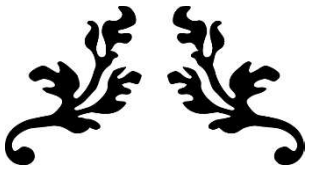


Ausgewählte Themen der Quanteninformatik

Hauptseminar Sommersemester 2026

Dieser Sammelband enthält die Ausarbeitungen des Hauptseminars **„Ausgewählte Themen der Quanteninformatik“**. Die Beiträge befassen sich mit aktuellen Fragestellungen und Methoden der Quanteninformatik und spannen den Bogen von Ansätzen zur Schaltkreisausführung bis hin zur Optimierung von Quantenalgorithmen.

Quantum Circuit Cutting

Benedikt Rösch
st201327@stud.uni-stuttgart.de

Hauptseminar Ausgewählte Themen der Quanteninformatik

In der Quanteninformatik hängt die nötige Qubitanzahl häufig von der Größe der Eingabedaten ab [1]. Dadurch benötigen Schaltkreise oft mehr Qubits als eine Quantum Processing Unit (QPU) bereitstellt. Quantum Circuit Cutting (QCC) stellt einen Ansatz zur Lösung dieses Problems dar [6].

Grundlegendes Schema QCC ist ein Anwendungsmuster der Quanteninformatik, welches einen Schaltkreis in mehrere Teilschaltkreise zerlegt. Dabei unterscheidet man zwischen *Wire Cuts*, bei denen Qubit-Kanäle zeitlich bzw. vertikal durchtrennt werden, sowie *Gate Cuts*, bei denen Mehr-Qubit-Gatter räumlich bzw. horizontal getrennt werden [7]. Die erhaltenen Teilschaltkreise werden unabhängig voneinander mehrfach ausgeführt, um ihre jeweiligen Erwartungswerte zu bestimmen. Diese werden klassisch durch eine gewichtete Linearkombination zum Erwartungswert des ungeschnittenen Schaltkreises rekombiniert [4]. Die verloren gegangene Quantenkorrelation der Schnitte wird so quasiprobabilistisch simuliert [7].

Wire Cut Beim Wire Cut wird ein Quantenkanal Φ in eine Linearkombination $\Phi = \sum_i a_i \Phi_i$ [6] von Mess- und Präparationskanälen Φ_i zerlegt. Da eine Quantenleitung dem Identitätskanal entspricht und eine Messung den Quantenzustand projiziert, muss mehrfach in verschiedenen Basen gemessen werden. Der Erwartungswert des Identitätskanals lässt sich so aus den Erwartungswerten der Mess- und Präparationskanäle rekonstruieren [8].

Ein Wire Cut erzeugt zunächst zwei Teilschaltkreise, in die entsprechend der Kanalzerlegung für jede Basis Mess- bzw. Präparationsoperatoren implementiert werden. Somit entsteht für jeden Summanden der Linearkombination ein Teilschaltkreispaar Φ_i [1, 3] in der entsprechenden Basis. Der erste Teilschaltkreis misst das geschnittene Qubit und übermittelt das Messergebnis klassisch an den zweiten. Dieser präpariert das geschnittene Qubit entsprechend vor der Ausführung und misst anschließend in der ursprünglichen Berechnungsbasis. Die bedingten Erwartungswerte der Teilschaltkreispaare bilden den Erwartungswert des jeweiligen Mess- und Präparationskanals. Diese werden anschließend mit den Linearkoeffizienten der Zerlegung gewichtet und zum Erwartungswert des ungeschnittenen Schaltkreises summiert [1].

Gate Cut Beim Gate Cut wird ein unitäres Mehr-Qubit-Gatter U in eine Linearkombination $U = \sum_i a_i F_i^A \otimes F_i^B$ von Tensorprodukten lokaler Ein-Qubit-Operatoren zerlegt, die nicht zwingend unitär sind [1, 2]. Hierzu wird U durch mehrere Ein-Qubit-Operatoren ersetzt, sodass für jeden Summanden der Linearkombination ein voneinander unabhängiges Teilschaltkreispaar entsteht. Diese werden separat ausgeführt, in der Berechnungsbasis gemessen und anschließend anhand der Linearkoeffizienten zum Erwartungswert des ursprünglichen Schaltkreises rekonstruiert. Die konkrete Zerlegung hängt sowohl vom geschnittenen Gatter [9] als auch von

den durch die QPU unterstützten Operationen ab. Daher können einige Gatterzerlegungen auf einer bestimmten Hardware nicht direkt implementiert werden [1]. Nicht unitäre Operatoren entsprechen Messoperatoren [1] und können daher nicht als deterministisches Quantengatter implementiert werden.

Gamma-Faktor Die Teilschaltkreise müssen mit einer hohen Shotzahl ausgeführt werden, um präzise Erwartungswerte zu erhalten [3]. Die ursprüngliche Shotzahl verteilt sich somit nicht über die Teilprobleme. Dieser Mehraufwand wird durch den Gamma-Faktor $\gamma = \sum_i |a_i|$ [7] als Summe der Beträge der Linearkoeffizienten beschrieben. Der eigentliche Messaufwand wächst proportional zu γ^2, um dieselbe Varianz des ungeschnittenen Schaltkreises zu erreichen. Bei mehreren Schnitten verhält sich der Gamma-Faktor multiplikativ, da alle Kombinationen der entstehenden Teilschaltkreise berücksichtigt werden müssen [7]. Die benötigte Shotzahl steigt daher mit jedem zusätzlichen Schnitt exponentiell.

Nachteile Die Wahl optimaler Schnittpositionen ist wegen des davon abhängigen exponentiellen Mehraufwands entscheidend. Ihre Bestimmung entspricht einem NP-schweren Graphenproblem, wodurch eine komplexe klassische Vorverarbeitung erforderlich ist [6, 8]. Der Quantenvorteil geht durch den exponentiellen Mehraufwand zwar nicht vollständig verloren, wird jedoch abgeschwächt [2].

QCC erhöht durch den zusätzlichen Mess- und Präparationsaufwand sowie die Gatterersetzungen die Schaltkreiskomplexität und führt weitere Fehlerquellen ein, welche die Auswirkungen von Rauschen verstärken können [5]. Zudem sind Gatterzerlegungen durch die verwendete QPU limitiert.

Die Skalierbarkeit von Circuit Cutting ist problemabhängig und wird insbesondere vom spezifischen Schaltkreisaufbau sowie der Anzahl der in der QPU verfügbaren Qubits beeinflusst [2]. Stark verschränkte Schaltkreise lassen sich bezogen auf den Gamma-Faktor häufig weniger effizient zerlegen als modulare Schaltkreise.

Fazit und Ausblick Circuit Cutting ermöglicht die Ausführung von Quantenschaltkreisen, deren Qubitbedarf die Kapazität der genutzten QPU übersteigt. Es ist somit eine Softwarelösung für ein Hardwareproblem, die lokale Teilprobleme berechnet und anhand deren Erwartungswerte den ursprünglichen Schaltkreis-Erwartungswert rekonstruiert. Die Nachteile des Anwendungsmusters sind eine exponentielle Steigerung der Gesamtshotzahl mit jedem Schnitt, sowie ein erhöhter Aufwand für die klassische Vor- und Nachverarbeitung. Die Effizienz ist dabei insbesondere von Position und Anzahl der Schnitte abhängig.

Bereits heute ermöglicht QCC die Ausführung großer Quantenprobleme auf QPUs der NISQ-Ära und bleibt auch darüber hinaus für modulare Quantenarchitekturen, deren Komponenten lediglich klassisch verbunden sind, relevant [1]. Dadurch kann die Bedeutung kleiner, lokaler Quantensysteme steigen. Insbesondere im Edge-Computing [4] können so neue Anwendungsmöglichkeiten für Quantenalgorithmen ermöglicht werden.

QCC eröffnet zudem neue Forschungsgebiete und motiviert bestehende neu. Hierzu zählen insbesondere die Bestimmung optimaler Schnittpositionen und die Reduktion des Samplingaufwands [7]. Außerhalb der Quanteninformatik sind beispielsweise die Entwicklung effizienter Rekombinationsalgorithmen sowie die Lösung von Graphen- und Optimierungsproblemen eng verbundene Forschungsfelder [6].

Literatur

1. Bechtold, M., Barzen, J., Beisel, M., Leymann, F., Weder, B.: Patterns for Quantum Circuit Cutting. In: Proceedings of the 30th Conference on Pattern Languages of Programs, PLoP '23, The Hillside Group, USA (2023), ISBN 9781941652190
2. Brandhofer, S., Polian, I., Krsulich, K.: Optimal Partitioning of Quantum Circuits Using Gate Cuts and Wire Cuts. IEEE Transactions on Quantum Engineering **5**, 1–10 (2024), https://doi.org/10.1109/TQE.2023.3347106
3. Brenner, L., Piveteau, C., Sutter, D.: Optimal Wire Cutting With Classical Communication. IEEE Transactions on Information Theory **71**(10), 7742–7752 (Oct 2025), ISSN 1557-9654, https://doi.org/10.1109/tit.2025.3601047, URL http://dx.doi.org/10.1109/TIT.2025.3601047
4. Furutanpey, A., Barzen, J., Bechtold, M., Dustdar, S., Leymann, F., Raith, P., Truger, F.: Architectural Vision for Quantum Computing in the Edge-Cloud Continuum. In: 2023 IEEE International Conference on Quantum Software (QSW), pp. 88–103 (2023), https://doi.org/10.1109/QSW59989.2023.00021
5. Meulen, M., Neumann, N.M.P., Verbree, J.: Evaluating Quantum Wire Cutting for QAOA: Performance Benchmarks in Ideal and Noisy Environments (2 2026)
6. Peng, T., Harrow, A.W., Ozols, M., Wu, X.: Simulating Large Quantum Circuits on a Small Quantum Computer. Phys. Rev. Lett. **125**(15), 150504 (2020), https://doi.org/10.1103/PhysRevLett.125.150504
7. Schmitt, L., Piveteau, C., Sutter, D.: Cutting circuits with multiple two-qubit unitaries. Quantum **9**, 1634 (Feb 2025), ISSN 2521-327X, https://doi.org/10.22331/q-2025-02-18-1634, URL http://dx.doi.org/10.22331/q-2025-02-18-1634
8. Uchehara, Gideon, M.M.u.A.A.: Quantum Circuit Cutting. Verfügbar: https://pennylane.ai/demos/tutorial_quantum_circuit_cutting (2022), Zuletzt aufgerufen: 27.04.2026
9. Ufrecht, C., Periyasamy, M., Rietsch, S., Scherer, D.D., Plinge, A., Mutschler, C.: Cutting multi-control quantum gates with ZX calculus. Quantum **7**, 1147 (Oct 2023), ISSN 2521-327X, https://doi.org/10.22331/q-2023-10-23-1147, URL https://doi.org/10.22331/q-2023-10-23-1147

Quanten Hauptkomponentenanalyse

Arif Iscak

Institut für Architektur von Anwendungssystemen

Zusammenfassung. Klassische Hauptkomponentenanalyse (PCA) skaliert mit $O(d^3)$ und wird bei hochdimensionalen Datensätzen zum Engpass. Lloyd et al. [3] zeigen mit der Quanten Hauptkomponentenanalyse (QPCA) einen Algorithmus, der die Hauptkomponenten einer Dichtematrix ρ in $O(R \log d)$ extrahiert, ein exponentieller Geschwindigkeitsvorteil bei niedrigem Rang R.

1 Einleitung

Reale Datensätze sind häufig hochdimensional: Jedes Objekt, etwa in einem Kostümdatensatz, wird durch zahlreiche Merkmale wie Farbe, Material oder Preis beschrieben. Mit wachsender Dimension d steigt der Rechenaufwand klassischer Analysemethoden erheblich, und viele Merkmale sind zudem korreliert. PCA skaliert mit $O(d^3)$, da die vollständige Eigenwertzerlegung der Kovarianzmatrix berechnet werden muss, was bei großen Datensätzen zum Engpass wird. Ziel ist eine Methode, die Dimensionen reduziert und die wesentliche Information erhält.

2 Grundlagen

PCA ersetzt korrelierte Variablen durch wenige unkorrelierte Hauptkomponenten, die Richtungen maximaler Varianz im Datenraum entsprechen den Eigenvektoren der Kovarianzmatrix[1]. Der limitierende Faktor bleibt die Laufzeit von $O(d^3)$.

Für die quantenmechanische Umsetzung lässt sich ein Satz klassischer Datenvektoren $a_1, \ldots, a_n$ als Kovarianzmatrix

$$C = \sum_i |\vec{a}_i|^2 a_i a_i \tag{1}$$

darstellen. Da C positiv semidefinit und hermitesch ist, erfüllt sie zentrale Eigenschaften einer Dichtematrix. Normierung auf die Spur, $\rho = C/\mathrm{tr}(C)$, liefert eine gültige Dichtematrix mit denselben Eigenvektoren wie C: Die Eigenvektoren von ρ sind damit automatisch die Hauptkomponenten.

3 Methode

Lloyd et al. [3] extrahieren die Eigenvektoren von ρ exponentiell schneller als klassisch möglich. Das zentrale Verfahren ist die Dichtematrix-Exponentiation, bei der $e^{-i\rho t}$ indirekt konstruiert wird, da ρ nur als viele physikalische Kopien vorliegt. Jede Kopie

führt dabei über eine Swap-Operation eine infinitesimale Rotation auf σ aus. Über n Kopien akkumuliert sich dies zu $e^{-i\rho t}\sigma e^{i\rho t}$. Bei QPCA wird $\sigma = \rho$ gesetzt, der Zustand wirkt also auf sich selbst.

Mit $e^{-i\rho t}$ liest die Quantenphasenschätzung (QPE) über Interferenz die Eigenwerte $\tilde{r}_i$ aus und erhält durch diese dann die Eigenvektoren χ_i, da jeder Eigenzustand einen eigenwertabhängigen Phasenfaktor erhält.

Der Algorithmus läuft wie folgt ab: Als erstes werden Daten über QRAM geladen und ρ implizit aufgebaut, dann dient die Dichtematrix-Exponentiation als Subroutine, die QPE wiederholt aufruft. Das Ergebnis ist

$$\sum_i r_i \chi_i \chi_i \otimes \tilde{r}_i \tilde{r}_i, \tag{2}$$

mit einer Laufzeit von $O(R \log d)$.

4 Ergebnisse

PCA benötigt $O(d^3)$, QPCA erreicht bei niedrigem Rang R praktisch $O(\log d)$. Bei $d = 10^3$ ergibt sich bereits ein Geschwindigkeitsvorteil enormer Größenordnung, algorithmischer und nicht hardwarebedingter Natur.

5 Diskussion

Der praktischen Umsetzung stehen zwei Hürden unterschiedlicher Natur entgegen. Erstens erfordert das Verfahren tiefe Schaltkreise und viele Kopien von ρ, was auf NISQ-Hardware wegen Dekohärenz und Gatterfehlern nicht robust erfüllbar ist, ein Hardware-Reifegrad-Problem. Variationelle Quanten Algorithmen (VQA) wie der Variational Quantum State Eigensolver (VQSE) [2] begegnen dem mit flachen, klassisch optimierten Schaltkreisen. Außerdem benötigt VQSE nur eine Kopie von ρ, ist jedoch heuristisch ohne Konvergenzgarantie, der Speedup nur empirisch nachweisbar.

Zweitens setzt QPCA QRAM voraus, um Daten in $O(\log d)$ zu laden. QRAM ist bislang rein theoretisch, ein fundamentaler Existenznachweis statt eines Optimierungsproblems. Quantum Associative Memory [4] als alternativer Ansatz bietet keinen adressbasierten Zugriff auf alle Datenvektoren und ist damit kein direkter QRAM-Ersatz.

Die Schaltkreistiefe ist ein Hardware-Reifegrad-Problem mit VQA als praktikabler Antwort, jedoch bleibt der Datenzugriff über QRAM ein ungelöstes Architekturproblem, für das QAM nur einen Denkansatz bietet.

6 Fazit und Ausblick

QPCA zeigt, wie ein Quantenzustand aktiv an seiner eigenen Analyse teilnehmen und dabei einen exponentiellen Speedup erzielen kann. Die praktische Umsetzung wird derzeit durch die begrenzten Möglichkeiten von NISQ-Hardware sowie das Fehlen eines effizienten QRAM eingeschränkt. Unabhängig davon stellt der Ansatz eine konzeptionelle Inspiration für den Entwurf neuer Quantenalgorithmen dar und dient insbesondere als Vorbild für Self-Analysis-Verfahren.

Literatur

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
2. Cerezo, M., Sharma, K., Arrasmith, A., Coles, P.J.: Variational quantum state eigensolver. npj Quantum Information **8**(1) (2022), ISSN 2056-6387, https://doi.org/10.1038/s41534-022-00611-6, URL http://dx.doi.org/10.1038/s41534-022-00611-6
3. Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nature Physics **10**(9) (2014), ISSN 1745-2481, https://doi.org/10.1038/nphys3029, URL http://dx.doi.org/10.1038/nphys3029
4. Ventura, D., Martinez, T.: Quantum associative memory (1998), URL https://arxiv.org/abs/quant-ph/9807053

Quantum Clustering with Q-K-Means

IAAS

Deciding on appropriate medical treatments is critical to patient health. While there are many traditional diagnostic approaches, new techniques are emerging such as clustering. When applying clustering to this problems we group patients with similar medical data, this allows for data driven treatments where we can utilize past results of patients in the same cluster, therefore with a similar medical history, to decide on effective treatments. To cluster the patients we need clustering algorithms that group data based on their similarity. Classical algorithms scale at least linear with the amount and dimension of the data. Which is a problem considering humans medical data is very complex and numerous with the UKs NHS alone holding an estimated 63 million patient records [4]. With the amount of medically relevant facts we end up with a large high dimensional data set that is expensive to cluster classically. Thus we would like a more efficient method for large clustering problems. One such approach are potentially Quantum-classical hybrid algorithms that compute the process partially on a quantum computer. In this paper we focus on the K-means algorithm with a quantum part, the Q-K-Means. First we review the K-Means algorithm. It can be broken down into several steps. It starts with the pre-processing where the Data is cleaned, we remove outliers, handle missing values and remove duplicated or possibly reduce the dimensionality with PCA, for which we have seen a quantum approach in the last chapter.
Then we initialize k Centroids usually with some minimum distance ϵ to ensure faster convergence.
Next we compute the euclidean distance of each data point to each centroid and assign the data points to the cluster with their closest centroid.
Lastly we recompute the centroids for each cluster by averaging the values of the data points in their corresponding cluster. If they stay the same or change below some threshold δ we terminate, if we have significant centroid changes we reiterate the process with the newly computed centroids.
We can see that the described algorithm runs in O(n*d*I*k) with n being the amount of data, d the dimensions of the data I being the iterations of the algorithm and k the amount of clusters we input. Additionally we usually need to run the algorithm several times to find the optimal k, where the clusters have maximal expression power, clusters aren't to loose but also not to specialized and small too lose generalization. One way to achieve this is the elbow method where we run the algorithm with increasing ks until we see a drop in decrease of the within sum squared distances [3]. The algorithm scales linearly with n and d and we need to run it additional times to find the optimal k. This gives us the motivation to try and improve upon the algorithm. By leveraging quantum computing to achieve a theoretical speedup. We take a look at Q-K-Means. This Q-K-Means works much like the K-Means with the classical part being virtually

identical, however we now want to compute the euclidean distances with help of a quantum computer. To do this we first need to initialize the data on the quantum computer and then execute a quantum circuit that lets us estimate the distance of the data points.
To initialize we need to normalize the data and encode it into a quantum state. Normalization means the squared amplitude sum of a quantum state needs to equal 1. We could normalize all our data vectors at once, by doing that however we could eliminate important distance relationships for clustering, as now the data is present in a more confined space were previously far apart data can be way closer together and get miss classified as distances shrink. In order to avoid this we can use the Inverse stereographic projection [2] which keeps distance relationships better intact. Or we can pairwise normalize for amplitude encoding [1]one of many different encoding patterns [6]. For the purpose of distance calculation we are going to focus on amplitude encoding we will see later why.
Amplitude encoding takes information in vector form and encodes it into the normalized amplitudes of the different quantum states. Notably we can encode an N dimensional Vector in $Log_2(N)$Qubits.

$(1,2,3,4)^T \implies \frac{1}{\sqrt{30}}|00\rangle + \frac{2}{\sqrt{30}}|01\rangle + \frac{3}{\sqrt{30}}|10\rangle + \frac{4}{\sqrt{30}}|11\rangle$
Here we encode the Vector $(1,2,3,4)^T$ with amplitude encoding.
To estimate the euclidean distance between two Vectors on a Quantum Computer, for this we leverage the swap test. By executing the swap test a large amount of times and averaging the results for measuring 0 we then can compute the distance estimation of vectors c and d as follows [1]:

$$|\psi\rangle := \frac{1}{\sqrt{2}}(|0\rangle|c\rangle + |1\rangle|d\rangle) \qquad |\phi\rangle := \frac{1}{\sqrt{Z}}(|c\rangle|0\rangle - |d\rangle|1\rangle)$$

$$Pr(0) = \frac{1}{2} + \frac{1}{2}|\langle\psi|\phi\rangle|^2$$

$$|c-d|^2 = 4Z(Pr(0) - 0.5)$$

With $Z := |a|^2 + |b|^2$. The time complexity of the swap test depends on the number of Qubits in the Vectors it is comparing. Thus amplitude encoding can bring us an exponential speedup when calculating distances [5] as it has a time complexity of $O(Log_2(D))$ over the classical computation which has a complexity of O(D) (Dimensions). This approach however is currently not efficient, since we can only achieve a speedup with data in amplitude encoding which currently needs $O(2^N)$ gates to create the encoding. Thus is unclear whether a real speed-up can be achieved[**?**]
Furthermore current Quantum computers have an error rate for each gate making accurate computing challenging for higher Dimensional data as it requires more gates. Additionally we need to load the data into the quantum computer which has a time complexity of O(n). Further improvements to current NISQ machines, quantum algorithms or potentially other hybrid approaches would be needed to achieve a realistic speedup in clustering with quantum algorithms over classical ones.

Bibliography

1. Stephen DiAdamo, Corey O'Meara, Giorgio Cortiana, and Juan Bernabé-Moreno. Practical quantum k-means clustering: Performance analysis and applications in energy grid classification. *IEEE Transactions on Quantum Engineering*, 3:1–16, 2022.
2. Furkan Semih Dündar. Qubit representation of motion on the 2d plane. 3:1–, 06 2021.
3. GeeksforGeeks. Elbow method for optimal value of k in k-means. `https://www.geeksforgeeks.org/machine-learning/elbow-method-for-optimal-value-of-k-in-kmeans/`, n.d. Accessed: 2026-07-04.
4. NHS England. National care records service (NCRS). `https://digital.nhs.uk/services/national-care-records-service`. Accessed: 2026-07-07.
5. Alessandro Poggiali, Alessandro Berti, Anna Bernasconi, Gianna M. Del Corso, and Riccardo Guidotti. Quantum clustering with k-means: A hybrid approach. *Theoretical Computer Science*, 992:114466, April 2024.
6. Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. Expanding data encoding patterns for quantum algorithms. In *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, pages 95–101, 2021.

Quanten Klassifikation mit Stützvektormaschinen

Orlando Ackermann

Um komplexe Datensätze zu analysieren, bestimmt man zunächst deren Hauptkomponenten, zum Beispiel mit dem bereits vorgestellten Quanten-PCA [7], um im Anschluss zugrundeliegende Strukturen zu identifizieren. Dies kann über Clustering geschehen, sodass mit dem eben vorgestellten Q-K-Means [11] beispielsweise zwei Klassen identifiziert werden. Anstatt für einen neu auftretenden Datenpunkt diese beiden Schritte zu wiederholen, möchte man nun einen Klassifikator trainieren, der neue Datenpunkte günstig in eine der beiden Klassen zuordnen kann.

Ein bewährtes und effektives Klassifikationsverfahren bilden Stützvektormaschinen (SVM) [8]. Gegeben M Trainingsdatenpunkte $(\vec{x}_i, y_i)$ mit $\vec{x}_i \in \mathbb{R}^N$ und Klassen $y_i \in \{\pm 1\}$, ermitteln SVMs eine Hyperebene zwischen beiden Klassen, welche den Abstand zu den Datenpunkten maximiert [12]. Dafür wird mit quadratischer Programmierung [2] eine Zielfunktion maximiert und die Gewichte der Hyperebene bestimmt, welche nur durch wenige, sogenannte Stützvektoren, beeinflusst werden, wie in Abbildung 1a zu sehen ist. Diese Eigenschaft macht das Modell robust gegenüber Ausreißern. Um auch komplexere Daten zu trennen, können diese mit einer Merkmalsfunktion φ in eine höhere Dimension abgebildet werden, in der sie durch eine Hyperebene wieder getrennt werden können, wie in Abbildung 1b dargestellt. Die Kosten belaufen sich dabei auf $\mathcal{O}(M^3)$ für das Bestimmen der Gewichte der Hyperebene, sowie $\mathcal{O}(M^2N)$, um die dafür nötige Kernelmatrix $K = \langle (x_i, x_j) \rangle$ aufzustellen [10]. Insbesondere wird das Bestimmen von K umso aufwändiger, desto größer die Dimension N der Daten ist, sodass für $N > M$ die Berechnung der Kernelmatrix die Kosten dominiert.

Genau an dieser Schwachstelle der klassischen Behandlung hochdimensionaler Räume setzen Quanten-Stützvektormaschinen (QSVM) an. Quantencomputer bieten einen natürlichen Vorteil, um Daten in höhere Dimensionen abzubilden, da ihr Zustandsraum mit der Qubit-Zahl exponentiell wächst. Dadurch können komplexe Kernel entworfen werden, die sich dennoch effizient auswerten lassen [6]. Allerdings treten auch Mehrkosten auf durch klassisches Vor- und Nachbereiten, sowie durch die hohe Anzahl an benötigten Messungen [6, 8]. Wie Tabelle 1 zusammenfasst, versuchen drei quantenbasierte Verfahren diese Vor- und Nachteile auf verschiedene Art und Weise zu bewältigen.

Ein hybrides QSVM-Verfahren nutzt Quanten Kernel-Estimation (QKE) um die Kernelmatrix $K = (\langle x_i, x_j \rangle)_{1 \leq i \leq M}$ zu berechnen [8]. Auch zum Klassifizieren eines neuen Datenpunkts x werden mithilfe des Quantenschaltkreises die Werte $K_i = \langle x_i, x \rangle$ berechnet. Dies spart die Kosten $\mathcal{O}(M^2N)$ des klassischen SVM Problems. Da allerdings die Gewichte der SVM-Hyperebene weiterhin klassisch in $\mathcal{O}(M^3)$ bestimmt werden müssen [6], lohnt sich dieser Ansatz vor allem bei hochdimensionalen Daten, wenn $N \gg M$ gilt.

Ein variationeller Quantenalgorithmus (VQA) basiert ebenfalls auf einer hybriden Architektur: Ein parametrisierter Quantenschaltkreis klassifiziert Trai-

	Klassisch [12]	QKE [6]	VQA [6]	Rein Quantum [10]
Verfahren zur Optimierung	Quadratische Programmierung		Iterativ	HHL-Algorithmus [5]
Kosten (Theorie) (Praxis)	Hoch	Mittel bis hoch	Sehr verschieden	Niedrig Mittel
Hardware	Klassisch	Fehlerhafte QPU (NISQ)		Fehlertolerante QPU
Kernel Trick	Eingeschränkt	Einfach		Ansatzabhängig
Einschränkung	Datenmenge M Dimension N		Viele Hyper-parameter	Daten effizient laden

Tabelle 1: Vergleich der QSVM-Verfahren und klassischen SVM.

ningsdaten und übergibt die Ergebnisse an einen klassischen Optimierer, welcher die Parameter ϑ basierend auf den Ergebnissen und einer Kostenfunktion anpasst. Nach der Konvergenz von ϑ dient der optimierte Schaltkreis direkt zur Klassifikation neuer Datenpunkte [6]. Auch hier ist das Ziel eine lineare Trennung der Daten zu erreichen, allerdings wird nicht unbedingt der Abstand zwischen den Klassen maximiert. Die Laufzeit bis zur Konvergenz der Parameter wird maßgeblich von der Wahl der Kostenfunktion und des klassischen Optimierers beeinflusst, sowie den Startwerten und der Größe des Schaltkreises [3].

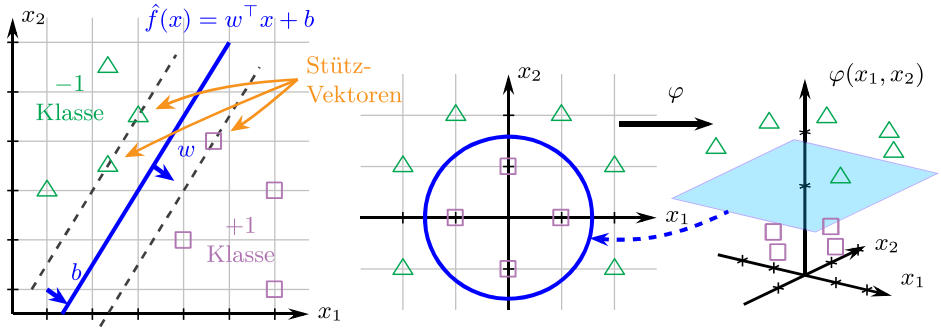
Der in [10] vorgestellte reine Quantenalgorithmus beruht im Gegensatz zu klassischen SVM auf Kleinste-Quadrate-SVM, welche die quadratische Distanz aller Datenpunkte zur Hyperebene minimiert und somit anfälliger gegenüber Ausreißern ist. Durch diese Problemredefinition werden die SVM-Bedingungen zu Gleichungen, sodass die Gewichte über ein lineares Gleichungssystem (LGS) bestimmt werden können. Eine zentrale Voraussetzung für das effiziente Ausführen des Algorithmus ist, dass man Trainingsdaten effizient in eine Superposition laden kann (z.B. über QRAM [4]). Daraus wird die Kernelmatrix K gebildet und benutzt um ein LGS aufzustellen, welches man mit dem HHL-Algorithmus [5] nach dem Zustandsvektor $|b, \alpha\rangle$ der Gewichte der Hyperebene auflöst. Die Klassifikation von einem x erfolgt dann über das Skalarprodukt $|x\rangle$ mit $|b, \alpha\rangle$. Die theoretischen Kosten von $\mathcal{O}(\log(MN))$ sind dabei realistisch nicht umzusetzen: NISQ Maschinen scheitern bereits an der Schaltkreistiefe [9], während selbst bei fehlertoleranten Quantenrechnern das Problem der Datenbereitstellung bestehen würde [1].

Abschließend lässt sich ein Gegensatz zwischen theoretischer Laufzeit und heutiger Anwendbarkeit der verschiedenen Ansätze für Quanten Stützvektormaschinen feststellen. Die reine QSVM skaliert theoretisch optimal, scheitert aber an fehlender Hardware. QKE ist bereits heute nutzbar, verringert die Laufzeit des SVM Problems allerdings nur mittelmäßig. Das variationelle Verfahren bildet einen Mittelweg: Es ist NISQ-geeignet, erfordert aber das Vermeiden karger Hochebenen, Regionen mit verschwindendem Gradienten, welche unter anderem durch einen “Warm-Start”, die Wahl passender Startwerte, umgangen werden können. Dies wird in den nachfolgenden Kapiteln behandelt [3, 13].

Literatur

1. Aaronson, S.: Quantum Machine Learning Algorithms: Read the Fine Print. Nature Physics **11**(4), 291–293 (2015)
2. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
3. Dallinger, M.: Auftreten Karger Hochebenen in variationellen Quantenalgorithmen. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)
4. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum Random Access Memory. Phys. Rev. Lett. **100**, 160501 (Apr 2008)
5. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum Algorithm for Linear Systems of Equations. Physical Review Letters **103**(15) (Oct 2009), ISSN 1079-7114
6. Havlíček, V., Córcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow, J.M., Gambetta, J.M.: Supervised Learning with Quantum-Enhanced Feature Spaces. Nature **567**(7747), 209–212 (Mar 2019), ISSN 1476-4687
7. Iscak, A.: Quantum Principal Component Analysis. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)
8. Leymann, F.: Quanteninformatik 2, Universität Stuttgart (Sommersemester 2025)
9. Preskill, J.: Quantum Computing in the NISQ Era and Beyond. Quantum **2**, 79 (Aug 2018), ISSN 2521-327X
10. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum Support Vector Machine for Big Data Classification. Physical Review Letters **113**(13) (2014), ISSN 1079-7114
11. Seiz, O.: Quantum Clustering with Q-K-Means. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)
12. Staab, S.: Foundations of Machine Learning, Universität Stuttgart (Sommersemester 2025)
13. Volpe, A.: Warm-Start for Quantum Algorithms. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)

Anhang



(a) (b)

Abb. 1: (a) Lösung der klassischen SVM für die gegebenen Daten. (b) Kernel-Trick mit Projektion der Datenpunkte in einen dreidimensionalen Merkmalsraum.

Auftreten Karger Hochebenen in Variationellen Quantenalgorithmen

Moritz Dallinger

In den beiden vorangegangenen Kapiteln wurden quantenbasiertes Clustering mit Q-K-Means [21] sowie Quanten Klassifikation mit Stützvektormaschinen [1] vorgestellt. Diese beiden Verfahren zählen zum Bereich des quantenmaschinellen Lernens und sind auf heutigen NISQ-Maschinen häufig mit variationellen Quantenalgorithmen (VQAs) umgesetzt [5, 13, 20]. Zu den bekanntesten VQAs gehören der Variationelle Quantum Eigensolver (VQE) und der Quantum Approximate Optimization Algorithm (QAOA) [9, 19]. VQAs kombinieren einen parametrisierten Quantenschaltkreis mit einer klassischen Optimierung der Parameter, um durch iterative Änderungen der Parameter eine gegebene Kostenfunktion zu minimieren oder zu maximieren [6].

Die Leistungsfähigkeit dieser Verfahren hängt dabei entscheidend von einer erfolgreichen klassischen Optimierung der Parameter ab. Allerdings können die Gradienten mit zunehmender Anzahl von Qubits exponentiell klein werden, wodurch die Optimierung erheblich erschwert oder sogar unmöglich wird [15, 17]. Dieses Phänomen wird als karge Hochebene, *engl: Barren Plateau*, bezeichnet und stellt eine der zentralen Herausforderungen von VQAs dar [17].

Eine karge Hochebene kann mathematisch dargestellt werden. Wir haben

$$\left\{ \left| \psi(\vec{\theta}) \right\rangle \mid \vec{\theta} \in \Omega \subset_{\text{kompakt}} \mathbb{R}^m \right\}, \qquad C\left(\vec{\theta}\right) = \left\langle \psi(\vec{\theta}) \right| \mathrm{O} \left| \psi(\vec{\theta}) \right\rangle$$

für eine Menge an Parametern $\vec{\theta} = (\theta_1, \ldots, \theta_m)$, die parametrisierte Menge an Zuständen $\psi\left(\vec{\theta}\right)$, eine differenzierbare Kostenfunktion C und eine Observable O. Dann hat C eine karge Hochebene $K \subseteq_{\text{kompakt}} \Omega$ bezüglich eines Parameters θ_j genau dann, wenn

$$\forall \varepsilon > 0 \; \exists \, 0 < b < 1 \; \forall \vec{\theta} \in K : \; p\left(\frac{\partial C}{\partial \theta_j}(\vec{\theta}) \geq \varepsilon \right) \in \mathcal{O}(b^n)$$

für ein Wahrscheinlichkeitsmaß $p : \Omega \to [0, 1]$ gilt. Erkennbar ist, dass die Gradienten der Kostenfunktion mit der Anzahl der Qubits n exponentiell klein werden [16]. Die Kostenfunktion konzentriert sich dabei zunehmend um ihren Erwartungswert [15].

Die Information über die Richtung eines möglichen Abstiegs der Kostenfunktion geht hierbei durch verschwindende Gradienten verloren [17]. Für klassische Optimierungsverfahren bedeutet dies, dass nur sehr kleine Änderungen der Parameter stattfinden können [17]. Dies stellt ein erhebliches Problem dar, da ein globales oder auch lokales Minimum oder Maximum eventuell nicht mehr gefunden werden kann [15, 17].

Aus der Definition wird deutlich, dass eine karge Hochebene keinerlei Einschränkung hinsichtlich der geometrischen Form der Kostenfunktion voraussetzt.

Sie charakterisiert ausschließlich das Verhalten der Gradienten innerhalb der betrachteten kargen Hochebene. Insbesondere lassen sich mithilfe glatter Bump-Funktionen C^∞-Kostenfunktionen konstruieren, deren karge Hochebenen eine beliebige Gestalt annehmen und auch mehrere lokale Minima und Maxima besitzen können [3, 4, 16].

Desweiteren kann bei kargen Hochebenen auch zwischen probabilistischen und deterministischen kargen Hochebenen unterschieden werden [15]. Bei probabilistischen kargen Hochebenen konzentriert sich die Kostenfunktion zwar zunehmend um ihren Erwartungswert, aber dies bedeutet nicht, dass die Kostenlandschaft überall flach ist [15, 17]. Es existieren Bereiche innerhalb einer probabilistischen kargen Hochebene, die ausreichend große Gradienten besitzen, die eine erfolgreiche Optimierung ermöglichen können [2]. Diese Bereiche besitzen jedoch ein exponentiell kleines Volumen und werden als enge Schluchten, *engl: narrow gorges*, bezeichnet [2, 7].

Im Gegensatz zu probabilistischen kargen Hochebenen konzentriert sich die Kostenfunktion bei deterministischen kargen Hochebenen für alle Parameterwerte um einen konstanten Wert [8, 17]. Die Kostenfunktion besitzt unabhängig von den Parametern exponentiell kleine Gradienten [15]. Eine solche Kostenlandschaft enthält keine Bereiche mit ausreichend großen Gradienten, wodurch eine klassische Optmierung keine ausreichenden Informationen über die Richtung einer Parameteränderung erhält [15]. Es entstehen also keine engen Schluchten [15].

Die Entstehung karger Hochebenen wird durch verschiedene Eigenschaften des Variationsansatzes und der Kostenfunktion begünstigt [15]. Die exponentiell wachsende Dimension des Hilbertraums bildet die Grundlage für das Auftreten exponentiell kleiner Gradienten [6, 8, 15]. Tiefe und stark parametrisierte Quantenschaltkreise verstärken diesen Effekt, da sie große Bereiche des Hilbertraums erkunden [8, 10, 17]. Auch ungeeignete Anfangszustände, insbesondere stark verschränkte Zustände, sowie ungeeignet gewählte Observablen können karge Hochebenen verursachen [8]. Darüber hinaus können Rauschen und Dekohärenz auf heutigen NISQ-Maschinen deren Entstehung begünstigen [24].

Eine wirksame Strategie zur Vermeidung karger Hochebenen besteht in der Verwendung flacher Quantenschaltkreise [15]. Für lokale Kostenfunktionen treten hierbei keine kargen Hochebenen auf [7], wohingegen bei globalen Kostenfunktionen weiterhin kargen Hochebenen auftreten können [7, 17].

Desweiteren können weniger expressive, symmetrieerhaltende Ansätze mit kleinen dynamischen Lie Algebren das Auftreten karger Hochebenen reduzieren [14, 15, 18]. Außerdem können adaptive Ansätze, wie ADAPT-VQE, wobei neue Gatter nur dann zum Ansatz hinzugefügt werden, wenn sie ausreichend große Gradienten liefern, das Auftreten karger Hochebenen reduzieren [11, 12].

Zudem verringern lokale Observablen das Risiko karger Hochebenen gegenüber globalen Observablen, können diese jedoch nicht grundsätzlich ausschließen, wenn der Quantenschaltkreis nicht flach ist [17]. Weitere Vermeidungsstrategieen umfassen verschiedene Warm-Start-Methoden [15, 22], doch dazu mehr im nächsten Kapitel *Warm-Start for Quantum Algorithms* [23].

Quellen

1. Ackermann, O.: Quanten Klassifikation mit Stützvektormaschinen. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)
2. Arrasmith, A., Holmes, Z., Cerezo, M., Coles, P.J.: Equivalence of quantum barren plateaus to cost concentration and narrow gorges. Quantum Science & Technology **7**(4), 045015 (2022)
3. Bröcker, T., Jänich, K.: Einführung in die Differentialtopologie. (No Title) (1973)
4. Bröcker, T., Lander, L.: Differentiable germs and catastrophes, vol. 17. Cambridge University Press (1975)
5. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., et al.: Variational quantum algorithms. Nature Reviews Physics **3**(9), 625–644 (2021)
6. Cerezo, M., Larocca, M., García-Martín, D., Diaz, N.L., Braccia, P., Fontana, E., Rudolph, M.S., Bermejo, P., Ijaz, A., Thanasilp, S., et al.: Does provable absence of barren plateaus imply classical simulability? Nature Communications **16**(1), 7907 (2025)
7. Cerezo, M., Sone, A., Volkoff, T., Cincio, L., Coles, P.J.: Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nature communications **12**(1), 1791 (2021)
8. Diaz, N., García-Martín, D., Kazi, S., Larocca, M., Cerezo, M.: Showcasing a barren plateau theory beyond the dynamical lie algebra. arXiv preprint arXiv:2310.11505 (2023)
9. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)
10. Fontana, E., Herman, D., Chakrabarti, S., Kumar, N., Yalovetzky, R., Heredge, J., Sureshbabu, S.H., Pistoia, M.: Characterizing barren plateaus in quantum ansätze with the adjoint representation. Nature Communications **15**(1), 7171 (2024)
11. Grimsley, H.R., Barron, G.S., Barnes, E., Economou, S.E., Mayhall, N.J.: Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus. npj Quantum Information **9**(1), 19 (2023)
12. Grimsley, H.R., Economou, S.E., Barnes, E., Mayhall, N.J.: An adaptive variational algorithm for exact molecular simulations on a quantum computer. Nature communications **10**(1), 3007 (2019)
13. Kerenidis, I., Landman, J., Luongo, A., Prakash, A.: q-means: A quantum algorithm for unsupervised machine learning. Advances in neural information processing systems **32** (2019)
14. Larocca, M., Czarnik, P., Sharma, K., Muraleedharan, G., Coles, P.J., Cerezo, M.: Diagnosing barren plateaus with tools from quantum optimal control. Quantum **6**, 824 (2022)
15. Larocca, M., Thanasilp, S., Wang, S., Sharma, K., Biamonte, J., Coles, P.J., Cincio, L., McClean, J.R., Holmes, Z., Cerezo, M.: Barren plateaus in variational quantum computing. Nature Reviews Physics **7**(4), 174–189 (2025)
16. Leymann, F.: Vorlesung: Quanteninformatik 1 (2025)
17. McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes. Nature communications **9**(1), 4812 (2018)
18. Meyer, J.J., Mularski, M., Gil-Fuster, E., Mele, A.A., Arzani, F., Wilms, A., Eisert, J.: Exploiting symmetry in variational quantum machine learning. PRX quantum **4**(1), 010328 (2023)

19. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.H., Zhou, X.Q., Love, P.J., Aspuru-Guzik, A., O'brien, J.L.: A variational eigenvalue solver on a photonic quantum processor. Nature communications **5**(1), 4213 (2014)
20. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Physical review letters **113**(13), 130503 (2014)
21. Seiz, O.: Quantum Clustering with Q-K-Means. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)
22. Truger, F., Barzen, J., Bechtold, M., Beisel, M., Leymann, F., Mandl, A., Yussupov, V.: Warm-starting and quantum computing: A systematic mapping study. ACM Computing Surveys **56**(9), 1–31 (2024)
23. Volpe, A.: Warm-Start for Quantum Algorithms. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026)
24. Wang, S., Fontana, E., Cerezo, M., Sharma, K., Sone, A., Cincio, L., Coles, P.J.: Noise-induced barren plateaus in variational quantum algorithms. Nature communications **12**(1), 6961 (2021)

Alle Links wurden zuletzt am 8. Juli 2026 geprüft.

Warm-Start for Quantum Algorithms

Anna Volpe (st177704@stud.uni-stuttgart.de)

Variational Quantum Algorithms (VQAs) play an important role in quantum computing and combine quantum and classical computation in an iterative process [1,2]. As discussed in the previous chapter [3], their optimization is affected by cost landscapes that can contain barren plateaus, i.e., flat areas [4].

One approach to guide the optimization process is *Warm-Start (WS)*, which uses known or efficiently generated information instead of executing the target algorithm without problem-specific prior information [5]. Different WS techniques can be applied depending on the algorithm and the specific problem [6]. We focus on the Biased Initial State pattern and compare *Quantum Approximate Optimization Algorithm (QAOA)* with WS-QAOA for MaxCut.

MaxCut asks for a partition of the vertices of a graph into two subsets such that the total weight of the edges connecting different subsets is maximized [7,8]. In QAOA, each vertex is represented by one qubit, and a measurement in the computational basis produces a bitstring representing one possible cut. The weighted MaxCut objective can be expressed by the cost Hamiltonian [7]:

$$H_C = \sum_{(i,j)\in E} \frac{w_{ij}}{2}\left(1 - Z_i Z_j\right).$$

For each edge, the corresponding term distinguishes whether the connected vertices belong to the same or to different subsets. QAOA starts from the uniform superposition $|+\rangle^{\otimes n}$ and alternates cost and mixer operators for p layers, while the parameters γ and β are optimized classically [9]. The two-qubit terms of the cost Hamiltonian couple qubits associated with adjacent vertices through ZZ interactions [7,9].

Efficient classical approximation algorithms such as Goemans-Williamson algorithm [8] are known for MaxCut. The Goemans-Williamson algorithm achieves an expected approximation ratio of about 0.878 for instances with non-negative edge weights [8]. QAOA alternates problem-dependent phase operations with mixing operations to prepare a parameterized distribution over candidate cuts [9]. QAOA lacks general performance guarantees, and for certain MaxCut instances, constant-depth QAOA cannot outperform Goemans-Williamson rounding [7]. Since classical approximation algorithms can already provide good solutions for MaxCut, this information can be used to initialize QAOA instead of starting without problem-specific prior information [5,7]. This idea leads to WS-QAOA.

Egger et al. [7] propose a WS-QAOA approach in which information from classical preprocessing is used in the quantum optimization. The continuous variant encodes a relaxed solution directly into the initial state, whereas the MaxCut variant uses a rounded classical cut [7]. In the continuous construction, a binary variable

$x_i \in \{0,1\}$ is relaxed to $c_i \in [0,1]$. Let c^* denote an optimal relaxed solution. The value c_i^* is encoded into qubit i using

$$\theta_i = 2\arcsin\sqrt{c_i^*}.$$

Applying $R_Y(\theta_i)$ to $|0\rangle$ prepares a state in which the probability of measuring $|1\rangle$ is c_i^* [7]. The initial state therefore reflects the relaxed solution instead of assigning equal probabilities to all computational basis states.
Since the QAOA mixer is not designed to perform selective phase changes with respect to the Biased Initial State, it must be adapted accordingly. In continuous WS-QAOA, the mixer Hamiltonian is chosen such that the Biased Initial State is its ground state [7], as shown in Figure 1.

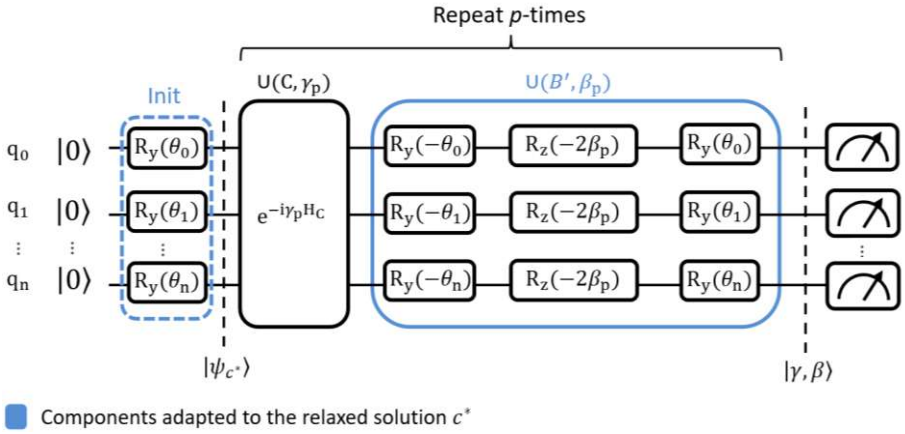


Figure 1. Circuit structure of WS-QAOA. The initial state and mixer operator are adapted to the relaxed solution c^*. Adapted from [7].

For MaxCut, Egger et al. [7] also consider semidefinite programming combined with Goemans-Williamson randomized rounding [7]. The classical procedure first produces a binary cut used to initialize the circuit. A regularization parameter prevents the qubits from being fixed to exact computational basis states and allows the circuit to represent different cuts. The classical approximation therefore provides an initial solution, while the variational circuit can explore alternatives [7].

WS uses prior information instead of starting an algorithm from scratch. Reported goals include reduced optimization time, lower resource consumption, and improved solution quality, although the observed effects depend on the algorithm and problem instance [5]. Some WS techniques, such as the one presented in this paper, aim to avoid or mitigate barren plateaus [5,6]. In the future, such knowledge could be used to provide decision support for selecting suitable WS techniques.

Disclaimer Grammarly was used during the writing of this paper to correct spelling, punctuation, and grammar.

References

1. Preskill, J.: Quantum Computing in the NISQ era and beyond. Quantum 2, 79 (2018).
2. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., Coles, P.J.: Variational Quantum Algorithms. Nature Reviews Physics 3, 625-644 (2021).
3. Dallinger, M.: Auftreten Karger Hochebenen in variationellen Quantenalgorithmen. Hauptseminar, Ausgewählte Themen der Quanteninformatik (2026).
4. McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes. Nature Communications 9, 4812 (2018).
5. Truger, F., Barzen, J., Bechtold, M., Beisel, M., Leymann, F., Mandl, A., Yussupov, V.: Warm-Starting and Quantum Computing: A Systematic Mapping Study. ACM Computing Surveys 56(9), 1-31 (2024).
6. Truger, F., Barzen, J., Beisel, M., Leymann, F., Yussupov, V.: Warm Starting Patterns for Quantum Algorithms. (2024).
7. Egger, D.J., Mareček, J., Woerner, S.: Warm-starting quantum optimization. Quantum 5, 479 (2021).
8. Goemans, M.X., Williamson, D.P.: Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. Journal of the ACM 42(6), 1115-1145 (1995).
9. Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm. (2014).

All links were last followed on July 08, 2026.