

VERBUNDPROJEKT
“QUANTUM ARTIFICIAL
INTELLIGENCE FOR THE
AUTOMOTIVE INDUSTRY
Q(AI)²”

Eingehender Sachbericht zum
Verwendungsnachweis – Robert Bosch GmbH
(Förderkennzeichen 13N15585)

1. Einleitung

Die nachfolgende eingehende Sachbericht zum Verwendungsnachweis des Teilvorhaben (TVH) der Robert Bosch GmbH (Förderkennzeichen 13NI5584) im Verbundprojekt "Quantum Artificial Intelligence for the Automotive Industry Q(AI)²" gliedert sich zum besseren Lesefluss nachfolgend in zwei wesentliche Kapitel, in denen jeweils die Ausarbeitung und Ergebnisse zu zwei wesentlichen industriellen Anwendungsfällen das TVH dargestellt werden. Abschließend folgt eine kompaktere Zusammenfassung von weiteren Arbeiten. Aus Platzgründen wird an geeigneter Stelle auf bereits veröffentlichte Arbeiten und Ergebnisse mit weiteren Details verwiesen.

2. Anwendungsfall „Segmentierung und Klassifikation von Bildern“ (TVH-AP4)

2.1 Zusammenarbeit mit anderen Partnern

Die nachfolgend dargestellten Ausarbeitungen und Ergebnisse entstanden in enger Kooperation mit dem Projektpartner Jülich Supercomputing Centre (JSC).

2.2 Einleitung zum Anwendungsfall (TVH-AP1-A1.1)

2.2.1 Allgemeine Beschreibung des Anwendungsfall

Für verschiedene industrielle Anwendungsfälle ist es nötig, Menschen und deren Pose in verschiedenen Situationen auf Bildern zu identifizieren. Ein Beispiel ist die kollaborative Fertigung wie sie in Abbildung 1 dargestellt ist. In diesem Anwendungsfall muss die *Pose* einer oder mehrere Personen bestimmt werden, was eine Identifizierung von verschiedenen Körperteilen (z.B. Kopf, Schultern, Hüfte, Hände, Beine, ...) auf Grundlage von aufgezeichneten Bildern notwendig macht.

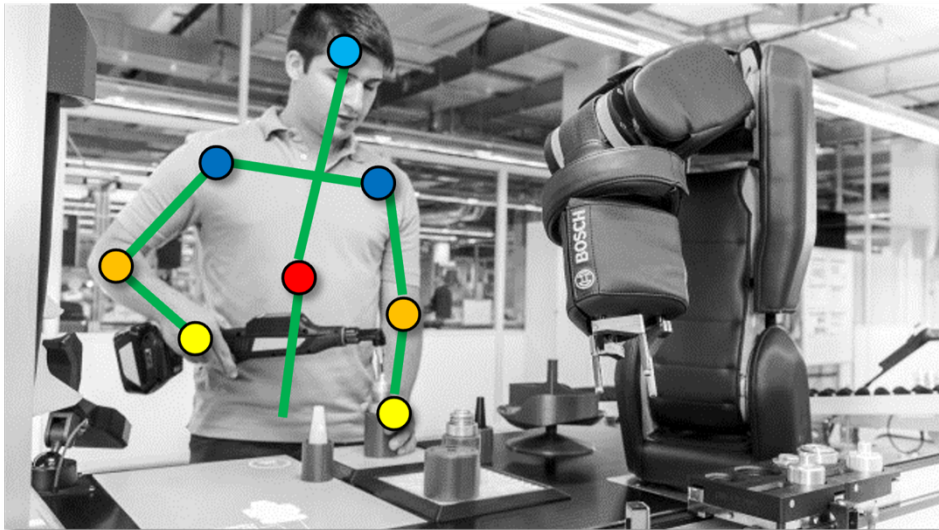


Abbildung 1: Kollaborative Fertigung bei der die Pose des mit der Maschine interagierenden Menschen erkannt werden muss.

2.2.2 Übersicht Quanten-hybrider Ablauf zur Lösung des Anwendungsfall

Abbildung 2 zeigt den Ablauf des im TVH untersuchten Anwendungsfalls der Segmentierung und Klassifizierung. Ein über eine Kamera aufgezeichnetes Bild (linke obere Ecke) ist der Ausgangspunkt. Dieses wird unter anderem von einem „faltenden neuronalen Netzwerk“ (engl. convolutional neural network (CNN)) auf einem klassischen Computer vorprozessiert. Die Ausgaben der Vorverarbeitung durch das CNN sind

- Pixelgenaue *Detektionen* (siehe Abbildung 4)
- Für jede Detektion und für jedes mögliche Körperteil (z.B. Kopf, Hüfte, Schulter, Hand, ...), die Wahrscheinlichkeit ρ_{dc} das diese Detektion ein Körperteil des Typs Kopf, Hüfte Schulter, Hand etc. ist.
- Für alle Paare von Detektionen und alle Paare von möglichen Körperteilen, eine Wahrscheinlichkeit $\rho_{dd'cc'}$ dass beide Detektionen zu einer Person und dabei jeweils vom Körpertyp c bzw. c' sind.

Mit den Ausgabedaten des CNN wird dann ein binäres Optimierungsproblem formuliert, dessen globales Optimum einer Segmentierung der Personen auf dem Bild und der Klassifizierung von erkannten Körperteilen entspricht.

Der in Abbildung 2 dargestellte Ablauf baut in den vorverarbeitenden klassischen Schritten teilweise auf bestehende Arbeiten auf, insbesondere die Vorverarbeitung des aufgezeichneten Bildes durch ein CNN [Insafutdinov16]. Da Quantencomputer wahrscheinlich mittelfristig nur eine überschaubare Anzahl von Qubit enthalten werden und somit ein ressourcenschonender Ansatz zur Lösung von Anwendungsfällen nötig ist, wurde im Rahmen des TVH diese Herangehensweise zur Lösung des Anwendungsfall ausgewählt (Präprozessierung von Bildern durch CNN und anschließendes Lösen eines binären Optimierungsproblems mit Quantencomputer/-annealer). Ein anderer Ansatz, z.B. vollständig aufbauend auf rein quantenphysikalischen Neuronalen Netzen (QNN), würde vermutlich deutlich mehr Qubits in Anspruch nehmen als in absehbarer Zeit verfügbar sein werden.

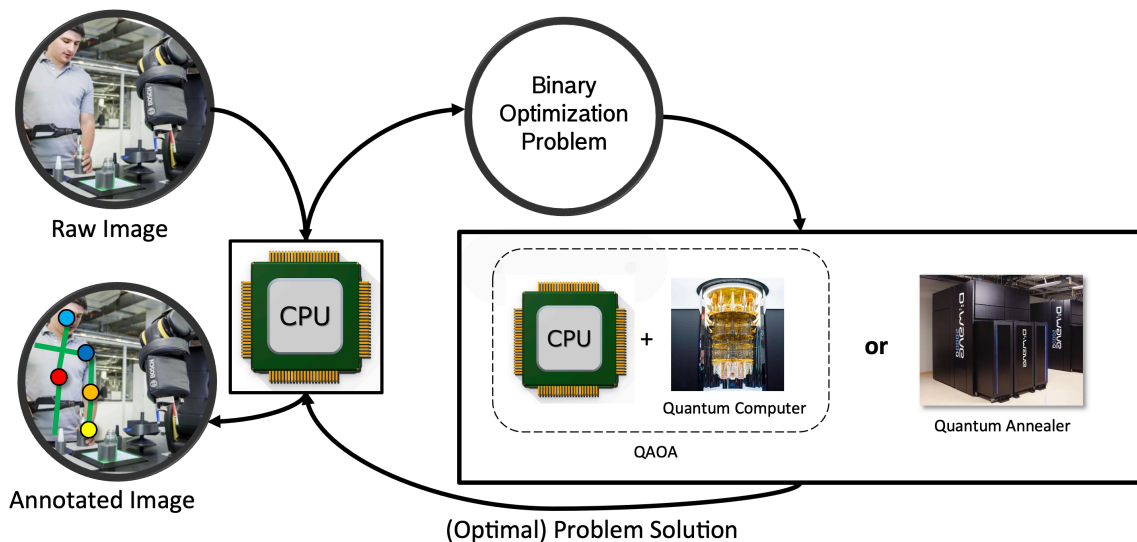


Abbildung 2: Workflow des Anwendungsfalls Segmentierung und Klassifizierung

Eine der Aufgaben im TVH bestand darin, das zu lösende binäre Optimierungsproblem für einen Quantencomputer/-annealer in geeigneter Abstraktion zu formulieren (TVH-AP1-A1.1), passende Quantenalgorithmen zu entwickeln bzw. zu implementieren (TVH-AP4-A4.1/A4.2) und mit diesen Algorithmen unter Hinzunahme von realistischen industriellen Instanzen eine technische Leistungsanalyse (engl. Benchmarking) durchzuführen (TVH-AP5-A5.1/A5.2).

2.3 Details der technischen Umsetzung des Anwendungsfalls (TVH-AP4-A4.1/A4.2)

2.3.1 Vorbereitende Arbeiten

Im Rahmen von vorbereitenden Arbeiten für das Teilvorhaben wurde das Problem des „korrelierten Zusammenfassens“ (engl. correlation clustering) [Bansal2004] untersucht. Dabei handelt es sich um ein generelles Teiloptimierungsproblem des nachfolgend beschriebenen Problems der Segmentierung und Klassifizierung, bei dem die Knoten eines Graphen auf Grundlage von annotierten Korrelationen auf den Kanten zwischen Knoten möglichst optimal zusammengefasst werden müssen. Im Rahmen des Teilvorhabens wurde eine geeignete Formulierung des korrelierten Zusammenfassens für Quantencomputer entwickelt und untersucht. Weiter wurde ein Quantensimulator auf Qudit-Basis beschrieben und der Einfluss von Gatterfehlern untersucht. Aus Platzgründen sei der interessierte Leser auf die im Rahmen des TVH publizierte Arbeit [Weggemans2022] verwiesen.

2.3.2 Lineare „klassische“ Kodierung des Anwendungsfall

Ausgangspunkt für den Umsetzung kann die lineare Kodierung aus der Literatur sein [Levinkov2017, Insafutdinov16]. Es sei jedoch angemerkt, dass auf Quantencomputern eine Beschränkung auf diese lineare Kodierung nicht zwingend notwendig ist. Daher wird im Folgenden auch eine andere für Quantencomputer geeignete mathematische Abstraktion gezeigt.

Die klassische lineare Kodierung baut auf den binären Variablen

$$x_{dc} : \mathbf{D} \times \mathbf{C} \rightarrow \{0, 1\}$$

auf, die den Wert 1 haben, wenn Detektion $d \in \mathbf{D}$ vom Körpertyp $c \in \mathbf{C}$ ist. Bei einem Wert von 0 gilt, dass die Detektion d unterdrückt und als Fehldetektion des CNN betrachtet werden.

Weiter beschreiben die binären Variablen

$$y_{dd'} : \binom{\mathbf{D}}{2} \rightarrow \{0, 1\}$$

dass die Detektionen d und d' zur selben Person gehören, wenn $y_{dd'} = 1$.

Im Weiteren werden die notwendigen Nebenbedingungen dargestellt. Jede Detektion kann höchstens ein Körpertyp sein. Wenn einer Detektion kein Körpertyp zugewiesen wird, wird diese Detektion unterdrückt.

$$\forall d \in \mathbf{D} \forall c, c' \in \mathbf{C} : x_{dc} + x_{dc'} \leq 1 \quad (1)$$

Wenn zwei Detektionen nicht unterdrückt werden, so können zwei unterschiedliche Detektionen d und d' zur selben Person gehören.

$$\forall d, d' \in \binom{\mathbf{D}}{2} : y_{dd'} \leq \sum_{c \in \mathbf{C}} x_{dc} \quad (2)$$

$$y_{dd'} \leq \sum_{c \in \mathbf{C}} x_{d'c}$$

Weiter muss die Transitivität betrachtet werden: wenn Detektionen d und d' zu selben Person gehören ($y_{dd'} = 1$), und d und d'' zur selben Person gehören ($y_{dd''} = 1$), folgt, dass d' und d'' zur selben Person gehören ($y_{d'd''} = 1$).

$$\forall (d, d', d'') \in \binom{\mathbf{D}}{3} : y_{dd'} + y_{d'd''} - 1 \leq y_{dd''} \quad (3)$$

$$y_{dd'} + y_{dd''} - 1 \leq y_{d'd''}$$

$$y_{dd''} + y_{d'd''} - 1 \leq y_{dd'}$$

Die Einführung von zusätzlichen Hilfsvariablen, wie in [Levinkov2017, Insafutdinov16] gezeigt, ist nicht notwendig. Stattdessen kann eine nicht-lineare Zielfunktion verwendet werden, deren globales Minimum die Segmentierung und Klassifizierung von Personen/Körperteilen angibt.

$$\min_{x,y} f(x,y,p_{dc},p_{dd'cc'})$$

$$f(x,y,p_{dc},p_{dd'cc'}) = \sum_{d \in \mathcal{D}} \sum_{c \in \mathcal{C}} \alpha_{dc} x_{dc} + \sum_{d,d' \in \binom{\mathcal{D}}{2}} \sum_{c,c' \in \mathcal{C}} \beta_{dd'cc'} x_{dc} x_{d'c'} y_{dd'} \quad (4)$$

$$\alpha_{dc} = \log \frac{1 - p_{dc}}{p_{dc}}$$

$$\beta_{dd'cc'} = \log \frac{1 - p_{dd'cc'}}{p_{dd'cc'}}$$

2.3.3 Energiefunktional in QAOA aus der klassischen linearen Kodierung

Für die Lösung des Segmentierungs- und Klassifikationsproblems mittels QAOA (engl. „Quantum Approximate Optimization Algorithm“) [Farhi2014] muss ein Funktional (engl. *Hamiltonian*) aus den oben gezeigten Nebenbedingungen und der Zielfunktion erstellt werden. Dafür wurde für die Nebenbedingungen folgendes Vorgehen angewendet: unter Verwendungen von σ^z Operatoren wurden unitäre Operatoren entwickelt, die jeweils den Wert von 1 ergeben, wenn die entsprechenden Qubit-Zustände der Variablen die Nebenbedingungen verletzen, sonst 0. Bei Verletzung der Nebenbedingungen trägt somit eine „Strafe“ im Energiefunktional bei. Die nachfolgende Tabelle zeigt für die Nebenbedingungen die Zustände, welche die jeweilige Nebenbedingung verletzen und den jeweiligen normierten Operator.

Nebenbedingung (siehe oben)	Nebenbedingung verletzender Zustand	Operator für Nebenbedingung
(1)	$ 1\rangle_{x_{dc}} 1\rangle_{x_{dc'}}$	$\frac{1}{4} (1 - \sigma_{x_{dc}}^z) (1 - \sigma_{x_{dc'}}^z)$
(2)	$ 1\rangle_{y_{dd'}} \otimes_{c \in \mathcal{C}} 0\rangle_{x_{dc}}$	$\frac{1}{2^{ \mathcal{C} +1}} (1 - \sigma_{y_{dd'}}^z) \prod_{c \in \mathcal{C}} (1 + \sigma_{x_{dc}}^z)$
(3)	$ 1\rangle_{y_{dd'}} 1\rangle_{y_{d'd''}} 0\rangle_{y_{dd''}}$ $ 1\rangle_{y_{dd'}} 0\rangle_{y_{d'd''}} 1\rangle_{y_{dd''}}$ $ 0\rangle_{y_{dd'}} 1\rangle_{y_{d'd''}} 1\rangle_{y_{dd''}}$	$\frac{1}{8} [(1 - \sigma_{y_{dd'}}^z) (1 - \sigma_{y_{d'd''}}^z) (1 + \sigma_{y_{dd''}}^z) + (1 - \sigma_{y_{dd'}}^z) (1 + \sigma_{y_{d'd''}}^z) (1 - \sigma_{y_{dd''}}^z) + (1 + \sigma_{y_{dd'}}^z) (1 - \sigma_{y_{d'd''}}^z) (1 - \sigma_{y_{dd''}}^z)]$

Tabelle 1: Übersicht der Übersetzung der Nebenbedingungen der linearen Kodierung in Operatoren

In der Tabelle fällt auf, dass für Nebenbedingung (2), je nach Mächtigkeit der Körpertypenklasse \mathcal{C} , ein Operator entsteht der auf mehr als drei Qubits wirkt. Damit eine Implementierung des QAOA auch auf NISQ-Plattformen ausführbar bleibt (insbesondere für supraleitende Plattformen), wurde die Entscheidung getroffen, dass die Mächtigkeit von $|\mathcal{C}| \leq 2$ im Weiteren eingeschränkt wird. Nachfolgend wird gezeigt, dass mit Hilfe von Gateoperationenidentitäten dadurch nur 2-Qubit-Gateoperationen benötigt werden, die auf NISQ-Plattformen verfügbar sind.

Ein Funktional für alle Nebenbedingungen entsteht dann aus der Summe der einzelnen unitären Operationen über die jeweiligen Variablenindices aus Gleichung (1)-(3). Die Zeitentwicklung $\exp[-i\gamma H_c]$ des Funktional der Nebenbedingungen H_c führt dann zu Pauli-Rotationen von einzelnen oder mehreren Qubits¹. Weiter können folgende Identitäten genutzt werden, um Pauli-Rotationen über zwei oder drei Qubits zu vereinfachen [Vatan2004]:

¹ Da σ_i^z und σ_j^z kommutieren kann die Summe in der e -Funktion in Produkten von e -Funktionen geschrieben werden.

- $\exp\left[-i\frac{\theta}{2}\sigma_i^z\sigma_j^z\right] = CNOT_{ij}R_j^z(\theta)CNOT_{ij}$
- $\exp\left[i\theta\sigma_i^z\sigma_j^z\sigma_k^z\right] = CNOT_{ik}CNOT_{jk}R_j^z(2\theta)CNOT_{jk}CNOT_{ik}$

Um ein Funktional für die Zielfunktion (4) zu entwickeln wurden analog die Variablen durch die entsprechenden σ^z Operatoren ersetzt und analog die Zeitentwicklung wie bei den Nebenbedingungen durchgeführt.

2.3.4 Effizientes Energiefunktional für Quantenannealer

Obige „klassische“ lineare Kodierung ist in der Lage, eine beliebige Anzahl von Personen auf einem Bild zu segmentieren, ohne dass eine maximale Anzahl an zu detektierenden Personen angegeben werden muss. Während dies ein Vorteil ist, müssen obige Ungleichungen unter Verwendung von Hilfsvariablen (engl. slack variables) umgeschrieben werden, wodurch wesentlich mehr Qubits benötigt werden. Ein Umschreiben der Ungleichungen ist nötig, da ein Energiefunktional für einen Quantenannealer, wie z.B. dem Annealer von D-Wave, ein Ising-Modell bzw. ein quadratisches unbeschränktes binäres Optimierungsproblem (engl. quadratical unconstraint binary optimization problem (QUBO)) sein muss. Im Folgenden werden zwei quadratische Kodierungen vorgestellt.

Kodierung *ohne* Unterstützung der Unterdrückung von Detektionen

Zunächst wird eine Menge von diskreten „Segmenten“ \mathcal{S} , z.B. einzelne Personen, eingeführt. Damit ergeben sich die binären Variablen zu

$$x_{dcs} \in \{0,1\}^{D \times C \times S},$$

wobei $x_{dcs} = 1$ nur wenn $C(d) = c$ und $S(d) = s$, sonst $x_{dcs} = 0$.

Weiter gilt die Nebenbedingung, dass eine Detektion $d \in \mathcal{D}$ nur maximal von einem Typ $c \in \mathcal{C}$ sein kann und zu maximal einem Segment $s \in \mathcal{S}$ gehören kann (engl. „one-hot“ constraint):

$$\forall d \in \mathcal{D} : \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} x_{dcs} = 1$$

Diese Nebenbedingung wird dann umgeschrieben zu einem quadratischen energetischen Strafterm (engl. penalty term) der in eine Zielfunktion/Energiefunktional übernommen werden kann:

$$f_{pen}^{hot} = \lambda \sum_{d \in \mathcal{D}} \left(1 - \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} x_{dcs}\right)^2$$

Das Energiefunktional aus (4) lässt sich umschreiben in

$$f_{obj}^{hot}(x_{dcs}, p_{dc}, p_{dd'cc'}) = \sum_{d \in \mathcal{D}} \sum_{c \in \mathcal{C}} \alpha_{dc} \sum_{s \in \mathcal{S}} x_{dcs} + \sum_{d, d' \in \binom{\mathcal{D}}{2}} \sum_{c, c' \in \mathcal{C}} \beta_{dd'cc'} \sum_{s \in \mathcal{S}} x_{dcs} x_{d'c's}$$

Letztlich ergibt sich die gesamte Zielfunktion eines quadratisches unbeschränktes binäres Optimierungsproblem (engl. quadratical unconstraint binary optimization problem (QUBO)) zu

$$f_{QUBO}^{hot} = f_{obj}^{hot} + f_{pen}^{hot}$$

Diese Zielfunktion kann dann, z.B., auf dem D-Wave Quantenannealer genutzt werden, in dem ein Energiefunktional durch Ersetzung der binären Variablen $x_{dcs} \rightarrow \frac{1}{2}(1 - \sigma_{dcs}^z)$ erstellt wird.

Kodierung *mit* Unterstützung der Unterdrückung von Detektionen

Ein Nachteil der oben gezeigten quadratischen Kodierung ist, dass diese nativ nicht die Unterdrückung von Detektionen unterstützt, d.h. alle Detektionen auf einem Bild segmentiert und klassifiziert werden müssen, auch wenn diese nur "Artefakte" sind. Die obige Kodierung kann jedoch durch Hinzufügen eines zusätzlichen Segments $s_{sup} \in \mathcal{S}$ erweitert werden, welches alle Detektionen enthält, die unterdrückt werden sollen. Damit steigt allerdings die Anzahl der Variablen um $|\mathcal{D}| \cdot |\mathcal{C}|$ im Vergleich zum oben gezeigten Fall ohne Unterdrückung.

Eine effizientere Lösung wurde entwickelt und besteht im Einführen einer zusätzlichen Variable

$$y_d = \begin{cases} 1, & d \in \mathcal{D} \text{ wird unterdrückt} \\ 0, & \text{sonst} \end{cases}$$

Damit steigen die zusätzlich benötigten Variablen dann „nur“ um $|\mathcal{D}|$ zusätzliche Variablen. Der Strafterm der Nebenbedingungen ändert sich zu

$$f_{pen}^{hot*} = \lambda \sum_{d \in \mathcal{D}} \left(1 - \left(\sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} x_{dcs} \right) - y_d \right)^2$$

Mit diesem Term für die Nebenbedingungen bleibt man im Formalismus des quadratischen unbeschränkten binären Optimierungsproblem. Allerdings muss der Hyperparameter $\lambda \geq 0$ durch Feineinstellung so angepasst werden, dass auch der Fall mit $\sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} x_{dcs} = 1$ und $y_d = 1$ zu falschen Lösungen führt.

2.3.5 Übersicht zur Anzahl von Variablen je nach Anwendungsfall und Kodierung

Betrachtet man zunächst die Anzahl der Gesamtvariablen/logischen Qubits für ein "vollwertiges" Problem (um den Bereich von $|\mathcal{D}| = 140$, $|\mathcal{C}| = 14$, $|\mathcal{S}| = 5$, der den Bereich umfasst, der auch in der klassischen Referenzliteratur betrachtet wird [Insafutdinov16]), zeigt Abbildung 3 die Anzahl der Variablen in Abhängigkeit von der Anzahl der Detektionen $|\mathcal{D}|$. Wie zu sehen ist, haben die entwickelten *one-hot* Kodierungen (mit bzw. ohne Unterdrückung von Detektionen) Vorteile in der Anzahl der Variablen ab $|\mathcal{D}| \approx 115$.

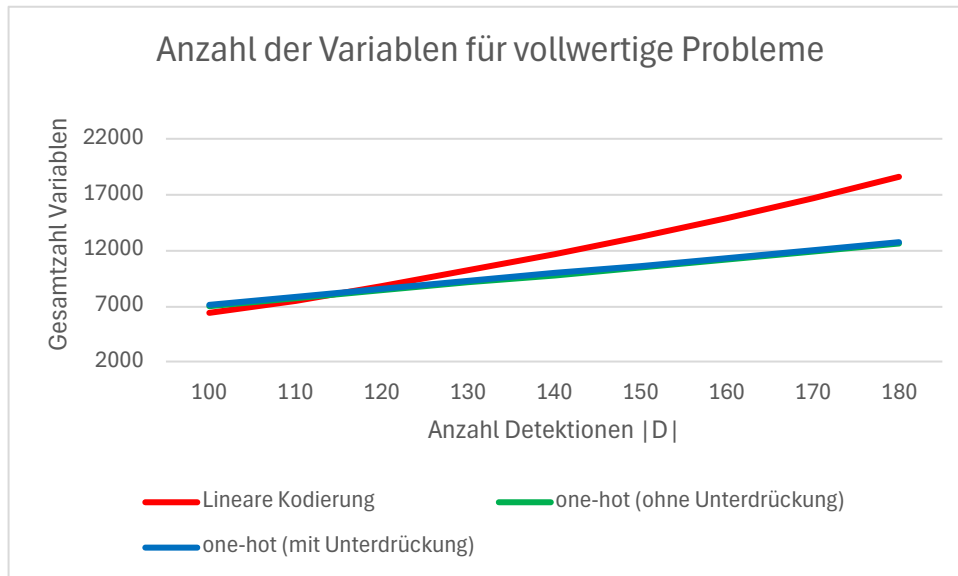


Abbildung 3: Übersicht der Gesamtzahl an Variablen / logischen Qubits in Abhängigkeit der Gesamtzahl der Detektionen $|D|$ je nach verwendeter Kodierung des Problems (mit $|C| = 14$, $|S| = 5$). Blaue und grüne Linie sind fast deckungsgleich in dieser Darstellung.

Tabelle 2 zeigt in Abhängigkeit der Kodierung die Berechnung der Gesamtzahl der Variablen je nach Komplexität des Segmentierungs- und Klassifizierungsproblems. Da zur Umsetzung der linearen klassischen Kodierung auf einem Quantenannealer viele Hilfsvariablen benötigt werden, ist diese Kodierung durch die folgende signifikante Steigerung der Gesamtvariablen für diesen Anwendungsfall nicht sinnvoll.

Kodierung	Formel Variablen (ohne Hilfsvariablen für QA)	Formel Variablen (mit Hilfsvariablen für QA)	Wert vollwertiges Problem ($ D = 140$, $ C = 14$, $ S = 5$)
Linear klassisch	$ D \cdot C + \binom{D}{2}$	$ D \cdot C + \binom{D}{2} + D \cdot \binom{C}{2} + 2 \cdot \binom{D}{2} + 3 \cdot \binom{D}{3}$	11.690 / 1.386.630
QUBO ohne Unterdrückung	$ D \cdot C \cdot S $	nicht nötig	9800
QUBO mit Unterdrückung	$ D \cdot C \cdot S + D $	nicht nötig	9940

Tabelle 2: Übersicht Anzahl der Variablen in Abhängigkeit der gewählten Kodierung des Problems

Passende Kodierung für verschiedene Problemgrößen und Zielplattformen (TVH-AP5-A5.1)

Wird als Zielplattform ein Quantencomputer gewählt, und kein Quantenannealer, und lässt man Beschränkungen in der Mächtigkeit von Gateoperationen außen vor, so würde man für $|D| \leq 115$ wahrscheinlich die quantenmechanische Übersetzung der linearen klassischen Kodierung bevorzugen.

Nimmt man zusätzlich in Betracht, dass NISQ-Plattformen aktuell um die 100 physikalische Qubits besitzen und man Testinstanzen/Benchmarkinginstanzen für diese verfügbaren Plattformen generieren möchte, zeigt nachfolgende Tabelle 3 jedoch dass, je nach Instanz, die one-hot Kodierung Vorteile bieten kann (im Vergleich zu Kodierungen, die auch eine Unterdrückung von Detektionen unterstützen).

D	C	S	Lineare Kodierung	one-hot (mit Unterdrückung)	one-hot (ohne Unterdrückung)	Lineare Kodierung als QUBO (für QA)
10	2	2	65	50	40	525
10	3	3	75	100	90	555
10	3	2	75	70	60	555
10	4	3	85	130	120	595
10	5	2	95	110	100	645
15	2	2	135	75	60	1725
15	2	3	135	105	90	1725
15	5	3	180	240	225	1905
20	2	2	230	100	80	4050
30	4	2	555	270	240	13785

Tabelle 3: Übersicht der Gesamtzahl an Variablen je nach Kodierung für Probleminstanzen, die auf NISQ-Quantencomputern und Quantenannealern ausgeführt werden können.

Ist jedoch die Zielplattform ein Quantenannealer, und werden Hilfsvariablen für eine gültige Kodierung der klassischen linearen Kodierung auf dieser Plattform benötigt, so steigt die Gesamtzahl der benötigten logischen Qubits immens (bis zu 2 Größenordnungen, siehe obige Tabelle). Daher wurden im Rahmen des TVH bei Benchmarks von möglichst realistische/realitätsnahe Problemgrößen auf Quantenannealern immer eine one-hot Kodierung gewählt.

2.3.6 Implementierung der Quantenschaltkreise

Bei der Implementierung und für die Simulation der Quantenschaltkreise wurde IBM Qiskit verwendet [Javadi-Abhari2024]. Bei der Umsetzung der one-hot Kodierung wurde zusätzlich noch die Zwischenschicht Docplex [Docplex] verwendet, da aus einer Docplex-Formulierung als klassischen Referenzlöser IBM CPLEX [CPLEX] aufgerufen werden kann und auch eine automatische Übersetzung in Ising-Modelle zur Lösung mit QAOA möglich ist. Teilweise wurden Skalierungsprobleme des Zustandsvektorsimulators von Qiskit beobachtet. In diesem Fall fand eine Simulation mit Hilfe von openQAOA [Sharma2022] statt. Im Fall der quantenmechanischen Umsetzung der linearen klassischen Kodierung wurde nur Qiskit (ohne Docplex-Zwischenschicht) verwendet.

2.3.7 Erstellung skalierbarer realistischer Instanzen zur Leistungsanalyse (TVH-AP5-A5.2)

Für die Entwicklung, das Testen und für die Leistungsanalyse der entwickelten hybriden Quantenalgorithmen wurden verschiedene Instanzen für Testzwecke begleitend zur Entwicklung der Implementierung der Quantenalgorithmen und deren Leistungsanalyse generiert.

Bei der Generierung dieser Instanzen wurde besonders darauf geachtet, dass zur Lösung der Segmentierungs- und Klassifikationsprobleme für diese Instanzen eine bestimmte Zahl von Qubits nötig sind, um auch ein Skalierungsverhalten des Quantenalgorithmus auf realen Quantencomputern/-annealern besser untersuchen zu können.

Dafür wurden die Ausgabedaten des klassischen CNN durch einen zusätzlichen Prozessierungsschritt beschränkt.

Diese Beschränkung verlief meist in zwei Schritten. Zuerst wurde die Anzahl an Detektionen reduziert. Bei einem manuell erstellten Datensatz wurden händisch die Position von Körperteilen von Menschen auf Bildern pixelgenau annotiert (soweit diese Annotationen nicht schon vorlagen). Die vom CNN vorgeschlagenen Detektionen können dann reduziert werden, indem die Detektionen mit dem kleinsten Abstand zu den manuell annotierten Detektionen ausgewählt werden. Abbildung 4 zeigt alle Detektionen, die vom CNN ausgegeben werden als schwarze Punkte pixelgenau auf dem unterliegenden Bild. Weiter zeigen die als Sterne gekennzeichneten Detektionen die Detektionen für

zwei Radfahrer an, die am nächsten zu deren annotierter Kopfspitze und linkem Knie liegen. Werden nur diese vier Detektionen und zwei Körperteilklassen (Kopf und linkes Knie) genutzt, um ein Optimierungsproblem zu erstellen, werden zur Lösung auf einem Quantencomputer 16 Qubits in der one-hot Kodierung ohne Unterdrückung benötigt ($|D| = 4$, $|C| = 2$, $|S| = 2$). Über ein ähnliches Vorgehen wurden dann auch weitere Testinstanzen erstellt, bei denen z.B. der Kopf und die Hüfte von drei Personen auf einem Bild segmentiert und klassifiziert werden müssen.

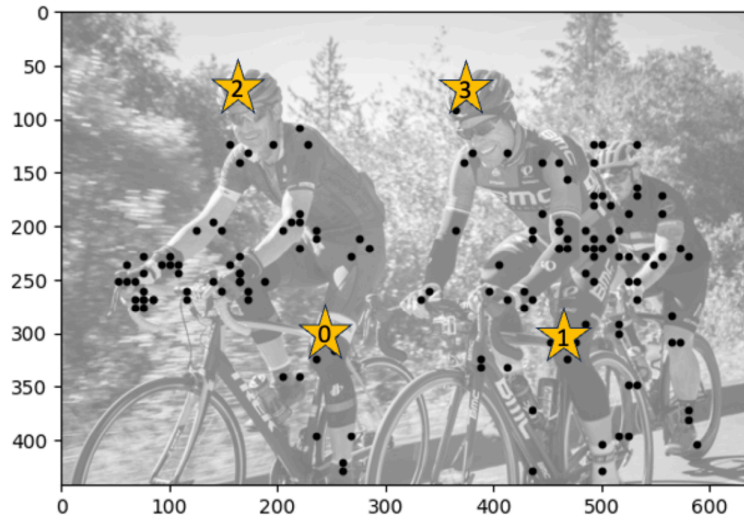


Abbildung 4: Die vom CNN ausgegebenen Detektionen werden als schwarze Punkte dargestellt. Die Sterne heben eine Reduktion der Instanzengröße hervor, die ein Teilproblem ergeben, bei dem lediglich der Kopf und das linke Knie von zwei Personen erkannt werden soll.

2.4 Referenzimplementierung über Antwortmengenprogrammierung

Um eine Alternative für die freie Implementierung des klassischen Referenzlösers CPLEX von IBM zu haben, der in seiner kostenlosen Testversion auf 1000 Variablen beschränkt ist, wurde eine Implementierung in der Modellierungssprache des Antwortmengenlösers *clingo* [Gebser2019] entwickelt. Dabei wurden Kodierungen in der Antwortmengenprogrammierung erstellt, die sowohl mit der klassischen linearen Kodierung bzw. der one-hot Kodierung vergleichbar sind. Mit Hilfe dieser Implementierungen konnten dann die Lösungen der Quantenalgorithmen validiert werden und optimale Lösungen nachgewiesen werden.

2.5 Ergebnisse für QAOA mit linearer Kodierung (TVH-AP4-A4.1)

Mit der linearen Kodierung des Anwendungsfalls wurden verschiedene Tests durchgeführt, um ein erstmaliges Verständnis der Herausforderungen bei der Verwendung von Quantencomputern bzw. QAOA zur Lösung des Anwendungsfalls besser zu verstehen. Abbildung 5 zeigt hier beispielhaft Ergebnisse für eine 14 Qubit Testinstanz. Weitergehende Abschätzungen zeigten jedoch, dass eine verbesserte Kodierung des Problems nötig ist, um erstens die Zahl der benötigten logischen Qubits zu senken und zweitens eine effiziente Ausführung auf einem Quantenannealer zu gewährleisten. Der Quantenannealer ermöglicht es dann auch, größere Probleminstanzen testen zu können (größere Anzahl physikalischer Qubits). Daher wurden die weiteren Arbeiten auf die Untersuchung der one-hot Kodierung fokussiert.

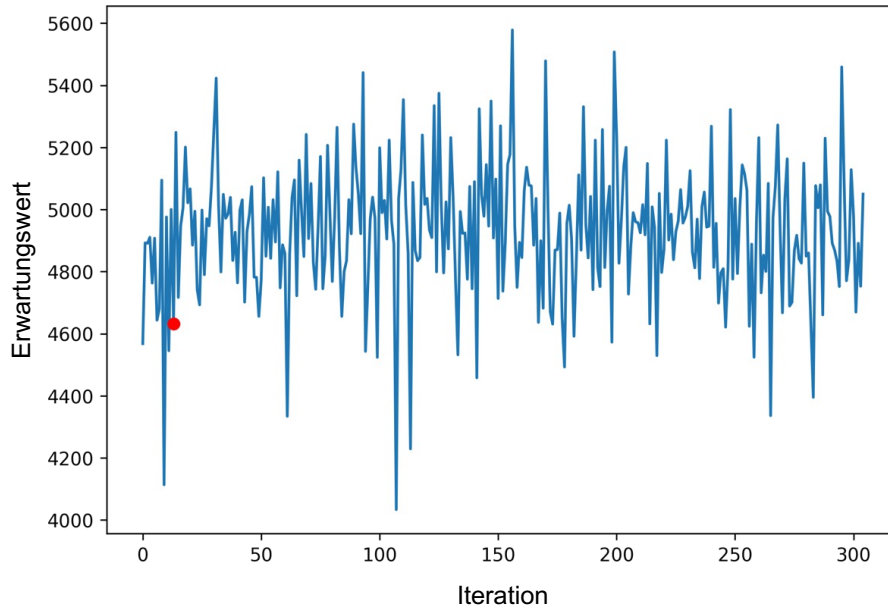


Abbildung 5: Verlauf des Erwartungswertes bei der Ausführung des QAOA ($p = 2$) für das Segmentierungs- und Klassifizierungsproblem für eine Probleminstanz mit 14 Qubits unter Verwendung von Bayesian Optimization als klassischer Optimierer. Der rote Punkt zeigt den Iterationsschritt bei dem die optimale Lösung zum ersten Mal gemessen wurde.

2.6 Technische Leistungsanalyse für QAOA mit one-hot Kodierung (TVH-AP5-A5.2)

Bei der Untersuchung von QAOA mit der one-hot Kodierung wurde der QASM-Simulator von openQAOA [openQAOA] gegenüber der Zustandsvektorsimulation von Qiskit bevorzugt, da dieser Laufzeitvorteile bei der Simulation der Quantenschaltkreise auf den zur Verfügung gestandenen klassischen Computern hatte. Initiale Experimente zeigten, dass eine Abtastung (engl. sampling) bei der Messung der Ergebnisse der Quantenschaltkreise von 10.000 Messungen erfolgen sollte, da bei ca. 1000 Messungen das numerische Rauschen noch zu groß war. Weiterhin zeigten Tests, dass es sinnvoll ist, hohe α_{dc} Werte auf den höchsten $\beta_{d'd'cc'}$ Wert zu setzen und weitere Gewichtungen im Ising-Hamiltonian auf den Maximalwert von 1 zu normalisieren. Dies führte bei der Untersuchung der Instanzen zu einer „Schmetterlings“/XOR-förmigen Energielandschaft für $p = 1$, was auch eine instanzunabhängige Parameterinitialisierungsstrategie begünstigen kann. Eine aus dem Quantenannealing bekannte Parameterinitialisierungsstrategie wurde näher untersucht [Willsch2020] und die beste gefundene Parametrisierung wurde in den nachfolgenden Experimenten genutzt.

Da der untersuchte Anwendungsfall Nebenbedingungen aufweist und zusätzlich eine one-hot Kodierung verwendet, konnte für die Verwendung eines XY-Mixers auf die Arbeiten von [Wang2020] aufgebaut werden.

2.6.1 Ergebnisse auf idealem Quantencomputer

Abbildung 6 zeigt beispielhaft den Vergleich von X - und XY -Mixern im QAOA für eine 16 Qubit Instanz ($|D| = 4$, $|C| = 2$, $|S| = 2$) des Segmentierungs- und Klassifizierungsproblems.

Wie erwartet werden bei der Verwendung des XY -Mixer nur gültige Lösungen gezogen, während für $p \leq 4$ beim X -Mixer auch mehr Lösungen mit hoher Wahrscheinlichkeit vorkommen, welche die Nebenbedingungen verletzen. Weiter ist für $p \leq 4$ die Wahrscheinlichkeit, die optimalen Lösungen bei der Verwendung des XY -Mixer zu finden, eine Größenordnung höher als bei der Verwendung des X -Mixers. Erst ab $p = 8$ zeigt der X -Mixer eine ähnliche Wahrscheinlichkeitsverteilung auf wie der XY -Mixer.

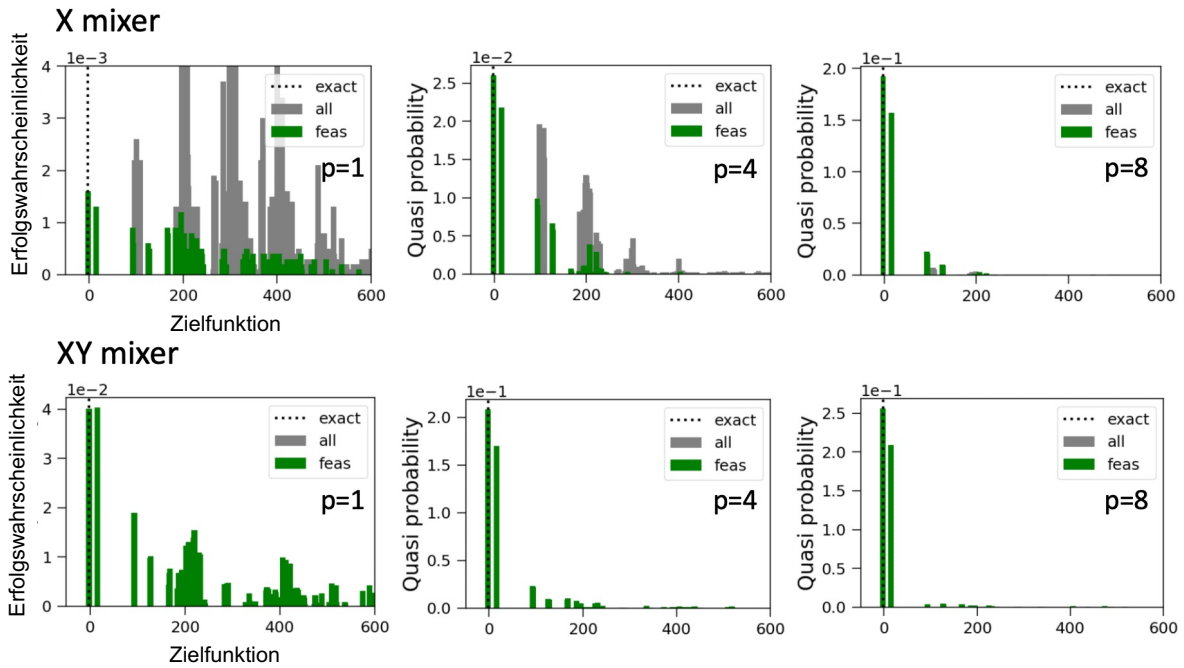


Abbildung 6: Darstellung der Wahrscheinlichkeitsverteilung für Lösungen verschiedener Optimalität bei verschiedenen Quantenschaltkreistiefen p unter Verwendung des X- bzw. XY-Mixers in der Simulation eines fehlerfreien Quantencomputers mittels Qiskit. Graue Balken zeigen Lösungen an bei denen Nebenbedingungen verletzt sind, grüne Balken zeigen Lösungen bei denen die Nebenbedingungen erfüllt sind. Die gestrichelte Linie zeigt die optimale Lösung an.

Um den X- bzw. XY-Mixer weiter zu vergleichen zeigt Abbildung 7 für 10 verschiedene Probleminstanzen je Problemgröße (16 und 24 Variablen/logische Qubits) jeweils die Erfolgswahrscheinlichkeit über der Quantenschaltkreistiefe p . Unabhängig von der Wahl der Parameter im Schaltkreis (ob über die standardmäßige QAOA-Parameteroptimierungsschleife oder der aus dem Quantenannealing bekannten Parameterinitialisierungsstrategie), zeigt der XY-Mixer immer eine höhere Erfolgswahrscheinlichkeit als der X-Mixer.

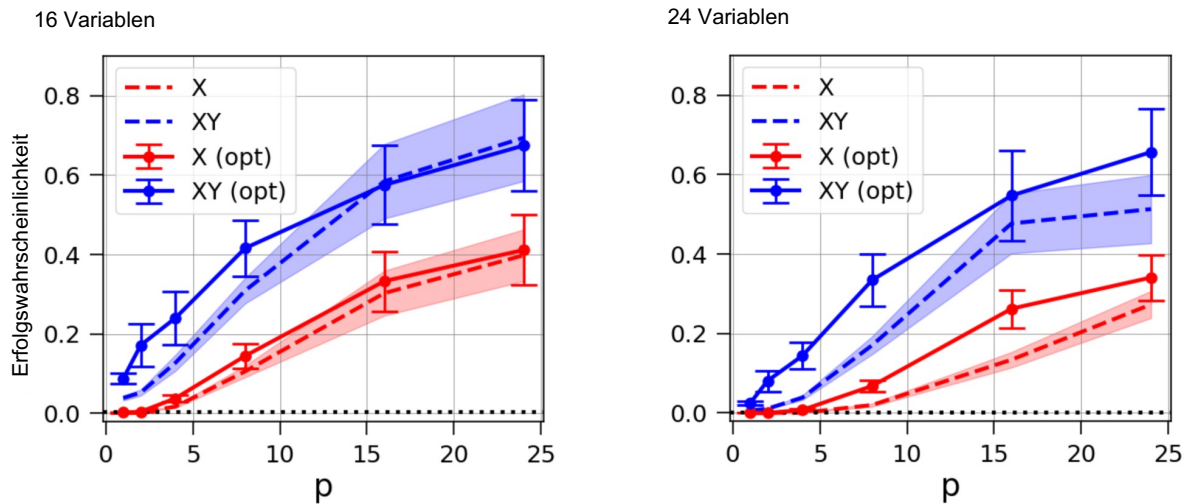


Abbildung 7: Erfolgswahrscheinlichkeit für X- bzw. XY-Mixer gemittelt je über 10 Probleminstanzen der jeweiligen definierten Problemgröße. Gestrichelte Linien bzw. durchgezogenen Linien zeigen den Mittelwert und farblich gefüllte Flächen bzw. Fehlerbalken die Standardabweichung. Gestrichelte Linien zeigen die Ergebnisse mit einer linearen Parameterinitialisierung des QAOA-Quantenschaltkreises (inspiriert aus dem Quantenannealing [Willsch2020]), während die durchgezogenen Linien die Ergebnisse der klassischen Parameteroptimierung bei der Verwendung von QAOA zeigen.

2.6.2 Simulation verrauschter Quantencomputer

Anknüpfend an die Ergebnisse der rauschfreien Simulation, die zeigt, dass die Verwendung eines XY -Mixers Vorteile bieten kann, sind im Quantenschaltkreis, im Vergleich zum X -Mixer, mehr Gateoperationen nötig. Dies ist unter anderem auf die aufwändigere Anfangszustandspräparierung und den Mix-Hamiltonian zurückzuführen. Für verschiedene Anzahlen von Qubits/Problemgrößen von Instanzen des Segmentierungs- und Klassifikationsproblems (bis 32 Qubits) und verschiedene Werte der QAOA-Tiefe p (mit $p \leq 15$) wurde die Quantenschaltkreistiefe nach einem Transpiling auf den von IBM in Qiskit bereitgestellten verrauschten *FakeGuadalupeV2* Emulator untersucht. Qualitativ ist die Schaltkreistiefe in Abhängigkeit der Problemgröße mehr als doppelt so groß bei der Verwendung eines XY -Mixer als beim X -Mixer. Das gleiche Verhalten wird auch in Abhängigkeit der QAOA-Tiefe p beobachtet. Es sei angemerkt, dass der Unterschied zwischen XY -Mixer und X -Mixer aber um mehr als das zweifache steigt, sobald die Anzahl der Qubits bzw. p größer wird.

Abbildung 8 zeigt exemplarisch die Ergebnisse, im Vergleich zu Abbildung 6, wenn die Berechnungen auf einem simulierten verrauschten Quantencomputer (*FakeGuadalupeV2*) durchgeführt werden. Es ist deutlich zu erkennen, dass sich die Wahrscheinlichkeit, die optimale bzw. gültige Lösungen zu ziehen verschlechtert. Durch Gatterfehler bei der Ausführung der Schaltkreise kommt es auch zu Fehlern im XY -Mixer, wodurch jetzt auch ungültige Lösungen gezogen werden können. Des Weiteren zeichnet sich ab, dass der X -Mixer unter Umständen besser geeignet ist, die optimale Lösung mit höherer Wahrscheinlichkeit auf verrauschten Quantencomputern zu finden.

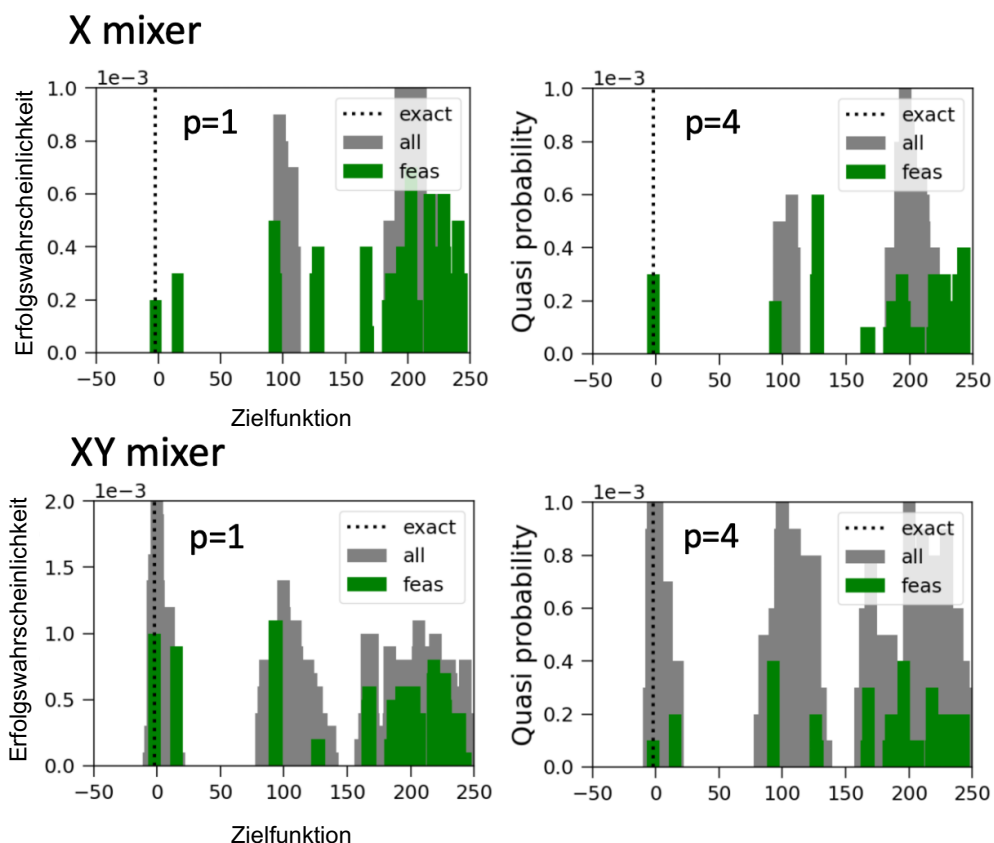


Abbildung 8: Darstellung der Wahrscheinlichkeitsverteilung für Lösungen verschiedener Optimalität bei verschiedenen Quantenschaltkreistiefen p unter Verwendung des X - bzw. XY -Mixers in der Simulation eines verrauschten Quantencomputers (*FakeGuadalupeV2*) mittels Qiskit. Graue Balken zeigen Lösungen an bei denen Nebenbedingungen verletzt sind, grüne Balken zeigen Lösungen bei denen die Nebenbedingungen erfüllt sind. Die gestrichelte Linie zeigt die optimale Lösung an.

2.6.3 Ergebnisse auf realen Quantencomputern

Anknüpfend an die simulierten Ergebnisse auf verrauschten Quantencomputern aus dem vorigen Unterkapitel wurde exemplarisch eine Instanz mit 16 Variablen/Qubits auf drei unterschiedlichen realen Quantencomputern berechnet. Dabei wurde der X-Mixer verwendet und 10.000 Messungen bei der Bestimmung des Erwartungswertes benutzt. Aus der fehlerfreien/rauschfreien Simulation in Abbildung 6 wurde weiter abgeleitet, dass $p = 8$ zu guten Ergebnissen führen kann.

Tabelle 4 zeigt die Erfolgswahrscheinlichkeit, die optimale Lösung je nach verwendeten realem Quantencomputer/-simulator bzw. mit fehlerfreier Simulation zu ziehen. Die Ergebnisse zeigen, dass die supraleitenden Quantencomputer von IBM und Rigetti eine deutlich schlechtere Erfolgswahrscheinlichkeit liefern, als der Ionenfallenquantencomputer von IonQ. Insgesamt zeigen jedoch alle realen Quantencomputer eine deutlich niedrigere Erfolgswahrscheinlichkeit als die Referenzberechnung mit Hilfe der fehlerfreien Quantensimulation.

Quantencomputer / -emulator	Erfolgswahrscheinlichkeit
IBM Qiskit fehlerfreier QASM Simulator	37.99 %
IBM Qiskit verrauschter FakeGuadelupeV2 Simulator	0.01 %
IBM Brisbane	0.04 %
Rigetti AspenM3	0.01 %
IonQ Aria2	2.00 %

Tabelle 4: Erfolgswahrscheinlichkeit, die optimale Lösung einer Benchmarkinginstanz auf verschiedenen realen Quantencomputern zu messen.

2.6.4 Ergebnisse auf realem D-Wave Quantenannealer

Da auf den verfügbaren Quantencomputern und insbesondere Quantensimulatoren nur mit einer sehr beschränkten Anzahl von Qubits gearbeitet werden konnte, wurden weitere Experimente mit größeren Probleminstanzen auf dem D-Wave Quantenannealer durchgeführt (QPU Advantage System 5.4 mit mindestens 5.000 physikalischen supraleitenden Qubits). Die Ausglühzeit (engl. annealing time) war $20\mu s$ und es wurden eine implizite Abtastung von fünf unabhängigen Läufen mit je 2.000 Messungen gewählt. Für die Einbettung (engl. embedding) wurde der stochastische *MinorMiner* Algorithmus von D-Wave verwendet. Dabei wurde aus zehn Läufen das Ergebnis mit der kleinsten Kettenlänge (engl. chain length) gewählt, das als Eingabe für die *FixedEmbeddingComposite* Routine fungierte. Des Weiteren wurden die Hyperparameter *chain_strength*, *penalty_strength* und *annealing_time* für verschiedene Problemgrößen getestet bzw. optimiert.

Abbildung 9 zeigt die Erfolgswahrscheinlichkeit gemittelt für jeweils zehn Probleminstanzen bei unterschiedlicher Problemgröße für das Segmentierungs- und Klassifizierungsproblem. Das klassisch simulierte Auskühlen (in der Implementierung von D-Wave, engl. simulated annealing (SA)) wurde als klassischer Referenzalgorithmus verwendet. Während für kleinere Probleminstanzen das Quantenannealing mit dem simulierten Abkühlen gleichaufliegt, steigt die Diskrepanz für größere Instanzen deutlich.

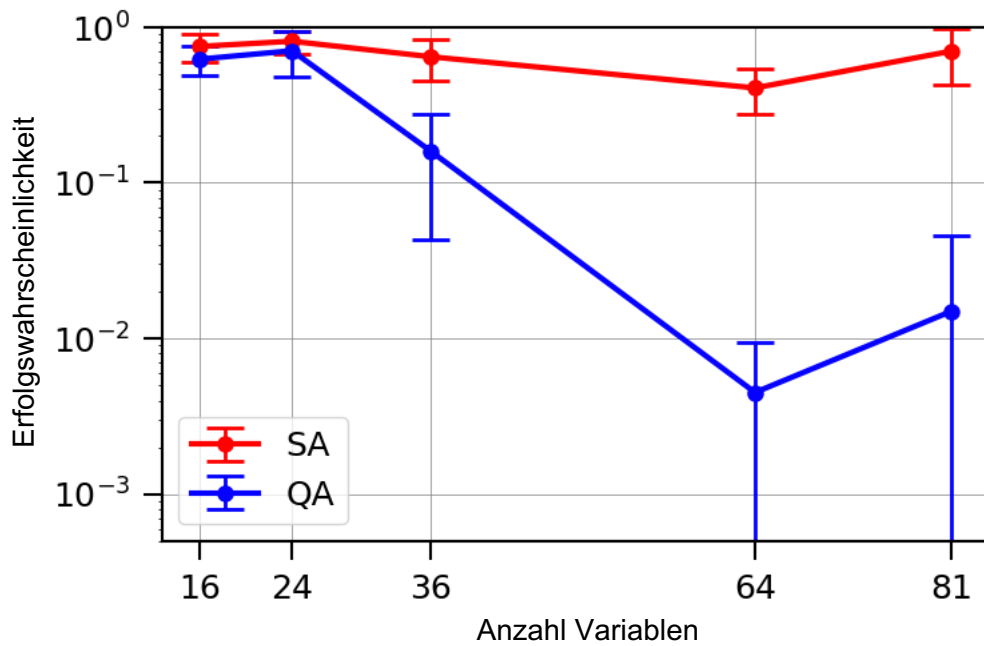


Abbildung 9: Darstellung der Erfolgswahrscheinlichkeit, die optimale Lösung in Abhängigkeit von der Problemgröße zu ziehen. Es wurde jeweils über zehn Probleminstanzen gemittelt. Fehlerbalken zeigen die Standardabweichung.

Eine Erklärung, warum bei steigender Anzahl von Variablen/logischen Qubits die Erfolgswahrscheinlichkeit abnimmt, könnte Abbildung 10 liefern. Hintergrund ist das die Wechselwirkungen der Qubits im Ising-Modell nicht nativ auf die Konnektivität des D-Wave Quantenannealer abgebildet werden können. Ab ca. 81 Variablen werden somit physische Qubitkettenlängen größer als zehn erreicht, wobei schon ab einer Kettenlänge von sieben Warnungen des D-Wave Programme ausgegeben werden.

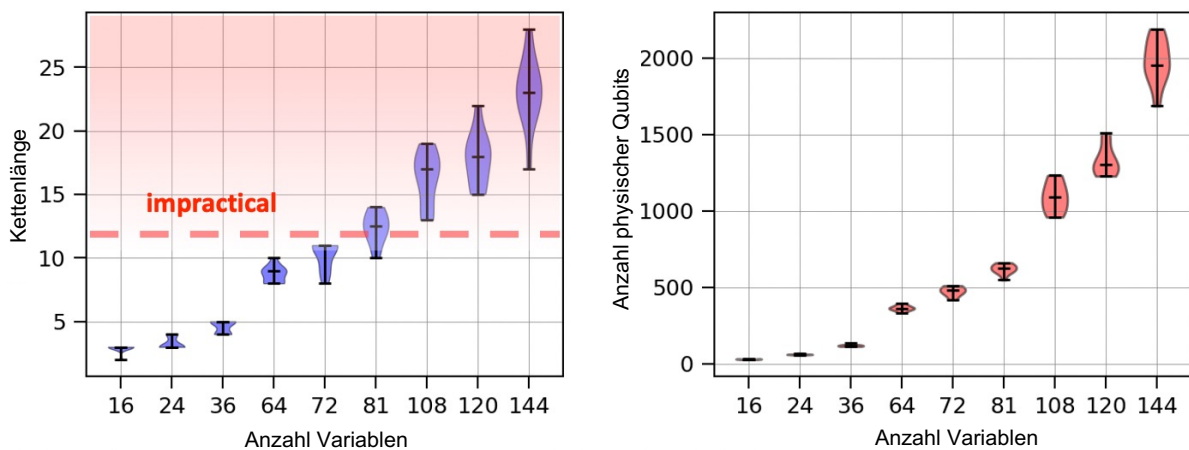


Abbildung 10: Qubitkettenlänge bzw. Anzahl der physischen Qubits nach der Einbettung in Abhängigkeit der Anzahl von Variablen/logischen Qubits/Problemgröße. Es werden Mittelwert gezeigt bzw. die Standardabweichung über 10 Ausführungen des stochastischen MinorMinor Algorithmus zur Einbettung von DWave.

2.6.5 Ergebnisse Vergleich QAOA, Quantenannealer und simulierte Abkühlung

Abbildung 11 zeigt den direkten Vergleich der bereits oben diskutierten (Quanten)Methoden QAOA mit X- und XY-Mixer auf idealem Quantencomputer, quantengestützte Abkühlung auf dem D-Wave Quantenannealer und klassischer simulierter Abkühlung. Es wurden jeweils 10 unterschiedliche Instanzen mit 16 bzw. 25 Qubits/Variablen untersucht. Wie sich erkennen lässt, ist die klassisch simulierte Abkühlung die Methode mit der höchsten mittleren Erfolgswahrscheinlichkeit. QAOA mit XY-Mixer zeigt für hohe Werte von p (hohe Quantenschaltkreistiefen) eine zum Quantenannealer

vergleichbare mittlere Erfolgswahrscheinlichkeit, und das trotz der Tatsache, dass dieser nicht fehlerkorrigiert ist und nicht intrinsisch die Nebenbedingungen („one-hot“) erfüllt.

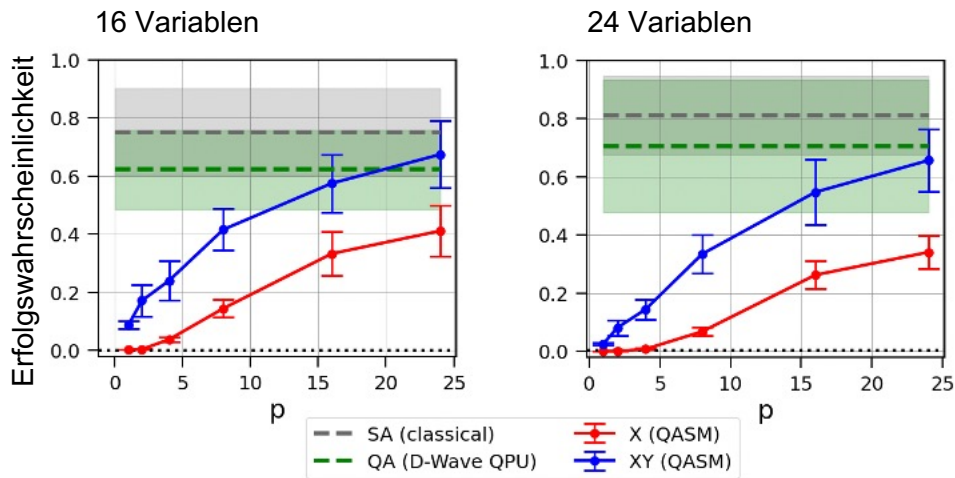


Abbildung 11: Mittlere Erfolgswahrscheinlichkeiten über 10 verschiedene Probleminstanzen, jeweils für die Problemgrößen von 16 bzw. 24 Variablen über Quantenschaltkreistiefen p für QAOA bei der Verwendung eines X- bzw. XY- Mixers bei der Simulation eines idealen Quantencomputers. Weiterhin sind die (konstanten) Erfolgswahrscheinlichkeiten für den D-Wave Quantenannealer (QA) und die klassische Referenzimplementierung der klassischen Abkühlung (SA) aufgetragen. Fehlerbalken und durchsichtige Flächen zeigen die Standardabweichung.

3. Anwendungsspezifische QSVM zur Klassifikation (TVH-AP3)

3.1 Zusammenarbeit mit anderen Partnern

Die nachfolgend dargestellten Ausarbeitungen und Ergebnisse entstanden in enger Kooperation mit dem Projektpartner Jülich Supercomputing Centre (JSC) und Mercedes Benz.

3.2 Kurzbeschreibung des Anwendungsfall

Der Anwendungsfall der binären Klassifikation von Schweißpunkten betrachtet, ob die Qualität von Schweißpunkten, die zwei Metallplatten verbinden, „ausreichend gut“ ist. Dies bedeutet letztlich, dass diese korrekt miteinander verbunden sind und ein Werkstück weiterverarbeitet werden kann. Während des Schweißvorgangs, für jeden individuellen Schweißpunkt, werden verschiedene physikalische Parameter aufgezeichnet, die im Folgenden in den (Quanten)Algorithmen als Merkmale (engl. features) genutzt werden. Im Gegensatz zu synthetischen Daten, wurde die im weiteren Verlauf betrachteten Daten(-sätze) in realen industriellen Fertigungsprozessen aufgezeichnet.

3.3 Software-Bibliothek zur effektiven Untersuchung von verschiedenen Quantenkernen

Um auf bestehender Literatur der quantengestützten Stützvektormaschine [Havlíček2019, Schuld2019] (engl. quantum support vector machine (QSVM)) aufzubauen und diese Arbeiten weiterzuführen, wurde eine Softwarebibliothek in Python entwickelt, mit deren Hilfe sich die kodierenden Quantenschaltkreise V (oder in diesem Falle „quantum feature maps“) in Kernelementen

$$\kappa(x, y) = |\langle 0 | V^\dagger(y)V(x) | 0 \rangle|^2$$

zwischen zwei Merkmalvektoren x und y (engl. feature vectors) einfach und effizient parametrisieren lassen.

Auf eine kombinatorische Art und Weise konnten dann einfach die Gesamtzahl der zu verwendenden Qubits festgelegt und die Quantengatter definiert werden, welche die klassischen Daten in Quantenzuständen kodieren. Für eine Auswahl an geeigneten kodierenden Quantengattern wurde

sich dabei an den Erfahrungen aus [Sim2019] orientiert. Weiterhin wurde parametrisiert, wie oft diese Kodierung wiederholt werden soll und ob klassische Daten noch mit einem Parameter α skaliert werden sollen.

Abbildung 12 zeigt exemplarisch einen parametrierbaren Quantenschaltkreis V über vier Qubits.

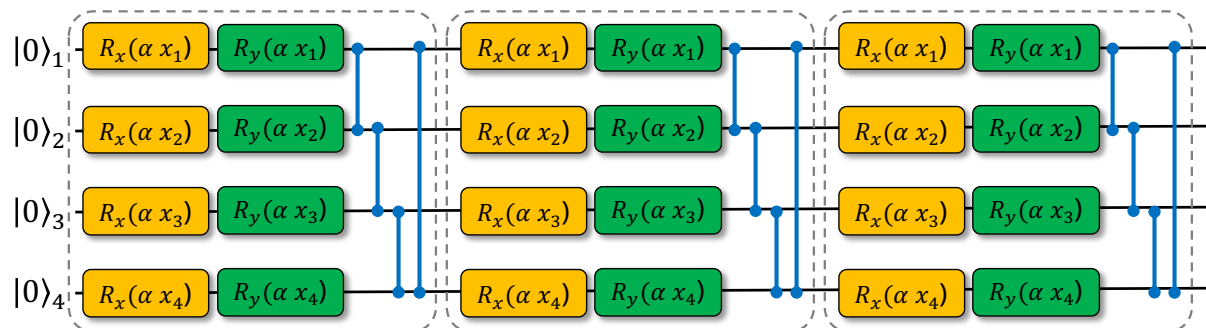


Abbildung 12: Beispiel eines Quantenschaltkreis V als Teil zur Berechnung eines Kernelements auf dem klassischen Datenvektor x (mit Elementen x_i). Der Quantenschaltkreis umfasst vier Qubits, in deren Zustand die klassischen Daten x_i in den gestrichelten Boxen kodiert werden, was dreimal wiederholt wird. Die Kodierung von klassischen Daten wird über eine Winkelkodierung durch Einqubitgatter (Einqubitrotationen) R_x (gelb) und R_y (grün) realisiert mit anschließenden paarweisen Gattern, die zu einer Verschränkung von Quantenzuständen führen können.

3.4 Datenvorverarbeitung, Training- und Testprotokoll

Für die Datenvorverarbeitung sei der Begriff des „Datensatz“ definiert. Dieser umfasst die Gesamtheit der Daten, die für *ein gesamtes Experiment* genutzt werden, um z.B. die Klassifikationsgüte eines Klassifikators für diesen Datensatz zu bestimmen. Jeder Datensatz enthält mehrere Datenpunkte.

Für den Anwendungsfall „Klassifikation von Schweißpunkten“ unterscheiden wir zwei Typen von Datensätzen: (1) ein Datensatz enthält nur Datenpunkte, die über den Schweißprozess von zwei definierten Metallplatten aufgezeichnet wurden – im Folgenden „paarweiser Datensatz“ genannt. (2) Ein Datensatz enthält Datenpunkte bei denen verschiedene Typen von Metallplatten über Schweißpunkte paarweise verbunden wurden.

Im Fall (1) wird somit ein Klassifikator bestimmt, der speziell auf zwei Typen von Metallplatten trainiert wurde. Im Fall (2) werden Klassifikatoren bestimmt die im Idealfall über mehrere Metallplattentypen generalisieren können. In den im Folgenden gezeigten Arbeiten wurde sich auf Datensätze von Fall (1) konzentriert, da keine zusätzlichen Qubits zur Kodierung des Materialtyps benötigt werden (welche linear steigt mit der Anzahl von Materialtypen).

Für den Anwendungsfall der Klassifikation von Schweißpunkten standen 20 verschiedene paarweise Datensätze zur Verfügung. Jeder der Datensätze umfasst Datenpunkte mit je 8 verschiedenen Merkmalen, wobei die Gesamtzahl von Datenpunkten je Datensatz im Bereich von 109 bis 789 Datenpunkten lag. Bei jedem Datensatz handelt es sich um einen nicht balancierten Datensatz, mit Dysbalancen zwischen guten und schlechten Schweißpunkten im Bereich von 3% bis 99%, je nach paarweisen Datensatz.

In der Datenvorverarbeitung werden zunächst die Merkmale (engl. features) der Datenpunkte eines Datensatz in ihrem Wertebereich über eine min-max-Skalierung² auf den Wertebereich $[-1,1]$ skaliert. Weiterhin wird jeder Datensatz zu 80% in eine Trainingsmenge von Datenpunkten und die übrigen 20% in eine Testmenge von Datenpunkten aufgeteilt. Ein Klassifikator wird auf der Trainingsmenge trainiert und abschließend seine Leistung auf der Testmenge von Datenpunkten evaluiert.

² sklearn.preprocessing.MinMaxScaler

Für jeden Datensatz wurde eine kombinatorischen Hyperparameteroptimierung per Gittersuche³ (engl. grid search) durchgeführt, die neben verschiedene Quantenkernelvorlagen (engl. quantum kernel templates) auch andere bekannte Hyperparameter aus der klassischen Stützvektormaschinentheorie abdeckte. Innerhalb der Gittersuche auf den Trainingsdaten wurde, zur potenziellen Steigerung der Generalisierung, eine Bewertung der verschiedenen Parameterbelegungen ein gleichgewichtet Kreuzvalidierungsverfahren⁴ (engl. stratified cross validation) genutzt.

Als entscheidende Metrik wurde dann die mittlere ausgewogene Genauigkeit (engl. mean balanced accuracy) aus dem Kreuzvalidierungsverfahren genutzt, um die beste Parameterkombination aus den Ergebnissen der Gittersuche zu bestimmen. Abschließend wurde dann ein Klassifikator mit der besten Parameterkombination auf allen Trainingsdaten eines Datensatz trainiert und die Klassifikationsgüte auf den jeweiligen Testdaten bestimmt.

3.5 Ergebnisse zur Stützvektormaschine mit Quantenkernel

Für die im folgenden gezeigten Ergebnisse der paarweisen Datensätze wurden bei der Gittersuche folgenden Parameterraum vollständig kombinatorisch exploriert:

- $\alpha \in [1.0, 2.0, 3.0, 4.0]$
- $n_{layer} \in [1,2]$ (Anzahl Wiederholungen der nachfolgenden *feature-maps*)
- *feature-maps* $\in [$
 - ['rx', 'cx']
 - ['rx', 'cz']
 - ['rx', 'swap']
 - ['rx', 'cry']
 - ['rx', 'crz']
 - ['rx', 'ryy']
 - ['rx', 'rzz']
 - ['rx', 'ry', 'cx']
 - ['rx', 'ry', 'cz']
 - ['rx', 'ry', 'crz']
 - ['rx', 'ry', 'swap']

Für den Parameter der *feature-maps* wurden jeweils bei Qubitdrehoperatoren (z.B. 'rx') auf den einzelnen Qubits q_i ($i = 1 \dots 8$) die klassischen Datenvektorenwerte x_i als Drehwinkel genutzt und die Verschränkung (z.B. durch 'cx') zwischen Qubits jeweils vollständig paarweise zwischen den Qubits durchgeführt.

Im klassischen Teil der hybriden quantengestützten Stützvektormaschine [Havlíček2019, Schuld2019] wurde der Parameter $C = 100$ gesetzt.

Als klassische Referenzimplementierung wurde die Stützvektormaschinenimplementierung von *scikit-learn* [Pedregosa2011] mit klassischem *rbf*-Kernel genutzt. Hierbei wurde auch eine Gittersuche über einen gängigen Parameterraum durchgeführt.

Tabelle 5 zeigt die gemittelten Ergebnisse über die Datensätze unter anderem für die quantengestützten Stützvektormaschine (siehe Spalte „QKE“) und die klassische Referenzimplementierung (siehe Spalte „RBF“).

Wie zu sehen ist, zeigen beide Methoden etwa die gleiche Klassifizierungsgüte, jedoch keinen deutlichen Vorteil für die Quantenmethode.

³ sklearn.model_selection. GridSearchCV

⁴ sklearn.model_selection. StratifiedKFold mit "3 folds" auf den Trainingsdaten und "shuffle=on"

Metrik	Mittlere ausgewogene Genauigkeit ⁵				Ausgewogene F1 Metrik (unterrepräsentierte Klasse) ⁶			
	QKE	pQKE	QKT	RBF	QKE	pQKE	QKT	RBF
Werte	0.66 ± 0.14	0.67 ± 0.15	0.65 ± 0.10	0.65 ± 0.13	0.57 ± 0.30	0.57 ± 0.32	0.55 ± 0.32	0.56 ± 0.30

Tabelle 5: Ergebnisse der quantengestützten Stützvektormaschine gemittelt über die verfügbaren paarweise Datensätze für verschiedene Quantenkernel-basierte Methoden.

3.6 Projizierende Quantenkernel

Aus der Literatur ist bekannt, dass zu expressive Quantenkernel die Klassifikationsgüte einer Quantenkernelstützvektormaschine verschlechtern können [Thanasilp2022].

In [Huang2021] wurden mit projizierenden Quantenkerneln (engl. projected quantum kernel estimation (pQKE)) eine Methode vorgestellt, bei der der Merkmalvektor, nach dem dieser in einen hochdimensionalen Hilbertraum kodiert wurde, wieder in einen niedrigdimensionalen Unterraum projiziert wird.

Tabelle 5 zeigt die über die paarweisen Datensätze gemittelten Ergebnisse einer Gittersuche mit dem gleichen Parameterraum wie die nicht-projizierenden Quantenkernel im vorangegangenen Unterkapitel. Wie aus Tabelle 5 zu entnehmen ist, ist die Verbesserung der Mittelwerte der Metriken, die in Tabelle 5 zu sehen sind, nur marginal gegenüber der rein klassischen Stützvektormaschine.

3.7 Quantenkerneltraining

Um die besten per Gittersuche gefunden Quantenkernel für die paarweisen Datensätze noch weiter zu verbessern, wurden die jeweils gefunden bestparametrisierten Quantenkernel einem Quantenkerneltraining (engl. quantum kernel training (QKT)) unterzogen [Glick2021]. Dafür wird der nun fest parametrisierte kodierende Quantenschaltkreis V um einen parametrisierten Quantenschaltkreis erweitert, dessen Parameter trainiert werden können. Ein parametrisiertes Kernelement ist dann wie folgt beschrieben:

$$\kappa_{\theta}(x, y) = |\langle 0 | R_y^{\dagger}(\theta) V^{\dagger}(y) V(x) R_y(\theta) | 0 \rangle|^2.$$

In unserem Fall wirkte die Einqubitrotation R_y mit demselben Winkel θ auf alle im Schaltkreis vorhandenen Qubits. Als Optimierer wurde COBYLA⁷ verwendet.

Tabelle 5 zeigt die über die paarweise Datensätze gemittelten Ergebnisse. Wie sich erkennen lässt, verbessern sich die Resultate im Mittel nicht entscheidend, sondern liegen leicht unter den bisher diskutierten Methoden.

4. Weitere Arbeiten im Teilvorhaben

4.1 Neue Methoden zur Schaltungseinbettung und Rauschminderung (TVH-AP4-A4.2)

Das TVH beschäftigte sich auch mit neuen Methoden zur Rauschminderung auf verrauschten Quantencomputern. Hierbei lag der Fokus auf neuen Methoden, die bei der Einbettung (engl. embedding) des logischen Quantenschaltkreises auf dem Zielquantencomputer den logischen Schaltkreis um möglichst wenige zusätzliche Quantengatter anreichern (z.B. SWAP-Gatter) [Cowntan2019, Wille2019]. Hierbei lag der Fokus auf der Entwicklung einer neuen Methode, welche die Optimalität von Lösungen, im Gegensatz zu Heuristiken [Cowntan2019], garantieren kann. Ausgangspunkt der neuen Methode ist die Antwortmengenprogrammierung zur Systemsynthese von klassischen Computern [Andres2013, Gebser2019]. In diesen Arbeiten wurde gezeigt, dass für klassische Computer bei einer hohen Dichte der Kommunikationsverbindungen die

⁵ sklearn.metrics.balanced_accuracy_score

⁶ sklearn.metrics.f1_score (unterrepräsentierte Klasse)

⁷ qiskit.algorithms.optimizers.COBYLA

Antwortmengenprogrammierung Vorteile in der Routenfindung von Nachrichten gegenüber SAT-Lösern (wie z.B. in [Wille2019]) haben kann. Dieser Vorteil in der Formulierung von Erreichbarkeit wird in der neuen Methode für die Ablauffolge von SWAP-Gatteroperationen auf Quantencomputern ausgenutzt.

Im Rahmen des TVH wurde eine Lösung mit Hilfe der Antwortmengenprogrammierung ausgearbeitet und die Ergebnisse mit den Lösungen anderer Arbeiten validiert [Wille2019]. Ein entsprechendes Patent wurde eingereicht. Für weitere Details zu den Arbeiten wird auf die Patentveröffentlichung verwiesen.

Die oben beschriebene Methode wurde in einem weiteren Schritt um ein noch weitergehendes Verfahren zur Rauschminderung erweitert. Ausgangspunkt kann hier eine Testroutine sein, die die Qualität oder Fehlerrate von physikalischen Qubits eines realen Quantencomputer charakterisiert. Anhand von priorisierten Metriken kann damit eine Liste von Qubits erstellt werden, die bevorzugt genutzt oder vermieden werden sollen (z.B. „stark verrauschte“ Qubits). Die oben beschriebene Methode zur Einbettung von logischen Schaltkreisen wird dann dahingehend erweitert, dass bei der Einbettung eines logischen Schaltkreises möglichst wenig zusätzlich Gateoperationen hinzukommen und anhand der Liste die Nutzung von physikalischen Qubits möglichst bevorzugt bzw. vermieden wird. Für weitere Details wird auf eine zweite Patentveröffentlichung verwiesen.

4.2 Quantengestützter Logik-Löser

Mit dem Ziel, die Leistung aktueller Logiklöser (z.B. SAT-Löser, SMT-Löser, Antwortmengenlöser) durch Hilfe von Quantentechnologie zu beschleunigen, wurde eine neue Methode patentiert, welche es erlaubt Quantencomputer und Quantenannealer mit klassischen Logiklösern zu koppeln. Hierbei löst der Quantencomputer/-annealer ein (Teil)Problem und übergibt seine Lösung an den Logiklöser, der die Variablenbelegung der Lösung verwendet, um nicht-deterministische Variablenbelegungen zu ersetzen. Aus Platzgründen wird der Leser für mehr Details auf die Patentanmeldung verwiesen.

Referenzen

[Insafutdinov16] Insafutdinov, Eldar et al., „DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model“, 2016.

[Bansal2004] Bansal, Nikhil et al., „Correlation clustering“, 2004

[Weggemans2022] Weggemans, Jordi et al., „Solving correlation clustering with QAOA and a Rydberg qudit system: a full-stack approach“, 2022.

[Levinkov2017] Levinkov, Evgeny et al., „Joint graph decomposition & node labeling: Problem, algorithms, applications“, 2017.

[Farhi2014] Farhi, Edward et al., „A quantum approximate optimization algorithm“, 2014.

[Vatan2004] Vatan, Farrokh et al., „Realization of a general three-qubit quantum gate“, 2004.

[Javadi-Abhari2024] Javadi-Abhari, Ali et al., „Quantum Computing with Qiskit“, 2024.

[Docplex] <https://ibmdecisionoptimization.github.io/docplex-doc/>

[CPLEX] <https://www.ibm.com/de-de/products/ilog-cplex-optimization-studio>

[Sharma2022] Sharma, Vishal et al., „OpenQAOA – An SDK for QAOA“, 2022.

- [Gebser2019] Gebser, Martin et al., „Multi-shot ASP solving with clingo“, 2019.
- [Willsch2020] Willsch, Madita et al., „Benchmarking the quantum approximate optimization algorithm“, 2020.
- [Wang2020] Wang, Zihui et al., „XY mixers: Analytical and numerical results for the quantum alternating operator ansatz“, 2020.
- [Havlíček2019] Havlíček, Vojtěch et al., „Supervised learning with quantum-enhanced feature spaces“, 2019.
- [Schuld2019] Schuld, Maria et al., „Quantum Machine Learning in Feature Hilbert Spaces“, 2019.
- [Sim2019] Sim, Sukin et al., „Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms“, 2019.
- [Pedregosa2011] Pedregosa, F. et al., „Scikit-learn: Machine Learning in Python“, 2011.
- [Thanasilp2022] Thanasilp, Supanut et al., „Exponential concentration and untrainability in quantum kernel methods“, 2022.
- [Huang2021] Huang, Hsin-Yuan et al., „Power of data in quantum machine learning“, 2021.
- [Glick2021] Glick, Jennifer et al., „Covariant quantum kernels for data with group structure“, 2021.
- [Cowtan2019] Cowtan, Alexander et al., „On the Qubit Routing Problem“, 2019.
- [Wille2019] Wille, Robert et al., „Mapping Quantum Circuits to IBM QX Architectures Using the Minimal Number of SWAP and H Operations“, 2019.
- [Andres2013] Andres, Benjamin et al., „Symbolic System Synthesis Using Answer Set Programming“, 2013.

Kurzbericht Verbundprojekt „QUANTUM ARTIFICIAL INTELLIGENCE FOR THE AUTOMOTIVE INDUSTRY Q(AI)²“ – Teilvorhaben Robert Bosch GmbH (Förderkennzeichen 13N15585)

Die im Teilvorhaben durchgeführten Arbeiten lassen sich im Wesentlichen in die vollständige Ausarbeitung von zwei neuen industriellen Anwendungsfällen für Quantencomputing gliedern. Der Anwendungsfall „Segmentierung und Klassifikation von Bildern“ umfasst die Identifikation der Pose eines oder mehrerer Menschen auf Bildern, z.B. für die Mensch-Maschine-Interaktion in der kollaborativen Fertigung. Weiterhin befasst sich der Anwendungsfall „Klassifikation von Schweißpunkten“ mit der Bestimmung der Qualität von Schweißpunkten zwischen zwei Metallplatten und ob diese bestimmten Qualitätskriterien genügen.

In enger Kooperation mit dem Projektpartner Jülich Supercomputing Centre (JSC) wurden Arbeiten zum Anwendungsfall „Segmentierung und Klassifikation von Bildern“ durchgeführt. Zuerst wurde der Anwendungsfall anhand existierender Literatur für klassische Verfahren (siehe ausführlicher Sachbericht) [Insafutdinov16] untersucht und der Anwendungsfall in geeigneter Form abstrahiert, sodass aus dieser Abstraktion ein quantenmechanischer Algorithmus ausgewählt bzw. entwickelt werden kann. In diesem Rahmen entstand auch eine Veröffentlichung zu einem Teilproblem der Segmentierung und Klassifikation [Weggemans2022]. Für den vollständigen Anwendungsfall wurden dann zuerst eine quantenphysikalische Implementierung unter Verwendung des QAOA („Quantum Approximate Optimization Algorithm“) [Farhi2014] entwickelt und getestet, welche auf einer linearen Problemkodierung abgeleitet aus der klassischen Literatur aufbaut. Da die Anzahl von *physikalischen* Qubits auf aktuellen verfügbaren Quantencomputern deutlich kleiner ist als die Anzahl der benötigten *logischen* Qubits für realistische Probleminstanzen, wurde ein Benchmarking von größeren Probleminstanzen mit dem D-Wave Quantenannealer in Jülich angestrebt. Bei genauer Untersuchung der linearen Kodierung stellte sich heraus, dass diese zu viele Hilfsvariablen bzw. zusätzliche Qubits benötigen würde, um die Kodierung des Problems auf dem Quantenannealer ausführen zu können. Daher wurde eine effizientere Kodierung entwickelt, welche sich nativ auf dem Quantenannealer umsetzen lässt. Abschließend wurden dann verschiedene Probleminstanzen in dieser Kodierung sowohl in quantenmechanischer Simulation als auch auf realen Quantencomputern und dem D-Wave Quantenannealer einem Benchmarking unterzogen.

Bei der rauschfreien Simulation der Quantenschaltkreise auf klassischen Computern wurde insbesondere der Einfluss von X- bzw. XY-Mixer im QAOA untersucht. Es zeigte sich, dass der Anwendungsfall der Segmentierung und Klassifizierung auch Nebenbedingungen enthält, bei denen der XY-Mixer eine bessere Performanz zeigt. Bei der Simulation mit einem Fehlermodell eines verrauschten Quantencomputer zeigte sich jedoch schnell, dass durch die zusätzliche benötigten Gatteroperationen im XY-Mixer dieser wegen des Einflusses von Gatterfehlern nicht mehr im Vorteil ist. Bei Experimenten auf verschiedenen realen Quantencomputern von IBM, Rigetti und IonQ wurde daher der X-Mixer genutzt und untersucht mit welcher Wahrscheinlichkeit die optimale Lösung von Probleminstanzen gefunden werden kann. Die Ergebnisse zeigen jedoch eine Erfolgswahrscheinlichkeit von weniger als 0,1% auf supraleitenden Quantencomputern und 2% auf einem Ionenfallen-Quantencomputer. Im Vergleich beträgt die Erfolgswahrscheinlichkeit knapp 40% im Rahmen einer fehlerfreien Simulation der Quantenschaltkreise. Desweiteren zeigten die Experimente auf dem Quantenannealer im Mittel bessere Ergebnisse, obwohl dieser nicht fehlerkorrigiert ist und nicht per Algorithmus (vgl. XY-Mixer) sichergestellt ist, dass die Nebenbedingungen des Anwendungsfall eingehalten werden. Ein direkter Vergleich eines fehlerkorrigierten Quantencomputers (per Simulation) mit dem nicht fehlerkorrigierten Quantenannealer zeigte,

dass erst bei Schaltkreistiefen von ca. $p = 10$ die Erfolgswahrscheinlichkeiten von QAOA mit dem XY-Mixer denen des Quantenannealers entsprechen.

Bei der Ausarbeitung des Anwendungsfalls der „Klassifikation von Schweißpunkten“ (in Kooperation mit JSC und Mercedes) wurde auf existierende Literatur aufgebaut, diese aber insbesondere um eine Softwarebibliothek erweitert, die es auf einfache Weise erlaubt verschiedene Quantenkernel-Templates und -Parameterkombinationen einer technischen Leistungsanalyse zu unterziehen. Neben der „Quanten Kernel Estimation“ (QKE) [Havlíček2019, Schuld2019] wurde auch die Methoden „Quantum Kernel Training“ (QKT) [Glick2021] und „Projecting Quantum Kernel Estimation“ (pQKE) [Huang2021] auf reale industrielle Daten angewendet und mit den Ergebnissen einer klassischen Implementierung (Support Vector Machine (SVM)) verglichen. Es zeigte sich, dass die Quantenkernelmethoden eine ähnliche Klassifikationsgüte wie die klassische Referenzimplementierung lieferten, allerdings keine entscheidenden Vorteile brachten.

Im Rahmen des Teilvorhabens wurde auch an neuen Methoden zur Einbettung von logischen Quantenschaltkreisen auf realen Quantencomputertopologien gearbeitet (unter anderem zur Rauschminderung). Hierbei wurden zwei Patentanmeldungen eingereicht, die zwei neue Methoden beschreiben, welche Antwortmengenprogrammierung nutzt, um optimale Einbettungen zu berechnen, die den ursprünglichen logischen Quantenschaltkreis nur um eine minimale Anzahl von benötigten Gitteroperationen anreichert (z.B. SWAP-Gatter).

Weiterhin wurde noch eine Patentanmeldung eingereicht, welche die Verknüpfung von klassischen Logiklösern mit Quantencomputern oder Quantenannealern beschreibt. Um den Lösungsvorgang von klassischen Logiklösern zu beschleunigen, können Quantencomputer und Quantenannealer hier genutzt werden um (Teil)Probleme (sub)optimal (heuristisch) zu lösen und die Lösungen dann als Ausgangspunkt für optimale Lösungen im Logiklöser dienen.

Referenzen

[Insafutdinov16] Insafutdinov, Eldar et al., „DeeperCut: A Deeper, Stronger, and Faster Multi-person Pose Estimation Model“, 2016.

[Weggemans2022] Weggemans, Jordi et al., “Solving correlation clustering with QAOA and a Rydberg qudit system: a full-stack approach”, 2022.

[Farhi2014] Farhi, Edward et al., „A quantum approximate optimization algorithm“, 2014.

[Havlíček2019] Havlíček, Vojtěch et al., „Supervised learning with quantum-enhanced feature spaces“, 2019.

[Schuld2019] Schuld, Maria et al., „Quantum Machine Learning in Feature Hilbert Spaces“, 2019.

[Huang2021] Huang, Hsin-Yuan et al., „Power of data in quantum machine learning“, 2021.

[Glick2021] Glick, Jennifer et al. „Covariant quantum kernels for data with group structure“, 2021.