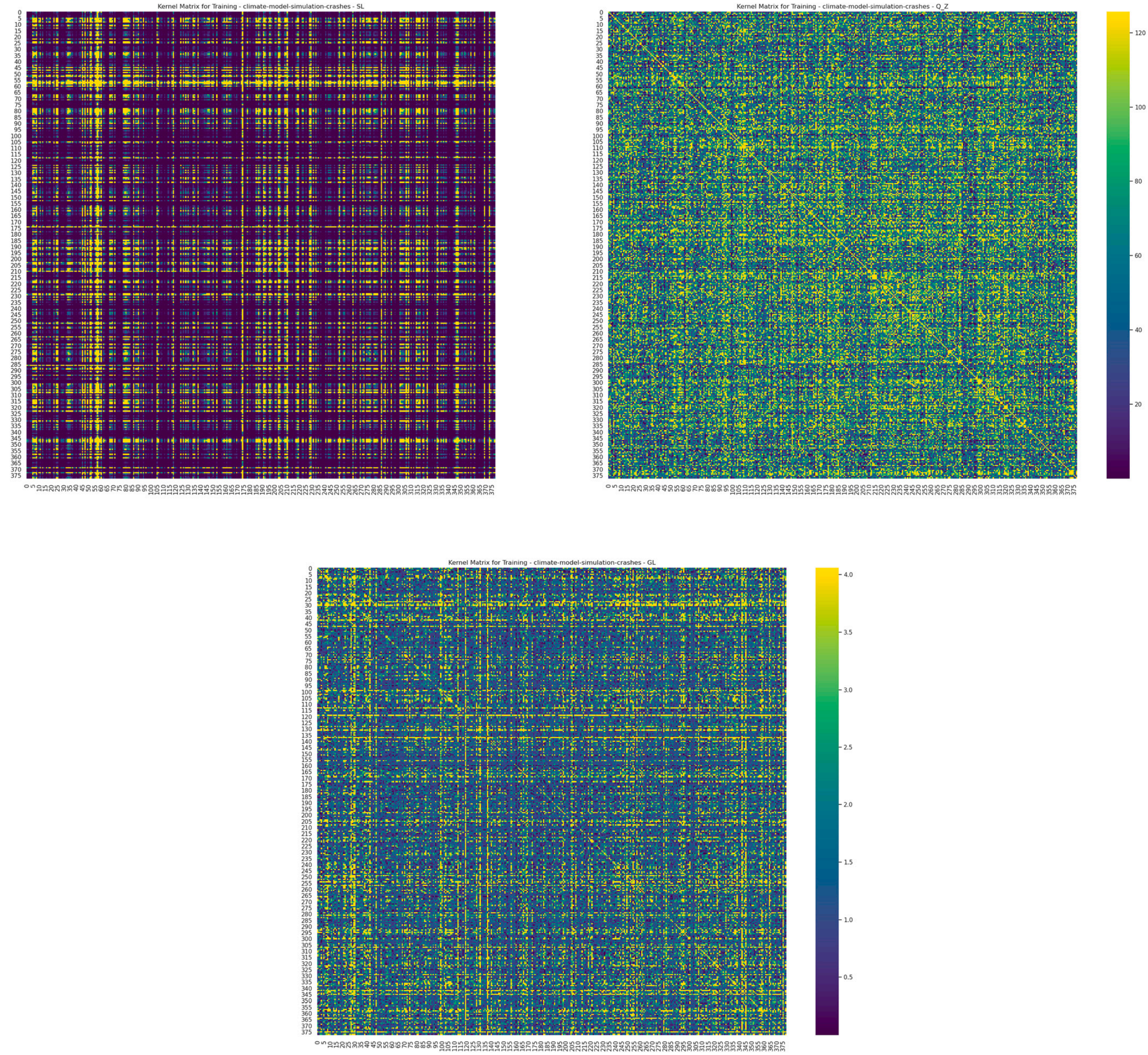


**Fig. 2.** Kernel matrices for the climate-model-simulation-crashes datasets. The left image depicts our SL(n) based kernel matrix, while the right image shows the kernel matrix generated using the Z-feature map. The bottom image is the GL(n) based Kernel matrix.

**Table 5**
Accuracy of various approaches across synthetic datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| Synthetic_100 | **1.0000** | **1.0000** | 0.9000 | 0.6000 | 0.8667 | 0.6000 | **1.0000** | **1.0000** |
| Synthetic_250 | **1.0000** | **1.0000** | 0.9733 | 0.4133 | 0.8800 | 0.4133 | **1.0000** | **1.0000** |
| Synthetic_500 | **1.0000** | **1.0000** | 0.9933 | 0.5867 | 0.8333 | 0.5867 | **1.0000** | **1.0000** |
| Synthetic_1000 | **1.0000** | 0.8667 | 0.9633 | 0.5233 | 0.8533 | 0.5233 | 0.8667 | **1.0000** |

**Table 6**
Precision of various approaches across synthetic datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| Synthetic_100 | **1.0000** | **1.0000** | 0.9250 | 0.5928 | 0.8667 | 0.5928 | **1.0000** | **1.0000** |
| Synthetic_250 | **1.0000** | **1.0000** | 0.9714 | 0.4192 | 0.8991 | 0.4192 | **1.0000** | **1.0000** |
| Synthetic_500 | **1.0000** | **1.0000** | 0.9936 | 0.6371 | 0.8526 | 0.6371 | **1.0000** | **1.0000** |
| Synthetic_1000 | **1.0000** | 0.8860 | 0.9669 | 0.5198 | 0.8704 | 0.5198 | 0.8860 | **1.0000** |

**Table 7**
Recall of various approaches across synthetic datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| Synthetic_100 | **1.0000** | **1.0000** | 0.8846 | 0.5928 | 0.8733 | 0.5928 | **1.0000** | **1.0000** |
| Synthetic_250 | **1.0000** | **1.0000** | 0.9762 | 0.4502 | 0.8669 | 0.4502 | **1.0000** | **1.0000** |
| Synthetic_500 | **1.0000** | **1.0000** | 0.9932 | 0.5942 | 0.8302 | 0.5942 | **1.0000** | **1.0000** |
| Synthetic_1000 | **1.0000** | 0.8630 | 0.9621 | 0.5165 | 0.8570 | 0.5165 | 0.8630 | **1.0000** |

**Table 8**
F1-score of various approaches across synthetic datasets.

| Data Set \ Approach | $Q_Z$ | SO | SL | SU | GL | U | O | CatBoost |
|---|---|---|---|---|---|---|---|---|
| Synthetic_100 | **1.0000** | **1.0000** | 0.8942 | 0.5928 | 0.8661 | 0.5928 | **1.0000** | **1.0000** |
| Synthetic_250 | **1.0000** | **1.0000** | 0.9731 | 0.3731 | 0.8743 | 0.3731 | **1.0000** | **1.0000** |
| Synthetic_500 | **1.0000** | **1.0000** | 0.9933 | 0.5550 | 0.8300 | 0.5550 | **1.0000** | **1.0000** |
| Synthetic_1000 | **1.0000** | 0.8640 | 0.9632 | 0.4983 | 0.8524 | 0.4983 | 0.8640 | **1.0000** |

able," we mean that when choosing the right symmetry group, we can achieve results comparable to or with the same accuracy as a powerful modern boost classifier; Tables 1, 3, 2, 4. We observed similar insights from the results for the synthetic datasets, where we produced an easy-to-solve classification with varying numbers of samples and inherent noise; Tables 5, 7, 6, 8. In many cases, the different symmetry feature maps are capable of encoding the data such that we can achieve a perfect score, just like our CatBoost baseline. However, there are certain cases where they do not provide good results, i.e., drastically underperform. In these cases, because of the composition of our synthetic dataset, we can conclude that the employed Lie group did not expressively map the information onto the qubit/vector. This means that due to the composition of our synthetic dataset and the parametrization of the different generators of the employed Lie groups, we see that the data maps onto certain regions of the Lie manifold, thus producing matrices that are certainly unique. However, as these obtained matrices from the parameterized generators in the exponential are applied to our qubits/vectors, we can lose information and the ability to properly express the encoded data. Contrary to this, we see that the standard Pauli-Z matrix feature map does not suffer from bad performance in this regard. Given this evidence, we conclude that the standard Pauli feature encoding does not compress the encoded data too much to blur information and further, this Pauli-Z matrix allows for expressive data encoding. For our approach and the other Lie groups, this is not the case, such that our approach always comes with a compression (assuming we have more than one feature) as all features are encoded onto one qubit/vector (with varying length), and further, we did not particularly choose the most expressive generators, i.e., parts of the Hilbert space. Thus, we conclude that information about the dataset at hand, about its symmetry properties in terms of the orthogonality of the different features to each other, and information about which generators of which group allow for expressive encodings can improve the performance of this Lie-group-based learning approach.

Several works emphasize that analyzing the data at hand and choosing an expressive encoding, and further altering the Hilbert space with respect to the symmetry properties at hand, can yield quantum machine learning approaches with superior performance compared to classical machine learning approaches. Here we want to mention the work by West et al. [12], where the researchers report drastically outperforming classical machine learning approaches by considering the symmetry of the data and identifying rotationally equivariant sub-Hilbert spaces for the discussed data, such that one can reduce the Hilbert space of the data at hand to trainable subspaces and feasible quantum machine learning approaches. This also emphasizes the need for developing highly specialized architectures for quantum machine learning, just as for neural networks. Our results and study tie into this work such that the presented approach can be used for a similar goal, i.e., to reduce the size of the Hilbert space and thus make the approach more feasible, but not by singling out relevant subspaces but by encoding multiple features onto the same qubit using a more dense encoding. We already discussed that this more dense encoding, and the resulting compression comes with drawbacks, but it also comes with opportunities to scale down the Hilbert space and thus the number of required qubits.

However, our approach still deals with all of this on an abstract level, and to the best of our knowledge, most of the regarded symmetries and encodings cannot be realized experimentally yet, as real-life quantum systems not only do not adhere to many of the regarded symmetries but are also bound to principles such as unitarity. Here, we need to mention another article that ties into this line of research. The work by [11] discusses ideas for non-unitary quantum machine learning. I.e., they present probabilistic quantum algorithms that overcome the normal unitary restrictions by employing a Linear Combination of Unitaries (LCU) method. Furthermore, they propose a framework for identifying irreducible quantum subspace projections for quantum encoded data. Thus, our approach might be linked to this work such that this identification of irreducible quantum subspaces could be used to identify the best level of compression of the presented encodings, and further since restrictions such as unitarity can be circumvented to some degree. This might pave the way for further breaking restrictive symmetry properties and thus allow for a greater variety of symmetry considerations in quantum information processing in general.

So, we ask one final question: Do quantum machine learning approaches always require sophisticated engineering solutions for the Hilbert space and feature engineering to show quantum advantage? Taking into account the results from [12,11], where the researchers report outperforming classical machine learning approaches by considering the symmetry of the data at hand, and our results, the answer is *yes*—for the current state-of-the-art. Considering this, our presented Quantum Kernel Estimator approaches are not nicely generalizable, such that one cannot simply apply a standard approach to arbitrary datasets and expect reasonable results. On the contrary, there seem to be particular Lie groups that are well-suited for certain datasets, allowing for the most expressive encoding. This is because quantum encodings span regions of the Lie manifold via the exponential map, which are then encoded onto a qubit. If the chosen encoding cannot separate the data adequately, i.e., distribute it onto the Hilbert space effectively, one might encounter non-expressive qubits. In such cases, even though the element of the Lie group is unique and expressive, the resulting qubit/vector after encoding isn't. This means that different Lie group elements might provide similar qubits/Hilbert space vectors. Thus, as with most learning-based approaches, the no-free-lunch theorem applies, implying that to obtain the best possible solution for a given task, one needs to explore the task, obtain information, and incorporate this information into the architecture of the learning approach to obtain quantum advantage [20]. In other words, no algorithm/approach fits every dataset best, and we have to choose the most suitable encoding and possibly manipulate the corresponding Hilbert space in order to obtain the best results and a trainable quantum classifier.

Still, sophisticated boost classifiers like CatBoost [17] are a strong baseline to which researchers should compare their results, as these classifiers not only perform well in terms of accuracy on a wide range of datasets but can also be easily implemented and tested against.

## 7. Conclusion

In this study, we explored the application of Lie group-based kernel matrices in a Support Vector Machine classifier framework. Our experiments confirmed that no single algorithm, including advanced non-quantum classifiers like CatBoost, consistently outperforms others on all metrics, which aligns with the “No Free Lunch” Theorem [20]. This variability underscores the importance of choosing appropriate model assumptions based on the specific data symmetries and features of each dataset. Among our presented ideas, the Special Linear Groups (SL($n$)) often showed superior performance for real-life data sets, suggesting that different symmetry groups might be suited for different data sets, as one would expect from a novel data-based learning approach.

A major limitation, however, is the incompatibility of arbitrary Lie groups with standard quantum architectures, which predominantly operate on $SU(2)$ symmetries and unitary transformations. Despite this, the diversity in quantum computing approaches provides a theoretical foundation for potential future hardware that can exploit, e.g. $SU(3)$ symmetries. This would allow us to construct quantum kernel matrices similar to those created using $SU(2)$ encodings, e.g. z-feature map encodings. The work by Chi et al. [25] on high-dimensional quantum information processing, particularly using qudits, highlights the potential of high-dimensional quantum states (qudits) to enhance quantum computation and quantum simulations beyond the standard qubit frameworks. Integrated photonic platforms, as discussed, offer a promising pathway for implementing scalable and programmable high-dimensional quantum architectures. Thus higher dimensional quantum machine learning approaches employing actual quantum hardware are not completely out-of-reach.

There is much freedom in choosing the particular quantum encoding and parametrizations, the standard z-feature map approach uses one parametrized Pauli matrix to encode one feature onto a qubit. Theoretically, one could use all three generators of the $SU(2)$ group to encode a feature onto a qubit using each generator. However, the standard approach leads to an expansion of the feature space, as one feature is projected onto a 2D complex vector. In contrast, using more generators of the $SU(2)$ group would lead to a more compressed approach. Following this line of thought, moving into higher dimensions with arbitrary Lie groups produces even greater compression. For example, with $SU(3)$, one can encode 8 features onto 3 complex components, thus compressing the feature space into fewer dimensions. However, this makes different approaches variably expressive and, considering that we are discussing machine learning approaches here, a compression of the analyzed data is sometimes not desirable as it might lead to diminished classification accuracies for certain datasets.

In general, the takeaway from this article is that one can choose arbitrary Lie groups to construct kernel matrices, thus performing quantum(-inspired) machine learning using arbitrary Lie groups and, respectively, their algebra to encode data onto $n$-dimensional real or complex vectors/qubits/qudits, [25]. In the future, this approach could utilize more exotic quantum hardware that allows for different data processing strategies, and particularly, encodings that diverge from current $SU(2)$-Pauli-matrices-based approaches.

Given further development in this direction and the previously demonstrated quantum advantages shown by quantum machine learning approaches that either break unitarity [11] and/or build particular symmetry-based data engineering for specific datasets [12], we estimate that the inclusion of a broader variety of Lie groups, and thus possibilities for different information processing, allows for better capturing the complexity inherent in real-life datasets.

## Code availability

The full Python program code contains scripts to create generators for all mentioned Lie groups, scripts to check their correctness, and the whole feature encoding using these generators. Further, we provide the full machine learning pipeline, including the classical ML approach. The full code—including an easy-to-use script to try the approach on your ideas—will be made available upon acceptance of this article.

## CRediT authorship contribution statement

**Sebastian Raubitzek:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sebastian Schrittwieser:** Writing – review & editing, Writing – original draft, Project administration, Funding acquisition. **Alexander Schatten:** Writing – review & editing, Writing – original draft, Project administration. **Kevin Mallinger:** Writing – review & editing, Writing – original draft, Project administration, Funding acquisition, Data curation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

All data is available from open data repositories

## Acknowledgements

The financial support by the Austrian Federal Ministry of Labour and Economy, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

## Appendix A. Symmetry groups

In our experiments, we consider various Lie groups in dimension $n$ to encode multidimensional data. This appendix provides a concise overview of these groups and their associated Lie algebras:

*Orthogonal Group, $O(n)$*

The Orthogonal Group in dimension $n$, denoted as $O(n)$, comprises all $n \times n$ orthogonal matrices that preserve a fixed point in an $n$-dimensional Euclidean space. The group operation is matrix multiplication. Orthogonal matrices, denoted by $Q$, satisfy $Q^\top Q = QQ^\top = I$, where $Q^\top$ is the transpose of $Q$, and $I$ is the identity matrix.

The Lie algebra associated with $O(n)$ is $\mathfrak{so}(n)$, consisting of all $n \times n$ skew-symmetric matrices, $A$, such that $A^\top = -A$. Key properties of the generators of $\mathfrak{so}(n)$ include:

- **Skew-Symmetry**: Each generator is a skew-symmetric matrix, equal to the negative of its transpose.
- **Closure under the Commutator**: The commutator of any two generators is a linear combination of the generators.
- **Orthogonality**: Exponentiating any generator results in an orthogonal matrix.
- **Dimensionality**: Each generator is an $n \times n$ matrix.
- **Number of Generators**: There are $\frac{n(n-1)}{2}$ generators.

The Special Orthogonal Group, $SO(n)$, is a normal subgroup of $O(n)$ consisting of matrices with determinant 1, preserving orientation. While the Lie algebra $\mathfrak{so}(n)$ is the same for both $O(n)$ and $SO(n)$, $SO(n)$ represents the connected component of the identity in $O(n)$.

*Special Orthogonal Group, $SO(n)$*

The Special Orthogonal Group in dimension $n$, denoted $SO(n)$, is a subgroup of the Orthogonal Group consisting of matrices with determinant 1. It represents the group of rotations in an $n$-dimensional Euclidean space.

The Lie algebra associated with $SO(n)$, denoted $\mathfrak{so}(n)$, consists of all $n \times n$ skew-symmetric matrices. The generators of $\mathfrak{so}(n)$ exhibit the following properties:

- **Skew-Symmetry**: Each generator is a skew-symmetric matrix.
- **Closure under the Commutator**: The commutator of any two generators is a linear combination of the generators.
- **Orthogonality**: Exponentiating any generator results in an orthogonal matrix.
- **Determinant +1**: The determinant of exponentiated generators is +1.
- **Dimensionality**: Each generator is an $n \times n$ matrix.
- **Number of Generators**: There are $\frac{n(n-1)}{2}$ generators.

*Special Linear Group, $SL(n, \mathbb{R})$*

The Special Linear Group, $SL(n, \mathbb{R})$, consists of $n \times n$ matrices with determinant 1, under matrix multiplication and inversion. It is a normal subgroup of the General Linear Group $GL(n, \mathbb{R})$.

The Lie algebra $\mathfrak{sl}(n, \mathbb{R})$ comprises traceless $n \times n$ matrices, with the Lie bracket defined as the commutator $[A, B] = AB - BA$. Properties of the generators of $SL(n, \mathbb{R})$ include:

- **Closure under the Commutator**: The commutator of any two generators remains within the span of the generators.
- **Determinant Condition**: Exponentiating any generator yields a matrix with determinant 1.
- **Dimensionality**: Each generator is an $n \times n$ matrix.
- **Number of Generators**: There are $n^2 - 1$ generators.

*General Linear Group, $GL(n, \mathbb{R})$*

The General Linear Group, $GL(n, \mathbb{R})$, includes all invertible $n \times n$ matrices over the real numbers, capturing all possible linear transformations such as rotations, reflections, scalings, and shears.

Its Lie algebra, $\mathfrak{gl}(n, \mathbb{R})$, is composed of all $n \times n$ matrices, representing all possible infinitesimal transformations. Properties of the generators of $GL(n, \mathbb{R})$ include:

- **Closure under the Commutator**: The commutator of any two generators remains within the span of the generator set.
- **Invertibility**: Matrices generated from any random linear combination of the generators are invertible.
- **Dimensionality**: Each generator is an $n \times n$ matrix.
- **Number of Generators**: The set contains $n^2$ generators.

*Unitary Group, $U(n)$*

The Unitary Group, $U(n)$, includes all $n \times n$ unitary matrices, which are crucial in quantum mechanics where transformations must preserve the inner product structure of state spaces.

The corresponding Lie algebra $\mathfrak{u}(n)$, composed of skew-Hermitian matrices, reflects the infinitesimal versions of these transformations. Properties of the generators of $\mathfrak{u}(n)$ include:

- **Skew-Hermitian**: Each generator is equal to the negative of its complex conjugate transpose.
- **Traceless (for $SU(n)$ subset)**: For the $SU(n)$ subgroup, the generators are traceless.
- **Number of Generators**: There are $n^2$ generators.
- **Algebra Closure**: The commutator of any two generators is a linear combination of the generators.
- **Unitarity**: Exponentiating any generator yields a unitary matrix.
- **Dimensionality**: Each generator is an $n \times n$ matrix.
- **Global Phase Factor**: The generators include a purely imaginary multiple of the identity matrix, reflecting the $U(1)$ subgroup within $U(n)$.

*Special Unitary Group, $SU(n)$*

The Special Unitary Group, $SU(n)$, contains all $n \times n$ unitary matrices with determinant 1, essential in quantum mechanics for symmetry operations that preserve probabilities.

The Lie algebra $\mathfrak{su}(n)$ includes traceless skew-Hermitian matrices. Properties of the generators of $\mathfrak{su}(n)$ include:

- **Skew-Hermitian**: Each generator is equal to the negative of its complex conjugate transpose.
- **Traceless**: The trace of each generator is zero.
- **Number of Generators**: There are $n^2 - 1$ generators.
- **Algebra Closure**: The commutator of any two generators is a linear combination of the generators.
- **Unitarity**: Exponentiating any generator yields a unitary matrix.
- **Determinant = 1**: The determinant of exponentiated generators is 1.
- **Orthogonality**: Generators are orthogonal under the Hilbert-Schmidt inner product.

## References

[1] Nandhini S, Harpreet Singh, Akash U. N, An extensive review on quantum computers, Adv. Eng. Softw. 174 (2022) 103337, https://www.sciencedirect.com/science/article/pii/S0965997822002381.
[2] P. Kong, A review of quantum key distribution protocols in the perspective of smart grid communication security, IEEE Syst. J. 16 (2022) 41–54.
[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (2017) 195–202, https://doi.org/10.1038/nature23474.
[4] M. Schuld, N. Killoran, Quantum machine learning in feature Hilbert spaces, Phys. Rev. Lett. 122 (2019) 040504, https://link.aps.org/doi/10.1103/PhysRevLett.122.040504, Visited on 2024-01-10.
[5] M. Schuld, F. Petruccione, Quantum ensembles of quantum classifiers, Sci. Rep. 8 (2018) 2772, https://doi.org/10.1038/s41598-018-20403-3.
[6] M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning, Contemp. Phys. 56 (2015) 172–185, https://doi.org/10.1080/00107514.2014.964942.
[7] Q. Community, Qiskit: an open-source framework for quantum computing, https://qiskit.org/, 2022, Visited on 2024-02-2024.
[8] S. Raubitzek, K. Mallinger, On the applicability of quantum machine learning, Entropy 25 (2023), https://www.mdpi.com/1099-4300/25/7/992.
[9] H. Georgi, Lie Algebras in Particle Physics: from Isospin to Unified Theories, CRC Press, 2019.
[10] J. Glick, T. Gujarati, A. Córcoles, Y. Kim, A. Kandala, J. Gambetta, K. Temme, Covariant quantum kernels for data with group structure, Nat. Phys. 20 (2024) 479–483, https://doi.org/10.1038/s41567-023-02340-9.
[11] J. Heredge, M. West, L. Hollenberg, M. Sevior, Non-unitary quantum machine learning, https://arxiv.org/abs/2405.17388, 2024.
[12] M. West, J. Heredge, M. Sevior, M. Usman, Provably trainable rotationally equivariant quantum machine learning, PRX Quantum 5 (2024) 030320, https://link.aps.org/doi/10.1103/PRXQuantum.5.030320.
[13] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297, https://doi.org/10.1007/BF00994018.
[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 282–290, https://scikit-learn.org/stable/. (Accessed 18 April 2023).
[15] B. Hall, Lie Groups, Lie Algebras, and Representations: An Elementary Introduction, Springer, Cham, 2015, https://link.springer.com/book/10.1007/978-3-319-13467-3. (Accessed 13 March 2024).
[16] S. Raubitzek, L. Corpaci, R. Hofer, K. Mallinger, Scaling exponents of time series data: a machine learning approach, Entropy 25 (2023), https://www.mdpi.com/1099-4300/25/12/1671.
[17] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, A. Gulin, CatBoost: unbiased boosting with categorical features, Adv. Neural Inf. Process. Syst. 31 (2018), https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.
[18] K. Mallinger, S. Raubitzek, T. Neubauer, S. Lade, Potentials and limitations of complexity research for environmental sciences and modern farming applications, Curr. Opin. Environ. Sustain. 67 (2024) 101429, https://www.sciencedirect.com/science/article/pii/S1877343524000162.
[19] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[20] D. Wolpert, W. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.

[21] V. Havlíček, A. Córcoles, K. Temme, A. Harrow, A. Kandala, J. Chow, J. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature 567 (2019) 209–212.

[22] Y. Hu, G. Morota, G. Rosa, D. Gianola, Prediction of plant height in Arabidopsis thaliana using DNA methylation data, Genetics 201 (2015) 779–793, https://doi.org/10.1534/genetics.115.177204.

[23] S. Amit Saxena, Pancreatic cancer data classification with quantum machine learning, J. Quant. Comput. 5 (2023) 1–13, http://www.techscience.com/jqc/v5n1/54536.

[24] P. Evangelista, M. Embrechts, B. Szymanski, Some properties of the Gaussian kernel for one class learning, Artif. Neural Netw. – ICANN 2007 (2007) 269–278.

[25] Y. Chi, Y. Yu, Q. Gong, J. Wang, High-dimensional quantum information processing on programmable integrated photonic chips, Sci. China Inf. Sci. 66 (2023) 180501, https://doi.org/10.1007/s11432-022-3602-0.