Warm-Starting Patterns for Quantum Algorithms

Felix Truger, Johanna Barzen, Martin Beisel, Frank Leymann, and Vladimir Yussupov

Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
http://www.iaas.uni-stuttgart.de

in: Proceedings of the 16th International Conference on Pervasive Patterns and Applications (PATTERNS 2024). See also BIBTEX entry below.

BIBTEX:

```
@inproceedings{Truger2024_WarmStartingPatterns,
    title         = {{Warm-Starting Patterns for Quantum Algorithms}},
    author        = {Truger, Felix and Barzen, Johanna and Beisel, Martin and Leymann, Frank
                      and Yussupov, Vladimir},
    year          = 2024,
    month         = apr,
    booktitle     = {Proceedings of the 16\textsuperscript{th} International Conference on
                      Pervasive Patterns and Applications (PATTERNS 2024)},
    publisher     = {Xpert Publishing Services (XPS)},
    pages         = {25--31},
    isbn          = {978-1-68558-161-9},
    url           = {https://www.thinkmind.org/index.php?view=article&articleid=
                      patterns_2024_2_10_70005},
}
```

Warm-Starting Patterns for Quantum Algorithms

Felix Truger, Johanna Barzen, Martin Beisel, Frank Leymann, and Vladimir Yussupov
Institute of Architecture of Application Systems, University of Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany
email: {*firstname.lastname*}@*iaas.uni-stuttgart.de*

Abstract—**Quantum computing promises considerable advantages in efficiency and accuracy over classical computing for certain problems. However, today's Noisy Intermediate-Scale Quantum (NISQ) computers are error-prone and limited in the number of qubits, which complicates leveraging them in practice. To mitigate these issues, multiple warm-starting techniques are being introduced in the quantum computing domain to improve the efficiency and accuracy of quantum algorithms by utilizing known or efficiently generated results as a starting point for the quantum computation. However, heterogeneous warm-starting techniques are often tailored for specific algorithms and require expertise in multiple domains, such as quantum computing and machine learning, thus complicating the choice of technique. Well-structured patterns that abstractly document proven solutions to recurring problems can help quantum software engineers in this decision-making process. In this work, we extend the existing pattern language for quantum algorithms with four novel warm-starting patterns that refine a more abstract pattern introduced in previous work and document how recurring problems in the design and execution of quantum algorithms can be solved with warm-starts. Thereby, the underlying methods are made available for interested parties in a concise and easily digestible manner.**

Keywords—Quantum Computing; Hybrid Algorithms; Quantum Software Engineering; Warm-Start; Patterns.

I. Introduction

On quantum computers, information is represented by the states of quantum bits (qubits), which possess unique properties, such as *superposition* and *entanglement*. Due to these properties, quantum computing promises advantages over classical computing for certain problems [1]. For example, factorization of composite numbers is theoretically feasible with the help of quantum computers, but not known to be tractable with classical computers [2]. Moreover, it has been shown that a significant speed-up over classical machine learning is possible in certain cases when utilizing quantum computers [3].

In the current *Noisy Intermediate-Scale Quantum (NISQ)* era, quantum computers offer a limited number of qubits that are prone to errors [4][5]. Therefore, quantum algorithms are limited to quantum circuits acting on few qubits and requiring only few operations. Moreover, many algorithms are designed as hybrid quantum-classical algorithms with the intention to utilize both classical and quantum computation in a fruitful combination that mitigates these current limitations. The most prominent examples are *Variational Quantum Algorithms (VQAs)* consisting of parameterized quantum circuits and a classical optimizer employed to search for viable parameter values for these circuits to solve a problem at hand [6]. Quantum algorithms can be further improved using so-called warm-starting techniques that utilize known or efficiently generated results as a starting point instead of starting from scratch.

However, *warm-starting* is an umbrella term for a heterogeneous set of techniques that affect quantum algorithms in fundamentally different ways and exhibit a multitude of properties and potential benefits [7]. For example, warm-starts can be realized by encoding information into a quantum algorithm's initial quantum state or a sophisticated parameter initialization. Techniques proposed in the literature are often specialized for specific algorithms and problems, which complicates reusing them or even deciding about their suitability for a certain use case. Moreover, they often require expertise in different domains, including quantum computing and machine learning.

Patterns document abstract solutions to recurring problems [8] and can help engineers understand and apply these solutions for their specific use case. To support quantum software engineers in better understanding the concepts, applicability, and benefits of different warm-starting techniques, we present four novel warm-starting patterns. With these patterns, we capture recurring solution strategies for warm-starting quantum algorithms and refine the more abstract warm-starting pattern that exists in the pattern language for quantum algorithms.

The remainder of the paper is structured as follows: We discuss related work in Section II, before fundamentals and the pattern format are introduced in Section III. Section IV introduces the four new warm-starting patterns in detail. In Section V, we discuss aspects of the application of the patterns and the evaluation of the warm-starts. Finally, Section VI concludes the paper with a summary and outlook.

II. Related Work

Leymann [9] proposed and initiated a pattern language for quantum algorithms. This pattern language has been continuously extended, e.g., with refined patterns for state preparation, hybrid quantum algorithms, error handling, and execution semantics [10]–[16]. In this work, we further extend it by documenting four novel patterns capturing different solutions for warm-starting quantum algorithms, which refine the existing, abstract WARM-START pattern. These new patterns were identified through an analysis of quantum-related warm-starting techniques encountered in the literature (cf. Section III). To the best of our knowledge, there exist no other works documenting patterns in the quantum computing domain and conforming to Alexander et al.'s notion of patterns [8].

Pattern languages, originally known from architecture [8], have been documented for various other domains, e.g., for software engineering [17], enterprise integration [18], and cloud computing [19]. Leymann and Barzen [20] propose Pattern Atlas, a repository and tool to visualize and link patterns of different pattern languages. Moreover, Falkenthal and Leymann [21] propose the concept of solution languages that interconnect concrete solutions for patterns, i.e., implementation artifacts, to systematically collect implementation knowledge and reduce the manual efforts of (re)implementing existing solutions. Such solutions are linked to the corresponding patterns and other solutions as per the relations in the pattern language.

Warm-starting techniques were proposed and examined in various previous works. Mari et al. [22] discuss and evaluate forms of quantum transfer learning, particularly different directions in which quantum transfer learning can be utilized in the context of *Quantum Neural Networks (QNNs)*. Egger et al. [23] and Tate et al. [24], respectively, describe and evaluate how classical approximation algorithms can be utilized in the *Quantum Approximate Optimization Algorithm (QAOA)*, while Galda et al. [25] and Shaydulin et al. [26] focus on transferring parameters across problem instances. Truger et al. [7] explore and analyze warm-starting techniques in the quantum computing domain in a literature study, thereby summarizing categories of such techniques. Beisel et al. [27] propose a workflow modeling extension to facilitate the integration and orchestrations of VQAs in workflows. This includes modeling constructs for warm-starting VQAs with initial parameter values and approximations incorporated into a biased initial state. However, none of these works formally document the warm-starting techniques as solutions to recurring problems in the form of patterns.

III. Fundamentals and Pattern Format

In this section, we discuss fundamentals of quantum algorithms and VQAs in particular. Moreover, we present the pattern format and authoring method used in this work.

A. Fundamentals of Quantum Algorithms

Quantum algorithms are implemented as quantum circuits describing manipulations of qubits similar to classical logic circuits. Quantum circuits consist of wires representing the underlying qubits and gates representing operations on the qubits. The number of wires is called the *width* of the circuit and the number of gates acting on a qubit determines the circuit *depth*. Gates can act on a single qubit or multiple qubits, e.g., the Hadamard gate (H) creates a superposition on a single qubit and the two-qubit controlled-not gate (CNOT) can be used to entangle or disentangle qubits. Some gates, such as the rotation gates RX, RY, and RZ, are parametrized, i.e., the intensity of the manipulation depends on parameter values set at runtime. Therefore, circuits can be parameterized as well and their output upon measurement depends on the parameter values. Such parameterized quantum circuits are the basis for VQAs, such as the QAOA [28], *Variational Quantum Eigensolvers (VQEs)* [29], and QNNs [30]. To determine viable values for the circuit parameters of VQAs, classical optimizers are employed. Quantum and classical execution are then executed in a loop, in which the output of the circuit run on a quantum device is evaluated for the optimizer to steer parameter values in a favorable direction. Once a termination condition is met, e.g., when the result has converged or a set time limit has expired, the circuit can be executed with the final set of optimized parameter values to retrieve the result of the overall quantum-classical algorithm. This way, the aforementioned QAOA can be used to approximate solutions to combinatorial optimization problems or VQEs can be used to approximate eigenvalues with the help of a quantum computer. More generally, QNNs can be trained to compute arbitrary functions for various purposes, e.g., classification or regression [30].

B. Pattern Format and Authoring Method

We follow the pattern format from previous work on quantum computing patterns [9]–[16] and rely on best practices for pattern writing [17]–[19]. Each pattern is identified by its *Name* and an *Icon* that serves as a mnemonic. The *Problem* targeted by the pattern is highlighted with a brief question. Known alternative names are optionally listed as *Aliases*. Afterward, the *Context* in which the pattern is applicable, i.e., the situation in which the problem may arise, is explained. Next, *Forces* that need to be considered when solving the problem are described. Then, we elaborate on the high-level *Solution* that is additionally illustrated by a *Solution Sketch*. In the *Results* paragraph, we discuss the consequences of the solution. Afterward, we draw connections between the new pattern and *Related Patterns*, before we summarize *Known Uses* by listing implementations of the pattern.

As patterns are abstractions of existing solutions, the patterns in this work were identified by exploring warm-starting techniques proposed and used in the literature. In previous work, we conducted a systematic mapping study to survey scientific literature on warm-starting techniques in the quantum computing domain in general, thereby identifying different warm-starting techniques [7]. Recurring approaches that are regarded promising were further analyzed, and the underlying solutions were abstracted and documented as patterns.

IV. Warm-Starting Patterns for Quantum Algorithms

In this section, we first give an overview of the patterns introduced in this work and align them w.r.t. the existing patterns for quantum algorithms. Afterward, we document the four novel warm-starting patterns for quantum algorithms.

A. Pattern Language for Quantum Algorithms

Figure 1 provides an overview of the pattern language for quantum algorithms proposed and initialized by Leymann [9] with its essential patterns for quantum states, unitary transformations, and the program flow of quantum algorithms. It aims to support scientists and software developers in building quantum algorithms. Weigold et al. [10] [11] extended the pattern language with state preparation patterns for quantum algorithms focusing on how data can be encoded in quantum algorithms. Also, additional patterns for the program flow of hybrid algorithms were documented [12].

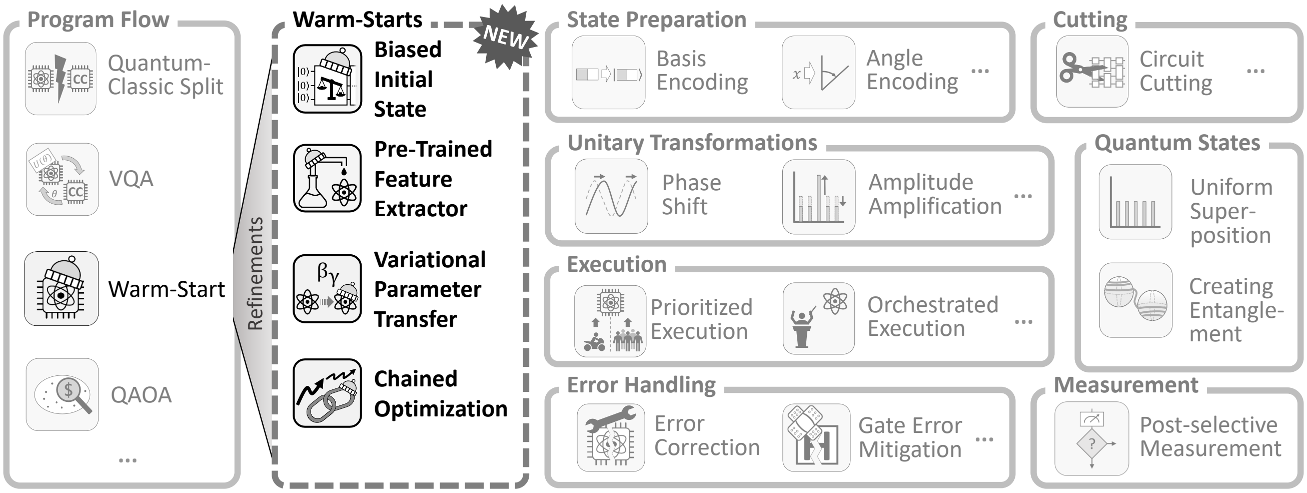

Figure 1. Overview of the pattern language for quantum algorithms, including the newly documented warm-starting patterns highlighted in bold.

Particularly, Weigold et al. [12] identified warm-starting as a general pattern applicable to quantum algorithms, which we aim to refine in this work with more concrete recurring solutions in that sense. Beisel et al. [13] describe patterns for quantum error handling and Georg et al. [14] document patterns for the execution of quantum applications. Moreover, patterns for the partitioning of quantum circuits, i.e., *circuit cutting*, have been introduced by Bechtold et al. [15].

B. Warm-Starting Techniques for Quantum Algorithms

The Warm-Start pattern identified by Weigold et al. refines the more general Quantum-Classic Split pattern, which summarizes splitting of computational workload between quantum and classical computers [9][12]. In this sense, it suggests to use classical methods to approximate a solution to the problem at hand and utilize the approximation as a starting point. As shown in Figure 1, the new patterns presented in this work further refine the Warm-Start pattern.

In our previous work [7], we identified different properties of warm-starting techniques, e.g., warm-starts can be applied in different directions, i.e., classical-to-quantum (C2Q), quantum-to-quantum (Q2Q), and quantum-to-classical (Q2C) [22]. Since this work focuses on warm-starting patterns *for* quantum algorithms in line with the pattern language, only C2Q and Q2Q cases are considered in the following.

C. Biased Initial State

Problem: How to utilize efficient approximations in quantum algorithms to improve the solution quality or speed up the computation?

Context: For many computationally hard problems, efficient approximation algorithms exist. However, typical quantum algorithms neglect these approximations and valuable information remains unused as the quantum algorithm starts from a neutral position. As a result, deep quantum circuits may be required, which increases accumulative error rates, and more quantum resources may be required to solve a problem. **Forces:** Moreover, current quantum devices are error-prone, thus, the depth of executable quantum circuits is limited. However, including approximations requires special care, as it can limit the quantum algorithm in an unintended way [31][32]. Also, changing the initial state may require additional adaptations of corresponding parts of the quantum circuit [23][33]. **Solution:** Encode approximations into the initial state of quantum circuits, thereby biasing the initial quantum state towards viable solutions. Hence, a chain of algorithm executions as depicted in Figure 2 is beneficial: First, an efficient algorithm is utilized to approximate a solution of a given problem instance. This can often be achieved at low cost on classical hardware. Then, the initial state $|\psi\rangle$ of the subsequent quantum algorithm is biased toward the approximation and the algorithm is

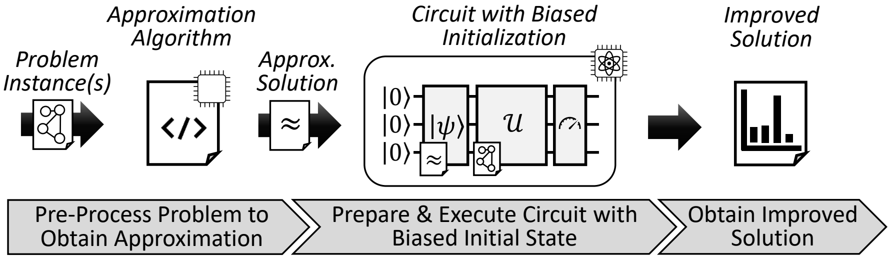

Figure 2. Solution sketch for the Biased Initial State pattern

executed on a quantum device to obtain an improved solution.
Result: The quantum algorithm employed in the second step utilizes the approximation as a starting point to improve upon. Due to the biased initial state, optimal solutions can be explored quicker and the solution quality achievable in a set amount of time may therefore increase. Moreover, this way the workload of the overall computation can be distributed to multiple devices, e.g., classical and quantum devices.
Related Patterns: This pattern is a refinement of the Warm-Start pattern and related to the state preparation patterns, e.g., Angle Encoding, since different encodings may be applied to prepare and bias the initial state of a quantum algorithm [11][12]. Moreover, it can be applied with the VQA pattern and its refinements, such as the QAOA [12].
Known Uses: Egger et al. [23] introduce a biased initial state for QAOA and the Maximum Cut problem (MaxCut) utilizing the classical Goemans-Williamson approximation algorithm. Similarly, Tate et al. [24] adapt QAOA for MaxCut with a Burer-Monteiro relaxation of the problem. QAOA was also adapted for a biased initial state for the Knapsack problem [34]. Wang [35] proposes a "classically-boosted" quantum algorithm for the Maximum 3-Satisfiability and Maximum Bisection problems based on biased initial states. Beisel et al. [27] propose a workflow modeling construct facilitating the integration of warm-starts via biased initial states in VQAs.

D. Pre-Trained Feature Extractor

Problem: How to process large data items through QNNs when the number of available qubits is lower than the size of a data item?

Aliases: Quantum Transfer Learning [22]
Context: A QNN shall be trained for a specific task, that requires the processing of large data items, e.g., images or multi-dimensional data. However, the number of qubits required to load such data items into the QNN is larger than the number of qubits of the available quantum devices.
Forces: The width of circuits implementing QNNs is limited by the number of available qubits. In addition, quantum devices are scarce resources that should be utilized as efficiently as possible. However, naively reducing the original data items may result in the loss of information relevant for the computation. Large pre-trained classical models for various general tasks, such as object recognition for images, are widely available or can be created at low cost.
Solution: Use a pre-trained classical model to reduce the dimensions of the data items and train the QNN based on the reduced data. As shown in Figure 3, a pre-trained classical model for a wide range purpose, such as a neural network trained for object recognition, can be utilized for a hybrid QNN to be trained for a related special purpose task. Intermediate values of inputs processed through such models, e.g., those present at a condensed next-to-last neural network layer, can be seen as a compressed representation of the original data exhibiting its most significant features. Thus, the pre-trained model serves as a feature extractor. These features can be encoded into a quantum state to train the QNN for the target task.
Result: Due to the compressed representation obtained from the pre-trained feature extractor, fewer qubits are required to process data in the QNN. Furthermore, the compressed nature of the data may reduce the QNN's training time, as irrelevant information has already been omitted from the training data.
Related Patterns: This pattern refines the Warm-Start pattern and is related to the state preparation patterns, e.g., Angle Encoding, [11][12]. Different encodings may be applied to encode the extracted features into a quantum state. It is typically applied in conjunction with QNNs, a form of VQA [6]. Furthermore, the Circuit Cutting pattern solves a similar problem by partitioning the computation of a large quantum circuit into computations of multiple smaller circuits [15].
Known Uses: Pre-Trained Feature Extractor is frequently used when image processing, particularly image classification, shall be enhanced with QNNs [22] [36]–[41]. It was also applied for text classification [42]. Moreover, autoencoders [43] can be considered a special case of Pre-Trained Feature Extractor, that are designed and trained specifically for the purpose of data compression.

E. Variational Parameter Transfer

Problem: How to obtain a problem-aware parameter initialization for VQAs that reduces the optimization runtime?

Context: A VQA needs to be executed on a quantum device, which encompasses the optimization of its variational parameters. Parameter optimization requires repeated access to the quantum device, typically starting with random initial parameter values [44], to sample solutions and determine a direction for their optimization, e.g., through gradient descent.

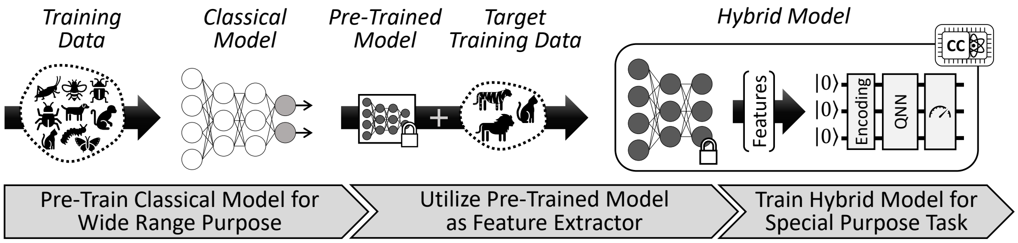
Training Data
Classical Model
Pre-Trained Model
Target Training Data
Hybrid Model
CC
Features
|0⟩
|0⟩
|0⟩
Encoding
QNN
Pre-Train Classical Model for Wide Range Purpose
Utilize Pre-Trained Model as Feature Extractor
Train Hybrid Model for Special Purpose Task

Figure 3. Solution sketch for the Pre-Trained Feature Extractor pattern

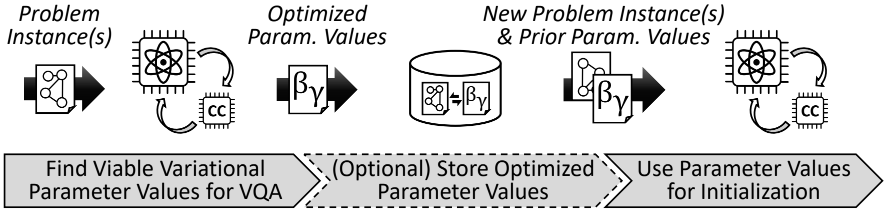
Problem Instance(s)
Optimized Param. Values
New Problem Instance(s) & Prior Param. Values
βγ
CC
Find Viable Variational Parameter Values for VQA
(Optional) Store Optimized Parameter Values
Use Parameter Values for Initialization

Figure 4. Solution sketch for the VARIATIONAL PARAMETER TRANSFER pattern

Forces: Obtaining viable parameter initializations for VQA is challenging due to large parameter spaces and effects, such as barren plateaus [45] and non-convex optimization landscapes [46]. Barren plateaus are areas with vanishing gradients in a cost function's parameter space that must be avoided, whereas local minima in non-convex optimization landscapes pose an additional challenge to efficient parameter initialization as they disturb the search for a global optimum.

Solution: Transfer viable variational parameter values from related problem instances. As shown in Figure 4, optimized parameter values may be stored or directly reused for new problem instances. In many cases, it can be expected that optimized parameter values for a solved problem instance are in proximity of viable parameter values for a related or similar new problem instance. Therefore, optimized parameter values from earlier executions may be utilized for a problem-aware parameter initializationinstead of a random initialization. Appropriate databases, toolkits, and provenance systems for quantum computing [47][48] facilitate the optional storage of optimized parameter values for their utilization in later executions.

Result: Parameter transfers can reduce the number of iterations of the optimization loop. A favorable parameter initialization can also increase the likelihood of finding globally optimal parameter values and thus increase the solution quality.

Related Patterns: This pattern is a refinement of the WARM-START pattern and can be applied in conjunction with VQA, including its refinements like QAOA, [12].

Known Uses: VARIATIONAL PARAMETER TRANSFER has been frequently proposed and applied for QAOA and Max-Cut [25][26][49][50]. Moreover, Shaydulin et al.'s repository of preoptimized parameters implements the storage option [47]. Beisel et al. [27] propose a modeling construct for workflows to integrate warm-starts via parameter initialization in VQAs.

F. Chained Optimization

Problem: How to avoid local optima and improve convergence when optimizing variational parameter values for VQAs?

Context: Optimal variational parameter values for a VQA need to be determined. The performance of the algorithm depends heavily on these values and a global optimum in the parameter space is needed to obtain optimal solutions.

Forces: Local minima in non-convex optimization landscapes and barren plateaus hinder the optimization, as the optimizer may be unable to reach a global optimum. Moreover, evaluating all possible parameter values is infeasibly expensive.

Solution: Chain different optimizers with different scopes or strengths together. As indicated in Figure 5, a global optimization strategy can be combined with a subsequent local optimizer. The former would determine a general area of interest in the overall optimization landscape. Afterward, the local optimizer is started from a point in this area of interest and searches on a smaller scale, aiming to find the global optimum.

Result: By chaining optimizers, the subsequent optimizers utilize previously obtained results as starting points to improve upon. Thereby, optimizers are combined to benefit from their respective strengths and achieve cost-efficient optimization.

Related Patterns: This pattern refines the WARM-START pattern and can be applied in conjunction with VQA, including QAOA, [12]. It is similar to the VARIATIONAL PARAMETER TRANSFER pattern documented above, with an unaltered problem instance, while the algorithm in use, specifically the optimization algorithm, is exchanged instead.

Known Uses: Rad et al. [51] use this method to avoid barren plateaus in VQAs. Tao et al. [52] apply it in a QNN optimization. Wauters et al. [53] supplement their Reinforcement Learning-based optimization approach for QAOA with subsequent gradient-based local optimization.

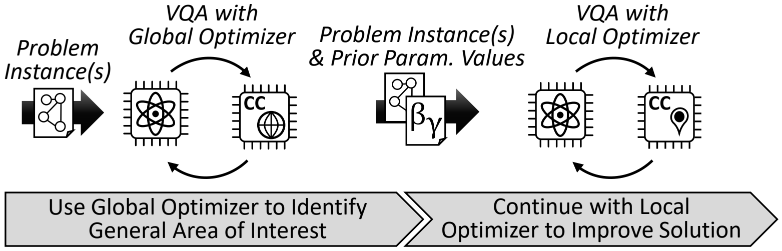
Problem Instance(s)
VQA with Global Optimizer
Problem Instance(s) & Prior Param. Values
VQA with Local Optimizer
βγ
CC
CC
Use Global Optimizer to Identify General Area of Interest
Continue with Local Optimizer to Improve Solution

Figure 5. Solution sketch for the CHAINED OPTIMIZATION pattern

V. Discussion

We discuss known and potential challenges and limitations, and evaluation criteria for the application of the patterns above.

The dependency of concrete warm-starting solutions on different problem-specific factors, such as the nature of the quantum algorithm and problem at hand, available approximation algorithms, and feasible quantum state preparation procedures, can complicate the pattern application. In particular, the Biased Initial State and Pre-Trained Feature Extractor patterns require the determination of suitable techniques for obtaining and incorporating starting points on a case-by-case basis. Moreover, it was shown that the success of warm-starts through a biased initial state can depend on the careful selection of approach-specific hyperparameters [23][54]. In addition, such warm-starts can unintentionally prevent improvements as was shown, for example, for a warm-started variant of the QAOA where replacing the initial uniform superposition with the encoding of a good solution fails with little to no improvements [31]. Furthermore, applying biased initial states can impose restrictions on the parameterized quantum circuit in VQAs and some state preparations are not feasible on current NISQ hardware [32]. More specifically, some circuit designs complicate or prevent retaining the solution quality associated with the encoded biased initial state. These challenges and limitations likely also apply to the Pre-Trained Feature Extractor pattern, which likewise requires encoding information in the initial state. However, it was shown for the Biased Initial State that both problems can be avoided in some cases by transforming the initial state into a parameter transfer in VQAs [32]. Incorporating starting points for the parameter-focused Variational Parameter Transfer and Chained Optimization patterns is trivial since it reduces to a parameter initialization. Nonetheless, these warm-starts via parameter initializations could also potentially restrict the subsequent optimization in an undesired way when applied improperly, especially by limiting the optimization to an unfavourable area of the parameter space.

Also the evaluation of warm-starting techniques is challenging, as it is problem-specific and likewise dependent on different factors, such as available approximations and state preparation procedures. As different warm-starting methods aim to improve upon different behaviours, e.g., a reduced need for quantum computational resources, reduced runtime, or increased accuracy (cf. [7]), different approaches and metrics are required for analyzing and comparing them. Moreover, in the case of hybrid warm-starts, the trade-off between classical and quantum computational efforts may be ambiguous and dependent on the use case and concrete resources at hand.

The broad spectrum of potential applications of the warm-starting patterns introduced in this work may become even more extensive when considering warm-starts in other contexts outside of the quantum computing domain. It is conceivable that some techniques are analogously applicable in similar contexts of classical computing and, particularly, the classical domains of machine learning and optimization (cf. [7]).

VI. Conclusion

In this work, we elaborated on warm-starting techniques for quantum algorithms. We documented four novel patterns, Biased Initial State, Pre-Trained Feature Extractor, Variational Parameter Transfer, and Chained Optimization, thereby expanding the existing pattern language for quantum algorithms and refining the Warm-Start pattern. By documenting and making the knowledge on these solutions to recurring problems easily accessible for interested parties, we hope to assist quantum software engineers in utilizing warm-starting techniques in their applications.

In future work, we aim to analyze additional warm-starting techniques and their implementations to evaluate their compatibility with each other. Moreover, we will incorporate the warm-starting patterns presented in this work into the publicly available Pattern Atlas on the PlanQK platform [55], where also the other patterns of the pattern language for quantum algorithms have been incorporated. The accessibility for a broad audience enables refinement of the patterns based on community feedback. Moreover, the platform facilitates linking related patters together, even across different pattern languages.

Acknowledgement

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

References

[1] J. Preskill, “Quantum computing and the entanglement frontier,” *arXiv:1203.5813*, 2012.

[2] P. Shor, “Algorithms for Quantum Computation: Discrete Logarithms and Factoring,” in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.

[3] Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning,” *Nature Physics*, vol. 17, no. 9, pp. 1013–1017, 2021.

[4] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, 2018.

[5] 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, p. 044007, 2020.

[6] M. Cerezo *et al.*, “Variational quantum algorithms,” *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[7] F. Truger *et al.*, “Warm-Starting and Quantum Computing: A Systematic Mapping Study,” *arXiv:2303.06133*, 2023.

[8] C. Alexander *et al.*, *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.

[9] F. Leymann, “Towards a Pattern Language for Quantum Algorithms,” in *First International Workshop, QTOP 2019, Munich, Germany, March 18, 2019, Proceedings*. Springer, 2019, pp. 95–101.

[10] M. Weigold, J. Barzen, F. Leymann, and M. Salm, “Data Encoding Patterns For Quantum Algorithms,” in *Proceedings of the 27th Conference on Pattern Languages of Programs (PLoP)*. HILLSIDE, 2020, pp. 1–11.

[11] ——, “Expanding Data Encoding Patterns For Quantum Algorithms,” in *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, 2021, pp. 95–101.

[12] 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 International Publishing, 2021, pp. 34–51.

[13] 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.

[14] 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.

[15] 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, [in press].

[16] 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.

[17] J. O. Coplien, *Software Patterns*, ser. SIGS management briefings. SIGS Books & Multimedia, 1996.

[18] G. Hohpe and B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional, 2004.

[19] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*. Springer, 2014.

[20] F. Leymann and J. Barzen, “Pattern Atlas,” in *Next-Gen Digital Services. A Retrospective and Roadmap for Service Computing of the Future: Essays Dedicated to Michael Papazoglou on the Occasion of His 65th Birthday and His Retirement*. Cham: Springer International Publishing, 2021, pp. 67–76.

[21] M. Falkenthal and F. Leymann, “Easing Pattern Application by Means of Solution Languages,” in *Proceedings of the 9th International Conference on Pervasive Patterns and Applications (PATTERNS)*, 2017, pp. 58–64.

[22] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, “Transfer learning in hybrid classical-quantum neural networks,” *Quantum*, vol. 4, p. 340, 2020.

[23] D. J. Egger, J. Mareček, and S. Woerner, “Warm-starting quantum optimization,” *Quantum*, vol. 5, p. 479, 2021.

[24] R. Tate, M. Farhadi, C. Herold, G. Mohler, and S. Gupta, “Bridging classical and quantum with SDP initialized warm-starts for QAOA,” *ACM Transactions on Quantum Computing*, vol. 4, no. 2, pp. 1–39, 2023.

[25] A. Galda, X. Liu, D. Lykov, Y. Alexeev, and I. Safro, “Transferability of optimal QAOA parameters between random graphs,” in *IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021, pp. 171–180.

[26] R. Shaydulin, P. C. Lotshaw, J. Larson, J. Ostrowski, and T. S. Humble, “Parameter Transfer for Quantum Approximate Optimization of Weighted MaxCut,” *ACM Transactions on Quantum Computing*, vol. 4, no. 3, pp. 1–15, 2023.

[27] M. Beisel *et al.*, “QuantME4VQA: Modeling and Executing Variational Quantum Algorithms Using Workflows,” in *Proceedings of the 13th International Conference on Cloud Computing and Services Science (CLOSER)*. SciTePress, 2023, pp. 306–315.

[28] E. Farhi, J. Goldstone, and S. Gutmann, “A Quantum Approximate Optimization Algorithm,” *arXiv:1411.4028*, 2014.

[29] A. Peruzzo *et al.*, “A variational eigenvalue solver on a photonic quantum processor,” *Nature communications*, vol. 5, no. 1, 2014.

[30] E. Farhi and H. Neven, “Classification with Quantum Neural Networks on Near Term Processors,” *arXiv:1802.06002*, 2018.

[31] M. Cain, E. Farhi, S. Gutmann, D. Ranard, and E. Tang, “The QAOA gets stuck starting from a good classical string,” *arXiv:2207.05089*, 2022.

[32] F. Truger, J. Barzen, F. Leymann, and J. Obst, “Warm-Starting the VQE with Approximate Complex Amplitude Encoding,” *arXiv:2402.17378*, 2024.

[33] R. Tate, J. Moondra, B. Gard, G. Mohler, and S. Gupta, “Warm-Started QAOA with Custom Mixers Provably Converges and Computationally Beats Goemans-Williamson’s Max-Cut at Low Circuit Depths,” *Quantum*, vol. 7, p. 1121, 2023.

[34] W. van Dam, K. Eldefrawy, N. Genise, and N. Parham, “Quantum Optimization Heuristics with an Application to Knapsack Problems,” in *IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021, pp. 160–170.

[35] G. Wang, “Classically-Boosted Quantum Optimization Algorithm,” *arXiv:2203.13936*, 2022.

[36] S. Mittal and S. K. Dana, “Gender Recognition from Facial Images using Hybrid Classical-Quantum Neural Network,” in *IEEE Students Conference on Engineering & Systems (SCES)*. IEEE, 2020, pp. 1–6.

[37] A. Gokhale, M. B. Pande, and D. Pramod, “Implementation of a quantum transfer learning approach to image splicing detection,” *International Journal of Quantum Information*, vol. 18, no. 05, p. 2050024, 2020.

[38] V. Azevedo, C. Silva, and I. Dutra, “Quantum transfer learning for breast cancer detection,” *Quantum Machine Intelligence*, vol. 4, no. 1, p. 5, 2022.

[39] M. J. Umer *et al.*, “An integrated framework for COVID-19 classification based on classical and quantum transfer learning from a chest radiograph,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 20, p. e6434, 2022.

[40] T. Kanimozhi, S. Sridevi, T. S. Manikumar, T. Dheeraj, and A. Sumanth, “Brain Tumor Recognition based on Classical to Quantum Transfer Learning,” in *International Conference on Innovative Trends in Information Technology (ICITIIT)*. IEEE, 2022, pp. 1–5.

[41] A. Furutanpey *et al.*, “Architectural Vision for Quantum Computing in the Edge-Cloud Continuum,” in *IEEE International Conference on Quantum Software (QSW)*. IEEE, 2023, pp. 88–103.

[42] C.-H. H. Yang, J. Qi, S. Y.-C. Chen, Y. Tsao, and P.-Y. Chen, “When BERT Meets Quantum Temporal Convolution Learning for Text Classification in Heterogeneous Computing,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8602–8606.

[43] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.

[44] A. Kulshrestha and I. Safro, “BEINIT: Avoiding Barren Plateaus in Variational Quantum Algorithms,” in *IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022, pp. 197–203.

[45] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, “Cost function dependent barren plateaus in shallow parametrized quantum circuits,” *Nature communications*, vol. 12, no. 1, p. 1791, 2021.

[46] P. Huembeli and A. Dauphin, “Characterizing the loss landscape of variational quantum circuits,” *Quantum Science and Technology*, vol. 6, no. 2, p. 025011, 2021.

[47] R. Shaydulin, K. Marwaha, J. Wurtz, and P. C. Lotshaw, “QAOAKit: A Toolkit for Reproducible Study, Application, and Verification of the QAOA,” in *IEEE/ACM 2nd International Workshop on Quantum Computing Software (QCS)*. IEEE, 2021, pp. 64–71.

[48] B. Weder, J. Barzen, F. Leymann, M. Salm, and K. Wild, “QProv: A provenance system for quantum computing,” *IET Quantum Communication*, vol. 2, no. 4, pp. 171–181, 2021.

[49] F. G. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, “For Fixed Control Parameters the Quantum Approximate Optimization Algorithm’s Objective Function Value Concentrates for Typical Instances,” *arXiv:1812.04170*, 2018.

[50] J. Wurtz and D. Lykov, “Fixed-angle conjectures for the quantum approximate optimization algorithm on regular MaxCut graphs,” *Phys. Rev. A*, vol. 104, p. 052419, 2021.

[51] A. Rad, A. Seif, and N. M. Linke, “Surviving The Barren Plateau in Variational Quantum Circuits with Bayesian Learning Initialization,” *arXiv:2203.02464*, 2022.

[52] Z. Tao, J. Wu, Q. Xia, and Q. Li, “LAWS: Look Around and Warm-Start Natural Gradient Descent for Quantum Neural Networks,” in *IEEE International Conference on Quantum Software (QSW)*. IEEE, 2023, pp. 76–82.

[53] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro, “Reinforcement-learning-assisted quantum optimization,” *Physical Review Research*, vol. 2, no. 3, p. 033446, 2020.

[54] F. Truger, M. Beisel, J. Barzen, F. Leymann, and V. Yussupov, “Selection and Optimization of Hyperparameters in Warm-Started Quantum Optimization for the MaxCut Problem,” *Electronics*, vol. 11, no. 7, 2022.

[55] PlanQK, “PlanQK - Pattern Atlas,” https://patterns.platform.planqk.de, 2023, [accessed: 2024.02.28].