

Benchmarking für Quantencomputing

Handlungsempfehlungen für ein
anwendungsbezogenes ideales Benchmark

QU

BIT





Impressum

Herausgeber

DLR Projektträger

Gesellschaft, Innovation, Technologie
Abteilung Digitale Strategien und Entwicklungen
Programmbegleitung „Quantentechnologien – Anwendungen für
die Wirtschaft“

Redaktion

Dr. Sophia Helmrich & Dr. Johann Schmidt, Abteilung Digitale
Strategien und Entwicklungen, DLR-PT

Mitwirkende

Hossam Ahmed, Mazen Ali, Abhishek Awasthi, Dimitris Badou-
nas, Valeria Bartsch, Colin Kai-Uwe Becker, Pallavi Bhardwaj,
Tim Bittner, Martin Braun, Sebastian Bock, Lukas Burgholzer,
Xiaolong Deng, Claudia Ehrig, Christoph Eichhammer, Domenik
Eichhorn, Marvin Erdmann, Christian Ertler, Fred Fiant, Sándor
Fekete, Taylor Garnowski, Alexander Geng, Ilie-Daniel Gheorghe-
Pop, Cristian Grozea, Wendelin Groß, Sascha Hauck, Dominik
Heldwein, Patrick Holzer, Michael Holzki, Luigi Iapichino, Matteo
Antonio Inajetovic, Michael Johanning, Caitlin Jones, Johannes
Jung, Matthias Kabel, Phillip Keldenich, Dominik Krupke, Georg
Kruse, Sophia Lahs, Jeanette Miriam Lorenz, Arcesio Castaneda
Medina, Ali Moghiseh, Andreas Mueller, Bharadwaj Chwodary
Mummaneni, Felix Paul, Maniraman Periyasamy, Sebastian
Rietsch, Marco Roth, Raja Seggoju, Sebastian Senge, Hendrik
Siebeneich, Theeraphot Sriarunothai, Jonas Stein, Rainer Sträter,
Nikolay Tcholtchev, Matthias Traube, Christian Tutschku, Friedrich
Wagner, Mareike Weule, Armin Wolf

Gestaltung

Kompetenzzentrum Öffentlichkeitsarbeit
des DLR-PT

Berlin, September 2024

Bildnachweise

Titel: Production Perig/ stock.adobe.com



Inhaltsverzeichnis

1. Executive Summary	4
2. Einleitung	5
3. Benchmarkarten	7
3.1. Vergleich Performance Hard- und Software	7
3.2. Vergleich Quantenalgorithmen vs. Klassischen Algorithmen	7
3.3. Identifizierung von Anwendungsproblemen	8
4. Benchmarkmetriken	8
4.1. Bewertungskriterien von Benchmarks	8
4.2. Beschreibung, Übersicht und Bewertung der Benchmarkmetriken	9
4.2.1. Vergleich Performance Hard- und Software	10
4.2.2. Vergleich Quantenalgorithmen vs. Klassischen Algorithmen	15
4.2.3 Identifizierung von Anwendungsproblemen	19
5. Handlungsempfehlungen	24
6. Literaturverzeichnis	26



1. Executive Summary

Quantencomputing wird intensiv erforscht, um herauszufinden, ob es in einigen Fällen effizienter als klassisches Computing ist. Es gibt potenzielle Anwendungsfälle, in denen Quantencomputing einen Vorteil bieten könnte. Die Entwicklung von Benchmarks und Werkzeugen zur Zertifizierung und Charakterisierung von Quantencomputern ist entscheidend, um die Leistungsfähigkeit in der von fehleranfälligen Quantenrechnern dominierten NISQ-Ära zu gewährleisten. Es gibt eine Vielzahl von Benchmarks, die sich teilweise stark ähneln oder für einen Vergleich zu stark unterscheiden. Oftmals ist es wichtig, die Hardware- und Softwareparameter zu verstehen, um die Ergebnisse richtig interpretieren zu können. Benchmarks und Benchmarkforschung sollten einem klaren Ziel folgen, d.h. praxisrelevante Probleme bewerten, die den realen Herausforderungen der Anwender entsprechen und die Skalierbarkeit von Lösungen einschätzen.

Hierzu haben wir im Rahmen der Programmbegleitung des vom BMWK geförderten Technologieprogramms „Quanten Computing - Anwendungen für die Wirtschaft“ dieses Übersichtspapier, gemeinsam mit den Expertinnen und Experten aus den neun geförderten Projekten, erarbeitet. Es gibt einen ersten Überblick zu drei verschiedenen Benchmarkarten und entsprechenden Benchmarkmetriken die in den Projekten zum Einsatz kommen. Daneben wurden anhand von sieben Kriterien die Benchmarks bewertet und Handlungsempfehlungen für eine Optimierung der Benchmarks gegeben.

Die Ergebnisse sind im Überblick in Abb. 1 dargestellt.

Benchmarkarten	Performance Hard- und Software	Vergleich Quantum- und klassische Algorithmen	Anwendungsbenchmark
Am besten bewertetes Benchmark	Kohärenzzeit	Time-to-solution	Factorization
Kriterien für die Bewertung	1 Gut definiert & Reproduzierbar 2 Unabhängig 3 Fair 4 Verfügbarkeit	5 Ganzheitlich 6 Klar & Spezifitär 7 Kein übertriebenes Finetuning	
Handlungsempfehlungen	<ul style="list-style-type: none">• Reale Anwendungen zur Basis für Benchmarks machen• Nutzerfreundlichkeit erhöhen• Benchmark-Community aufbauen• Benchmarks unabhängiger gestalten• Besonderheiten des Hybriden Computing berücksichtigen• Nutzen von unterschiedlichen Datensätzen für Training und Benchmarking		

Abbildung 1: Übersicht über Kernergebnisse.



2. Einleitung

Ist Quantencomputing effizienter als klassisches Computing? Für welche Anwendungsfälle gibt es einen voraussichtlichen Quantenvorteil? Auf welcher Hardware soll gerechnet werden? Welche Softwarekomponenten brauchen die Anwendenden? Wie hoch ist die Erfolgswahrscheinlichkeit eines Quantenalgorithmus? – Solche und weitere Fragen sollen mit Hilfe von Benchmarks, d.h. standardisierten Arbeitslasten zum Vergleich von Computersystemen, beantwortet werden. Mit der rasanten Entwicklung des Quantencomputing ist ein dringender Bedarf an einer breiten Palette von Werkzeugen, für die Zertifizierung und Charakterisierung, entstanden. Solche Werkzeuge sind entscheidend, da leistungsstarke Anwendungen in der NISQ-Ära¹ nur realisiert werden, wenn ein hohes Maß an Präzision der Komponenten erreicht und deren Funktion gewährleistet werden kann. (Eisert, et al., 2020)

Um diesen Bedarfen gerecht zu werden, gibt es eine Vielzahl an Benchmarks. Diese ähneln sich teilweise stark oder unterscheiden sich zu stark, um einen Vergleich zu gewährleisten. Oft sind, ohne vertieftes mathematisches Verständnis der Hardware- und Softwareparameter, die Aussagen der Ergebnisse eines Benchmarks, schwer zu verstehen. Somit lassen sich gleiche Benchmarks auf unterschiedlichen Systemen und ähnliche Benchmarks auf gleichen Systemen, nur schwer in Beziehung setzen und hinreichend und fair vergleichen. Das beginnt schon bei einfachen Maßzahlen, wie der Anzahl der einsetzbaren Qubits, die als Vergleichsgröße wenig geeignet sind, da sie keine Informationen über die Qualität beinhalten. Auch das heute, als Leistungs-Benchmark verbreitete Quantenvolumen, hat als Maß seine Tücken. (Wormsbecher, 2024)

Zudem ist es essentiell, dass Benchmarks und Forschung an Benchmark, nicht zum Selbstzweck vorangetrieben werden, sondern einem klaren Ziel folgen (Aaronson, 2020). Benchmarkings sollten zwei Ziele erreichen: Einerseits die Leistung eines praxisrelevanten

Problems, das den realen Herausforderungen der Anwendenden, wie beispielsweise in der Industrie entspricht, messen und bewerten zu können. Und andererseits messbar zu machen, wie eine entsprechende Lösung skaliert. Daher sollte, insbesondere von unnatürlich leichten Benchmarks, Abstand genommen werden. Es ist wichtig, Probleme für Benchmarks zu nutzen, die einige Beschränkungen enthalten, mehrere Parameter und Dimensionen aufweisen sowie eine gewisse Vielschichtigkeit bereithalten. Gute Ausgangspunkte sind beispielsweise Anwendungsproblemsammlungen (Luckow & Klepsch, 2021). Außerdem sollten Vergleiche anhand von Benchmarks durch Messungen (Anzahl der Operationen, Subroutinen-Profiling usw.) gestützt werden, die im Detail zeigen, warum eine Verbesserung eingetreten ist (Johnson, 2002).

Folglich sind die Definition, die Arten, die Eigenschaften und die Nutzbarkeit von Benchmarks, Kern vieler Forschungsfragen. So wird im Rahmen der Projekte, des vom BMWK geförderten Technologieprogramms „Quanten Computing - Anwendungen für die Wirtschaft“², ein starker Fokus auf Benchmarking gesetzt. Im Rahmen des Projekts „EniQmA“³ entsteht eine DIN-Spezifikation – DIN Spec 91480 „Benchmarking quantum computers with determined KPIs“ (DIN SPEC 91480, 2024). Die DIN-Spec spezifiziert Anforderungen an Kriterien für die Bewertung von Gatebasierten Quantencomputern, Quantenalgorithmen und deren Implementierungen sowie die Implementierung anhand einer standardisierten Reihe von Kernmetriken (KPIs) und zugehörigen Benchmark-Spezifikationen. Die Umsetzung dieser Spezifikationen, in einer Benchmark-Suite, soll den entwickelten Standard praktisch anwendbar machen.

1 Noisy Intermediate-Scale Quantum

2 Alle Projekte des Förderprogramms sind auf der Projektwebsite zu finden: Digitale Technologien: Quantencomputing

3 Informationen zum Projekt unter: EniQmA (digitale-technologien.de)



Weitere, in den Projekten angedachte Schwerpunkte zum Benchmarking, sind:

- Benchmarking von Quantensimulatoren
- Benchmarking und Vergleich von gatterbasierten (IBM, Google, Rigetti, IonQ) Algorithmen, wie QAOA bzw. VQEs, im Gegensatz zu (simulated) Annealing-Methoden (D-Wave, Fujitsu, Toshiba)
- Benchmark von Quantenalgorithmen (Volumetric Benchmarking, Application Performance, Randomized Benchmarking Framework, Quantum Mirror Circuits, Entanglement Benchmarks, Circuit optimization Benchmarks, Idle Tomography, Ableiten von QUBO Benchmarks aus SAT Benchmarks)
- Charakterisierung anhand von Qualitäts- und Performancemetriken (z.B. Quantum Volume, Datenqualität, Schaltkreistiefe, algorithmische Performanz, Randomized Benchmarking Framework etc.)
- Vergleich von Quantencomputing Ansätzen gegenüber existierenden Lösungen auf klassischen High-Performance Computern (Berechnungsdauer und Lösungsgenauigkeit)
- Benchmarking-Frameworks zur fundierten Bewertung von Performance und Anwendbarkeit der Lösungen von Use Cases (Zeit, Aufwand, Ergebnisqualität/Fehler, Interpretierbarkeit, aber auch nicht quantitativ-messbare Aspekte wie Akzeptanz, Tool-Handling oder notwendiges Knowhow)

Ausgehend von diesen Aktivitäten haben die Expertinnen und Experten aus den Projekten, eine Reihe von Herausforderungen identifiziert, die einen breiteren Nutzen aktueller Benchmarks verhindern. Es fehlen beispielsweise Standardsets und Benchmarks, die fair und einfach verfügbar sind. Auch realistische Testinstanzen für Benchmarks, sind zum einen schwer zu beschaffen und zum anderen noch zu groß für aktuelle Hardware. Das frühe Stadium der Technologie zeigt sich auch daran, dass der Zugriff auf verschiede-

ne Quantencomputer zum Benchmarken schwierig, teuer und zeitaufwendig ist. Zu guter Letzt existiert teilweise eine Unkenntnis über den State-of-the-art, sowohl mit Quantencomputern als auch im Vergleich zu klassischen Methoden.

Aufbauend auf diesem Lagebild, hat dieses Papier das Ziel, eine übergreifende Sammlung von Benchmarks für Quantencomputern zu erstellen. Angelehnt an die Schwerpunktsetzung aus den Projekten werden wir uns in diesem Papier auf drei Benchmarkarten konzentrieren: (i) Benchmarking von Hard- und Software, (ii) Benchmarking von Quantenalgorithmen und klassischen Algorithmen sowie (iii) Identifizierung von Anwendungsproblemen. Zunächst geben wir in Kapitel 3 eine Übersicht dieser drei betrachteten Benchmarkarten. Für diese drei Arten identifizieren und bewerten wir Benchmarkmetriken in Kapitel 4. Auf Basis der Bewertung werden Handlungsempfehlungen zur Spezifizierung dieser Metriken in Kapitel 5 gegeben. Dieses Papier entsteht im Rahmen der Programmbegleitung zum Technologieprogramm, „Quanten Computing – Anwendungen für die Wirtschaft“, des BMWK.



3. Benchmarkarten

Im Rahmen des Kapitels zu Benchmarkarten, geben wir kurze Erklärungen, welche verschiedenen Perspektiven beim Benchmarking von Quantencomputing bestehen und ordnen diese technologisch ein. Ergänzt wird diese Beschreibung dann in Kapitel 4, durch eine Beschreibung und Bewertung von konkreten Benchmarkbeispielen, innerhalb dieser Arten.

3.1. Vergleich Performance Hardware und Software

Mikrobenchmarks und Charakterisierungsprotokolle

Im Gegensatz zu modernen klassischen Computern sind aktuelle Quantencomputer in erster Linie durch Fehler und nicht durch ihre Größe oder Geschwindigkeit begrenzt. Bei Quantencomputern kann es zu einer Vielzahl komplexer Fehler kommen. Viele dieser Fehler resultieren aus Rauschen oder Fehlern der untersten Komponenten im System. Mikrobenchmarks und Charakterisierungsprotokolle auf Komponentenebene, wie z. B. randomisiertes Benchmarking, Entanglement Benchmarking (Zander & Becker, 2024) oder Gate-Set-Tomography, können Aufschluss über die Art und das Ausmaß dieser Fehler geben, sind jedoch nicht für Nutzende, die nur Anwendungen ausführen wollen, relevant (Lubinski, et al., 2023). Diese Charakterisierungen sind zudem, für die Funktionalität und das Design von hardwarenaher Softwarekomponenten, entscheidend (Gheorghe-Pop, et al., 2022).

Makrobenchmarks

Die begrenzte Vorhersagekraft von Leistungsmetriken auf niedriger Ebene hat dazu beigetragen, dass der Fokus zunehmend auf Benchmarks liegt, die die ganzheitliche Leistung eines Quantencomputers prägnant und direkt zusammenfassen (Lubinski, et al., 2023). Solche Makrobenchmarks wie das Quantenvolumen oder die Metrik der algorithmischen Qubits, ermöglichen zudem die Bewertung der Fähigkeit des gesamten Quantencomputing-Stacks. Somit entsteht ein Maßstab für den Stand der Technik des Quantencomputings und eine ungefähre Grundlage für den

Vergleich verschiedener Systeme (Gheorghe-Pop, et al., 2022). Diese Benchmarks sammeln verschiedene Algorithmen mit unterschiedlichen Quantenschaltungsstrukturen, um die Leistung von Quantenhardware zu erfassen. Da sie einen Durchschnitt über eine Vielzahl von Schaltkreisen bilden, können sie je nach Interpretation der Ergebnisse bei einer Gesamtausführung, das Niveau echter Benchmarks erreichen (Amico, et al., 2023).

3.2. Vergleich Quantenalgorithmen vs. Klassischen Algorithmen

Qualitätsmetriken für Algorithmen

Benchmarking für Quantencomputer wird häufig verwendet, um die Performance von Algorithmen zu bewerten (Gheorghe-Pop, et al., 2022). Es wird kontinuierlich an neuen Quantenalgorithmen und dazugehörigen Anwendungen geforscht, deren Funktionalität und Nutzen häufig anhand einfacher Benchmarks, auf Simulatoren oder echter Hardware, getestet wird. Qualitätsmetriken für Quantenalgorithmen konzentrieren sich darauf, einen Rahmen für den Vergleich zweier hybrider Algorithmen zu schaffen, die dasselbe Problem behandeln. Dieser Vergleich umfasst Aspekte wie Anwendung, Konvergenz, Skalierbarkeit und Nutzung von Quantenressourcen, wobei der Schwerpunkt auf der Qualität der Algorithmen liegt.

Ein Ansatz ist hierbei Algorithmen in der gleichen Programmiersprache zu konfigurieren und auf Hardware / Simulatoren oder Annealern auszuführen. Hierbei kann insbesondere die Wahl der Hardwareeinstellungen von Relevanz sein, angefangen von Standardeinstellungen über Experteneinstellungen (Mundada, et al., 2023). Verglichen werden unter anderem Laufzeit, Problemkomplexität, als auch auf Qualität der berechneten Lösungen, time-to-solution oder Erfolgswahrscheinlichkeit. Daneben kann aber auch die erfolgreiche Implementierung des Quantenalgorithmus, die Abschätzung von z. B. nötiger Schaltkreistiefe die für einen Quantenvorteil erforderlich ist sowie die Wirtschaftlichkeit eine wichtige Rolle spielen.



3.3. Identifizierung von Anwendungsproblemen

Vergleich von Systemen für den jeweiligen Anwendungsbereich

Anwender von Quantencomputern benötigen Wissen darüber, welche Fähigkeiten und Vorteile diese bezogen auf unterschiedliche Anwendungsbereiche haben. Dafür möchten sie auch die verschiedenen Systeme unterschiedlicher Hersteller für diesen Anwendungsbereich sinnvoll miteinander vergleichen können und in der Lage sein, zu identifizieren, ob es ein geeignetes System für den geplanten Nutzungszweck gibt. Hierzu wurden verschiedene Anwendungsbezogene Benchmarks eingeführt, z.B. Q-Score Metrik oder QED-C.

Performance und Anwendbarkeit der Use Cases

Daneben gibt es Benchmarking-Frameworks zur fundierten Bewertung von Performance und Anwendbarkeit der Lösungen von Use Cases, wie beispielsweise Zeit, Aufwand, Ergebnisqualität/Fehler, Interpretierbarkeit, aber auch nicht quantitativ-messbare Aspekte wie Akzeptanz, Tool-Handling oder notwendiges Knowhow. Hier steht die Frage nach dem erwarteten Quantenvorteil des jeweiligen Use Cases im Fokus.

4. Benchmarkmetriken

Auf Basis einer Literaturrecherche wurden, gemeinsam mit den Expertinnen und Experten, sieben Bewertungskriterien von Benchmarks identifiziert. Zudem wurden im Rahmen von zwei Workshops Benchmarkmetriken für die in Kapitel 3 vorgestellten Arten identifiziert, beschrieben, eingeordnet und anhand der Kriterien evaluiert. Ableitend aus der Bewertung konnten Handlungsempfehlungen zur Anpassung oder Erweiterung der Benchmarks getroffen werden. Kapitel 4.1. beschreibt die Bewertungskriterien, Kapitel 4.2. gibt die Beschreibung und Bewertung der Metriken und jeweiligen Benchmarks wieder.

4.1. Bewertungskriterien von Benchmarks

Um Benchmarks zu diskutieren, müssen wir zunächst definieren, was einen Benchmark ausmacht. Auch wenn es auf diese Frage keine eindeutige Antwort gibt, weisen allgemein akzeptierte Benchmarks im Allgemeinen einen bestimmten Funktionsumfang auf. Abstrakt muss sich die Leistung auf die Dreifaltigkeit von Skalierung, Qualität und Geschwindigkeit konzentrieren (Wack, et al., 2021): (i) **Skalierung** bezieht sich auf die Größe (z. B. in Anzahl der Qubits) der Probleme, die gelöst werden können; (ii) **Qualität** ist ein Maß dafür, wie zuverlässig eine bestimmte Aufgabe ausgeführt wird, wobei Abweichungen vom Idealverhalten auf das Vorhandensein einer oder mehrerer Fehlerquellen hinweisen; (iii) **Geschwindigkeit** stellt die zeitliche Leistung eines Systems dar, gemessen entweder anhand der Anzahl der pro Zeiteinheit ausgeführten Operationen oder durch Berechnung der Laufzeit eines vollständigen Quantenalgorithmus.

Erweitert wird diese Betrachtung durch Kriterien für Benchmarks. Das Papier konzentriert sich hierbei auf die Ergebnisse von (Amico, et al., 2023) und (DIN SPEC 91480, 2024) sowie auf die Erfahrung der Quantenbenchmark-Expertinnen und Experten aus den Projekten:

- **Gut definiert & Reproduzierbar:** Benchmarks sollten über klare Regeln für die Definition und Ausführung des Protokolls verfügen, damit es keine Mehrdeutigkeit gibt und andere die Tests reproduzieren können. Hierfür muss Daten- und Metadatenverfügbarkeit gewährleistet sein.
- **Ganzheitlich:** Die Benchmarking-Ergebnisse sollten Aufschluss über die Leistung einer großen Anzahl von Attributen, in möglichst wenigen Metriken, geben, d. h. die Skalierung sollte im Benchmark enthalten sein.
- **Unabhängig:** Das Protokoll sollte nicht auf einen bestimmten Gate-Satz, Algorithmus oder Plattform zugeschnitten sein und möglichst unabhängig sein.



- **Klar & Spezifitär:** Bessere Erklärung und Funktionalität des Benchmarks/der Metrik für den Laien.
- **Fair:** Die zum Vergleich herangezogenen State-of-the-Art-Methoden sollten nicht auf unfaire Weise verzerrt werden.
- **Kein übertriebenes Finetuning:** Reproduzierbarkeit ist nützlich, kann aber auch zu Problemen führen. Wenn beispielsweise in QML immer dieselben Datensätze verwendet werden, entsteht eine Reihe von Algorithmen, die für diesen Datensatz gut funktionieren, aber nicht unbedingt generalisiert auf anderen Datensätzen vergleichbar besser funktionieren.
- **Verfügbarkeit von Baselines:** Möglichkeit zum Vergleich mit den Ergebnissen anderer auf Basis eines vergleichbaren Referenzwertes.

Daneben wollen wir im Rahmen des Papiers beleuchten für welche Benutzerzielgruppe und welchen Nutzungsbereich der jeweilige Benchmark von **Relevanz** ist. Zudem ist Relevanz ein wichtiges Kriterium, um abzuschätzen, ob wichtige Anwendungen adressiert werden oder typische Bausteine für Algorithmen verwendet werden können.

Weitere durch die Experten benannte Kriterien sind **Hardware-übergreifende Anwendbarkeit** [Ob die Metrik auf ein anderes Quanten-Backend mit einer ähnlichen oder anderen Technologie angewendet werden kann (z. B. Quantenvolumen auf Ionenfallen vs. Quantenvolumen auf supraleitenden Systemen)] und **Umfang des Benchmarks** [Bestmögliches System vs. Benutzeranforderungen]. Diese Benchmarks wurden im Vergleich als weniger relevant für die Evaluation von Quantencomputing-Benchmarks eingestuft.

Ein in der Literatur häufig genanntes Kriterium: **Zufällig** [Benchmarks sollten eine zufällige Komponente haben (z. B. die Schaltkreise, Eingaben und/oder Ausgaben), und die, über diese Zufallsverteilung aggregierte endgültige Metrik, sollte zur Messung eines durchschnittlichen Ergebnisses verwendet

werden] wurden von den Expertinnen und Experten als weniger relevant eingeschätzt. Zufällig ist erfahrungsgemäß oft kein guter Ansatz zur Erzeugung von Benchmarks. In der klassischen Optimierung ist bekannt, dass dies zu künstlich einfachen Instanzen führen kann. Stattdessen ist es besser, die Instanzen sinnvoll & realistisch, zu gestalten. Das kann immer noch ein gewisses Maß an Zufälligkeit beinhalten, muss aber auf eine strukturierte Art und Weise angewendet werden, die widerspiegelt, was in der realen Welt wichtig ist. Daher werden wir „Zufällig“ im Folgenden als Kriterium nicht berücksichtigen.






4.2. Beschreibung, Übersicht und Bewertung der Benchmarkmetriken

Die allgemeinen Bewertungskriterien verschiedener Benchmarks können nun dafür genutzt werden, verschiedene Metriken zu bewerten. Die folgenden Metriken sind als besonders relevant für die Arbeit in den Projekten des Technologieprogramms „Quantencomputing – Anwendungen für die Wirtschaft“, durch 30 Expertinnen und Experten, bewertet worden. Die Benchmarks sind von der höchsten zur niedrigsten Relevanz gelistet.

Legende für die Bewertung in den einzelnen Kriterien:

- 1 – erfüllt
- 2 – teilweise erfüllt
- 3 – teilweise nicht erfüllt
- 4 – nicht erfüllt

Daraus leitet sich eine Gesamtbewertung ab.

-  Sehr gut
-  Gut
-  Mittelmäßig
-  Nicht geeignet
-  Gar nicht geeignet



4.2.1. Vergleich Performance Hard- und Software

Konnektivität



Zuordnung: Hardware

Familie: Mikrobenchmark, Performancemetric

Zielgruppe: Anwender, Soft- und Hardware-Entwickler

Kurzbeschreibung: In einem Algorithmus ist es wichtig, alle beteiligten Qubits miteinander rechnerisch, in einer Zwei-Qubit-Operation, verknüpfen zu können. Bei Ionenfallen-Quantencomputern ist stets eine maximale Konnektivität zwischen den Qubits gegeben. Bei supraleitenden Rechnern ist dies nicht möglich, da die Qubits in einer Architektur fest verbaut sind (Wormsbecher, 2024). Das Benchmark ist relevant für die Entwicklung von Quantenanwendungen und kann für verschiedene Quantum-Backend-Technologien verwendet werden.

Bewertung:

Gut Definiert	Ganzheitlich	Unabhängig	Klar & Spezifitär
1	2	1	2

Handlungsempfehlung:

- > Anwendungsbasierte Konnektivitätsanforderungen (abgeleitete/zusätzliche Metriken) können zur Klarheit beitragen und können Anwendungsfall für Anwendungsfall spezifisch erweitert werden.

Crosstalk Errors



Zuordnung: Hardware

Familie: Mikrobenchmark, Performancemetric

Zielgruppe: Entwickler, Quantum Error Correction und Quantum Error Mitigation Nutzer

Kurzbeschreibung: Crosstalk tritt in den meisten Quantencomputersystemen mit mehr als einem Qubit auf. Crosstalk-Fehler sind unerwünschte Dynamiken, die eines (oder beide) der Prinzipien Lokalität und Unabhängigkeit verletzen. Das heißt, wenn es beispielsweise unerwünschte Korrelationen zwischen Qubits gibt oder daraus resultierende nicht wohldefinierte Operationen (z. B. Gates und Messungen) erfolgen. (Sarovar, Proctor, Rudinger, Young, & Blume-Kohout, 2020) Die Entwickelnden können mit Benchmarks bewerten, ob die Fehlerwahrscheinlichkeit für die Schaltung, das Ergebnis plausibel macht.

Bewertung:

Gut Definiert	Ganzheitlich	Unabhängig	Klar & Spezifitär
2	2	1	2

Handlungsempfehlung:

- > Anwendungsbasierte Anforderungen entwickeln.
- > Nicht alle Plattformen unterstützen diese Metrik > Ausrollen auf weitere Plattformen.
- > Weiterentwicklung über NISQ-Zeitalter hinaus.



Kohärenzzeit



Zuordnung: Hardware

Familie: Mikrobenchmark, Performancemetrik

Zielgruppe: Soft- und Hardware-Entwickler, Data Scientist, Anwender (Auswahl von Hardware für bestimmten Algorithmus)

Kurzbeschreibung: Das ist die Zeitspanne, für die die Qubits während einer Berechnung ihren Quantenzustand beibehalten können, ehe sie zu Bits kollabieren. Die Länge eines Berechnungsalgorithmus ist begrenzt durch diesen Wert (Wormsbecher, 2024). Dies kann wichtige Hinweise für das Design eines Quantencircuits bieten (maximale Länge und Breite des Circuits).

Bewertung:

Gut Definiert	Ganzheitlich	Unabhängig	Klar & Spezifitär
1	1	1	1

Operationsgeschwindigkeit (Gate-Speed)



Zuordnung: Hardware

Familie: Mikrobenchmark, Performance- und Charakterisierungsmetrik

Zielgruppe: Anwender

Kurzbeschreibung: Die Zeit, die es braucht, um eine Rechenoperation auf Qubits durchzuführen. Die Zeit aller Operationen darf nicht die Kohärenzzeit übersteigen, da die Rechnung sonst kollabiert. Eine schnelle Operationsgeschwindigkeit ist essenziell für komplexe Berechnungen (Wormsbecher, 2024).

Bewertung:

Gut Definiert	Ganzheitlich	Unabhängig	Klar & Spezifitär
1	2	3	2

Handlungsempfehlung:

- > Benötigt mehr Spezifität, d. h. eine bessere Erklärung und Funktionalität für den Laien.
- > Sollte in Kombination mit anderen Benchmarks (z. B. Fidelity) oder für bestimmte Hardware-Modalitäten in Betracht gezogen werden.



Randomisiertes Benchmarking



Zuordnung: Hardware

Familie: Mikrobenchmark, Charakterisierungsprotokolle

Zielgruppe: Anwender

Kurzbeschreibung: Um das durchschnittliche Rauschen zu bestimmen, das mit jedem Gatter innerhalb eines Gattersatzes verbunden ist, wird eine Sequenz von (nahezu) gleichmäßig zufälligen Gattern, die als randomisierte Benchmarking-Sequenz bezeichnet werden, aus diesem Gattersatz auf einen Satz von Ziel-Qubits angewendet (DIN SPEC 91480, 2024). Wenn es sich um lange Folgen zufällig ausgewählter Gatter handelt, wird auch überprüft, ob das Fehlerverhalten bei langen Berechnungen stabil ist (Knill, et al., 2008). Es kann zudem zwischen Zustandsvorbereitungs- und Messfehlern und den Gatterfehlern unterschieden werden.

Bewertung:

Gut Definiert	Ganzheitlich	Unabhängig	Klar & Spezifitär
2	1	1	3

Handlungsempfehlung:

- > Die Spezifität für bestimmte Anwendungen kann schwach sein.
- > Anstatt den Durchschnitt über einen Gattersatz zu nehmen, indem Gatter nach dem Zufallsprinzip ausgewählt werden, können bestimmte Gatter mit einem ML-Algorithmus ausgewählt werden.

Circuit Layer Operation per Second (CLOPS)



Zuordnung: Hardware

Familie: Mikrobenchmark

Zielgruppe: Anwender

Kurzbeschreibung: CLOPS ist ein Beispiel für einen Benchmark, zur Messung der Geschwindigkeit eines Quantenprozessors. CLOPS untersucht die Anzahl der QV-Schichten pro Sekunde, die von einem Satz parametrisierter QV-Schaltkreise ausgeführt werden können, die während der Laufzeit aktualisiert werden, ohne den besten für QV erzielten Wert zu opfern. Ein hoher CLOPS-Wert weist darauf hin, dass die Hardware eine große Anzahl von Vorgängen pro Sekunde ausführen kann (Amico, et al., 2023).

Bewertung:

Gut Definiert	Ganzheitlich	Unabhängig	Klar & Spezifitär
1	2	1	2

Handlungsempfehlung:

- > Kein Hinweis darauf, wie relevant es für bestimmte Anwendungen ist.
- > Eine Spezifizierung würde die ganzheitlichen Kriterien beeinträchtigen.

Daneben wurden weitere Benchmarks benannt, jedoch durch die Experten und Expertinnen aus den Projekten, als weniger relevant eingestuft und

demnach nicht anhand der Kriterien bewertet. Für diese geben wir hier nur eine kurze Definition ohne numerische Bewertung.



Quantenvolumen (QV)

Zuordnung: Hardware, Algorithmus

Familie: Makrobenchmark

Zielgruppe: Anwender

Kurzbeschreibung: Das Quantenvolumen quantifiziert den größten zufälligen Schaltkreis, mit gleicher Breite und Tiefe, den der Computer erfolgreich implementiert (Cross, Bishop, Sheldon, Nation, & Gambetta, 2019). Um das Quantenvolumen zu ermitteln, durchmischt der Algorithmus abwechselnd eine feste Menge an Qubits (sogenannte Permutationen) und steuert sie danach mit zufälligen Zwei-Qubit-Operationen an. Dieser Vorgang wird so lange wiederholt, bis das Ergebnis von den idealen Werten abweicht. Daraus leitet sich dann eine Obergrenze der Qubit-Operationen, abhängig von der Problemgröße, ab (Wormsbecher, 2024).

Bewertung: Die Metrik wird von den Experten und Expertinnen als etwas veraltet bewertet, da das Quantenvolumen mit dem Aufkommen von Ionen-basierten Quantencomputern, exorbitante Werte annahm und die Metrik dadurch unausgewogen wurde.

Zwei-Qubit-Operationsgenauigkeit (Gate Fidelity)

Zuordnung: Hardware

Familie: Mikrobenchmark, Performancemetrik

Zielgruppe: Entwickler

Kurzbeschreibung: Berechnungen auf Quantencomputern bestehen aus Einzeloperationen, die entweder auf einem Qubit oder auf zwei Qubits ausgeführt werden. Häufig ist der limitierende Faktor für Berechnungen die Genauigkeit (Fidelity) von Zwei-Qubit-Operationen (Wormsbecher, 2024).

Bewertung: Grundlegende Metrik von entscheidender Bedeutung für die Entwicklung von Quantenalgorithmen.

Anzahl Physical Qubit

Zuordnung: Hardware

Familie: Mikrobenchmark, Performancemetrik

Zielgruppe: Anwender, Entwickler

Kurzbeschreibung: Einzelnes greifbares Quantensystem, das zur Kodierung der beiden Basiszustände eines Qubits oder eines Qubits an Quanteninformation verwendet wird (DIN SPEC 91480, 2024). Viele Qubits bedeuten aber nicht, dass die Anwender diese auch gleichzeitig nutzen können (siehe Konnektivität).



Anzahl Logical Qubit

Zuordnung: Hardware

Familie: Mikrobenchmark, Performancemetrik

Zielgruppe: Anwender, Entwickler

Kurzbeschreibung: Logische Qubits sind typischerweise so konzipiert, dass sie als vollständig quantenfehlerkorrigierte physikalische Qubits fungieren (DIN SPEC 91480, 2024).

Kosteneffiziente Quantum Error Correction

Zuordnung: Hardware

Familie: Mikrobenchmark

Zielgruppe: Entwickler

Kurzbeschreibung: Verfahren zur Diagnose und Korrektur von Fehlern in den Bestandteilen eines logischen Qubits, ohne die logisch kodierte Quanteninformation zu messen, indem die Symmetrien des logischen Qubits ausgenutzt werden (DIN SPEC 91480, 2024).

Gate-Set-Tomography

Zuordnung: Hardware

Familie: Mikrobenchmark, Charakterisierungsprotokolle

Zielgruppe: Entwickler

Kurzbeschreibung: Methode zur vollständigen Charakterisierung von Quantenlogikgattern. Sie berücksichtigt Fehler bei der Zustandsvorbereitung und -messung (Greenbaum, 2015).



4.2.2. Vergleich Quantenalgorithmen vs. Klassischen Algorithmen

Qualitätsmetriken für Quantenalgorithmen konzentriert sich darauf einen Rahmen für den Vergleich zweier Algorithmen zu schaffen, die dasselbe Problem behandeln. Dieser Vergleich umfasst

Aspekte wie Anwendung, Konvergenz, Skalierbarkeit und Nutzung von Quantenressourcen, wobei der Schwerpunkt auf der Qualität der Algorithmen liegt.

Qualität der berechneten Lösungen



Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Software- und Algorithmenentwickler, Anwender, Hardwareentwickler

Kurzbeschreibung: Die Qualität der Lösung ist ein Maß dafür, wie zuverlässig ein Quantenalgorithmus eine bestimmte Aufgabe ausführt, wobei Abweichungen vom idealen Verhalten auf das Vorhandensein einer oder mehrerer Fehlerquellen hinweisen (Amico, et al., 2023), z. B. Erfolgswahrscheinlichkeit der Rückgabe des Zustands „1“ (Mundada, et al., 2023) oder Fidelity des Quantenzustands, der die ideale Lösung repräsentiert.

Bewertung:

Gut Definiert	Unabhängig	Ganzheitlich	Nicht überfintunen	Fair
1	1	3	3-4	2

Handlungsempfehlung:

- > Nutzen von Daten, um die Qualität der Lösung abzuschätzen, die nicht im Trainingsprozess verwendet wurden und zufällig ausgewählt wurden.
- > Berücksichtigung einer breiteren Palette von Problemen/Datensätzen, um überfintuning der Algorithmen zu vermeiden.
- > Sollte optimal mit Time-to-Solution kombiniert werden.
- > MaxCut-Problem als Referenz für den Vergleich zu nutzen ist nicht zielführend, denn ein Algorithmus, der bei MaxCut sehr gut funktioniert, versagt bei anderen Problemen wie TSP, MIS oder dem Knapsack-Problem.



Time-to-solution



Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Anwender, bei Hard- und Softwareentwicklern als Optimierungsgröße

Kurzbeschreibung: Die time-to-solution ist definiert als die erwartete Rechenzeit, die benötigt wird, um einen Grundzustand für eine bestimmte Probleminstanz mit 99%iger Sicherheit zu finden (Sankar, et al., 2021). Dieses Benchmark ist stark problemabhängig.

Bewertung:

Gut Definiert	Unabhängig	Ganzheitlich	Nicht überfintunen	Fair
1	2	3	2	2-3

Handlungsempfehlung:

- > Bei time-to-solution muss auch die Abhängigkeit von der Hardware berücksichtigt werden, d. h. eine Abdeckung des gesamten Benchmarks von der Hardware bis zum Algorithmus und den Wartezeiten für „queuing“.
- > Verwenden einer Metrik, die für die gewünschte praktische Anwendung relevant ist, z. B. wall-clock time.
- > Sollte optimal mit der Lösungsqualität kombiniert werden.
- > Skalierung der Problemgröße sollte sorgfältig erfolgen.

Skalierbarkeit und Nutzung von Quantenressourcen



Zuordnung: Algorithmen, Anwendung

Familie: Qualitätsmetriken für Algorithmen, Anwendungsmetrik

Zielgruppe: Hard- und Softwareentwickler, für Anwender wichtig, um die aktuellen Limitierungen abschätzen zu können.

Kurzbeschreibung: Gibt an, welche Ressourcen für eine bestimmte Qualität des Ergebnisses erforderlich sind, z. B. Anzahl der Qubits oder Gates (DIN SPEC 91480, 2024). Hängt sehr stark an der verwendeten Hardware und den spezifisch genutzten Algorithmen

Bewertung:

Gut Definiert	Unabhängig	Ganzheitlich	Nicht überfintunen	Fair
1	4	2	3	2

Handlungsempfehlung:

- > Sollte unbedingt mit time-to-solution kombiniert werden, da ohne eine Abschätzung der Qualität oder der Zeit eine Skalierung schwierig wird.
- > Es müssen hier die Fortschritte in der Skalierung der Hardware berücksichtigt werden.
- > Kanonische Probleme zur Abschätzung der Skalierung nötig.



Dateneffizienz / Size of the dataset



Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Anwender (da oft nur beschränkte Datensätze zur Verfügung), Software- und Algorithmenentwickler

Kurzbeschreibung: Dateneffizienz beschreibt die Anzahl der Datenpunkte, für einen bestimmten Datensatz, die benötigt werden, um den Algorithmus für eine bestimmte Leistung auf den Testdaten beim maschinellen Lernen, zu trainieren (Alchieri, Badalotti, Bonardi, & Bianco, 2021).

Bewertung:

Gut Definiert	Unabhängig	Ganzheitlich	Nicht überfittet	Fair
1	2	3	4	1

Handlungsempfehlung:

- > Nutzen von Datensätzen, die sich in gewisser Weise voneinander unterscheiden (Anwendung, Dimension usw.).
- > Kombination mit weiteren Benchmarks, wie time-to-solution oder Qualität der Lösung.

Daneben wurden weitere Benchmarks benannt, jedoch durch die Experten und Expertinnen aus den Projekten als weniger relevant eingestuft und dem-

nach nicht anhand der Kriterien bewertet. Für diese geben wir hier nur eine kurze Definition ohne Bewertung.

Numerical accuracy of the solution

Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Entwickler

Kurzbeschreibung: Numerische Genauigkeit der Lösung, z. B. die Epsilon-Nähe zu einer Lösung einer Gleichung.

Fraction of feasible solutions

Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Entwickler

Kurzbeschreibung: Dieser Wert gibt uns Aufschluss darüber, wie schnell einer der Algorithmen das Sampling, im Vergleich zum schlechtesten Algorithmus, für jede Instanz abschließt (Andoin, Oregi, Villar-Rodriguez, Osaba, & Sanz, 2022).



Cross talk effects

Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Entwickler, Anwender

Kurzbeschreibung: Misst inwiefern cross talk effects die Qualität der Lösung beeinflussen.

Generalization Ability

Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Entwickler

Kurzbeschreibung: Die Fähigkeit zur Generalisierung ist ein Wert, der für QML, die Fähigkeit des trainierten Modells, auch bei ungesehenen Daten gut abzuschneiden, angibt (Gil-Fuster, Eisert, & Bravo-Prieto, 2024).

Energy efficiency

Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Anwender, Entwickler

Kurzbeschreibung: Energieeffizienz gibt die Menge an Energie an, die benötigt wird, um eine Lösung zu finden.

Cost-to-solution

Zuordnung: Algorithmen

Familie: Qualitätsmetriken für Algorithmen

Zielgruppe: Entwickler

Kurzbeschreibung: Dieser Wert gibt den tatsächlichen Preis, für das Erreichen einer bestimmten Genauigkeit/Qualität der Lösung, mit einer bestimmten Konfiguration, an.



4.2.3. Identifizierung von Anwendungsproblemen

Es besteht ein dringender Bedarf an vielfältigen anwendungsorientierten Metriken und Benchmarks, die die Leistung von Quantencomputern bei praxisrelevanten Aufgaben testen. Mit solchen Benchmarks können Hardwareentwickelnde ihre Fortschritte auf

kommerziell relevante Weise quantifizieren und Endnutzende können genauer vorhersagen, wie die verfügbare Hardware in ihrer Anwendung funktionieren wird (Lubinski, et al., 2023), (Becker, et al., 2022).

Q-Score Metrik



Zuordnung: Algorithmen

Familie: Anwendungsbenchmark

Zielgruppe: Anwender, Hardwareentwickler

Kurzbeschreibung: Der Q-Score misst die größte durchschnittliche Problemistanzgröße N , eines NP-schweren Optimierungsproblems, bei der ein QPU einen Zufallsalgorithmus, deutlich übertrifft (DIN SPEC 91480, 2024). Die Q-Score-Metrik ist definiert als die Größe des größten Graphen, für den das Quantengerät das Max-Cut-Problem, näherungsweise mit einer Lösung lösen kann, die einen zufälligen Ratealgorithmus deutlich übertrifft (Leermakers, Wezeman, Neumann, & Phillipson, 2022).

Bewertung:

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
1	1	2	2

Handlungsempfehlung:

- > Die Metrik ist sehr spezifisch in ihrer Anwendbarkeit, die Metrik ist gut geeignet für Quantum Annealing, nicht so gut für andere Hardware-Methoden. Die Metrik könnte entsprechend auf andere Probleme z. B. QED erweitert werden.
- > Muss von einer breiten Community genutzt werden, um etabliert zu werden und eine passende Baseline zu identifizieren.



Factorization



Zuordnung: Anwendung

Familie: Anwendungsbenchmark

Zielgruppe: Anwender, Entwickler

Kurzbeschreibung: Der Faktorisierungsalgorithmus von Shor und der zugehörige Periodenfindungsalgorithmus können als Grundlage für Benchmarks dienen, die Schaltkreise als Ganzes testen. Für das Benchmarking ist die Abhängigkeit der Erfolgswahrscheinlichkeit des Periodenfindungsalgorithmus von der Periode relevant (Davis, 2021).

Bewertung:

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
1	1	1	1

Handlungsempfehlung:

- > Durch die Festlegung eines Zeitlimits wird verhindert, dass Kosten und Zeit unendlich sind.
- > Muss von einer breiten Community genutzt werden, um etabliert zu werden und eine passende Baseline zu identifizieren.

Generative Quantum Machine Learning



Zuordnung: Anwendung

Familie: Anwendungsbenchmark

Zielgruppe: QML-Entwickler

Kurzbeschreibung: Im Allgemeinen zielt ein generatives QML-Modell darauf ab, eine Zielwahrscheinlichkeitsverteilung zu approximieren, indem eine parametrisierte Quantenschaltung so optimiert wird, dass jedes Ergebnis auf eine eindeutige Bitfolge abgebildet wird. (DIN SPEC 91480, 2024).

Bewertung:

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
3	3	4	1-2

Handlungsempfehlung:

- > Integrieren eines Pass/Fail-Tests, um Fälle zu berücksichtigen, in denen das Problem nicht in einer angemessenen Zeit gelöst werden kann.
- > Muss von einer breiten Community genutzt werden, um etabliert zu werden und eine passende Baseline zu identifizieren.
- > Die Anwendbarkeit ist stark auf gate-based Algorithmen ausgelegt. Hier wäre eine Erweiterung wünschenswert.



QED-C



Zuordnung: Anwendung

Familie: Anwendungssuite

Zielgruppe: Anwender

Kurzbeschreibung: Benchmark-Ansatz vom Quantum Economic Development Consortium. Diese Benchmarking-Suite basiert auf einer Vielzahl bekannter Quantenalgorithmen und wurde entwickelt, um die Effektivität von Quantenhardware, bei der Ausführung von Anwendungen, zu untersuchen und Proxys für die Qualität und die Zeit bis zur Lösung, für die entsprechende Anwendung, zu messen.) (Lubinski, et al., 2023) (Lubinski, et al., 2024).

Bewertung:

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
3	1	2	1

Handlungsempfehlung:

- > Integrieren eines Pass/Fail-Tests, um Fälle zu berücksichtigen, in denen das Problem nicht in einer angemessenen Zeit gelöst werden kann.
- > Muss von einer breiten Community genutzt werden, um etabliert zu werden und eine passende Baseline zu identifizieren.
- > Ausbau der Verfügbarkeit von Baseline Werten.

Minizinc2QUBO und das QUBO Solver Benchmark



Zuordnung: Anwendung

Familie: Anwendungssuite

Zielgruppe: Entwickler

Kurzbeschreibung: MiniZinc-Modelle können in QUBOs umgewandelt werden, die für das Benchmarking verwendet werden können, z. B. um die Leistung verschiedener Solver zu vergleichen und eine alternative Lösung zu entwickeln und zu testen (Wolf & Grozea, 2023). Eine Sammlung von QUBO-Problemen, von denen viele durch die Anwendung von Minizinc2QUBO gewonnen wurden, ist online, als „QUBO Solver Benchmark“, verfügbar (Grozea, 2024). Sowohl die Anbieter von Quantenannealing-Hardware als auch die Anbieter von Optimierungsoftware können dort gegeneinander antreten, wenn es darum geht, QUBO-Probleme, die aus industriellen Optimierungsaufgaben, aus SAT- und Minizinc-Modellen abgeleitet wurden, am besten zu lösen.

Bewertung:

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
2	1	2	1

Handlungsempfehlung:

- > Integrieren von Minizinc2QUBO in die PlanQK-Plattform, um seine Verfügbarkeit zu erhöhen.
- > Ausbau der Verfügbarkeit von Baseline Werten.
- > Hohe Eintrittsbarrieren müssen überwunden werden, durch einfachere Formulierung des Benchmarks.



QUARK-Framework



Zuordnung: Anwendung

Familie: Anwendungssuite

Zielgruppe: Anwender, Entwickler für die gemeinsame Weiterentwicklung

Kurzbeschreibung: Das QUARK-Framework zielt darauf ab, die Entwicklung von Benchmarks auf Anwendungsebene zu erleichtern und flexibel auf verschiedene Anwendungen anpassbar zu sein (Finžgar, Ross, Hölscher, Klepsch, & Luckow, 2022).

Bewertung: Der Fokus dieses open-source Frameworks ist die kollaborative Weiterentwicklung, um so – mit wenig Aufwand – neue relevante Anwendungsfälle modular hinzuzufügen. Darin unterscheidet sich QUARK maßgeblich von anderen Frameworks.

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
2	2	2	1

Handlungsempfehlung:

- > Integrieren eines parametrisierbaren Pass/Fail-Tests, um individuell einstellen zu können, welche Rechenzeiten als angemessen gelten und um Benchmarking-Szenarien, die nicht innerhalb dieser Rechenzeit gelöst werden können, vorzeitig abzubrechen, um Rechenzeit und -ressourcen zu sparen.
- > Das Framework um zusätzliche Anwendungsfälle, Problemklassen und Lösungsmethoden wie Quantum Annealing erweitern.

Quantum Image Encoding Fidelity



Zuordnung: Anwendung

Familie: Anwendungsbenchmark

Zielgruppe: Entwickler

Kurzbeschreibung: Ein Quantencomputer muss eine Mindestanforderung erfüllen, um für die Quantenbildverarbeitung nützlich zu sein. In diesem Anwendungsbenchmark wird das Eingabebild durch einen Algorithmus kodiert, z. B. „flexible representation of quantum images“ (FRQI) (Le, Dong, & Hirota, 2011) um dann die Qubits direkt messen zu können. Das bedeutet, dass der Quantenschaltkreis nur das Encoding, den Identitätsoperator und die Messung enthält. Der Vergleich der Eingangs- und Ausgangsbilder liefert ein Gesamtmaß für die Leistung des Quantencomputers. Dieses Maß könnte als ein auf der Pixeldifferenz basierendes Maß oder als die maximale Größe der Quantenbildkodierungstreue formuliert werden (Geng, Moghiseh, Redenbach, & Schladitz, 2023).

Bewertung:

Verfügbarkeit von Baselines	Gut Definiert	Unabhängig	Vertretbarer Aufwand
1	1	2	2

Handlungsempfehlung:

- > In Geng et al. wird nur FRQI untersucht und könnte auch auf andere Kodierungsmethoden ausgedehnt werden.
- > Könnte auch auf Quantum Annealer ausgedehnt werden.



Daneben wurden weitere Benchmarks benannt, jedoch durch die Experten und Expertinnen aus den Projekten als weniger relevant eingestuft und dem-

nach nicht anhand der Kriterien bewertet. Für diese geben wir hier nur eine kurze Definition ohne numerische Bewertung.

Optimality Gap

Zuordnung: Anwendung

Familie: Anwendungsbezogenes Benchmarking, Qualitätsmetrik für Algorithmen

Zielgruppe: Anwender, Entwickler

Kurzbeschreibung: Gibt an, wie nah die Zielwerte der Lösungen für ein Optimierungsproblem, dem wahren Optimum (oder, falls unbekannt, einer Grenze), dran sind (Lubinski, et al., 2024).

Bewertung: Gut geeignet für einige kombinatorische Optimierungsprobleme, da „Optimality“ als Metrik für verschiedene Probleme unterschiedliches Gewicht hat. Es gibt beispielsweise kombinatorische Optimierungsprobleme, bei denen eine suboptimale Lösung „gut genug“ ist, während bei anderen Problemen die optimale Lösung das einzige Ziel ist.

Quanten-LINPACK-Benchmark

Zuordnung: Anwendung

Familie: Anwendungsbenchmark

Zielgruppe: Anwender

Kurzbeschreibung: Wird verwendet, um die Gesamtleistung von Quantencomputern zu messen. Sie ist die Mindestanforderung an einen Quantencomputer, um eine nützliche Aufgabe wie das Lösen linearer Algebra-Probleme (z. B. linearer Gleichungssysteme) zu erfüllen (Dong & Lin, 2021).

Variational Quantum Eigensolver (VQE)

Zuordnung: Anwendung

Familie: Anwendungsbenchmark

Zielgruppe: Anwender im Bereich quantenchemischer Probleme

Kurzbeschreibung: Der Variational Quantum Eigensolver hat sich als einer der populärsten hybriden Algorithmen, zur Lösung quantenchemischer Probleme, mit Hilfe eines Quantencomputers, erwiesen. (DIN SPEC 91480, 2024). VQE ist zudem im QED-C Benchmark berücksichtigt.



SupermarQ

Zuordnung: Anwendung

Familie: Anwendbarkeit von Use Cases

Zielgruppe: Anwender

Kurzbeschreibung: Inspiriert von früheren Benchmarking-Bemühungen, für neue Technologien, ist SupermarQ eine sich entwickelnde Benchmark-Suite, die für die Skalierbarkeit mit Quantengeräten entwickelt wurde und viele Anwendungen (z. B. QAOA, GHZ-Zustände) und viele verschiedene Qualitätsmaße umfasst (Tomes, et al., 2022).

5. Handlungsempfehlungen

Ausgehend von den Erfahrungen, der Expertinnen und Experten in den Projekten, können generelle Handlungsempfehlungen abgeleitet werden.

Reale Anwendungen zur Basis für Benchmarks

- Anwendungsbasierte Anforderungen (abgeleitete/zusätzliche Metriken) können zur Klarheit beitragen und können Anwendungsfall für Anwendungsfall spezifisch erweitert werden.
- Benchmarks sollten sich an praktischen Anwendungsproblemen orientieren und so zur praktischen Erkenntnis beitragen. Daher sollte, insbesondere von unnatürlich leichten Benchmarks, Abstand genommen werden.

Nutzerfreundlichkeit erhöhen

- Benchmarks müssen mehr Spezifität aufweisen, d. h. eine bessere Erklärung und Funktionalität für den Laien. Hohe Eintrittsbarrieren für die Nutzung müssen überwunden werden, durch einfachere Formulierung des Benchmarks.
- Integration eines Pass/Fail-Tests, einschließlich eines anpassbaren Zeitlimits für Benchmarking-Szenarien, um Rechenzeit und Ressourcen zu sparen.
- Benchmarks sollten ganzheitlicher gestaltet werden, um ein Cherry-picking der Aspekte, die für eine gewisse Hardware/Algorithmus/Anwendung geeignet sind, zu verhindern.



Benchmark-Community aufbauen

- Benchmarks müssen von einer breiten Community genutzt werden, um etabliert zu werden und passende Baseline-Werte sowie Anwendungsbereiche zu identifizieren.
- Die Community sollte außerdem aktiv an der Weiterentwicklung der verfügbaren Benchmarks, Anwendungssuites und Metriken beteiligt sein, um sie bedarfsgerecht erweitern zu können.
- Die Community sollte an der Schaffung von Standards im Benchmarking arbeiten.

Benchmarks unabhängiger gestalten

- Nicht alle Plattformen unterstützen alle Metriken bzw. sind abhängig von der verwendeten Plattform, hier kann entweder ein Ausrollen auf weitere Plattformen erfolgen oder eine Kombination mit anderen komplementären Benchmarks.
- Aktuell liegt der Fokus sehr stark auf gate-based Algorithmen und Hardware. Eine Erweiterung ist anzustreben.

Besonderheiten des Hybriden Computing berücksichtigen

- Im NISQ-Zeitalter und insbesondere bei hybriden Computinganwendungen müssen, neben den quantenbezogenen Benchmarks, auch die klassischen Hard- und Softwareanteile berücksichtigt werden.
- Die Weiterentwicklung von NISQ spezifischen Benchmarks über das NISQ-Zeitalter hinaus muss mitgedacht werden. Es müssen hierbei die Fortschritte in der Skalierung der Hardware berücksichtigt werden.

Nutzen von unterschiedlichen Datensätzen für Training und Benchmarking

- Nutzen von Daten, um die Qualität der Lösung abzuschätzen, die nicht im Trainingsprozess verwendet wurden und zufällig ausgewählt wurden.
- Berücksichtigung einer breiteren Palette von Problemen/Datensätzen, die sich in gewisser Weise voneinander unterscheiden (Anwendung, Dimension usw.).
- Kanonische geeignete Datensätze entwickeln, um Vergleichbarkeit zu erhöhen.



6. Literaturverzeichnis

- Aaronson, S. (5. 3 2020). Scott Aaronson Blog - Turn down the quantum volume. Von <https://scottaaronson.blog/?p=4649> abgerufen
- Alchieri, L., Badalotti, D., Bonardi, P., & Bianco, S. (2021). An introduction to quantum machine learning: from quantum logic to quantum deep learning. *Quantum Machine Intelligence*, *S. 3*, 28. doi:<https://doi.org/10.1007/s42484-021-00056-8>
- Amico, M., Zhang, H., Jurcevic, P., Bishop, L. S., Nation, P., Wack, A., & McKay, D. C. (March 2023). Defining Standard Strategies for Quantum Benchmarks. <https://arxiv.org/pdf/2303.02108.pdf>.
- Andoin, M. G., Oregi, I., Villar-Rodriguez, E., Osaba, E., & Sanz, M. (2022). Comparative Benchmark of a Quantum Algorithm for the Bin Packing Problem. *IEEE Symposium Series on Computational Intelligence (SSCI)*, *S. 930-937*. doi:10.1109/SSCI51031.2022.10022156
- Becker, C. K.-U., Tcholtchev, N. V., Gheorghe-Pop, I.-D., Bock, S., Seidel, R., & Hauswirth, M. (2022). Towards a Quantum Benchmark Suite with Standardized KPIs. *IEEE 19th International Conference on Software Architecture Companion*, *S. ICSA-C 2022*.
- Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D., & Gambetta, J. M. (2019). Validating quantum computers using randomized model circuits. *Phys. Rev. A*, *S. 100*, 032328. doi:<https://doi.org/10.1103/PhysRevA.100.032328>
- Davis, E. D. (2021). Benchmarks for quantum computers from Shor's algorithm. *arXiv:2111.13856*. doi:10.48550/arXiv.2111.13856
- DIN SPEC 91480. (2024). Benchmarking quantum computers with determined KPIs.
- Dong, Y., & Lin, L. (2021). Random circuit block-encoded matrix and a proposal of quantum LINPACK benchmark. *Phys. Rev. A*, *S. 103*, 062412. doi:<https://doi.org/10.1103/PhysRevA.103.062412>
- Eisert, J., Hangleiter, D., Walk, N., Roth, I., Markham, D., Parekh, R., . . . Kashefi, E. (2020). Quantum certification and benchmarking, <https://doi.org/10.1038/>. *Nature Technical Reviews*, *S. 2*, 382.
- Finžgar, J. R., Ross, P., Hölscher, L., Klepsch, J., & Luckow, A. (2022). QUARK: A Framework for Quantum Computing Application Benchmarking. *arXiv:2202.03028*. doi:<https://doi.org/10.48550/arXiv.2202.03028>
- Geng, A., Moghiseh, A., Redenbach, C., & Schladitz, K. (2023). Improved FRQI on superconducting processors and its restrictions in the NISQ era. *Quantum Information Processing*, *104*. doi:<https://doi.org/10.1007/s11128-023-03838-0>
- Gheorghe-Pop, I.-D., Paschke, A., Mattern, D., Martyniuk, D., Becker, C. K.-U., & Tcholtchev, N. (2022). Ansätze für die strukturierte Entwicklung, das Testen und den Betrieb quantensicherer ICT-Systeme. In R. Neugebauer, *Quantentechnologien* (S. 439 ff). Fraunhofer Verlag.
- Gil-Fuster, E., Eisert, J., & Bravo-Prieto, C. (2024). Understanding quantum machine learning also requires rethinking generalization. *Nature Communications*, *S. 15*, 2277. doi:<https://doi.org/10.1038/s41467-024-45882-z>
- Greenbaum, D. (2015). Introduction to Quantum Gate Set Tomography. *arXiv:1509.02921* .
- Grozea, C. (2024). QUBO Solver Benchmark, online website. Von visited August 28th 2024: <https://qubobenchmark.fokus.fraunhofer.de/> abgerufen



- Johnson, D. S. (2002). A Theoretician's Guide to the Experimental Analysis of Algorithms. In Michael H. Goldwasser, D. S. Johnson, & C. C. McGeoch, Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. doi:<https://doi.org/10.1090/dimacs/059>
- Knill, E., Leibfried, D., Reichle, R., Britton, J., Blakestad, R. B., Jost, J. D., . . . Wineland, D. J. (Jan. 2008). Randomized benchmarking of quantum gates. *Phys. Rev. A*, *S. 77*, 012307. doi:<https://doi.org/10.1103/PhysRevA.77.012307>
- Le, P., Dong, F., & Hirota, K. (2011). A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Information Processing*, 63-84. doi:<https://doi.org/10.1007/s11128-010-0177-y>
- Leermakers, W. v., Wezeman, R., Neumann, N., & Phillipson, F. (2022). Evaluating the Q-score of Quantum Annealers. *IEEE International Conference on Quantum Software (QSW)*, *S. 9-16*. Von <https://arxiv.org/pdf/2208.07633.pdf> abgerufen
- Lubinski, T., Coffrin, C., McGeoch, C., Sathe, P., Apanavicius, J., & Neira, D. E. (2024). Optimization Applications as Quantum Performance Benchmarks. arXiv:2302.02278.
- Lubinski, T., Johri, S., Varosy, P., Coleman, J., Zhao, L., Necaie, J., . . . Proctor, T. (vol. 4 2023). Application-Oriented Performance Benchmarks for Quantum Computing. *IEEE Transactions on Quantum Engineering*, *S. 1-32*. doi:10.1109/TQE.2023.3253761
- Luckow, A., & Klepsch, J. (16. 4 2021). Quantum Computing: Towards Industry Reference Problems. Von <https://digitaleweltmagazin.de/fachbeitrag/quantum-computing-towards-industry-reference-problems/> abgerufen
- Mundada, P. S., Barbosa, A., Maity, S., Wang, Y., Merkh, T., Stace, T., . . . Baum, Y. (August 2023). Experimental Benchmarking of an Automated Deterministic Error-Suppression Workflow for Quantum Algorithms. *Phys. Rev. Applied*, *S. 20*, 024034. doi:<https://doi.org/10.1103/PhysRevApplied.20.024034>
- Sankar, K., Scherer, A., Kako, S., Reifenstein, S., Ghadermarzy, N., Krayenhoff, W. B., . . . Yamamoto, Y. (2021). Benchmark Study of Quantum Algorithms for Combinatorial Optimization: Unitary versus Dissipative. arXiv:2105.03528.
- Sarovar, M., Proctor, T., Rudinger, K., Young, K., & Blume-Kohout, E. N. (2020). Detecting crosstalk errors in quantum information processors. *Quantum*, *S. 4*, 321. doi:<https://doi.org/10.22331/q-2020-09-11-321>
- Tomesh, T., Gokhale, P., Omole, V., Ravi, G. S., Smith, K. N., Vizslai, J., . . . Chong, F. T. (2022). SupermarQ: A Scalable Quantum Benchmark Suite. *IEEE International Symposium on High-Performance Computer Architecture*, *S. 587-603*. doi:<https://doi.org/10.48550/arXiv.2202.11045>
- Wack, A., Paik, H., Javadi-Abhari, A., Jurcevic, P., Faro, I., Gambetta, J. M., & Johnson, B. R. (Oktober 2021). Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers. arXiv:2110.14108.
- Wolf, A., & Grozea, C. (2023). Automatic Conversion of MiniZinc Programs to QUBO. arXiv:2307.10032. doi: 10.48550/arXiv.2307.10032
- Wormsbecher, W. (Februar 2024). <https://www.heise.de/select/ct/2024/2/2332811143798946743>. Abgerufen am 13. 2 2024 von <https://www.heise.de/select/ct/2024/2/2332811143798946743>
- Zander, R., & Becker, C. K.-U. (2024). Benchmarking Multipartite Entanglement Generation with Graph States.