Institute of Architecture of Application Systems

Patterns for Quantum Machine Learning

Lavinia Stiliadou, Johanna Barzen, Martin Beisel,
Frank Leymann, and Benjamin Weder

Institute of Architecture of Application Systems,
University of Stuttgart, Germany,
[firstname.lastname]@iaas.uni-stuttgart.de

BIBTEX:

```
@inproceedings{Stiliadou2025_QMLPatterns,
  author    = {Stiliadou, Lavinia and Barzen, Johanna and Beisel, Martin and
               Leymann, Frank and Weder, Benjamin},
  title     = {{Patterns for Quantum Machine Learning}},
  booktitle = {Proceedings of the 17\textsuperscript{th} International
               Conference on Pervasive Patterns and Applications (PATTERNS
               2025)},
  year      = {2025},
  month     = apr,
  pages     = {7-14},
  publisher = {Xpert Publishing Services (XPS)},
  Isbn      = {78-1-68558-263-0}
}
```


Universität Stuttgart
Germany

Patterns for Quantum Machine Learning

Lavinia Stiliadou, Johanna Barzen, Martin Beisel, Frank Leymann, and Benjamin Weder
Institute of Architecture of Application Systems, University of Stuttgart
Universitätsstrasse 38, 70569 Stuttgart, Germany
e-mail: {lastname}@iaas.uni-stuttgart.de

***Abstract*—The tremendous success of applying machine learning techniques in science and industry leads to new challenges: Developing new, faster, or more precise algorithms can not compete with the continuously growing data volumes to process. Thus, the computational resources must be increased, leading to high costs and energy consumption. To overcome these issues, quantum machine learning promises to utilize quantum mechanical phenomena to train machine learning models more efficiently. However, realizing such quantum machine learning techniques is time-consuming and complex. This is especially the case as new techniques are typically published as scientific papers without suitable documentation for software developers. Patterns are a well-established concept to document proven solutions to recurring problems. Although a pattern language for quantum computing has been introduced, it currently misses patterns documenting how quantum machine learning can be successfully applied. To bridge this gap, this paper presents three novel patterns, focusing on quantum machine learning techniques.**

Keywords-Quantum Computing; Pattern Language; QML.

I. Introduction

In recent years, machine learning has revolutionized the industry by providing new means for solving problems in various domains, e.g., natural language processing or medicine [1]. However, a significant part of this progress was achieved by increasing the computational resources [2]. Thus, costs and energy consumption for reaching new milestones in the machine learning domain increased dramatically. For example, the training of modern neural networks, such as GPT4 [3], takes months and costs millions of dollars [4]. Quantum machine learning provides new concepts and algorithms taking advantage of quantum mechanical phenomena, such as superposition and entanglement, that promise to more efficiently train machine learning models [5][6]. Hence, quantum machine learning vastly differs from classical machine learning, e.g., in the structuring of the training data [7][8]. However, these concepts of quantum machine learning are complex, particularly for software developers without deep knowledge of physics and quantum computing. Additionally, there currently exists no structured documentation on how software developers can employ quantum machine learning to benefit from its advantages.

A well-established concept for documenting proven solutions to recurring problems in the software engineering domain is patterns [9][10]. Patterns facilitate understanding the origin of a problem and the forces that complicate solving it. Furthermore, they describe an abstract solution to the problem, enabling software developers to implement and customize the presented solution strategy for their given application domain.

To support developers in building quantum applications, Leymann [11] has introduced the quantum computing pattern language, which was continuously expanded to cover different areas of quantum computing. It documents fundamental concepts of quantum computing and discusses various techniques, e.g., error handling [12] or warm-starting [13]. While there are already patterns for solving optimization problems [14], there are no patterns documenting how quantum machine learning techniques can be successfully applied. In this paper, we extend the quantum computing pattern language by documenting three novel patterns for quantum machine learning, describing well-known types of quantum algorithms.

The remainder of this paper is structured as follows: In Section II, we introduce fundamentals about quantum algorithms and hybrid quantum applications. Furthermore, we present the utilized pattern format, as well as the authoring method. Section III extends the quantum computing pattern language by documenting the newly introduced quantum machine learning patterns. In Section IV, the presented quantum machine learning patterns are discussed and evaluated. Finally, Section V presents related work, and Section VI concludes the paper.

II. Fundamentals

In this section, we discuss the fundamentals of quantum algorithms and their use in quantum applications. Furthermore, we present our pattern format and the pattern authoring method.

A. Quantum Algorithms & Applications

Quantum computing achieves speed-ups compared to classical computations by leveraging quantum mechanical phenomena [15]: Unlike classical computers that operate on 0 s and 1 s, quantum devices utilize qubits that can exist in a superposition of states $|0\rangle := (1, 0)^T$ and $|1\rangle := (0, 1)^T$. Further, qubits can be entangled with each other, making their states inseparable. Various technologies to realize quantum devices exist, e.g., superconducting or photonic quantum devices, that utilize different computation models for quantum computing. In this paper, we focus on the gate-based computation model, which is used by quantum hardware providers such as IBM and Rigetti. To perform computations on gate-based quantum devices, the states of the qubits are manipulated utilizing different quantum gates, which are specified by a quantum program, a so-called quantum circuit [16]. There are single-qubit gates, which only affect the state of a single qubit, and multi-qubit gates, which affect multiple qubits and can entangle multiple qubits with each other. To retrieve a classical solution when executing a quantum circuit, the state of the quantum system must be measured. Since the state of the system collapses when measuring it and quantum computing is probabilistic by nature, quantum executions are typically executed multiple times to retrieve the expectation value of the computation [17].

Quantum algorithms are often hybrid, comprising classical pre- and post-processing steps, e.g., encoding data into a quantum circuit or mitigating occurring errors [16][18]. A special class of quantum algorithms is so-called *Variational Quantum Algorithms (VQAs)* [19]. VQAs typically utilize shallow circuits, i.e., they have a low number of consecutive gates and use a small number of qubits. Therefore, they enable the execution of practically relevant quantum computations on today's error-prone and intermediate-size quantum devices. The execution of VQAs alternates between executing parameterized quantum circuits and classically optimizing their parameters, e.g., using gradient-based or gradient-free optimization approaches. In each iteration of the VQA, the execution results are used to estimate the value of the cost function that encodes the problem. Its value is utilized by an optimizer to find more suitable circuit parameters, e.g., by minimizing the cost function. The quantum cost function is generally defined as: $C(\vec{\theta}) = f(\{\rho_i\}_{i=1}^{k}, \{O_i\}_{i=1}^{k}, U(\vec{\theta}))$, where $\vec{\theta}$ is a vector of parameters, f is some function, k is the size of the training set T. The set $\{\rho_i\}$ represents the input states from the training set, while $\{O_i\}$ denotes the set of observables. The parametrized quantum circuit $U(\vec{\theta})$, defined by the parameters $\vec{\theta}$, is iteratively optimized classically.

B. Patterns & Authoring Method

Patterns are an established concept for documenting proven solutions to commonly recurring problems in a well-structured manner [9]. Typically, a uniform pattern format is used for all patterns of a domain to facilitate understanding the patterns within a pattern language [20]. Therefore, we use the pattern format utilized by the previously published quantum computing patterns, which is structured as follows: Each pattern is identified by a unique *name* within the pattern language. Furthermore, patterns are associated with a mnemonic *icon* to enable a visual recognition of the pattern. The problem solved by the pattern is concisely summarized in a *problem* statement. Next, the *context* in which the problem appears is described and the *forces* complicating the solution of the problem are discussed. In the *solution* section, a proven strategy for solving the previously discussed problem is presented alongside a corresponding *solution sketch*. Subsequently, *examples* of the solution are explained. The *result* section describes the consequences of applying the solution and discusses what further steps might be necessary to handle them. Each pattern is semantically linked to *related patterns* in the eponymous section, e.g., to patterns that are commonly used in combination or that are alternatives to each other. Finally, the *known uses* section showcases real-world occurrences of the pattern. To identify and document the patterns for quantum machine learning, we applied the pattern authoring method introduced by Fehling et al. [20]. First, we analyzed the literature, as well as the documentation of current quantum software development kits for best practices and established solution strategies for quantum machine learning. Subsequently, the collected information was filtered based on its relevancy for practically applying quantum machine learning. The identified solutions were documented and iteratively refined to extract patterns comprising quantum machine learning algorithms. For the documentation of patterns, their previously described format is utilized.

III. Quantum Machine Learning Patterns

In this section, we first provide a brief overview of the existing quantum computing pattern language and subsequently introduce three novel patterns for quantum machine learning.

A. Quantum Computing Pattern Language Overview

Figure 1 gives an overview of the quantum computing pattern language. It comprises both the already existing, as well as the newly added patterns from this work. The different patterns are assigned to one category, depending on the phase of the hybrid quantum application lifecycle they belong to [18]:

First, the *unitary transformations* patterns [21] describe best practices for transformations after an initial state has been created. The *warm-starting* patterns [13] show various techniques to improve the performance of quantum algorithms. Next, the *program flow* patterns [14] summarize concepts to split computations between quantum and classical hardware. The *circuit cutting* patterns [22] document techniques to cut large quantum circuits into smaller circuits that can be successfully executed on today's quantum devices. To encode classical data into quantum circuits, the *data encodings* patterns [21] describe so-called state preparation routines. The *error handling* patterns [12] summarize approaches to reduce noise on today's quantum devices. Fundamental *quantum states* [11], how they are created, and for which quantum algorithms they are used as a basis are discussed in the eponymous category. For executing quantum circuits and hybrid quantum applications, different quantum cloud offerings are available. These offerings provide heterogeneous features, e.g., the execution via a queue or the exclusive reservation of a quantum device, and the *execution* patterns [23] document these execution styles, as well as their benefits and disadvantages. The *development* patterns [24] provide solutions and best practices for typical problems when developing hybrid quantum applications. Complementary, the *operations* patterns [25] cover abstract solutions for operating and managing hybrid quantum applications. Finally, the *measurement* patterns [21] present concepts and techniques for extracting classical data from quantum states.

In this work, we introduce three novel patterns describing well-known approaches for tackling crucial problems from the quantum machine learning domain: To overcome the difficulties when clustering large and complex data sets, Quantum Clustering algorithms have been introduced. They partition data sets into different clusters based on their similarity, which is calculated using quantum devices. The Quantum Classification pattern provides a means for

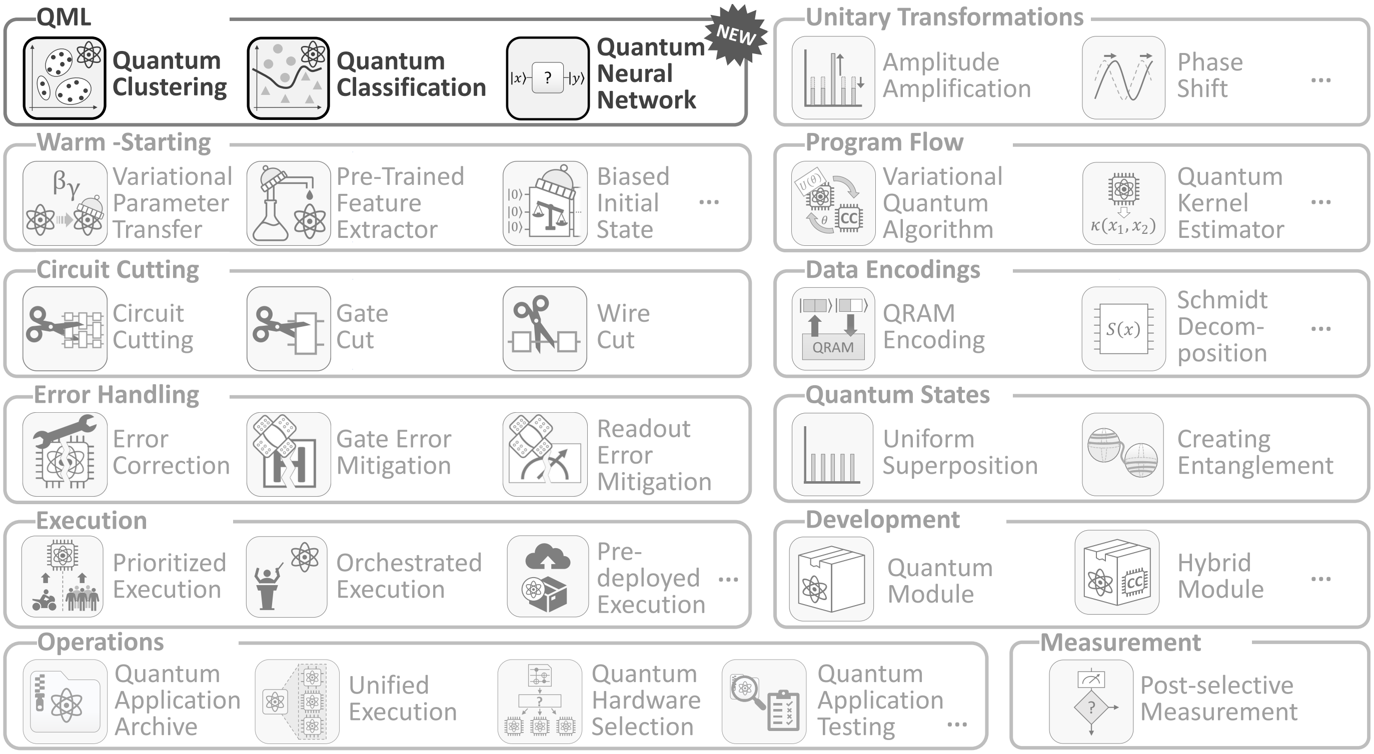
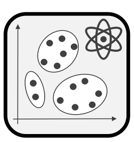


Figure 1. Overview of the quantum computing pattern language with some existing (light gray) and the new patterns proposed in this work (dark gray).

training a classifier based on a set of labeled data. This classifier enables assigning new data points to one of multiple classes. Finally, the QUANTUM NEURAL NETWORK pattern documents how quantum devices can be used to improve the accuracy and performance of neural networks by leveraging quantum mechanical phenomena.

B. Quantum Clustering



Problem: How to partition a data set into different clusters based on their similarity utilizing a quantum device?

Context: A set of unlabeled data needs to be grouped into different clusters. The clusters should organize the data points according to identified similarities.

Forces: Data sets may exhibit non-linear separability, which increases the complexity when clustering. Moreover, data sets utilized in machine learning continuously grow in size, leading to increased training times [3]. Therefore, algorithms whose runtime scales well with the number of data points in the data set and the dimensions of the feature space are required. In addition to the general machine learning forces, also quantum-specific forces have to be taken into account. For example, loading large data sets consisting of many tuples into current quantum devices is difficult due to the high circuit depth of the required state preparation routines [26][27].

Solution: Figure 2 gives an overview of the general clustering process. Use a quantum device to cluster the m data points $\{x_i\}_{i=1}^m$ of a data set. First, the classical data points are encoded into quantum states $\{|x_i\rangle\}_{i=1}^m$ by applying a unitary transformation U_ϕ, enabling the quantum computer to process the data. Once the data points are encoded, a given ansatz $V(\vec{\theta})$ is used to calculate the similarity between data points. The ansatz is a parameterized quantum circuit designed to approximate the quantum state that captures the relevant features of the data for similarity measurement. The ansatz computes the similarity either between pairwise data points or between all data points, depending on its structure [28]. The cost function used in this approach is designed to assign similar points to the same cluster and points with low similarity to different clusters. In the cost function, this is represented by a penalty term that penalizes distant points that are assigned to the same cluster. Additionally, the clustering process is controlled by adding constraints. To ensure that each data point is assigned to exactly one cluster, the following condition must hold $\sum_{a=1}^k q_i^a = 1$. Thereby, classical variables q_i^a are introduced to denote whether a data point x_i is assigned to cluster a. The quantum circuit parameters are updated iteratively to minimize the cost function.

Examples: An exemplary quantum clustering algorithm is the quantum k-means algorithm [29][30]: First, k initial data points are randomly selected as centroids for the clustering. Then, the states for both the centroids, as well as the remaining data points are prepared. The number k, which corresponds to the number of clusters, can either be specified by the user or automatically determined [31]. Subsequently, for each data point, the distance to all centroids is calculated utilizing a distance metric, e.g.,

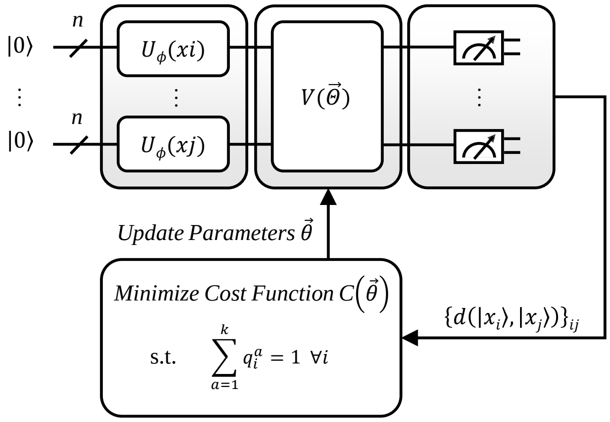


Figure 2. Solution sketch for the QUANTUM CLUSTERING pattern.

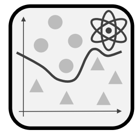
the Manhattan distance [30] or Euclidean distance [32]. For example, the SWAP test [33] can be used to determine the Euclidean distances efficiently on a quantum device. Each data point is assigned to the cluster corresponding to the centroid with the smallest distance to the data point. Afterward, the new centroids are calculated classically by computing the mean of all data points assigned to that cluster. If the retrieved centroids differ substantially, i.e., more than a certain threshold specified by the user, from the previous iteration, the previously described procedure is performed again utilizing the new centroids.

Result: Utilizing a quantum clustering algorithm may enable identifying clusters in a data set exponentially faster than with a classical clustering algorithm [32]. Often, the computational advantage of quantum clustering algorithms relies on the availability of the input data in a suitable format. Once an implementation of *Quantum Random Access Memory (QRAM)* is available, data can be encoded efficiently, enabling the full potential of quantum clustering.

Related Patterns: The QRAM ENCODING pattern [21] can be used to efficiently encode the data points for a quantum device. To facilitate the clustering of complex data sets, the data points can be mapped into a higher dimensional feature space using the QUANTUM KERNEL ESTIMATOR pattern. The QUANTUM CLUSTERING pattern uses the QUANTUM-CLASSICAL SPLIT pattern [11] to efficiently distribute the computations using quantum and classical hardware and can be realized as a HYBRID MODULE [24].

Known Uses: Ramirez [34] presents different quantum clustering techniques, such as quantum spectral clustering and quantum hierarchical clustering. Kavitha et al. [35] utilize quantum k-means clustering for detecting heart diseases. Patil et al. [36] introduce two measurement-based quantum clustering algorithms. The first algorithm follows a hierarchical clustering approach. The second algorithm uses unsharp measurements for the clustering process. Gopalakrishnan et al. [37] propose a quantum clustering algorithm that achieves linear scalability with respect to both the number of data points and their density.

C. Quantum Classification



Problem: How to train a classifier to assign new data points to one of multiple classes using a quantum device?

Context: New data points need to be classified into one of several different classes. A labeled set of training data is given.

Forces: Classifying data is getting increasingly more difficult when the feature space becomes larger [38]. While quantum computing enables solving this problem by utilizing efficient quantum algorithms, it also leads to additional challenges. For example, high-dimensional data sets can lead to large quantum circuits that may not be successfully executable on today's *Noisy Intermediate-Scale Quantum (NISQ)* devices [27]. Additionally, quantum approaches can suffer from exponential cost concentration, which makes models less sensitive to input data, leading to generalization problems [39][40].

Solution: Train a classifier using a quantum device to classify new data points precisely. In Figure 3, an overview of two different approaches for training a classifier is depicted. Classifiers can either be trained using (i) a kernel-based method or (ii) a variational method. Generally, the input for training a classifier is an initial set of labeled data $\{(x_i, y_i)\}_{i=1}^n$, where x_i are the feature vectors, y_i are the labels, i.e., real numbers, and n is the size of the training set. In the kernel-based approach, a quantum kernel is used to measure the similarities between data points by mapping them into a high-dimensional Hilbert space and computing the inner product of their corresponding quantum states. This quantum kernel is computed for all pairs of training data by applying a unitary $U_\phi(x_i)$ to encode each data point x_i into a quantum state. The adjoint operation $U_\phi^\dagger(x_j)$ is then applied to calculate the overlap between the states corresponding to x_i and x_j. Then, a classical algorithm, e.g., a classical support vector machine [41], is used for computing the classifier based on the previously calculated kernel. Alternatively, the variational method optimizes the parameters of a quantum circuit to directly realize the classifier. In this approach, a data point x is first encoded into a quantum state using a unitary $U_\phi(x)$, which maps the classical data into

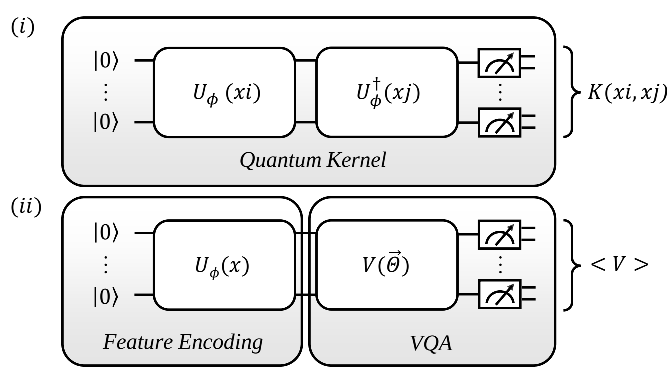
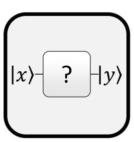


Figure 3. Solution sketch for the QUANTUM CLASSIFICATION pattern.

a quantum state. Once the data are encoded, a parameterized quantum circuit is applied. The circuit produces an output whose expectation value $< V >$ determines the predicted label for a given data point x. The parameters of the circuit are iteratively optimized by a classical optimizer that minimizes a quantum cost function that calculates the differences between the predicted and actual labels from the data set.
Examples: An example is the kernel-based quantum support vector machine [42], achieving logarithmic complexity with respect to both the data dimension and the number of training examples. Another example is the variational quantum support vector machine [38][43], which uses a parameterized quantum circuit to directly implement a SVM on a QPU.
Result: After the training process, the classifier can be used to assign new data points to one of the existing classes. Utilizing a quantum classifier may enable training a more accurate classifier than using a classical classification technique [44]. Quantum classifiers still function under the influence of noise as they are resistant to a small number of misclassifications [38]. This is particularly important with the noisiness of today's quantum devices. However, mitigation mechanisms must be implemented to address the challenges, such as cost concentration in kernel values or flatness in the optimization landscape [19][39].
Related Patterns: The QRAM ENCODING pattern [21] can be utilized to achieve a speed-up when encoding data. A quantum classifier can be realized using a VARIATIONAL QUANTUM ALGORITHM [14].
Known Uses: To train the quantum classifier, different approaches can be used, e.g., variational quantum support vector machines [34][38], quantum decision trees [45], and quantum nearest neighbor classification [46]. Furthermore, quantum classifiers have been applied in different application areas, e.g., image recognition [47], analyzing the sentiments of sentences [48], and predicting air pollution [49].

D. Quantum Neural Network (QNN)



Problem: How to learn an unknown unitary operator using a quantum device?

Context: An unknown unitary operator needs to be learned from a training set containing the quantum inputs and the expected quantum outputs.
Forces: Identifying a unitary operator that is capable of mapping input data to their respective output is getting increasingly more difficult with the complexity and variety of the data. Determining such a mapping requires a lot of input data, and the training procedure requires significant computational power [3]. However, this number can be reduced as outlined by the Quantum No-Free-Lunch Theorem since only obtaining a subset of the training samples as entangled quantum states is already beneficial [50].
Solution: Figure 4 shows the training process of a QNN to learn an unknown unitary operator U: To realize a corresponding quantum circuit, first, the input data are encoded to a quantum state $|x\rangle$. Similarly to classical neural networks, quantum circuits realizing a QNN comprise various parameterized hidden layers $V_i(\vec{\theta})$ to approximate U. The parameters $\vec{\theta}$ are iteratively adjusted by an optimizer, which minimizes a cost function until the quantum circuit produces approximately the expected outputs. The cost function uses the expected outputs and similarity measures, such as fidelity, to evaluate how closely the produced outputs $|\tilde{y}\rangle$ match the expected ones $|y\rangle$.

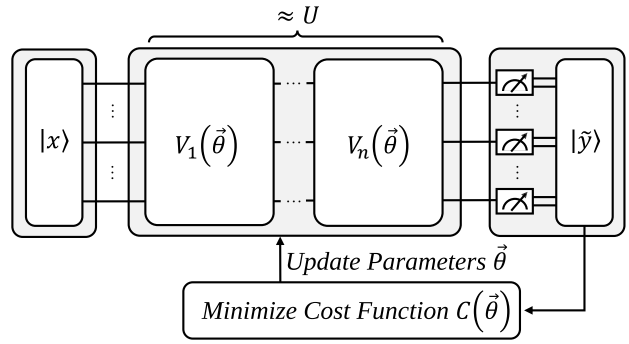


Figure 4. Solution sketch for the QUANTUM NEURAL NETWORK pattern.

Examples: A special kind of QNNs are quantum convolutional neural networks, which are utilized for processing structured grid data, e.g., for image processing [51][52]. It comprises four different types of layers: (i) First, state preparation layers are used to encode the classical input data. (ii) The convolutional layers enable the detection of spatial patterns within the input data. (iii) Pooling layers reduce some of the spatial dimensions to focus on the optimization of the most important features. (iv) Finally, fully connected layers are used to produce the final output of the quantum convolutional network.
Result: The result is a set of parameters that configures the QNN to approximate the expected output of the unknown unitary operator. The quality of the approximation depends on the size of the training set, its linear structure, and the degree of entanglement [53]. While entangled data provides benefits for the training of QNNs, a too high level of entanglement can lead to barren plateaus [54].
Related Patterns: Quantum neural networks are a realization of the VARIATIONAL QUANTUM ALGORITHM pattern [14]. Different state preparation routines, such as ANGLE ENCODING or BASIS ENCODING [21], can be utilized to encode the input data of the QNN. To integrate a QNN in existing applications, it can be provided as a HYBRID MODULE [24].
Known Uses: Jeswal et al. [55] and Vasuki et al. [56] provide surveys overviewing the various application areas of QNNs, ranging from prediction to pattern recognition problems. Kashif et al. [57] present an approach for efficiently training QNNs in the presence of noise when using NISQ devices.

IV. DISCUSSION

In this section, we discuss the challenges and limitations of applying the presented quantum machine learning patterns and elaborate on the validity of the documented patterns.

Quantum machine learning is a rapidly evolving field that promises to overcome the limitations of state-of-the-art classical machine learning methods [58]. However, the high error rates and low number of qubits of today's quantum devices prevent the application of the introduced concepts for many real-world problems. The roadmaps of quantum device providers, such as IBM [59] and QuEra [60], promise that in the near future, the potential of quantum machine learning can be demonstrated for practically relevant use cases.

An essential part of achieving speed-ups with quantum devices is efficient access to classical and quantum data [54]. Thereby, the classical data is prepared for the quantum device utilizing state preparation routines [21]. While there are efficient implementations for many state preparation routines, QRAM has not yet been successfully implemented [61]. This limits the effectiveness of many quantum machine learning algorithms, such as the quantum support vector machine, as the assumption of the algorithms is that data is available via QRAM [38]. Therefore, alternative, less efficient state preparation routines must be utilized until an efficient QRAM implementation is available.

To confirm the validity of patterns in the software engineering domain, a number of different real-world implementations of the patterns are identified [62]. Hence, several occurrences of each quantum machine learning pattern have been documented in the known uses section of the respective pattern. To ease the configuration and abstract technical details, patterns can be used to automate the generation of quantum applications [63].

V. Related Work

The patterns for quantum machine learning introduced in this work extend the existing quantum computing pattern language presented in Section III-A. Perez-Castillo et al. [64] analyze code repositories for the occurrences of different patterns of the quantum computing pattern language and identify a lack of abstraction mechanisms. The patterns presented in this paper aim to bridge this gap in the quantum machine learning domain. Aside from the quantum computing patterns, there are other works presenting patterns in the quantum computing domain that do not follow the pattern format introduced by Alexander et al. [9]: Baczyk et al. [65] document different patterns that aim to facilitate architectural design decisions when building quantum applications. Khan et al. [66] identify various architecture design patterns for quantum applications via a systematic literature survey. Huang and Martonosi [67] utilize quantum programming patterns to find bugs in quantum circuits. Gilliam et al. [68] and Perdrix [69] present patterns for building quantum circuits. However, none of these papers focus on patterns in the quantum machine learning domain.

Guo et al [70] present a set of patterns for defining ansätze in variational quantum algorithms. These ansätze can also be used in quantum machine learning, e.g., to implement the hidden layers of quantum neural networks.

Lakshmanan et al. [71] document various machine learning design patterns. They focus on different aspects that should be regarded when utilizing machine learning in practice, e.g., reproducibility or responsible artificial intelligence. Although the machine learning design patterns do not consider quantum machine learning, various patterns, such as Model Versioning can also be applied to quantum machine learning.

Falkenthal et al. [72] introduce the concept of solution languages to facilitate the application of patterns for real-world use cases. Solution languages comprise so-called concrete solutions, i.e., implementations of a pattern for a specific use case, e.g., a quantum circuit or a Python program. These concrete solutions are associated with the corresponding pattern, enabling developers to reuse existing implementations for their applications. Thereby, the manual effort of implementing the abstract solution described by the pattern can be reduced.

VI. Conclusion & Future Work

Machine learning has revolutionized research and industry by providing new means for solving various problems. However, a significant part of this progress was achieved by increasing the computational resources, leading to high costs and energy consumption. A promising technology providing additional computational power is quantum computing. In this paper, we capture existing concepts from the literature for utilizing quantum devices to tackle crucial machine learning problems efficiently. We document these concepts in an easily digestible manner as patterns that enable understanding the problem and solving it using proven solution strategies. The introduced patterns are publicly available via Pattern Atlas [73], an open-source tool for authoring, managing, and visualizing patterns [74].

Since quantum machine learning is a rapidly evolving and highly active area, we will continue investigating the progress achieved by researchers and companies. Hence, in future work, we plan to identify new solution strategies in the quantum machine learning domain and document them as novel patterns.

ACKNOWLEDGEMENTS

This work was partially funded by the BMWK projects *EniQmA* (01MQ22007B) and *SeQuenC* (01MQ22009B).

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects", *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
[2] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The Computational Limits of Deep Learning", *arXiv:2007.05558*, 2020.
[3] OpenAI, "GPT-4 Technical Report", *arXiv:2303.08774*, 2023.
[4] B. Cottier, R. Rahman, L. Fattorini, N. Maslej, and D. Owen, "The rising costs of training frontier AI models", *arXiv:2405.21015*, 2024.
[5] J. Biamonte *et al.*, "Quantum Machine Learning", *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
[6] J. Barzen, "From Digital Humanities to Quantum Humanities: Potentials and Applications", in *Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications*, Springer, 2022, pp. 1–52.
[7] T. Gabor *et al.*, "The Holy Grail of Quantum Artificial Intelligence: Major Challenges in Accelerating the Machine Learning Pipeline", in *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 2020, pp. 456–461.

[8] S. Thanasilp, S. Wang, N. A. Nghiem, P. Coles, and M. Cerezo, "Subtleties in the trainability of quantum machine learning models", *Quantum Machine Intelligence*, vol. 5, no. 1, p. 21, 2023.
[9] C. Alexander, S. Ishikawa, and M. Silverstein, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.
[10] G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional, 2004.
[11] F. Leymann, "Towards a Pattern Language for Quantum Algorithms", in *Proceedings of the 1st International Workshop on Quantum Technology and Optimization Problems (QTOP)*, Springer, 2019.
[12] M. Beisel *et al.*, "Patterns for Quantum Error Handling", in *Proceedings of the 14th International Conference on Pervasive Patterns and Applications (PATTERNS)*, Xpert Publishing Services (XPS), 2022, pp. 22–30.
[13] F. Truger, J. Barzen, M. Beisel, F. Leymann, and V. Yussupov, "Warm-Starting Patterns for Quantum Algorithms", in *Proceedings of the 16th International Conference on Pervasive Patterns and Applications (PATTERNS)*, Xpert Publishing Services (XPS), 2024, pp. 25–31.
[14] M. Weigold, J. Barzen, F. Leymann, and D. Vietz, "Patterns for Hybrid Quantum Algorithms", in *Proceedings of the 15th Symposium and Summer School on Service-Oriented Computing (SummerSOC)*, Springer, 2021, pp. 34–51.
[15] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
[16] F. Leymann and J. Barzen, "The bitter truth about gate-based quantum algorithms in the NISQ era", *Quantum Science and Technology*, vol. 5, no. 4, pp. 1–28, 2020.
[17] F. Leymann *et al.*, "Quantum in the Cloud: Application Potentials and Research Opportunities", in *Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER)*, SciTePress, 2020, pp. 9–24.
[18] B. Weder, J. Barzen, F. Leymann, and D. Vietz, "Quantum Software Development Lifecycle", in *Quantum Software Engineering*, Springer, 2022, pp. 61–83.
[19] M. Cerezo *et al.*, "Variational quantum algorithms", *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.
[20] C. Fehling, J. Barzen, U. Breitenbücher, and F. Leymann, "A Process for Pattern Identification, Authoring, and Application", in *Proceedings of the 19th European Conference on Pattern Languages of Programs (EuroPLoP)*, ACM, 2014.
[21] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Encoding patterns for quantum algorithms", *IET Quantum Communication*, vol. 2, no. 4, pp. 141–152, 2021.
[22] M. Bechtold, J. Barzen, M. Beisel, F. Leymann, and B. Weder, "Patterns for Quantum Circuit Cutting", in *Proceedings of the 30th Conference on Pattern Languages of Programs (PLoP)*, HILLSIDE, 2023.
[23] D. Georg *et al.*, "Execution Patterns for Quantum Applications", in *Proceedings of the 18th International Conference on Software Technologies (ICSOFT)*, SciTePress, 2023, pp. 258–268.
[24] F. Bühler *et al.*, "Patterns for Quantum Software Development", in *Proceedings of the 15th International Conference on Pervasive Patterns and Applications (PATTERNS)*, Xpert Publishing Services (XPS), 2023, pp. 30–39.
[25] M. Beisel, J. Barzen, F. Leymann, and B. Weder, "Operations Patterns for Hybrid Quantum Applications", in *Proceedings of the 15th International Conference on Cloud Computing and Services Science (CLOSER)*, SciTePress, 2025.
[26] V. Akshay, H. Philathong, M. Morales, and J. Biamonte, "Reachability deficits in quantum approximate optimization", *Physical review letters*, vol. 124, no. 9, p. 090 504, 2020.
[27] J. Preskill, "Quantum Computing in the NISQ era and beyond", *Quantum*, vol. 2, p. 79, 2018.
[28] A. Poggiali, A. Berti, A. Bernasconi, G. M. Del Corso, and R. Guidotti, "Quantum clustering with k-means: A hybrid approach", *Theoretical Computer Science*, vol. 992, p. 114 466, 2024.
[29] S. U. Khan, A. J. Awan, and G. Vall-Llosera, "K-Means Clustering on Noisy Intermediate Scale Quantum Computers", *arXiv:1909.12183*, 2019.
[30] Z. Wu, T. Song, and Y. Zhang, "Quantum k-means algorithm based on Manhattan distance", *Quantum Information Processing*, vol. 21, no. 1, p. 19, 2021.
[31] I. Khan, Z. Luo, J. Z. Huang, and W. Shahzad, "Variable Weighting in Fuzzy k-Means Clustering to Determine the Number of Clusters", *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 9, pp. 1838–1853, 2019.
[32] S. U. Khan, A. J. Awan, and G. Vall-Llosera, "K-Means Clustering on Noisy Intermediate Scale Quantum Computers", *arXiv:1909.12183*, 2019.
[33] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, "Quantum Fingerprinting", *Physical review letters*, vol. 87, no. 16, p. 167 902, 2001.
[34] J. G. C. Ramírez, "Advanced Quantum Algorithms for Big Data Clustering and High-Dimensional Classification", *Journal of Advanced Computing Systems*, vol. 4, no. 6, 2024.
[35] S. S. Kavitha and N. Kaulgud, "Quantum K-means clustering method for detecting heart disease using quantum circuit approach", *Soft Computing*, vol. 27, no. 18, pp. 13 255–13 268, 2023.
[36] S. Patil, S. Banerjee, and P. K. Panigrahi, "Measurement-Based Quantum Clustering Algorithms", *arXiv:2302.00566*, 2023.
[37] D. Gopalakrishnan *et al.*, "qCLUE: a quantum clustering algorithm for multi-dimensional datasets", *Frontiers in Quantum Science and Technology*, 2024.
[38] V. Havlíček *et al.*, "Supervised learning with quantum-enhanced feature spaces", *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
[39] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, "Exponential concentration in quantum kernel methods", *Nature Communications*, vol. 15, no. 1, p. 5200, 2024.
[40] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles, "Equivalence of quantum barren plateaus to cost concentration and narrow gorges", *Quantum Science and Technology*, vol. 7, no. 4, p. 045 015, 2022.
[41] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
[42] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum Support Vector Machine for Big Data Classification", *Physical review letters*, vol. 113, no. 13, p. 130 503, 2014.
[43] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, "The complexity of quantum support vector machines", *Quantum*, vol. 8, p. 1225, 2024.
[44] S. S. Kavitha and N. Kaulgud, "Quantum machine learning for support vector machine classification", *Evolutionary Intelligence*, vol. 17, no. 2, pp. 819–828, 2024.
[45] S. Lu and S. L. Braunstein, "Quantum decision tree classifier", *Quantum information processing*, vol. 13, no. 3, pp. 757–770, 2014.
[46] W. Roga, B. Chevalier, and M. Takeoka, "Fully Quantum Classifier", *arXiv:2307.03396*, 2023.
[47] J. Li, Y. Li, J. Song, J. Zhang, and S. Zhang, "Quantum Support Vector Machine for Classifying Noisy Data", *IEEE Transactions on Computers*, vol. 73, no. 9, pp. 2233–2247, 2024.
[48] F. Z. Ruskanda, M. R. Abiwardani, R. Mulyawan, I. Syafalni, and H. T. Larasati, "Quantum-Enhanced Support Vector Ma-

chine for Sentiment Classification", *IEEE Access*, vol. 11, pp. 87 520–87 532, 2023.
[49] O. Farooq *et al.*, "An enhanced approach for predicting air pollution using quantum support vector machine", *Scientific Reports*, vol. 14, no. 1, p. 19 521, 2024.
[50] A. Mandl, J. Barzen, F. Leymann, and D. Vietz, "On Reducing the Amount of Samples Required for Training of QNNs: Constraints on the Linear Structure of the Training Data", *arXiv:2309.13711*, 2023.
[51] T. Hur, L. Kim, and D. K. Park, "Quantum convolutional neural network for classical data classification", *Quantum Machine Intelligence*, vol. 4, no. 1, p. 3, 2022.
[52] S. Wei, Y. Chen, Z. Zhou, and G. Long, "A quantum convolutional neural network on NISQ devices", *AAPPS Bulletin*, vol. 32, pp. 1–11, 2022.
[53] A. Mandl *et al.*, "Linear Structure of Training Samples in Quantum Neural Network Applications", in *International Conference on Service-Oriented Computing*, Springer, 2023, pp. 150–161.
[54] S. Thanasilp, S. Wang, N. A. Nghiem, P. Coles, and M. Cerezo, "Subtleties in the trainability of quantum machine learning models", *Quantum Machine Intelligence*, vol. 5, no. 1, 2023.
[55] S. K. Jeswal and S. Chakraverty, "Recent Developments and Applications in Quantum Neural Network: A Review", *Archives of Computational Methods in Engineering*, vol. 26, no. 4, pp. 793–807, 2019.
[56] M. Vasuki, A. Karunamurthy, R. Ramakrishnan, and G. Prathiba, "Overview of Quantum Computing in Quantum Neural Network and Artificial Intelligence", *Quing International Journal of Innovative Research in Science and Engineering*, vol. 2, pp. 117–127, 2023.
[57] M. Kashif and M. Shafique, "HQNET: Harnessing Quantum Noise for Effective Training of Quantum Neural Networks in NISQ Era", *arXiv:2402.08475*, 2024.
[58] K. A. Tychola, T. Kalampokas, and G. A. Papakostas, "Quantum Machine Learning—An Overview", *Electronics*, vol. 12, no. 11, p. 2379, 2023.
[59] IBM, *Quantum Roadmap*, https://www.ibm.com/roadmaps/quantum, 2024, [accessed: 2025.03.01].
[60] QuEra, *Our Quantum Computing Roadmap*, https://www.quera.com/qec, 2024, [accessed: 2025.03.01].
[61] S. Jaques and A. G. Rattew, "QRAM: A Survey and Critique", *arXiv:2305.10310*, 2023.
[62] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, 1995.
[63] D. Vietz *et al.*, "Harnessing Patterns to Support the Development of Hybrid Quantum Applications", in *Proceedings of the 14th International Conference on Software and Computer Applications (ICSCA)*, ACM, 2025, pp. 1–11.
[64] R. Pérez-Castillo, M. Fernández-Osuna, J. A. Cruz-Lemus, and M. Piattini, "A preliminary study of the usage of design patterns in quantum software", in *Proceedings of the 5th ACM/IEEE International Workshop on Quantum Software Engineering (QSE)*, 2024, pp. 41–48.
[65] M. Baczyk, R. Pérez-Castillo, and M. Piattini, "Towards a Framework of Architectural Patterns for Quantum Software Engineering", in *Proceedings of the 4th International Workshop on Quantum Software Engineering and Technology (Q-SET)*, vol. 2, 2024, pp. 228–233.
[66] A. A. Khan *et al.*, "Software architecture for quantum computing systems — A systematic review", *Journal of Systems and Software*, vol. 201, p. 111 682, 2023.
[67] Y. Huang and M. Martonosi, "Statistical Assertions for Validating Patterns and Finding Bugs in Quantum Programs", in *Proceedings of the 46th International Symposium on Computer Architecture (ISCA)*, ACM, 2019, pp. 541–553.
[68] A. Gilliam *et al.*, "Foundational Patterns for Efficient Quantum Computing", *arXiv:1907.11513*, 2019.
[69] S. Perdrix, "Quantum Patterns and Types for Entanglement and Separability", *Electronic Notes in Theoretical Computer Science*, vol. 170, pp. 125–138, 2007.
[70] X. Guo, T. Muta, and J. Zhao, "Quantum Circuit Ansatz: Patterns of Abstraction and Reuse of Quantum Algorithm Design", *arXiv:2405.05021*, 2024.
[71] V. Lakshmanan, S. Robinson, and M. Munn, *Machine learning design patterns*. O'Reilly Media, 2020.
[72] M. Falkenthal, J. Barzen, U. Breitenbücher, and F. Leymann, "Solution Languages: Easing Pattern Composition in Different Domains", *International Journal on Advances in Software*, vol. 10, no. 3, pp. 263–274, 2017.
[73] Kipu Quantum, *PlanQK Pattern Atlas*, https://patterns.platform.planqk.de/pattern-languages/af7780d5-1f97-4536-8da7-4194b093ab1d, 2025, [accessed: 2025.03.01].
[74] F. Leymann and J. Barzen, "Pattern Atlas", in *Next-Gen Digital Services. A Retrospective and Roadmap for Service Computing of the Future*, Springer, 2021, pp. 67–76.