

# Abschlussbericht

zum Teilvorhaben „Quanten-Machine-Learning für Klassifikation und Regression“

im Verbundprojekt „Quantum Artificial Intelligence for the Automotive Industry (Q(AI)2)“

## Zuwendungsempfänger:

Mercedes-Benz AG  
Dr. Markus Leder  
HPC L423  
Mercedesstraße 120  
70372 Stuttgart

**Förderkennzeichen des Teilvorhabens:** 13N15582

**Laufzeit des Vorhabens:** 01.04.2021 bis 31.03.2024

**Berichtszeitraum:** 01.04.2021 bis 31.03.2024

Das Ziel des vorliegenden Teilvorhabens bestand einerseits darin, Fortschritte bei Quanten-Klassifikations- und -Regressionsalgorithmen zu erzielen (AP 3). Andererseits wurde angestrebt, Verbesserungen von Quantenoptimierungsalgorithmen, insbesondere des QAOA-Algorithmus zu erarbeiten (AP 5).

Im Folgenden soll zunächst auf den erstgenannten Teilbereich eingegangen werden, welcher sich mit Quanten-Klassifikations- und -Regressionsalgorithmen befasst.

Quanten-Klassifikations- und -Regressionsalgorithmen (Ausführung: Markus Leder)

## Anwendungsfall

Das vorliegende TVH trägt als Industriepartner gemäß der TVB eine der wirtschaftlich bedeutsamen Anwendungsmöglichkeiten für die im Rahmen des Projekts entwickelten Algorithmen bei. Der eingebrachte Anwendungsfall namens "Deep Mesh" (Beitrag zu AP 1) bezieht sich auf den digitalen Prozess der Produktentwicklung, bei dem in vielen Fällen die Finite-Elemente-Methode (FEM) zur numerischen Lösung partieller Differentialgleichungen eingesetzt wird. Dabei werden zu berechnende Bauteile als Gitter (auch "Mesh" genannt) aus geometrischen Elementen dargestellt. Eine wesentliche Voraussetzung für die Erzielung guter Ergebnisse besteht darin, dass die Gitter für das jeweilige Bauteil und den jeweiligen Anwendungszweck optimiert werden (z.B. Crashsimulationen, Umformsimulationen für die Produktion). Hierbei kommen verschiedene Algorithmen zum Einsatz, die auf industrielle Anwendungen angepasst sind. Jedoch liefern diese nicht immer an allen Stellen eines bestimmten Bauteils und für alle Einsatzzwecke der verschiedenen Simulationsdisziplinen optimale Ergebnisse. Nach Durchführung einer FEM-Berechnung kann es somit vorkommen, dass die Ergebnisse nicht mit denen des realen Prozesses (Umformung oder Crashtest) übereinstimmen. In diesen Fällen muss das Gitter manuell angepasst und die FEM-

Berechnung erneut durchgeführt werden, was ein zeitraubender und kostentreibender Prozess ist. Wünschenswert wäre daher eine frühzeitige und automatisierte Qualitätsbeurteilung der FEM-Gitter, sodass problematische Stellen noch vor der eigentlichen FEM-Simulation geeignet umgearbeitet werden können. Daher wird versucht, mithilfe maschinellen Lernens die einzelnen Elemente von FEM-Gittern zu beurteilen. Erfahrene Ingenieure haben in der Vergangenheit beanstandete Stellen bisheriger FEM-Gitter markiert und so eine Datenbank von Gittern erstellt, deren Elemente als "gut" oder "schlecht" gekennzeichnet sind (Abbildung 1).

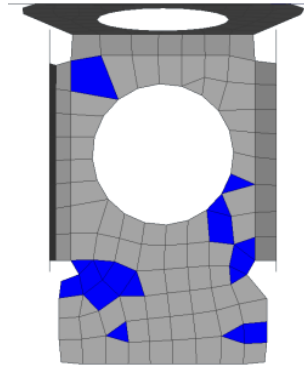


Abbildung 1: FEM-Gitter eines kleinen Fahrzeugbauteils. Die blauen Elemente zeigen Stellen in diesem automatisch erstellten Gitter, die von Ingenieuren zur manuellen Nacharbeit markiert wurden. Künftig soll diese Markierung von einem (Quanten-)Machine-Learning-Modell vorgenommen werden.

Die Datenbank ist in dieser Form allerdings noch nicht als Trainingsgrundlage für ein Machine-Learning-Modell geeignet. Dies liegt daran, dass die Qualität eines FEM-Elements maßgeblich von den benachbarten Elementen beeinflusst wird und die Anzahl der an ein Element angrenzenden Elemente variiert, was zu Merkmalsvektoren unterschiedlicher Länge führt. Dies stellt für viele Machine-Learning-Algorithmen ein Problem dar. Aus diesem Grund wird wie folgt vorgegangen [SD21]: Für jedes Element werden die "Ringe" von Elementen mit jeweils fixem Nachbarschafts-Abstand bestimmt (Abbildung 2).

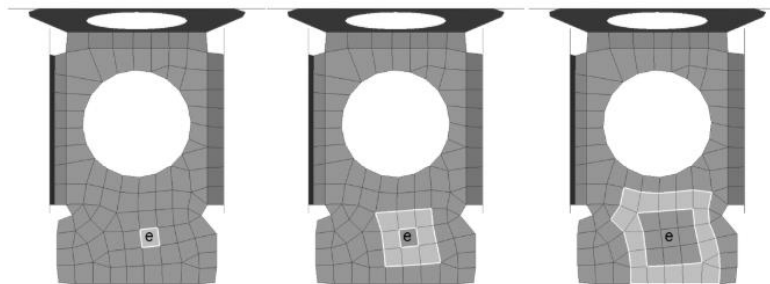


Abbildung 2: Ringe um das Element „e“ mit Nachbarschafts-Abstand 0, 1 und 2.

Anschließend werden für die Elemente, die zu diesem jeweiligen Ring gehören, verschiedene geometrische Eigenschaften der Einzelemente (z.B. Seitenverhältnis, Krümmung etc.) auf unterschiedliche Weisen aggregiert (Maximums-, Minimumsbildung,

arithmetischer Mittelwert). Dadurch entsteht für jedes Element ein Feature-Vektor, dessen Features durch den Abstand zum betrachteten Element, durch die Art der geometrischen Eigenschaft und durch die Art der Aggregation bestimmt sind. Somit kann jedem Element ein Feature-Vektor derselben Länge zugeordnet werden. Es können unterschiedliche Modelle trainiert werden, bei denen die Länge des Featurevektors durch einfache Beschränkung auf bestimmte Werte der drei genannten Merkmale (Abstand, Eigenschaft, Aggregation) flexibel angepasst werden kann, je nach gewünschter Balance zwischen numerischem Trainingsaufwand und Güte des Modells. Insbesondere der Abstand eignet sich hervorragend, um die Menge an Features zu beschränken, da hier keine große Gefahr besteht, eventuell bedeutsame Features ungewollt außer Betracht zu lassen, da es plausibel ist, dass die Elemente mit geringem Abstand zum jeweils betrachteten Element einen höheren Einfluss auf seine Güte haben als Elemente mit größerem Abstand. Diese Eigenschaft des betrachteten Anwendungsfalls macht ihn besonders geeignet für die Anwendung von Quantenalgorithmien zum maschinellen Lernen, da aufgrund der Beschränkungen, denen man bei der Verwendung aktueller Quantenrechner noch unterworfen ist, eine sinnvolle Reduktion der Features vonnöten ist.

## Algorithmenentwicklung

Aufgrund einer Recherche nach geeigneten Algorithmen gemäß der TVB als Ausgangspunkt für die Entwicklung von verbesserten Algorithmen zur Klassifikation von Datensätzen wurden Quanten-Kernel-Methoden und darauf aufbauende Quanten-Support-Vector-Machines (QSVM) ausgewählt. Obwohl Quanten-Neuronale-Netzwerke (QNN) eine Alternative darstellten, fiel die Entscheidung auf QSVM aufgrund der besseren theoretischen Untersuchungen, die in der Literatur zu Kernel-Methoden durchgeführt wurden, und der Konvergenzgarantie beim Training von QSVM, da es sich um konvexe Optimierungsprobleme handelt. [S21].

Die hauptsächlich verwendete Methode war Quantum Kernel Estimation (QKE) [HC+19]. Hierbei wird ein Quantenschaltkreis entwickelt, der den Merkmalsvektor eines Datenpunkts in einen Quantenzustand kodiert, indem der Schaltkreis auf den Ausgangszustand mit allen Qubits im Zustand Null wirkt. Ein Kernel-Element wird als das Quadrat des Betrags des Skalarprodukts zweier solcher Zustände für im allgemeinen unterschiedliche Merkmalsvektoren definiert, das auf einem Quantencomputer ausgewertet werden kann. Der Quantenalgorithmus trägt somit lediglich zur Bereitstellung des Kernels bei, während das Training des Modells wie bei der klassischen SVM auf einem klassischen Rechner erfolgt.

In einem explorativen Forschungsansatz wurde zunächst die Bedeutung der Quantenverschränkung bei der Kernel-Berechnung für die Fähigkeit der QSVM, einen Datensatz zu lernen, eingehend untersucht. Da die Verschränkung eine rein quantenmechanische Eigenschaft ist, erschien sie als besonders geeigneter Ansatzpunkt, um einen Vorteil gegenüber klassischen Kernen zu zeigen. Hierfür wurde ein Datensatz mit zwei Features betrachtet, wobei zu seiner Erzeugung ein Kernel mit einem Qubit pro Feature verwendet wurde, welcher die beiden Qubits verschränkt. Die Ergebnisse zeigten, dass sich SVMs mit klassischen Kernen (wie dem Radial-Basis-Function-(RBF)-Kernel oder dem Cosinus-Kernel) mit etwa demselben Erfolg wie SVMs mit Quanten-Kernen, die die beiden Qubits nicht verschränken, auf diese Daten trainieren lassen. Dies entsprach der Erwartung, da die typisch quantenmechanische Eigenschaft der Verschränkung nicht vorlag. Jedoch zeigte sich, dass Quanten-Kernel, die zwar die Qubits verschränken, sich

aber von dem Kernel unterscheiden, mit dem die Daten erzeugt wurden, demgegenüber nicht zu signifikanten Verbesserungen führten. Diese Erkenntnisse verdeutlichen, dass die Verschränkung alleine kein Garant für eine hohe Klassifikationsgüte einer QSVM ist. Es wurde zudem festgestellt, dass durch eine Erhöhung der Menge an Trainingsdaten, die mit dem verschränkenden Quantenkernel erzeugt werden, eine SVM mit einem klassischen RBF-Kernel eine ebenso gute Klassifikationsgüte erreichen kann wie eine SVM mit dem datenerzeugenden Quantenkernel selbst. Diese Ergebnisse unterstreichen die Bedeutung von umfangreichen Trainingsdaten und zeigen, dass die Verschränkung und damit die Quanteneigenschaften bei der Kernel-Berechnung nicht zwangsläufig eine entscheidende Rolle spielen. Diese Beobachtungen setzten sich ähnlich auf den industriellen Datensätzen fort.

Weiterhin wurden QSVM anhand einiger zweidimensionaler Standarddatensätze untersucht. Dabei wurde festgestellt, dass die Klassifikationsgüte bei den QSVM stark von der Skalierung der Daten abhängt, im Gegensatz zur SVM mit RBF-Kernel. Zudem zeigte sich, dass die SVM auf den Standarddatensätzen generell eine bessere Güte aufweist als die QSVM. Interessanterweise klassifizierten QSVM mit unverschränktem Codier-Schaltkreis, der die Kernelfunktion bestimmt, besser als solche mit verschränktem. Diese Ergebnisse stellen erneut die Bedeutung der Verschränkung in Frage.

Um über die Ergebnisse dieser explorativen Forschung hinaus systematisch nach neuen Algorithmen zu suchen, die die Kernelfunktionen für die QSVM erzeugen, wurde ein Generator eingesetzt, der vom Projektpartner Jülich Supercomputing Centre (JSC) des Forschungszentrums Jülich (FZJ) entwickelt wurde. Damit ließen sich unter Vorgabe diverser Parameter Quantenschaltkreise erzeugen, die den jeweiligen Merkmalsvektor in einen Quantenzustand kodieren.

Diese Parameter sind die Anzahl der zu verwendenden Qubits, die Typen von Quantengattern, die jeweils zum Aufbau einer algorithmischen Schicht verwendet werden, die Anzahl dieser Schichten, die hintereinander ausgeführt werden, sowie ein Skalierungskoeffizient für den gesamten Merkmalsvektor, mit dem er vor seiner Kodierung multipliziert wird (Abbildung 3). Als Quantengatter wurden verschiedene daten-kodierende Gattermuster verwendet, die von [SJA19] inspiriert waren. Hiermit konnten zahlreiche verschiedene Quantenkernel nach Bedarf erzeugt werden. Auf den Standarddatensätzen konnten jedoch trotz der großen verfügbaren Varianz an Kodierungsschaltkreisen keine signifikanten Vorteile gegenüber klassischen Referenzimplementierungen festgestellt werden.

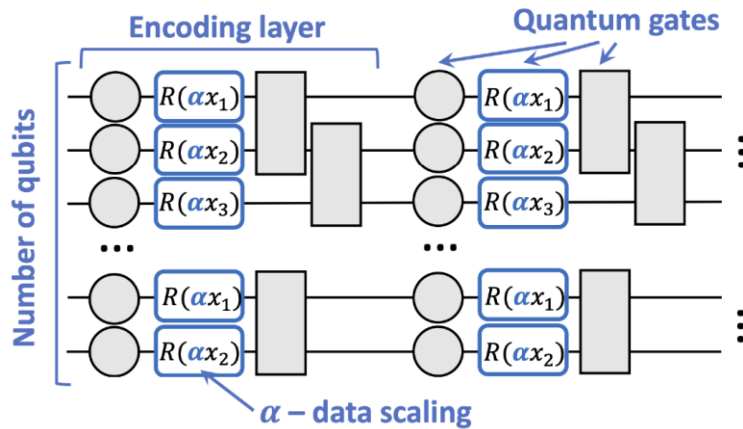


Abbildung 3: Datenkodierender Quantenschaltkreis. Blau dargestellt sind die Elemente, die kombinatorisch variiert wurden, um eine Fülle unterschiedlicher Schaltkreise und damit Kernel zu erzeugen.

Als zweite Kernel-Methode betrachteten wir neben der reinen QKE auch die "Projected Quantum Kernel Estimation" (pQKE) [HB+21]. Bei der reinen QKE wird der Merkmalsvektor in einen Quantenzustand eines Hilbertraums eingebettet, dessen Dimensionalität mit der Anzahl der Qubits exponentiell wächst. Dies führt dazu, dass das Skalarprodukt zweier zufällig gewählter Zustände nahezu null ist, da die einzelnen Einträge des Zustandsvektors sich nach Länge und Richtung in der komplexen Zahlenebene nahezu ausgleichen. Somit sind die meisten Datenpunkte im Zustandsraum orthogonal zueinander, sofern keine besondere Einbettung gewählt wird. Damit liegt der Quantenkernel nahe an der Einheitsmatrix und wird daher unbrauchbar. Um den Klassifikationserfolg nicht zu beeinträchtigen, wird bei der pQKE deshalb der Merkmalsvektor in einen Quantenzustand im hochdimensionalen Hilbertraum kodiert und anschließend mittels speziell ausgewählter Projektionsoperatoren in einen niederdimensionalen Teilraum projiziert, um die Expressivität der Kodierung zu reduzieren, die den Klassifikationserfolg gefährden kann [TW+22].

Schließlich wurde als dritte Methode das Quantum Kernel Training (QKT) angewendet [GG+21]. Bei QKT, basierend auf der analogen klassischen Methode der Kernel-Target-Alignment [CK+06], wird der den Kernel definierende Schaltkreis (also der Schaltkreis, der den Merkmalsvektor in einen Quantenzustand kodiert) mit zusätzlichen Gattern ausgestattet, die von trainierbaren Parametern abhängen. Durch das Training des Kernels selbst wird die Kernelfunktion zunächst an die vorhandenen Trainingsdaten angepasst, bevor sie zum Training der SVM verwendet wird. Da der Kernel als Metrik im Merkmalsraum fungiert, bedeutet dies, dass die einzelnen Datenpunkte bezüglich dieser Metrik durch geeignetes Training des Kernels in zwei idealerweise zueinander nahezu orthogonale Gruppen aufgeteilt werden, ihrem binären Label entsprechend, soweit dies durch das Training der Kernelparameter möglich ist, während die Datenpunkte mit demselben Label ein von Null möglichst weit entferntes Skalarprodukt besitzen sollten.

Bei diesen gemäß den Arbeitszielen der TVB durchgeführten Funktionstests auf den Standarddatensätzen konnten die erzeugten Kernel allerdings weder durch pQKE noch durch QKT signifikante Vorteile in der Klassifikationsgüte gegenüber einer klassischen SVM mit einem RBF-Kernel erzielen.

## Anwendung auf industriellen Datensätzen

Auf diversen Standarddatensätzen konnten die entwickelten Algorithmen keine Vorteile zeigen. Dies muss allerdings nicht bedeuten, dass sie nicht auf anderen Daten besser abschneiden können. Daher wurden die Kodierschaltkreise auf den eigentlichen Zieldatensätzen dieses TVH, den Daten des Anwendungsfalles „Deep Mesh“ getestet.

Die Daten bestehen aus einzelnen Datenpunkten, wobei jeder Datenpunkt einem der finiten Elemente entspricht. Jeder Datenpunkt besteht aus einem Merkmalsvektor identischer Länge. Die Gesamtheit der Daten wurde in einzelne Datensätze unterteilt, wobei diese Unterteilung insofern „natürlich“ ist, als jeder Datensatz einem der berechneten Bauteile entspricht, wobei die Anzahl der Datenpunkte pro Datensatz je nach Größe der Bauteile sehr unterschiedlich sein kann. Zunächst wurden die Datenpunkte eines Datensatzes in der Datenvorverarbeitung einer Min-Max-Skalierung unterzogen, um ihre Merkmale einheitlich auf den Wertebereich  $[-1, 1]$  zu skalieren. Anschließend wurde der Datensatz in eine Trainingsmenge und eine Testmenge aufgeteilt, wobei die Datenpunkte zufällig ausgewählt wurden. Die Trainingsmenge umfasst 80% der Datenpunkte, während die Testmenge 20% der Datenpunkte enthält.

Der Klassifikator wurde auf der Trainingsmenge trainiert und seine Leistung wurde anhand der Testmenge bewertet. Die Gesamtzahl der Datenpunkte betrug ca. 6000, um eine handhabbare Berechnungsdauer auf einem Simulator zu gewährleisten. Es wurde eine kombinatorische Hyperparametersuche auf einem Gitter nach den oben genannten Parametern durchgeführt. Das bedeutet, es wurden bestimmte Wertemengen für die Hyperparameter festgelegt, die dann in allen Kombinationen durchlaufen wurden. Für jede dieser Kombinationen wurde ein stratifiziertes dreifaches Kreuzvalidierungs-Training durchgeführt, um eine Überanpassung des Modells an die Trainingsdaten zu verhindern („Overfitting“). Dabei wurde die Trainingsmenge zufällig in drei gleichgroße Teilmengen unterteilt, die so gewählt sind, dass die Verteilung der Labels („gut“ und „schlecht“) in jeder der drei Teilmengen nach Möglichkeit jener der gesamten Trainingsmenge entspricht („stratifiziert“). Sodann wurden drei Klassifikatoren (hier QSVM) trainiert, wobei für alle die identische, gerade betrachtete Kombination an Hyperparametern verwendet wurde. Jeder Klassifikator wurde auf zweien der drei Teilmengen trainiert, wobei die dritte Teilmenge zur Validierung verwendet wurde, d.h., dort wurde die Leistungsfähigkeit des Klassifikators gemessen. Als Metrik für die Leistungsfähigkeit wurde die balancierte Genauigkeit eingesetzt, da die beiden Klassen sehr ungleiche Größe haben. (Die Klasse der „guten“ finiten Elemente ist selbstredend in aller Regel deutlich größer als die der „schlechten“. Andernfalls wäre die bisher verwendete Software zur Erstellung der FEM-Netze unbrauchbar.) Die balancierte Genauigkeit ist dabei das über beide Klassen genommene arithmetische Mittel des Anteils der korrekt eingestufteten Datenpunkte für die jeweilige Klasse an den in diese Klasse eingestufteten Datenpunkten. Über die drei balancierten Genauigkeiten wurde wiederum das arithmetische Mittel genommen. Diese Zahl wurde als Maß für die Güte der jeweiligen Hyperparameterkombination verwendet. Die beste Kombination wurde schließlich verwendet, um den Klassifikator auf der gesamten Trainingsmenge (80%) des betreffenden Datensatzes zu trainieren und auf der Testmenge (20%) die Klassifikationsgüte (ebenfalls balancierte Genauigkeit) zu bestimmen. Dieses Trainings- und Testprotokoll ist in dieser oder ähnlicher Form das übliche im Bereich des maschinellen Lernens.

Bei den Methoden pQKE und QKT wurde das gleiche Trainings- und Testprotokoll wie bei QKE verwendet. Bei der pQKE-Methode wurden für die Projektion der Zustände die Pauli-Operatoren X, Y und Z eingesetzt. Beim Quanten Kernel Training (QKT), das sehr ressourcenintensiv ist, wurde vor den besten, in der Kreuzvalidierung ermittelten kodierenden Schaltkreis des Datensatzes eine Lage aus Ry-Gattern eingefügt, die alle denselben trainierbaren Parameter enthalten. Beim Training wurden der Kernel selbst und die darauf aufbauende QSVM gleichzeitig trainiert, entsprechend [GG+21].

Tabelle 1 zeigt den Mittelwert und die Standardabweichung der balancierten Genauigkeiten, die über die verschiedenen Datensätze hinweg auf die oben beschriebene Weise mit den Methoden QKE, pQKE und QKT ermittelt wurden. Zum Vergleich werden auch die Ergebnisse des klassischen RBF-Kernels angegeben, bei dem die Gittersuche über den Kernel-Parameter (Weite der RBF) und einen SVM-Regularisierungsparameter C durchgeführt wurde. Die Ergebnisse zeigen, dass alle drei Quantenmethoden etwa auf dem gleichen Niveau wie die klassische Referenz-SVM liegen, unabhängig von den jeweils besten kodierenden Quantenschaltkreisen (die bei den verschiedenen Datensätzen in der Regel unterschiedlich waren) und anderen Parametern, jedoch mit einer hohen Standardabweichung. Bemerkenswert ist, dass dies auch für die Ergebnisse des QKT gilt. Die Ergebnisse des QKT können (abgesehen von statistischen Schwankungen beim Training der SVM durch die Datenaufteilung) im Vergleich zur QKE lediglich besser werden, da die Wahl des Winkels Null in den Ry-Gattern wieder auf den Ausgangskernel der QKE zurückführt. Die Tatsache, dass durch das Training kein verbesserter Kernel gefunden wird, legt nahe, dass die Möglichkeiten der SVM für diese Datensätze ausgereizt sind. Dies umso mehr, als die Ergebnisse der QKE bereits aus einem Kombinations- und Selektionsprozess hervorgegangen sind, aber im Vergleich zur besten klassischen RBF-SVM dennoch keine Verbesserung gefunden werden konnte.

use case	Balanced Accuracy (averaged)			
	QKE	pQKE	QKT	RBF
FEM	$0.73 \pm 0.13$	$0.73 \pm 0.13$	$0.72 \pm 0.14$	$0.75 \pm 0.14$

*Tabelle 1: Ergebnisse (über die betrachteten Bauteile gemittelte ausgeglichene Genauigkeit) für drei verschiedene Quantenkernel-Methoden (QKE, pQKE, QKT) und eine klassische Referenzmethode (SVM mit RBF-Kernel) auf mehreren Datensätzen des Anwendungsfalls „FEM-Gitter“.*

Entsprechend der Vorhabensbeschreibung in der TVB wurde ein Vergleich mit der besten bekannten klassischen ML-Methode durchgeführt. Diese ist die "Extremely Randomized Trees"-Methode (Extra Trees) [SD21].

Für den FEM-Datensatz wurde mit QKT eine Recall-Rate (Sensitivität) von 54% und eine Präzision von 63% gefunden. Mithilfe der Extra Trees wurde bei einer Recall-Rate von 54% nur eine Präzision von etwa 50% gefunden, und bei einer Präzision von 63% nur eine Recall-Rate von etwa 20%. Daher scheinen die QKT die Extra Trees zu übertreffen. Dieser Schluss wäre jedoch falsch, denn es konnten bei der Quantenmethode aus Ressourcengründen (SVM skalieren quadratisch mit der Zahl der Datenpunkte) nur insgesamt 6000 Datenpunkte verwendet werden, aufgeteilt auf mehrere Modelle, während bei den Extra Trees 1,6 Millionen für das Training eines einzigen Modells verwendet wurden.

Zusammenfassend ergibt sich das Bild, dass Quanten-Kernel-Methoden grundsätzlich für industrielle Anwendungsfälle geeignet sind, da sie zumindest für relativ kleine Datensätze Ergebnisse auf dem gleichen Niveau wie Standardmethoden liefern. Allerdings konnten keine praktischen Vorteile für Datensätze festgestellt werden, die so klein sind, dass sie noch mit klassischen Simulationen der Quantenschaltkreise bearbeitet werden können. Es gibt jedoch theoretische Hinweise darauf, dass die Situation bei größeren Datensätzen ähnlich sein wird [KBS21]. Aus diesem Grunde ist das Ergebnis noch nicht industriell verwertbar. Es sind hierzu andere Forschungsansätze vonnöten, wie etwa variationelle Schaltkreise/Quanten-Neuronale-Netze (QNNs). Die Berechnungen in diesem TVH wurden mangels Verfügbarkeit geeigneter Quantenrechner nur auf klassischer Simulationshardware ausgeführt. Es ist allerdings davon auszugehen, dass die Ergebnisse aufgrund der Verrauschtheit der Quantenrechner nicht besser ausfallen würden als in der Simulation mit perfekten Qubits.

### **Fortschritte außerhalb des Projekts**

Während der Laufzeit des Projektes wurden Arbeiten veröffentlicht [KBS21, BW+23, CP+23, TW+22], die nahelegen, dass das Fehlen eines Vorteils des Quantenmodells gegenüber dem klassischen Modell auf das Fehlen eines geeigneten "Inductive Bias" zurückzuführen ist. Das bedeutet, dass die verwendeten Modelle falsche oder nicht genügend Vorannahmen enthalten, die mit dem zugrundeliegenden Modell der Datenerzeugung übereinstimmen. Es gibt keinen Grund zu der Annahme, dass der kernelerzeugende Quantenschaltkreis bestimmte Eigenschaften enthält, die strukturelle Übereinstimmung mit dem Prozess der Erzeugung der Daten der FEM-Netze aufweisen. Dies gilt allerdings ebenso für die klassische SVM sowie für das beste klassische Modell, die Extra Trees. Da das zugrundeliegende Modell der Datenerzeugung für viele industrielle Daten nicht bekannt ist (wenn überhaupt sinnvoll davon gesprochen werden kann), so können nur sehr große Datenmengen beim Training einem fehlenden richtigen Bias des Modells entgegenwirken. Bei der SVM skaliert jedoch das Training quadratisch mit der Anzahl der Datenpunkte, da zur Berechnung eines Kernelements zwei Datenpunkte erforderlich sind, was die Verwendung größerer Datenmengen zur Erreichung einer höheren Klassifikationsgüte der SVM erschwert. Die Extra Trees skalieren dagegen nur leicht stärker als linear mit der Datenmenge, sodass hier große Datensätze verwendet werden können. Dies unterstreicht die in der Machine-Learning-Community verbreitete Einsicht, dass mehr Daten häufig wichtiger sind als die Modellklasse, um ein gutes trainiertes Modell zu erhalten, zumindest sofern die Modellklasse keinen zu den Daten passenden inductive Bias besitzt. Eine mögliche Lösung besteht also darin, variationelle Schaltkreise anstelle von Quantum Kernel Estimation zu verwenden, da diese nur linear mit der Anzahl der Datenpunkte skalieren. (Hierin liegt mutmaßlich der Grund für den Erfolg der klassischen neuronalen Netze gegenüber den klassischen SVM.) Eine weitere Möglichkeit besteht darin, vorhandene Strukturen in den Daten als induktiven Bias in die Modelle zu integrieren. Dies wird bei industriell erzeugten Daten jedoch voraussichtlich nur in Ausnahmefällen gelingen.

Anmerkung: Das vorliegende TVH hatte primär Klassifikationsprobleme im Fokus, welche mittels Quantenalgorithmen gelöst werden sollten. Sekundär wurden auch Regressionsprobleme betrachtet, mit dem Ziel, die Quantenkernel auch für diese einzusetzen. Einige Testdatensätze wurden hierfür herangezogen, jedoch ohne

signifikanten Erfolg. Da die Ansätze zur Klassifikation mittels Quantenkernel hinter den besten klassischen Methoden zurückblieben bzw. auf dem Niveau der klassischen RBF-SVM war, konnte bei Einsatz derselben Kernel zur Regression ebenfalls kein großer Erfolg erwartet werden. Daher wurden die Regressionsprobleme nicht weiter betrachtet.

Quantenoptimierungsalgorithmen (Ausführung: David Headley)

### **Quantum Approximate Optimization Algorithm (QAOA)**

Der Quantum Approximate Optimization Algorithm (QAOA) ist ein Algorithmus, der dazu dient, Lösungen für kombinatorische Optimierungsprobleme zu finden. Dieses Gebiet ist sehr heuristisch und Versuche, echte Quantenbeschleunigung mathematisch rigoros nachzuweisen, sind bisher gescheitert, auch an der Entdeckung quanten-inspirierter klassischer Algorithmen.

Bei der Verwendung von QAOA kann die Suche nach Parametern, die den erwarteten Wert der Lösungen maximieren, ein Hindernis für den effektiven Einsatz darstellen. Üblicherweise wird der Quantencomputer dabei als Blackbox betrachtet, deren Eingabeparameter so optimiert werden müssen, dass die Messung des Zustands des Quantenregisters nach Ausführung des Satzes von Quantengattern, die den QAOA ausmachen, zu guten Lösungen führt. Ein aktuelles Schlüsselproblem besteht darin, dass es in einigen Fällen schwieriger sein könnte, gute Werte für diese Eingabeparameter zu finden, als das Problem einfach auf einem klassischen Computer zu lösen. Es ist daher wertvoll, mathematische Methoden zur Bestimmung optimaler Parameter zu erforschen, indem man allgemeine Eigenschaften eines Problems oder einer Klasse von Problemen ausnutzt. Der zukünftige Nutzen von QAOA hängt davon ab, ob es möglich ist, gute Parameter für bestimmte Probleme zu finden, ohne für jede Instanz eine große Menge an klassischer Verarbeitung aufwenden und die Kosten eines iterativen Schleifenoptimierungsverfahrens auf einem verrauschten Quantenprozessor oder der ineffizienten klassischen Simulation tragen zu müssen.

Es gibt eine wachsende Anzahl von Forschungsarbeiten, die sich mit der Suche nach diesen Parametern für bestimmte Probleme beschäftigen, insbesondere wenn die Probleminstanzen so groß sind, dass gewisse Näherungen vorgenommen werden können. Während andere Arbeiten in diese Richtung [FG+22, CD21] sich auf die Berechnung der erwarteten Lösungsqualität für spezifische Problemklassen konzentriert haben, gehen wir den Ansatz, Probleme als Zufallsvariablen mit einer gegebenen Verteilung von Kosten zu modellieren [HW23]. Durch diese Formulierung und in Kombination mit der Verwendung eines hochsymmetrischen Treibers, wie der Grover-Algorithmus ihn verwendet, wird es möglich, einen geschlossenen Ausdruck für die erwartete Lösungsqualität zu schreiben, ohne eine rechnerisch prohibitive Abhängigkeit von der Größe des zu lösenden Problems in Kauf nehmen zu müssen. Arbeiten in diese Richtung könnten die Anwendbarkeit von QAOA auf Industrieprobleme durch die Bereitstellung von variationalen Parametern verbessern, die auf statistischen Eigenschaften des betreffenden Problems basieren.

In unseren Arbeiten haben wir einen neuen Problemtyp und eine neue Art des QAOA entdeckt, bei denen die Leistung des QAOA leicht quantifizierbar ist und optimale Winkel für eine bestimmte Klasse von Problemen gefunden werden können. Kombinatorische Optimierungsprobleme einer bestimmten Größe, die mit QAOA gelöst werden sollen, bestehen aus einer Gruppe potenzieller Lösungen. Für binäre Optimierungsprobleme

können diese Lösungen mit der Menge der binären Zeichenfolgen bestimmter Länge identifiziert werden, die eine Variable  $z$  annehmen kann. Jeder dieser Zeichenfolgen ist ein Wert einer Zielfunktion  $C(z)$  zugeordnet. Das Ziel von QAOA besteht darin, eine Zeichenfolge zu finden, die einem möglichst großen oder kleinen Wert von  $C(z)$  entspricht, je nach dem, ob ein Maximierungs- oder Minimierungsproblem gelöst werden soll.

Eine offene Frage in Bezug auf QAOA betrifft die Quantifizierung seiner Leistung. Es ist schwierig, die Leistung von QAOA zu bestimmen, da ein Teil des Algorithmus die Struktur der Probleme explizit ausnutzt. Insbesondere ist der Einsatz von Ein-Qubit-Treibern speziell auf Probleme zugeschnitten, die in Bezug auf das Umklappen von Bits strukturiert sind. Wenn also bei solchen Problemen der Wert eines Bits in  $z$  geändert wird, ändert sich die Zielfunktion um einen kleineren Wert, als wenn zufällig eine neue Kandidatenzeichenfolge ausgewählt wird. Wir haben die Leistung von QAOA für Probleme unabhängig von ihrer Struktur untersucht, indem wir eine Version von QAOA verwendet haben, die für die Problemstruktur unempfindlich ist. Dazu haben wir den Grover Driver verwendet, also einen nichtklassischen Term, der vollkommen austauschsymmetrisch ist, da er am  $n$ -Qubit-Hadamard Zustand reflektiert wird und so Übergänge zwischen allen möglichen Zuständen bewirkt, anders als die normalerweise verwendeten Ein-Qubit-Übergänge.

Die Verwendung des Grover Drivers bricht den Algorithmus in eine Samplingaufgabe um, die von im generischen Fall abschätzbaren Größen wie z.B. Zustandsdichten abhängt. Somit spielt die Problemstruktur für die Leistung des QAOA keine Rolle mehr. Aus Sicht des QAOA-Algorithmus besteht ein Problemtyp nur aus einer Wahrscheinlichkeitsverteilungsfunktion (PDF)  $P(c)$ , die die Wahrscheinlichkeit beschreibt, mit der ein Wert  $c$  der Zielfunktion von einer zufällig ausgewählten Zeichenfolge angenommen wird. Obwohl diese PDF für endliche Probleme diskret ist, kann sie im Grenzfall unendlicher Problemgröße als kontinuierlich angenähert werden, was die Berechnungen weiter vereinfacht.

Wir haben analytische Ausdrücke für die erwartete Qualität von Stichprobenlösungen berechnet, die vom QAOA für eine PDF geliefert werden. Diese Ausdrücke ermöglichen die Berechnung der Leistung von QAOA unabhängig von der Problemgröße. Das bedeutet, dass wir optimale Winkel für große kombinatorische Optimierungsprobleme finden können, ohne eine kostspielige Optimierungsschleife über die Berechnung auf dem Quantenprozessor oder eine Simulation einer großen Anzahl von Qubits auf einem klassischen Computer durchführen zu müssen.

Wir berechneten den Erwartungswert des QAOA-Hamiltonoperators, der das Problem beschreibt, für den Grover Driver bei verschiedenen Schaltkreistiefen. Es ist instruktiv, verschiedene Probleme zu betrachten und die PDF ihrer charakteristischen Funktionen zu ermitteln. Ein einfaches Beispiel ist das Random Energy Model, für das die PDF gaußförmig ist. Andere Probleme, die im Zusammenhang mit QAOA umfassend untersucht werden, sind Probleme basierend auf dem Sherrington-Kirkpatrick-Modell (SK) und Max-Cut-Probleme. Die Wahrscheinlichkeitsverteilung für SK-Modell-Probleme kann als gaußförmig angenähert werden, obwohl diese Näherung für QAOA mit hoher Schaltkreistiefe zusammenbricht, da sich die Ausläufer der Wahrscheinlichkeitsverteilung der Zielfunktion erheblich unterscheiden.

Der Erwartungswert eines gaußschen Problems bei Schaltkreistiefe 1 kann analytisch hergeleitet werden und muss nur differenziert werden, um die Parameter zu finden, die ihn maximieren oder minimieren. Die Standardabweichung kann an das jeweils vorlie-

gende Problem angepasst werden. Es zeigt sich, dass die Winkel, die durch die Minimierung des Erwartungswerts einer gaußschen PDF gefunden wurden, den Winkeln entsprechen, die sich durch Minimierung der erwarteten QAOA-Zielfunktion für SK-Instanzen endlicher Größe ergeben, wie in Abbildung 4 dargestellt.

SK Model Optimised Angles vs Analytical Angles  $p = 1$  SK Model Optimised Angles vs Analytical Angles  $p = 2$

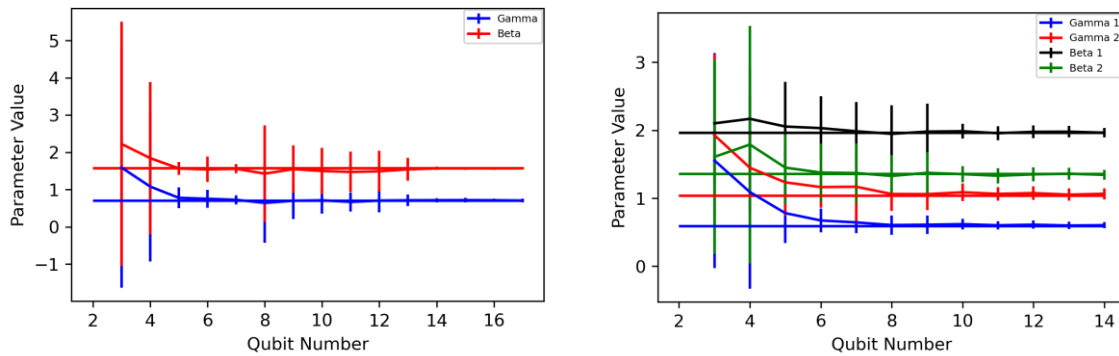


Abbildung 4: Vergleich der optimalen Parameter des SK-Modells für wachsende Qubitzahl mit den analytisch berechneten Parametern für die gaußsche Näherung. Die Parameter konvergieren zu den analytisch angenäherten Werten für Tiefe 1 und 2. Es wird jedoch erwartet, dass sie für größere Tiefe divergieren.

Im nächsten Schritt wurde dieses Verfahren auf beliebige Schaltkreistiefen verallgemeinert und das Partitionierungsproblem als eine Aufgabe identifiziert, die auf einem klassischen Rechner schwer zu lösen ist, für das jedoch die PDF genau bekannt ist. Abbildung 5 zeigt, dass mit den analytisch bestimmten Parametern bereits bei moderater Schaltkreistiefe (eine oder zwei Schichten) Konvergenz stattfindet. Außerdem hat sich der Grover Driver als etwas leistungsfähiger als der Ein-Qubit-X-Driver erwiesen.

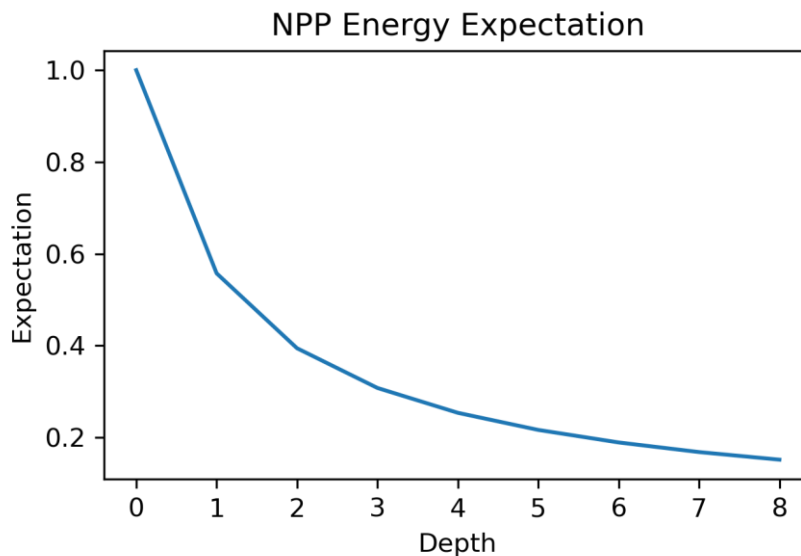


Abbildung 5: Erwartungswert des Partitionierungsproblems mit analytisch bestimmten Parametern des Grover-Drivers: Konvergenz bei moderater Schaltkreistiefe.

Sodann arbeiteten wir an einer Verallgemeinerung für leichter zu implementierende Driver. Darüber hinaus haben wir ein kaskadiertes Protokoll für optimierte annealing schedules in der LHZ-Architektur entwickelt, das schrittweise schnelle Lösungen für

leichte bis mittelschwere Instanzen liefert. Dieses dient dazu, in realistischen Workflows einfache und mittelschwere Instanzen schnell zu lösen, was bisher in einem blinden Verfahren nicht möglich war [BMW23].

Einer schnellen Verwertung der gesamten Methode steht (neben dem Mangel an geeigneter Hardware) bisher vor allem Tatsache entgegen, dass für industrielle Problemstellungen die PDF des Problems in der Regel nicht bekannt sind. Hierzu könnten allerdings Annahmen getroffen werden, mit denen die PDF und damit die zu verwendenden Parameter näherungsweise bestimmt werden können. Darüberhinaus könnte durch Anpassung einer geeigneten Fitfunktion an eine Stichprobe aus der Menge der Variablen  $z$  und ihrer Kostenfunktionswerte  $C(z)$  die PDF abgeschätzt werden.

Angesichts des Fehlens von Quantenrechnern wurden die Algorithmen lediglich auf konventioneller Simulationshardware ausgeführt. Folglich war es nicht möglich, die Auswirkungen von tatsächlichem Rauschen zu ermitteln.

### **Ko-Design von digital-analogen QAOA-Geräten**

In einem weiteren Forschungsstrang haben wir eine Hardwarearchitektur entwickelt, die auf induktiv gekoppelten Flux-Qubits basiert. Diese Architektur wurde mit dem Ziel entworfen, das Digital-Analog-Schema des Quantencomputings [PL+20] ideal umzusetzen. Dieses Kompilierungsschema ist wiederum optimal geeignet für die Ausführung des QAOA-Algorithmus.

Die gängige Designwahl für Quantencomputer basiert auf einem gatterbasierten Modell, bei dem jede Wechselwirkung zwischen Qubits unabhängig voneinander gesteuert wird. Bisher gibt es nur wenige Versuche, Hardware zu entwickeln, die speziell auf die zu lösenden Problemstellungen zugeschnitten ist. Supraleitende Qubit-Bauelemente sind in der Regel planar angeordnet, wodurch die entsprechenden planaren Probleme in der kombinatorischen Optimierung meist einfach zu lösen sind.

In vorherigen Arbeiten [HM+22] haben wir gezeigt, dass die Verwendung eines anderen Schemas zur Wechselwirkung zwischen Qubits Vorteile bieten könnte. Wir haben einen theoretischen Gerätetyp betrachtet, bei dem eine All-to-All-Zweikörperwechselwirkung, die als Ressourcenwechselwirkung bezeichnet wird, durch Ein-Qubit-Operationen zu einem Hamilton-Operator geformt werden kann, der für die Ausführung des QAOA benötigt wird. Unsere Untersuchungen haben gezeigt, dass ein solches Kompilierungsschema gut zum QAOA passt und dass die durch das Schema eingeführten Fehler die Ergebnisse der Berechnung nicht so stark beeinflussen sollten wie in anderen algorithmischen Kontexten. Dies liegt daran, dass QAOA ein Variationsalgorithmus ist und kohärente Fehler zwar die optimalen Parameterwerte ändern können, aber nicht unbedingt die Qualität der entsprechenden Lösung beeinträchtigen.

Im Anschluss daran haben wir uns mit den verschiedenen Arten von Quantencomputerarchitekturen beschäftigt, die man verfolgen sollte, um ein solches Gerät zu entwerfen. Die beiden wichtigsten Optionen sind Flux- und Transmon-Qubits, wobei Flux-Qubits die schnellsten Ein-Qubit-Gatter aufweisen, die für das Schema benötigt werden. Ein Schema wie das, welches wir vorschlagen, ist durch bestimmte Zeitskalen eines Geräts begrenzt: zum einen die Kohärenzzeit, das ist die Zeitspanne, über der die Qubits dekohärieren und somit ihre Informationen verlieren; zum anderen die Ausführungszeit einer Ein-Qubit-Operation, also die Zeit, die benötigt wird, um eine Ein-Qubit-Rotation

auf einem Qubit durchzuführen. Da Ein-Qubit-Gatter auf der Zeitskala von etwa 20 Nanosekunden [OM+99] ausgeführt werden können und die Kohärenzzeiten um die 40 Mikrosekunden [YG+16] liegen, ergibt sich ein Unterschied von mehreren tausend. Frühere Arbeiten haben ergeben, dass die Ressourcenwechselwirkung etwa 100-mal langsamer sein sollte als die einzelnen Qubit-Gatter, damit Fehler die Lösungsqualität nicht negativ beeinflussen. Das bedeutet, dass eine Ressourceninteraktion im Bereich von etwa 1 MHz verwendet werden sollte. Etwa 20 Zyklen mit wechselwirkenden Gattern können innerhalb der Kohärenzzeit durchgeführt werden, was eine Anordnung ermöglicht, die einen QAOA mit geringer Tiefe für bis zu 10 Qubits ermöglicht. Dies ergibt sich aus unseren früheren Arbeiten zu diesem Thema.

Eine schematische Darstellung der vorgeschlagenen Flux-Qubit-Schaltung, die die gewünschte Wechselwirkung realisiert, ist in Abbildung 6 dargestellt. Die Schaltung besteht aus einer zentralen Schleife, die alle Qubits, die als horizontal angeordnete Quadrate abgebildet sind, induktiv koppelt. Zukünftige Arbeiten in diesem Bereich werden untersuchen, ob es realistisch ist, kleine Demonstratorvorrichtungen zu entwerfen und Designentscheidungen zu treffen, um Inhomogenität in den Interaktionen zu minimieren.

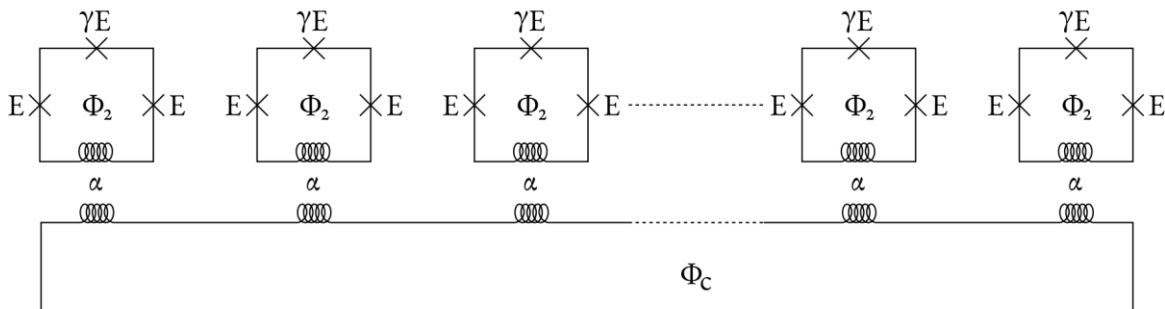


Abbildung 6: Ein Schaltplan für ein Flux-Qubit-Gerät zur Realisierung einer All-to-All-Zweikörperwechselwirkung.

## Veröffentlichungen des vorliegenden Teilvorhabens

1. David Headley and Frank K. Wilhelm. Problem-size-independent angles for a Grover-driven quantum approximate optimization algorithm. Phys. Rev. A 107, 012412 (2023). <https://doi.org/10.1103/PhysRevA.107.012412>
2. Beitrag zu: Stollenwerk, Tobias et al. (2024). Q(AI)<sup>2</sup> — Quantum Artificial Intelligence for the Automotive Industry. To appear in Special Issue on QKI of the German Journal of Artificial Intelligence - Organ des Fachbereichs "Künstliche Intelligenz" der Gesellschaft für Informatik e.V.

## Literaturverzeichnis

- [BMW23] G. Bishop, S. Montangero, and F.K. Wilhelm, A Set of Annealing Protocols for Optimized System Dynamics and Classification of Fully Connected Spin Glass Problems, arXiv:2310.10442

- [BW+23] Bowles, Joseph et al. (2023). Contextuality and inductive bias in quantum machine learning. arXiv:2302.01365
- [CD21] J. Claes, W.v. Dam, Quantum 5, 542 (2021). DOI 10.22331/q-2021-09-15-542. URL <https://doi.org/10.22331/q-2021-09-15-542>
- [CK+06] N. Cristianini, J. Kandola, A. Elisseeff, J. Shawe-Taylor, Innovations in Machine Learning pp. 205–256 (2006). DOI 10.1007/3-540-33486-6\_8. URL [https://link.springer.com/chapter/10.1007/3-540-33486-6\\_8](https://link.springer.com/chapter/10.1007/3-540-33486-6_8)
- [CP+23] Abdulkadir Canatar and Evan Peters and Cengiz Pehlevan and Stefan M. Wild and Ruslan Shaydulin. Bandwidth Enables Generalization in Quantum Kernel Models. Transactions on Machine Learning Research, 2835-8856 (2023). <https://openreview.net/forum?id=A1N2qp4yAq>
- [FG+22] E. Farhi, J. Goldstone, S. Gutmann, L. Zhou, Quantum 6, 759 (2022). DOI 10.22331/q-2022-07-07-759. URL <http://dx.doi.org/10.22331/q-2022-07-07-759>
- [GG+21] J.R. Glick, T.P. Gujarati, A.D. Corcoles, Y. Kim, A. Kandala, J.M. Gambetta, K. Temme, arXiv:2105.03406 (2021). URL <https://arxiv.org/abs/2105.03406v1>
- [HB+21] Huang, HY., Broughton, M., Mohseni, M. et al. Power of data in quantum machine learning. Nat Commun 12, 2631 (2021). <https://doi.org/10.1038/s41467-021-22539-9>
- [HC+19] Havlíček, V., Córcoles, A.D., Temme, K. et al. Supervised learning with quantum-enhanced feature spaces. Nature 567, 209–212 (2019). <https://doi.org/10.1038/s41586-019-0980-2>
- [HM+22] David Headley, Thorge Müller, Ana Martin, Enrique Solano, Mikel Sanz, and Frank K. Wilhelm. Approximating the quantum approximate optimization algorithm with digital-analog interactions. Phys. Rev. A 106, 042446 (2022). <https://doi.org/10.1103/PhysRevA.106.042446>
- [HW23] David Headley and Frank K. Wilhelm. Problem-size-independent angles for a Grover-driven quantum approximate optimization algorithm. Phys. Rev. A 107, 012412 (2023). <https://doi.org/10.1103/PhysRevA.107.012412>
- [KBS21] J. Kübler, S. Buchholz, B. Schölkopf, in Advances in Neural Information Processing Systems, vol. 34 (Curran Associates, Inc., 2021), vol. 34, pp. 12,661–12,673. URL

- [OM+99] T. P. Orlando, J. E. Mooij, Lin Tian, Caspar H. van der Wal, L. S. Levitov, Seth Lloyd, and J. J. Mazo. Superconducting persistent-current qubit. *Phys. Rev. B* 60, 15398 (1999).  
<https://doi.org/10.1103/PhysRevB.60.15398t>
- [PL+20] Adrian Parra-Rodriguez, Pavel Lougovski, Lucas Lamata, Enrique Solano, and Mikel Sanz. Digital-analog quantum computation. *Phys. Rev. A* 101, 022305 (2020). <https://doi.org/10.1103/PhysRevA.101.022305>
- [S21] Schuld, Maria (2021). Supervised quantum machine learning models are kernel methods. *arXiv:2101.11020*
- [SD21] Sprave, Joachim; Drescher, Christian (2021). Evaluating the Quality of Finite Element Meshes with Machine Learning. *arXiv:2107.10507*
- [SJA19] Sim, S., Johnson, P.D. and Aspuru-Guzik, A. (2019). Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Adv. Quantum Technol.*, 2: 1900070.  
<https://doi.org/10.1002/qute.201900070>
- [TW+22] S. Thanasilp, S. Wang, M. Cerezo, Z. Holmes. Exponential concentration and untrainability in quantum kernel methods (2022). DOI 10.48550/arXiv.2208.11060. URL <http://arxiv.org/abs/2208.11060>
- [YG+16] Yan, F., Gustavsson, S., Kamal, A. et al. The flux qubit revisited to enhance coherence and reproducibility. *Nat Commun* 7, 12964 (2016).  
<https://doi.org/10.1038/ncomms12964>

# Kurzbericht

zum Teilvorhaben „Quanten-Machine-Learning für Klassifikation und Regression“

im Verbundprojekt „Quantum Artificial Intelligence for the Automotive Industry (Q(AI)2)“

## Zuwendungsempfänger:

Mercedes-Benz AG  
Dr. Markus Leder  
HPC L423  
Mercedesstraße 120  
70372 Stuttgart

**Förderkennzeichen des Teilvorhabens:** 13N15582

**Laufzeit des Vorhabens:** 01.04.2021 bis 31.03.2024

**Berichtszeitraum:** 01.04.2021 bis 31.03.2024

Das Ziel des vorliegenden Teilvorhabens bestand einerseits darin, Verbesserungen von Quanten-Klassifikations- und -Regressionsalgorithmen zu erarbeiten (AP 3). Andererseits wurde angestrebt, Verbesserungen von Quantenoptimierungsalgorithmen, insbesondere des QAOA-Algorithmus zu erzielen (AP 5).

Im Beitrag des vorliegenden TVH zu AP 3 wurde die Entwicklung von Algorithmen für die Optimierung von Finite-Elemente-Gittern in der digitalen Produktentwicklung (genannt „Deep Mesh“) verfolgt. Das konkrete Ziel war der Einsatz quantenmaschinellem Lernverfahren, um die Qualität der Gitter automatisiert zu beurteilen. Dazu wurden Feature-Vektoren erstellt, die geometrische Eigenschaften der Gitterelemente aggregieren und als Trainingsdaten für ML-Verfahren geeignet sind. Zur Klassifikation der Gitterelemente wurden Quanten-Kernel-Methoden und darauf aufbauende Quanten-Support-Vector-Machines (QSVM) untersucht. Es wurde festgestellt, dass umfangreiche Trainingsdaten entscheidend sind und die Verschränkung alleine kein Garant für eine hohe Klassifikationsgüte einer QSVM ist. Ein vom Projektpartner Jülich Supercomputing Centre (JSC) des FZJ erstellter Generator wurde eingesetzt, um Quantenschaltkreise zu erzeugen, die den jeweiligen Merkmalsvektor in einen Quantenzustand kodieren und daraus Quantenkernel ableiten. Es konnten allerdings auf Standarddatensätzen trotz Verwendung zahlreicher unterschiedlicher Quantenkernel keine signifikanten Vorteile gegenüber klassischen Referenzimplementierungen festgestellt werden. Die diversen Quantenkernel wurden auf den industriellen Daten des Anwendungsfalls "Deep Mesh" getestet. Es wurden drei Quantenkernel-Methoden eingesetzt, Quanten-Kernel-Estimation (QKE), projizierte QKE (pQKE) und Quanten-Kernel-Training (QKT), sowie eine klassische Referenzmethode (SVM mit RBF-Kernel). Zum Training wurde eine Kreuzvalidierungsverfahren mit kombinatorischer Hyperparametersuche durchgeführt und die Leistung des trainierten Klassifikators auf einem Testset bewertet. Die Ergebnisse zeigen, dass alle drei Quantenmethoden in etwa auf dem gleichen Niveau wie die klassische Referenz-SVM liegen, unabhängig vom kodierenden Quantenschaltkreis und anderen Parametern, jedoch mit einer hohen

Standardabweichung. Es konnten keine praktischen Vorteile für Datensätze festgestellt werden, die so klein sind, dass sie noch mit klassischen Simulationen der Quantenschaltkreise behandelt werden können. Die Ergebnisse basierten jedoch auf einem vergleichsweise kleinen Datensatz, um den Simulationskapazitäten für Quantenschaltkreise Rechnung zu tragen, und weitere Forschung ist erforderlich, um festzustellen, ob Quantenmethoden für größere Datensätze praktische Vorteile bieten können. Fortschritte außerhalb des Projekts deuten allerdings darauf hin, dass das Fehlen eines Vorteils des Quantenmodells gegenüber dem klassischen Modell auf das Fehlen eines geeigneten "Inductive Bias" zurückzuführen ist, d.h., die Vorannahmen des Modells stimmen nicht mit den zu lernenden Daten überein. Dies ist allerdings auch im klassischen maschinellen Lernen ein häufiger Fall und kann durch eine sehr große Datenmengen ausgeglichen werden, was aufgrund der schlechten Skalierung der (Q)SVM mit der Menge der Trainingsdaten kein gangbarer Weg ist. Eine mögliche Lösung besteht darin, variationelle Schaltkreise anstelle von Quantum Kernel Estimation zu verwenden.

Im Beitrag dieses TVH zu AP 5 wurde das Problem der kombinatorischen Optimierung mittels Quantenalgorithmien betrachtet, genauer der Quantum Approximate Optimization Algorithm (QAOA). Die Suche nach Parametern, die den erwarteten Zielfunktionswert der Lösungen maximieren, kann ein Hindernis für seinen effektiven Einsatz darstellen. Es ist schwierig, die Leistung des QAOA zu bestimmen, da ein Teil des Algorithmus die Struktur der Probleme explizit ausnutzt. Es gibt eine wachsende Anzahl von Forschungsarbeiten, die sich mit der Suche nach diesen Parametern für bestimmte Probleme beschäftigen, insbesondere wenn die Probleminstanzen so groß sind, dass gewisse Näherungen vorgenommen werden können. In unseren Arbeiten haben wir einen neuen Problemtyp und eine neue Art des QAOA entdeckt, bei denen die Leistung des QAOA leicht quantifizierbar ist und optimale Winkel für eine bestimmte Klasse von Problemen gefunden werden können. Wir haben analytische Ausdrücke für den Erwartungswert der Qualität von Lösungen berechnet, die vom QAOA für eine Wahrscheinlichkeitsverteilung geliefert werden. Diese Ausdrücke ermöglichen die Berechnung der Leistung von QAOA unabhängig von der Problemgröße. Das bedeutet, dass wir optimale Winkel für große kombinatorische Optimierungsprobleme finden können, ohne eine kostspielige Optimierungsschleife über die Berechnung auf dem Quantenprozessor oder eine Simulation einer großen Anzahl von Qubits auf einem klassischen Computer durchführen zu müssen. Darüber hinaus haben wir ein kaskadiertes Protokoll für optimierte annealing schedules in der LHZ-Architektur entwickelt, das schrittweise schnelle Lösungen für leichte bis mittelschwere Instanzen liefert. Dieses dient dazu, in realistischen Workflows solche Instanzen schnell zu lösen, was bisher in einem blinden Verfahren nicht möglich war.

Desweiteren wurde eine Hardwarearchitektur für Quantencomputer entwickelt, die auf induktiv gekoppelten Flux-Qubits basiert und das Digital-Analog-Schema des Quantencomputings ideal umsetzt. Diese Architektur ist optimal geeignet für die Ausführung des QAOA-Algorithmus. Bisherige Designwahl für Quantencomputer basiert auf einem gatterbasierten Modell, bei dem jede Wechselwirkung zwischen Qubits unabhängig voneinander gesteuert wird. Die Verwendung eines anderen Schemas zur Wechselwirkung zwischen Qubits, die als Ressourcenwechselwirkung bezeichnet wird, bietet Vorteile für die Ausführung des QAOA-Algorithmus. Die vorgeschlagene Flux-Qubit-Schaltung besteht aus einer zentralen Schleife, die alle Qubits induktiv koppelt, und ermöglicht einen QAOA mit geringer Tiefe für bis zu 10 Qubits.