Quantum Computing, Software Engineering, and Artificial Intelligence

Shaukat Ali

Head of Department/Research Professor/Chief Research Scientist

Simula Research Laboratory &

Oslo Metropolitan University

Oslo, Norway

shaukat@simula.no

IPSJ/SIGSE Software Engineering Symposium (SES2024)

September 19, 2024, Yokohama, Japan

simula

Quantum Computing (QC)

QC promises to solve many complex problems, possibly together with classical computing.

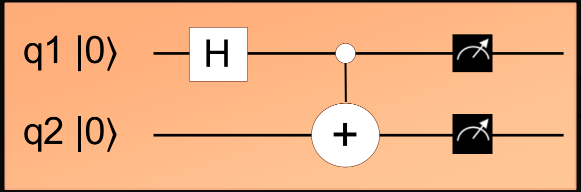
Many technologies are being tried out to build quantum computers.

Superconducting	**Quantum Annealers**	**Photonic**	**Silicon**	**Trapped Ion**	**Neural Atom**
IBM	D-Wave	Xanadu	SemiQon	Quantinuum	PASQAL
IQM		PsiQuantum	Equal 1	IonQ	Atom Computing
Google		QuiX Quantum			ColdQuanta
Chalmers University					QuEra

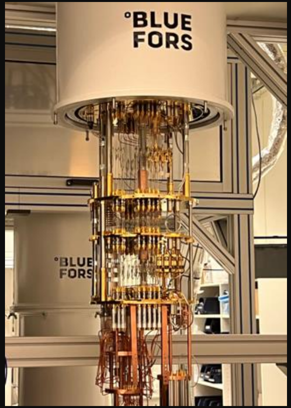
Irrespective of technology, we need to program quantum computers.

simula

Currently, we program quantum computers with quantum circuits.

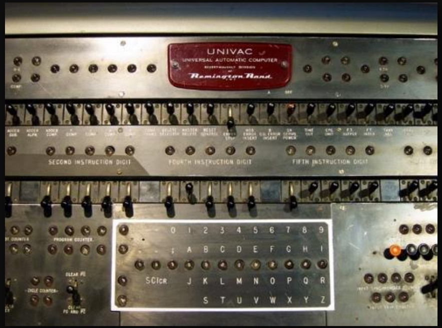


executes on



QAL 9000 Quantum computer,
Chalmers/Wallenberg Centre for Quantum Technologies, Sweden

≈



UNIVAC I
Universal Automatic Computer I
(1950's)

No abstraction!

Nothing intuitive!

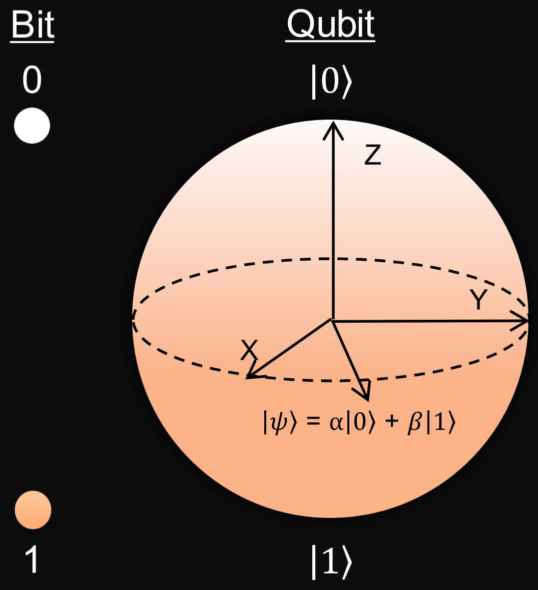
Richard Feynman

People with specialized background can program quantum computers.

"If you think you understand quantum mechanics, you don't understand quantum mechanics."

simula

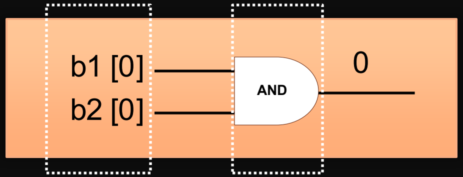
Bit vs. Qubit



Superposition

Classical vs. Quantum Circuit

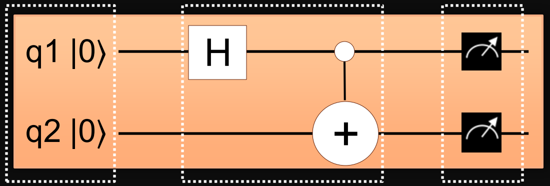
Classical Circuit



Initialization (assign 0/1 for each bit)

Classical Operations

Quantum Circuit

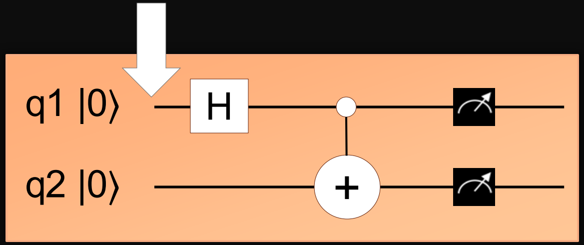


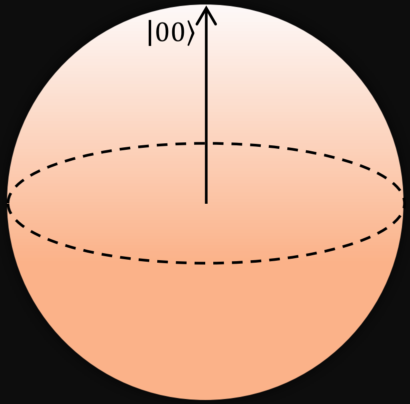
Initialization (assign 0/1 for each qubit)

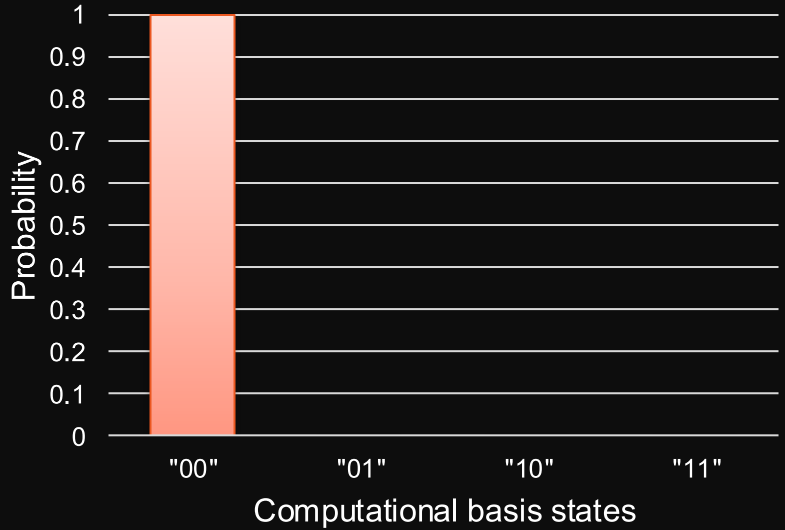
Quantum Operations

Measurement

Quantum Circuit Execution (1/4)

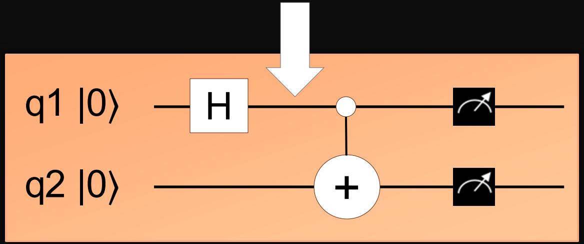


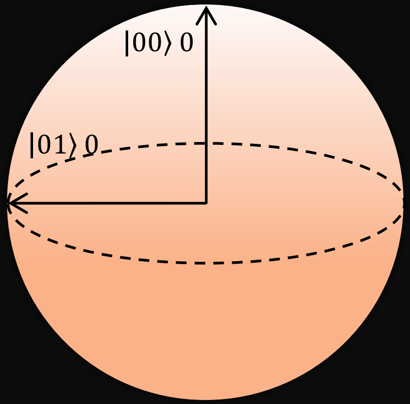


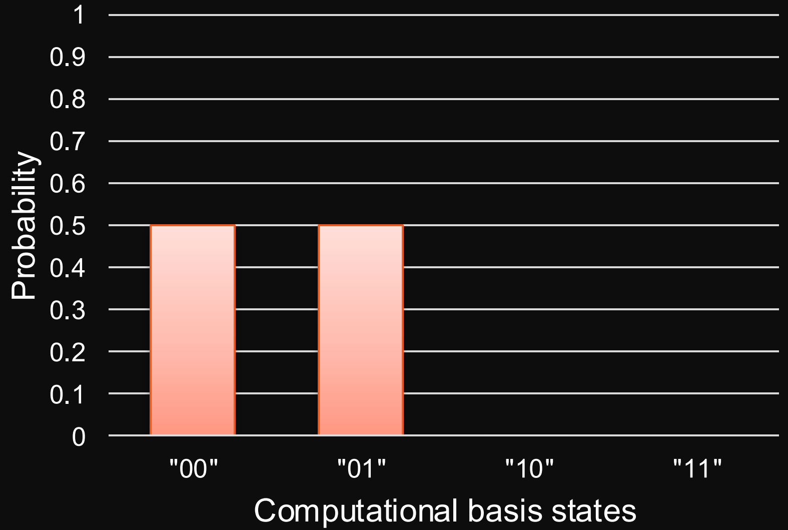


simula

Quantum Circuit Execution (2/4)

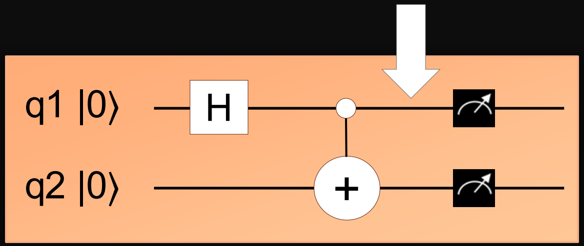


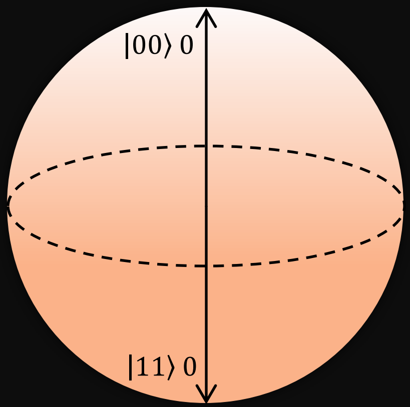


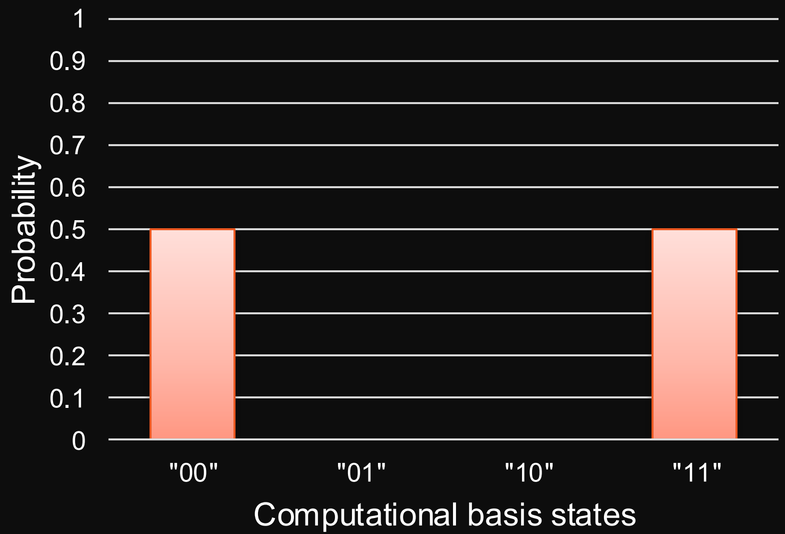


simula

Quantum Circuit Execution (3/4)

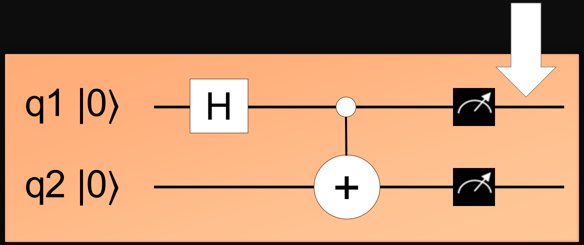


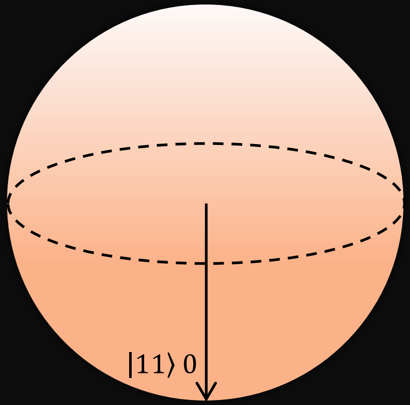


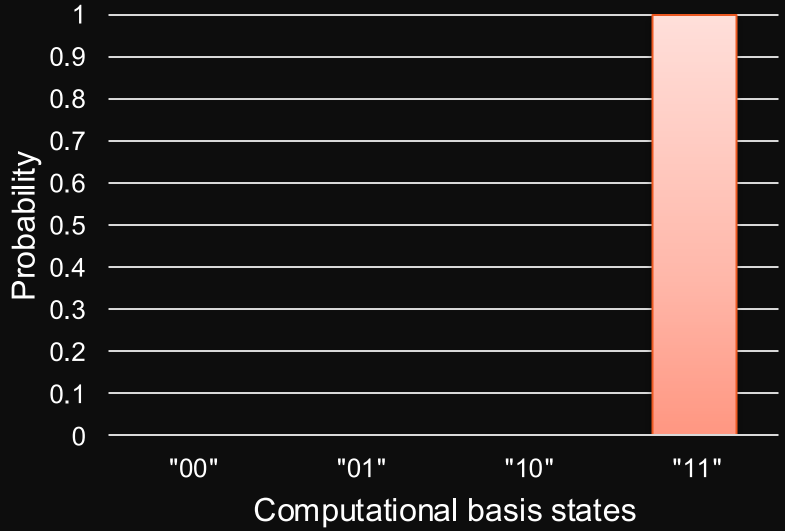


simula

Quantum Circuit Execution (4/4)-1st Shot

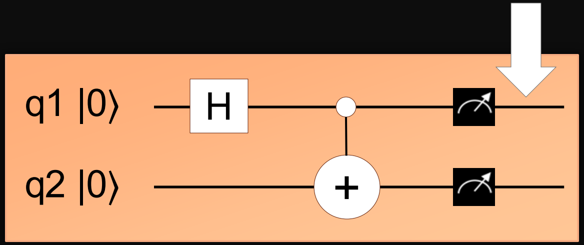


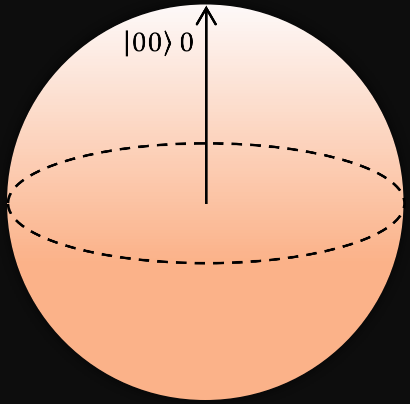


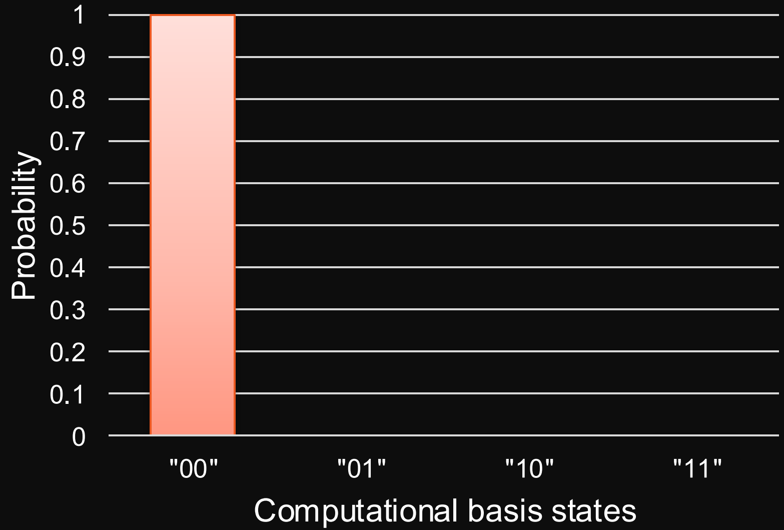


simula

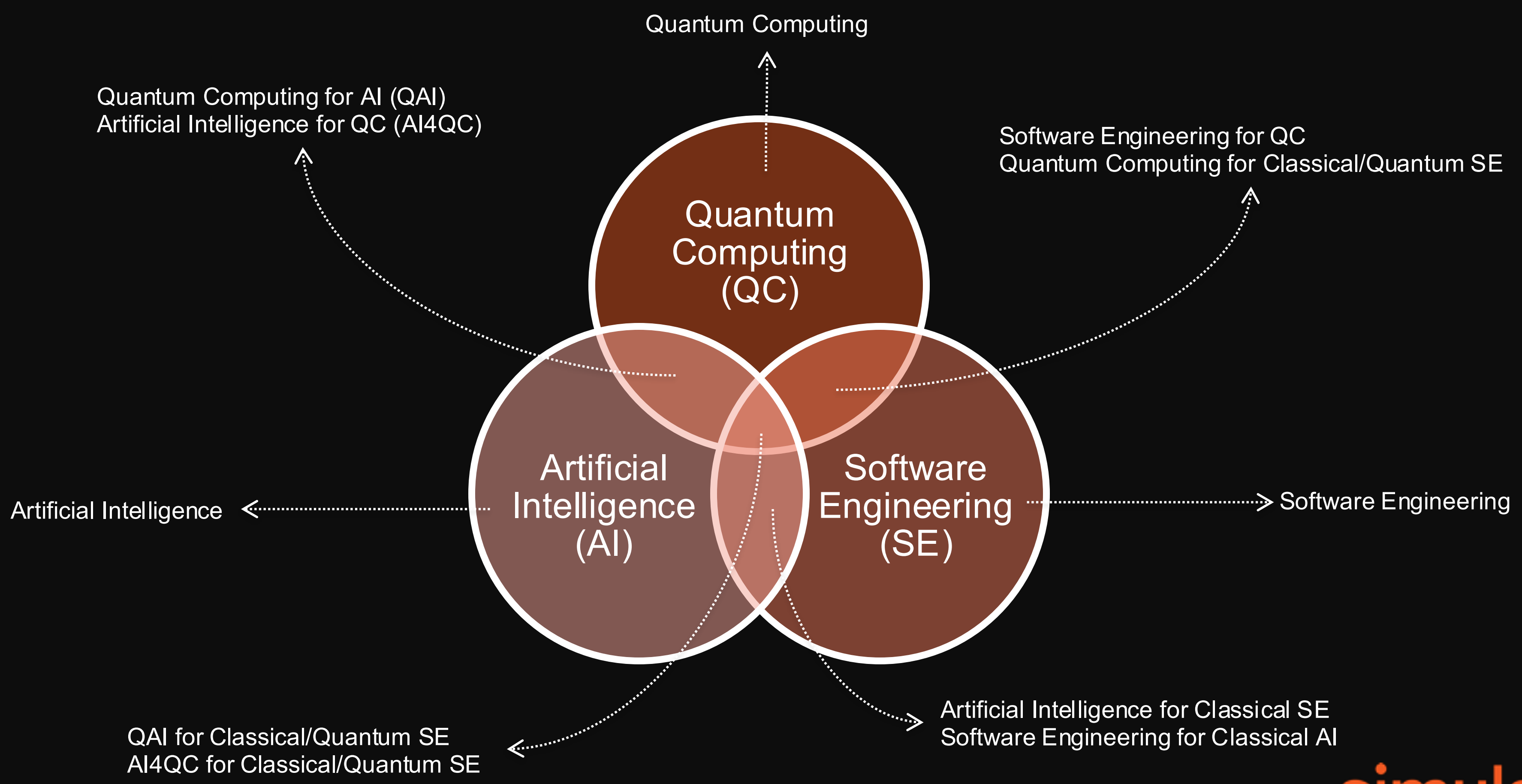
Quantum Circuit Execution (4/4)- 2nd Shot



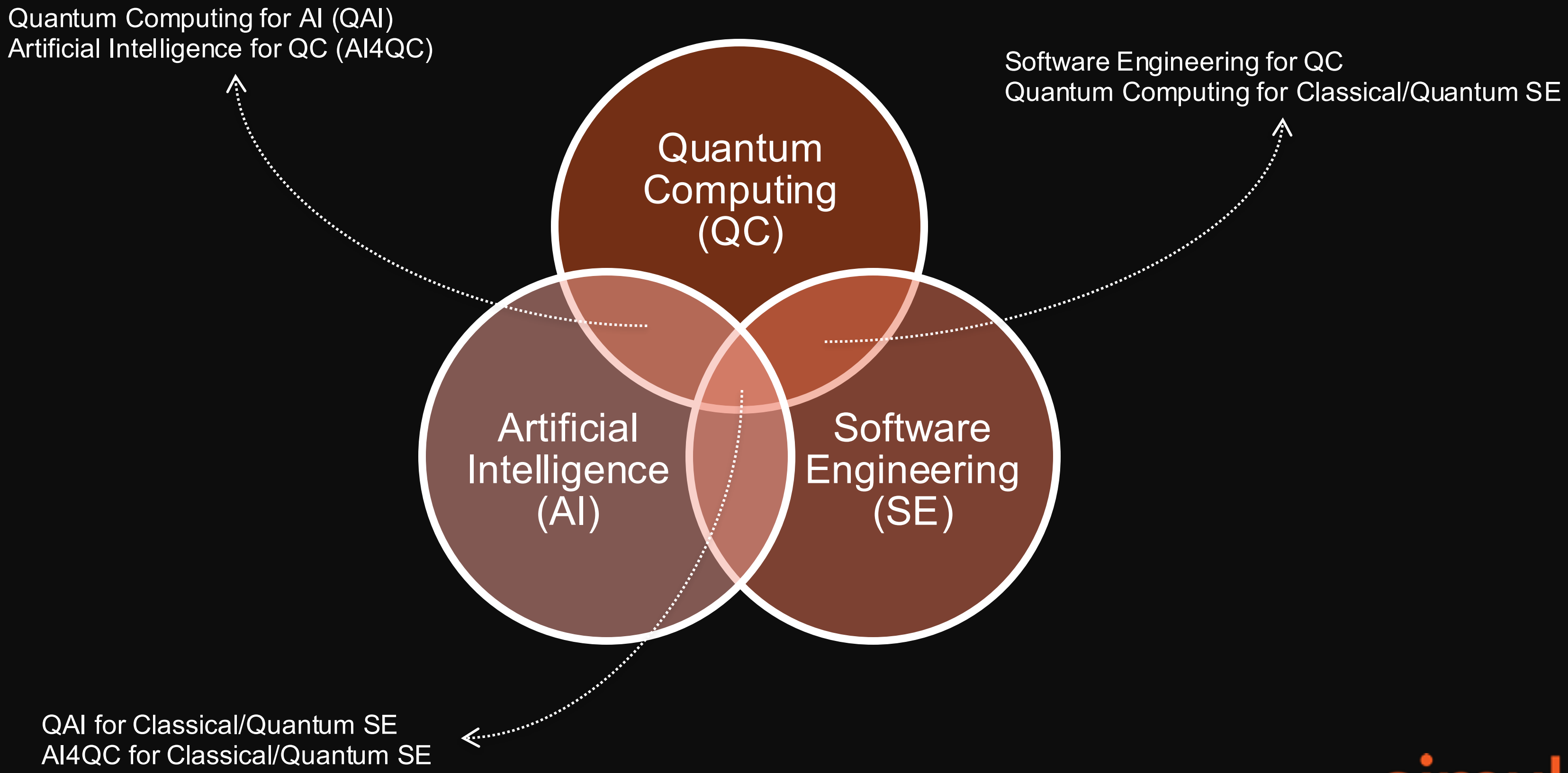




simula


Quantum Computing
Quantum Computing for AI (QAI)
Artificial Intelligence for QC (AI4QC)
Software Engineering for QC
Quantum Computing for Classical/Quantum SE
Quantum Computing (QC)
Artificial Intelligence (AI)
Software Engineering (SE)
Artificial Intelligence
Software Engineering
QAI for Classical/Quantum SE
AI4QC for Classical/Quantum SE
Artificial Intelligence for Classical SE
Software Engineering for Classical AI

simula

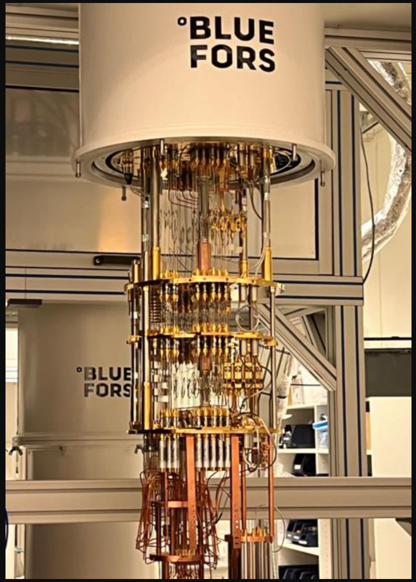

Quantum Computing for AI (QAI)
Artificial Intelligence for QC (AI4QC)
Software Engineering for QC
Quantum Computing for Classical/Quantum SE
Quantum Computing (QC)
Artificial Intelligence (AI)
Software Engineering (SE)
QAI for Classical/Quantum SE
AI4QC for Classical/Quantum SE

What is Quantum Software Engineering?

Quantum computers irrespective of their technology are programmed with quantum software.

Quantum software is at the core of the promised revolutionary QC applications.

Quantum software engineering enables cost-effective and scalable development of dependable quantum software.



QAL 9000 Quantum computer
Chalmers/Wallenberg Centre for Quantum Technologies, Sweden

simula

Shaukat Ali, Tao Yue, and Rui Abreu. 2022. When software engineering meets quantum computing. Commun. ACM 65, 4 (April 2022), 84–88. https://doi.org/10.1145/3512340

Quantum Software Engineering Challenges

Model-Driven Engineering

Need for high-level design methodologies for hybrid software systems

Scalable quantum software maintenance and evolution

Intelligent code generation and orchestration

Testing and Debugging

Efficient test oracles

Test scalability

From simulators to real quantum computers

Test strategies

Programming Paradigms

Complexity of circuit

Composable and reusable quantum software

Abstractions for quantum software

Software Development Process

Managing iterative development

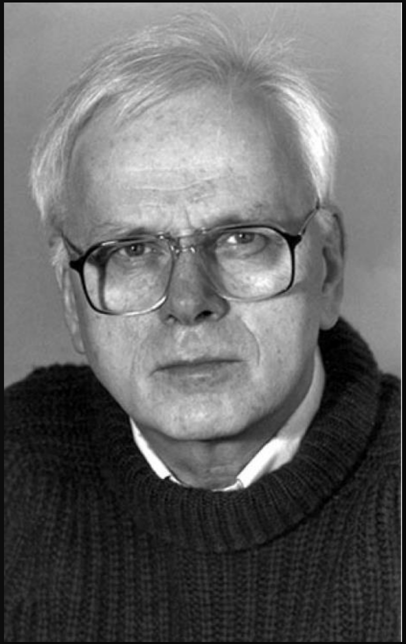
Risk management

Project management

Do we even need quantum software requirements engineering?

Murillo, Juan M., et al. "Challenges of quantum software engineering for the next decade: The road ahead." arXiv preprint arXiv:2404.06825 (2024).

Norwegian Computing Center (Norsk Regnesentral)



Kristen Nygaard

Ole-Johan Dahl

1960's simula

Object-Oriented Paradigm

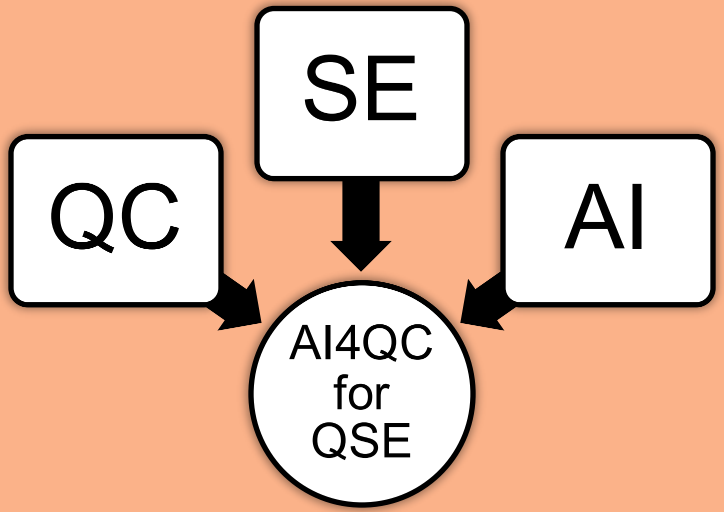
Class | Object | …

2024 Java C++ C# …

We need a novel quantum-oriented paradigm!

simula

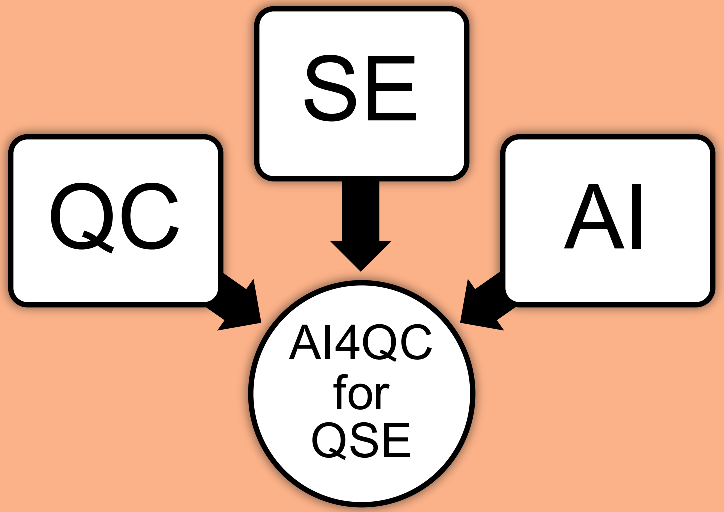
Shaukat Ali and Tao Yue. 2023. On the Need of Quantum-Oriented Paradigm. In Proceedings of the 2nd International Workshop on Quantum Programming for Software Engineering (QP4SE 2023). Association for Computing Machinery, New York, NY, USA, 17–20. https://doi.org/10.1145/3617570.3617868



Quantum Search-based Software Testing (Qu-SBT)

Classical AI-based Noise Reduction for Reliable Quantum Software Development (Q-LEAR)

simula

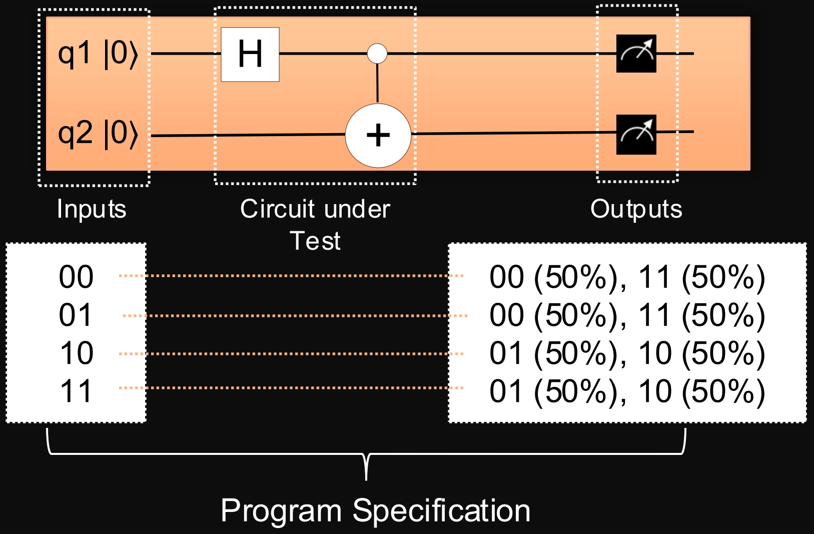


X. Wang, P. Arcaini, T. Yue, S. Ali, Generating Failing Test Suites for Quantum Programs with Search, in 13th Symposium on Search-Based Software Engineering (SSBSE 2021), Bari, Italy, October 11-15, 2021

Quantum Search-based Software Testing (Qu-SBT)

simula

Quantum Software Testing Preliminaries



- Test Case: Input (e.g., 00)
- Test Oracle
  - **Wrong Output (WOO):** For example, 01 output is produced by 00 input
  - **Wrong Output Distribution (WOD):** The observed output distribution is significantly different from the expected distribution tested with a statistical test (e.g., Chi-squared test)

simula

Quantum Search-based Testing (Encoding)

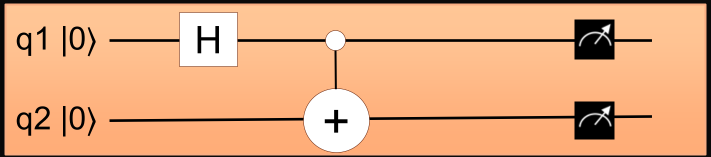
Integer Encoding:

x_1	x_2	..	x_m

- $m=\lceil \beta \times | D_I | \rceil$, where D_I is the domain of possible valid inputs, β is a parameter whose value can be selected
- x_i ranges from 0 to 2^{n-1}, where n is the number of qubits

Inputs

00
01
10
11



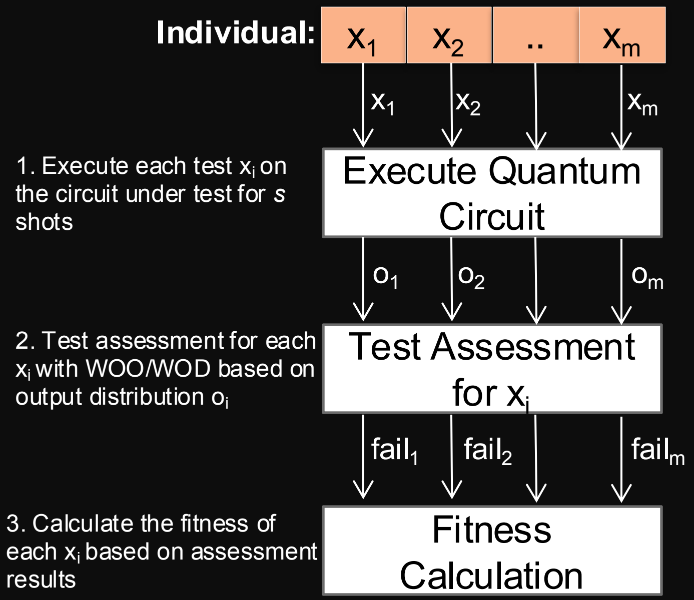
Individual:

0	2

- m=2, D_I is 4, β is 0.5
- x_i ranges from 0 to 3, n is 2

simula

Quantum Search-based Testing (Fitness Function)



- Let ta=[$fail_1$,…,$fail_m$] be the assessments of m tests with WOO or WOD test oracles; $fail_i$ is boolean
- Fitness function: max: $f=|\{fail_j \in ta \mid fail_i = true\}|$

Key Results

Comparison between Genetic Algorithm (GA) and Random Search (RS)

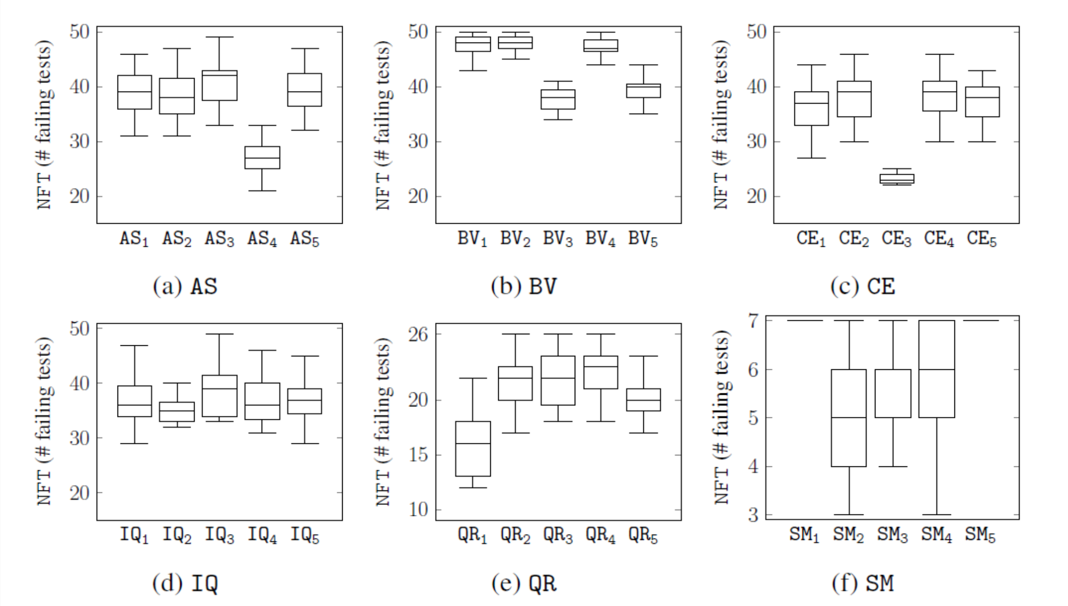
Program	✓	≡
AS	4	1
BV	5	0
CE	5	0
IQ	4	1
QR	5	0
SM	3	2

✓ GA is significantly better than RS

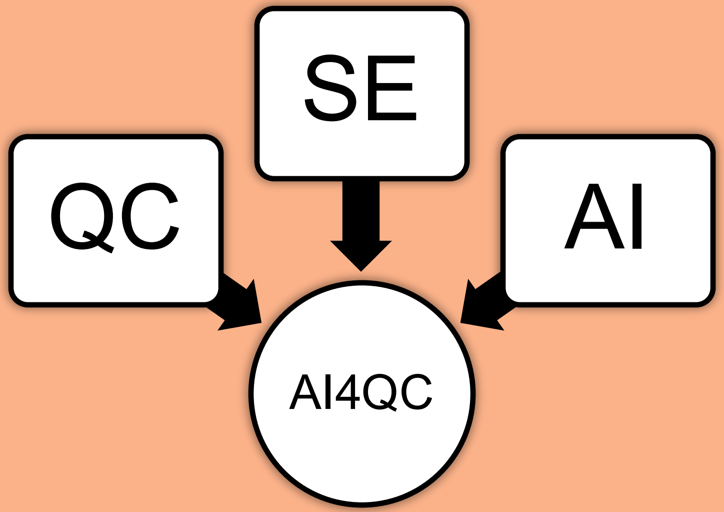
≡ No significant differences between GA and RS

GA outperforms RS for 87% of the faulty quantum programs.

Number of Failing Tests (NFT) of GA across 30 runs for 6 quantum programs



For both simple and complex programs, QuSBT can find maximum failing tests



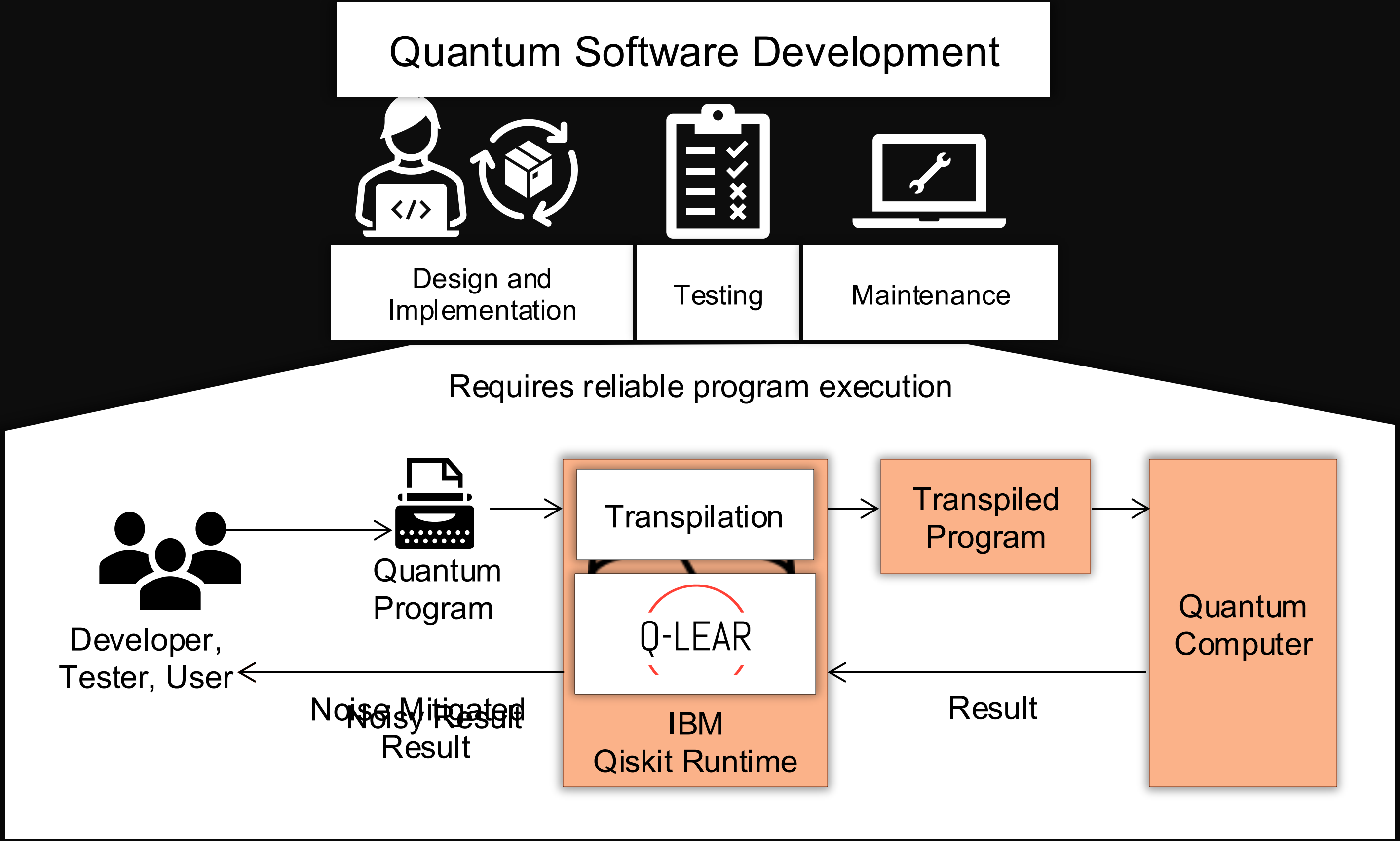
Classical AI-based Noise Reduction for Reliable Quantum Software Development (Q-LEAR)

A. Muqeet, S. Ali, T. Yue, P. Arcaini, A Machine Learning-Based Error Mitigation Approach for Reliable Software Development on IBM's Quantum Computers, in 32nd ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2024), 2024

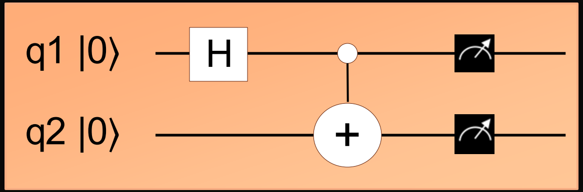
A. Muqeet, S. Ali, T. Yue, P. Arcaini, Mitigating Noise in Quantum Software Testing Using Machine Learning, Accepted in IEEE Transactions on Software Engineering, 2024

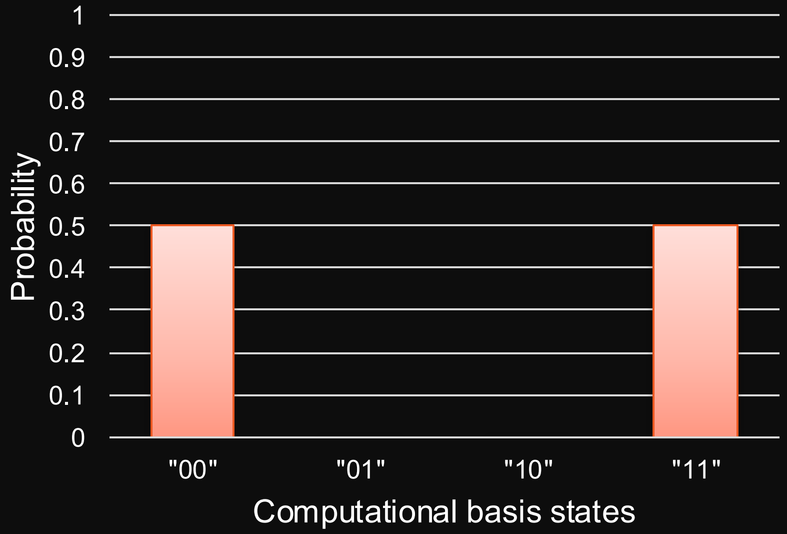
simula

Context

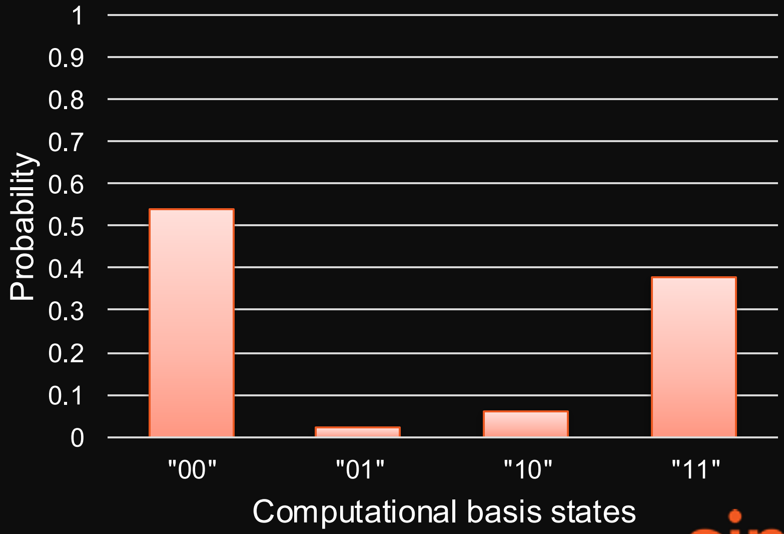

Quantum Software Development
Design and Implementation
Testing
Maintenance
Requires reliable program execution
Developer, Tester, User
Quantum Program
Transpilation
Q-LEAR
IBM Qiskit Runtime
Transpiled Program
Quantum Computer
Result
Noise Mitigated Result
Noisy Result

Effect of Noise





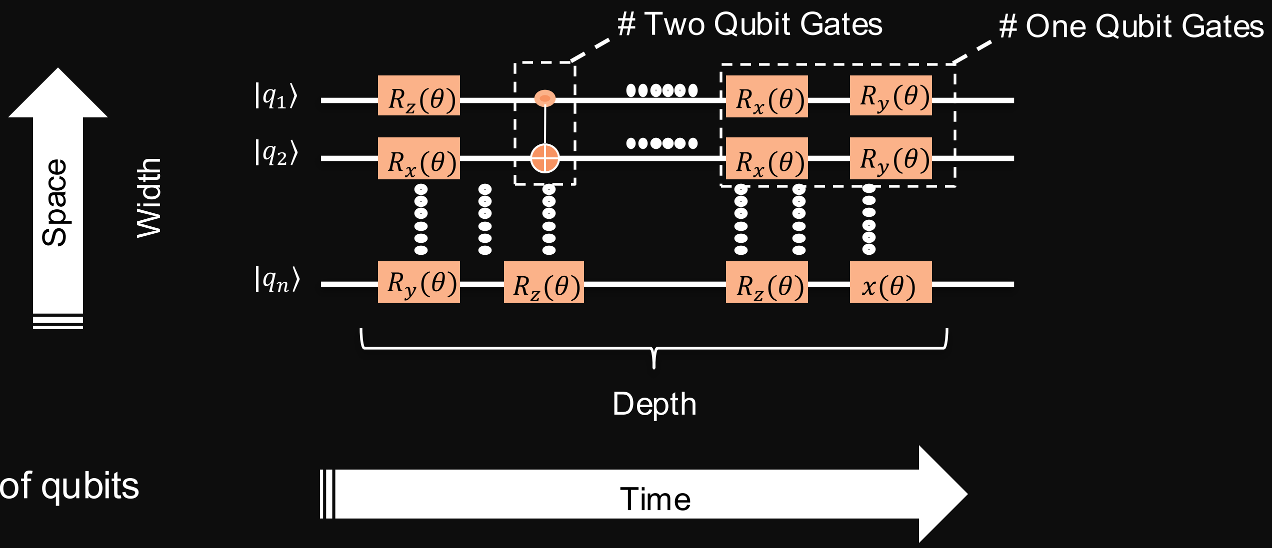
Ideal



Noisy

Circuit Level Features

Quantum circuit properties calculated without circuit execution. These properties directly correlate with quantum noise.



Width: Number of qubits

Depth: Number of steps

simula

Output Level Features

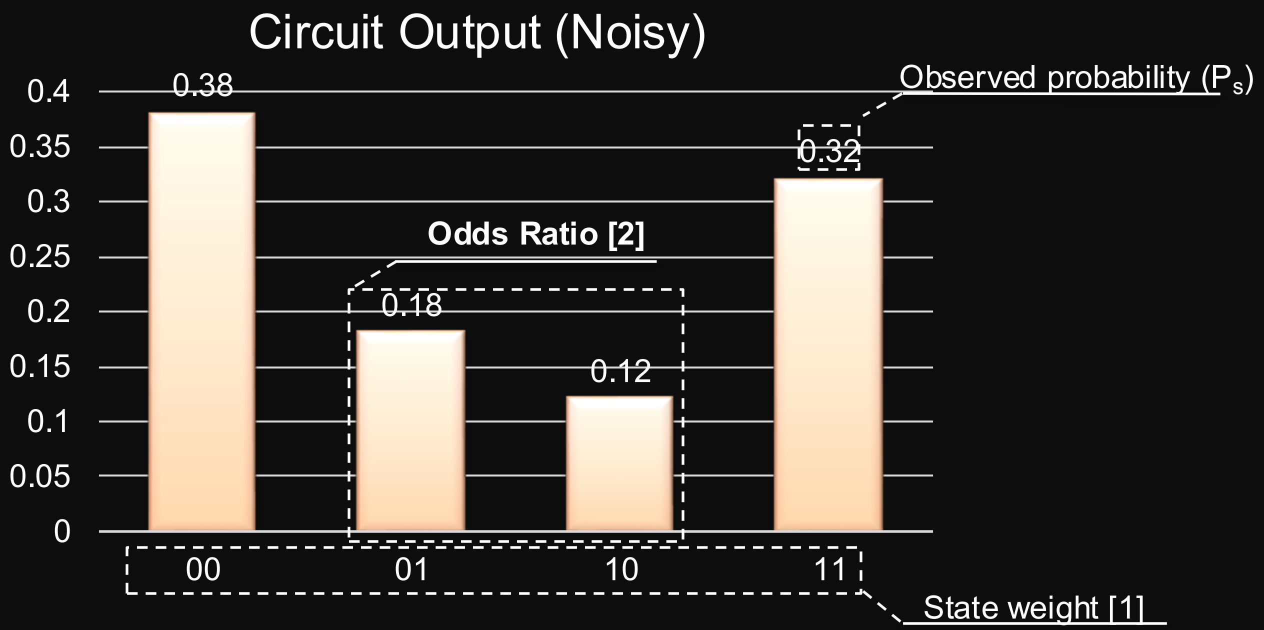
Quantum circuit properties derived from the output of a quantum circuit.

Odds Ratio: Quantifies the strength of association between each output state of two consecutive quantum circuit executions

$$Odds\ Ratio = \frac{odds_r}{odds_{r+1}}$$

$$odds_r = \frac{P_s}{1 - P_s}$$

State weight: Number of qubits in $|1\rangle$

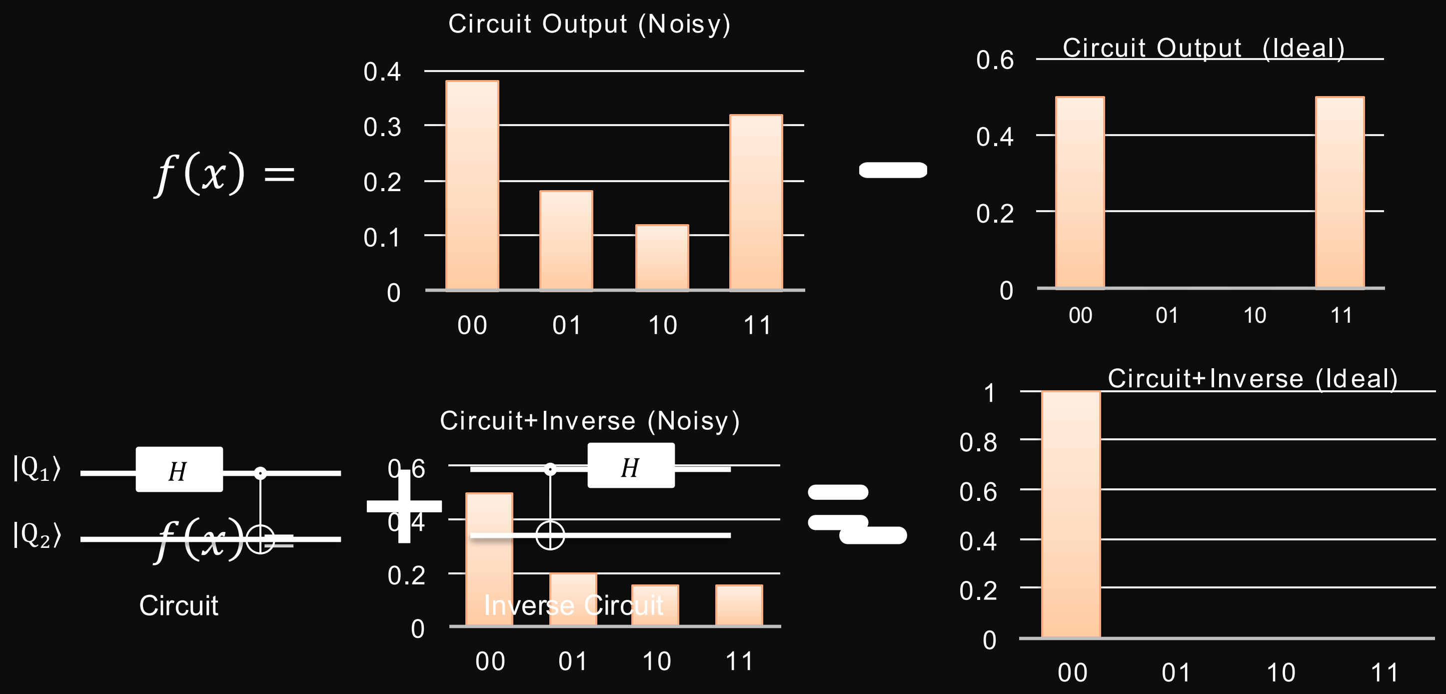


simula

[1] Qraft: Reverse Your Quantum Circuit and Know the Correct Program Output. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems.
[2] Muqeet, A., Yue, T., Ali, S., & Arcaini, P. (2023). Mitigating Noise in Quantum Software Testing Using Machine Learning. arXiv preprint cs.SE/2306.16992.

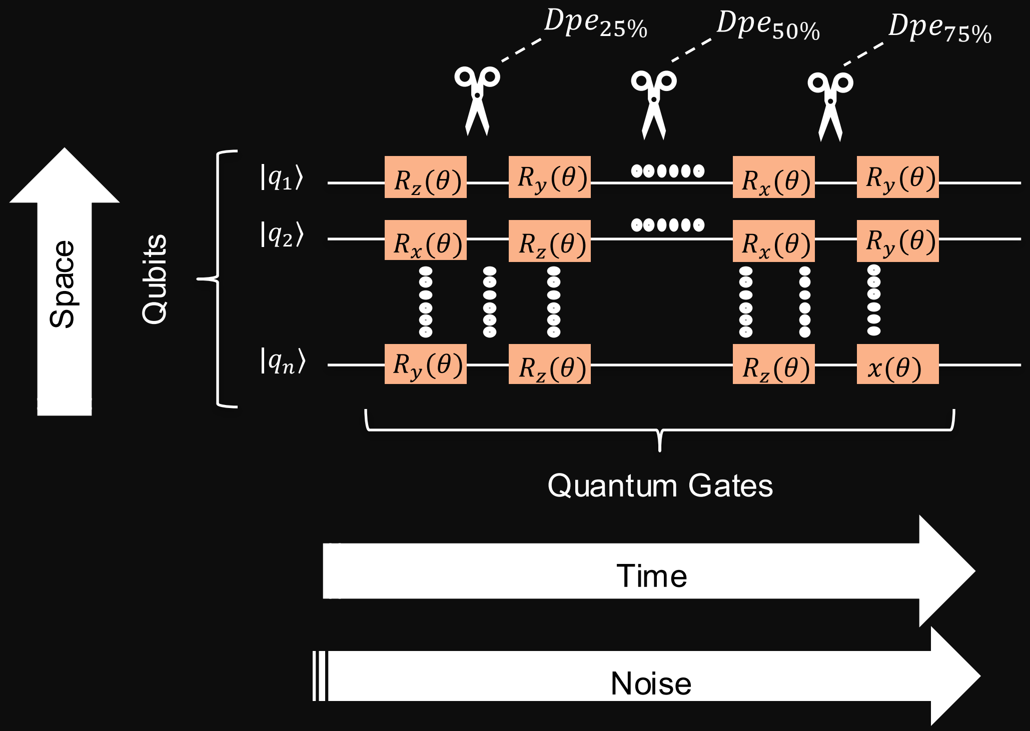
Depth-Cut Error

Assess the amount of noise present in the output of a quantum program after its execution.



Depth-Cut Error

Assesses the amount of noise present in the output of a quantum program after its execution.



simula

Results

Correlations

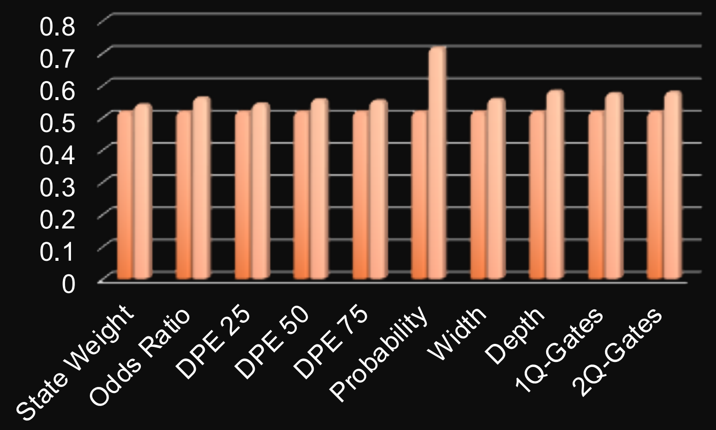
Relationship between Q-LEAR's feature set and error in circuit output due to noise.

$Circuits$	Stw	Odr	DPE_{25}	DPE_{50}	DPE_{75}
GS	0.003	−0.27	0.40	0.64	0.51
PC	−0.44	−0.16	0.49	0.94	0.95
PP	−0.43	−0.19	0.63	0.88	0.89
$QAOA$	−0.10	−0.19	0.58	0.57	0.52
RT	0.61	−0.22	0.94	0.91	0.92
TSP	−0.11	−0.20	0.70	0.59	0.62

All features show either a positive or negative correlation with output error. DPE features have the highest correlations.

Feature Importance

Do all features play an important role in mitigating errors from circuit output?



All features show a median error increase of at least 5%. The observed probability is the most important feature, followed by circuit depth and two-qubit gates.

simula

Results

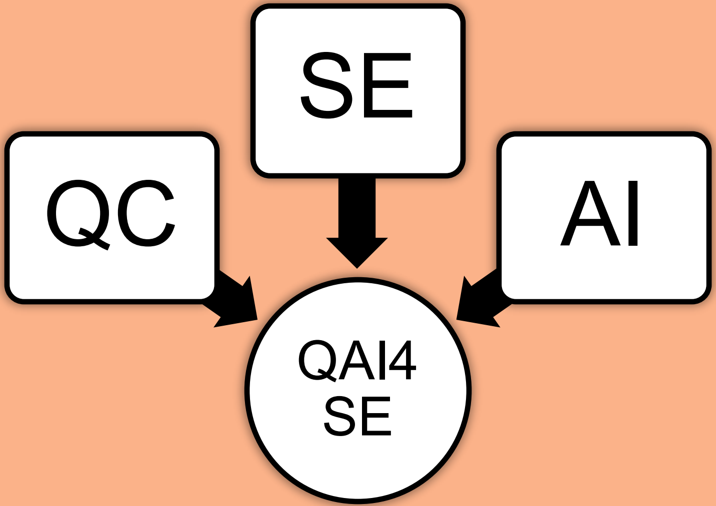
Baseline Comparison

How effective is Q-LEAR in training ML models for error mitigation compared with the state-of-the-art?

Computer	*Simulator*		*Real Computer*	
	QLEAR	*Qraft*	*QLEAR*	*Qraft*
Lagos	**52.0**	12.0	**19.0**	−11.0
Nairobi	**41.0**	17.0	**18.0**	−9.0
Perth	**48.0**	18.0	**16.0**	−14.0
Belem	23.0	48.0	6.0	12.0
Jakarta	**53.0**	24.0	**27.0**	−17.0
Lima	**44.0**	17.0	**20.0**	−15.0
Manila	**39.0**	37.0	**30.0**	−26.0
Quito	**49.0**	28.0	**31.0**	−19.0
Average	**38.1**	25.1	**21.0**	−12.3

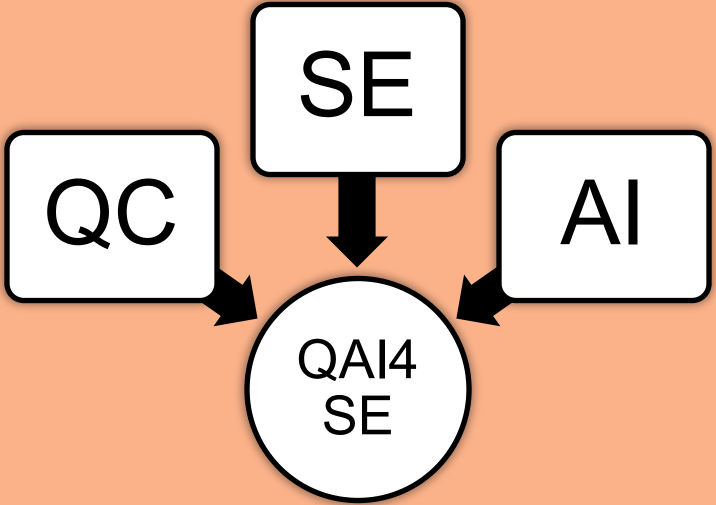
Compared to baseline, Q-LEAR effectively reduces output errors on simulators and real quantum computers.

simula



Classical Test Optimization with Quantum Annealing and Quantum Approximate Optimization Algorithm

Classical Regression Testing with Quantum Extreme Learning Machines



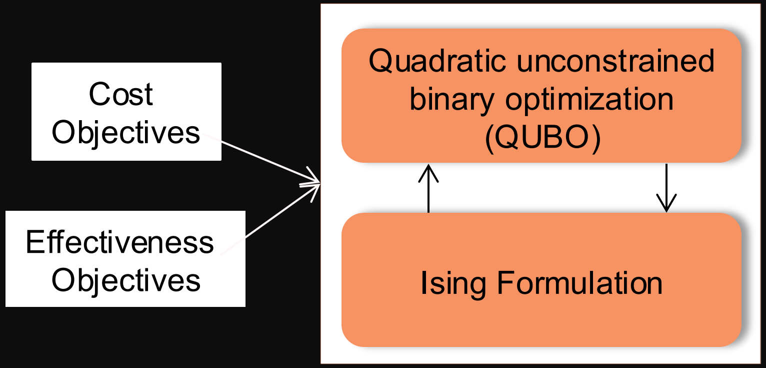
Classical Test Optimization with Quantum Annealing and Quantum Approximate Optimization Algorithm

X. Wang, A. Muqeet, T. Yue, S. Ali, P. Arcaini, Test Case Minimization with Quantum Annealers, in ACM Transactions on Software Engineering and Methodology, 2024 (to appear)

Wang, Xinyi, et al. "Guess What Quantum Computing Can Do for Test Case Optimization." arXiv preprint arXiv:2312.15547 (2023).

Problem Encoding

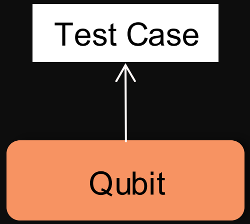
Optimization Objectives



Cost Objective: Minimize the execution time of the selected subset of test cases

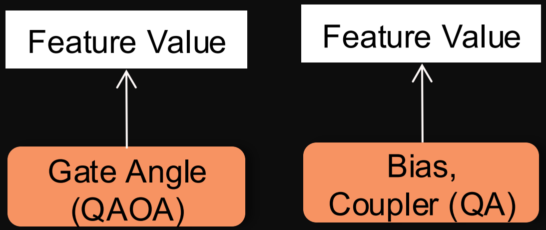
Effectiveness Objective: Maximize the code coverage of the selected subset of test cases

Test Cases to Physical Qubits

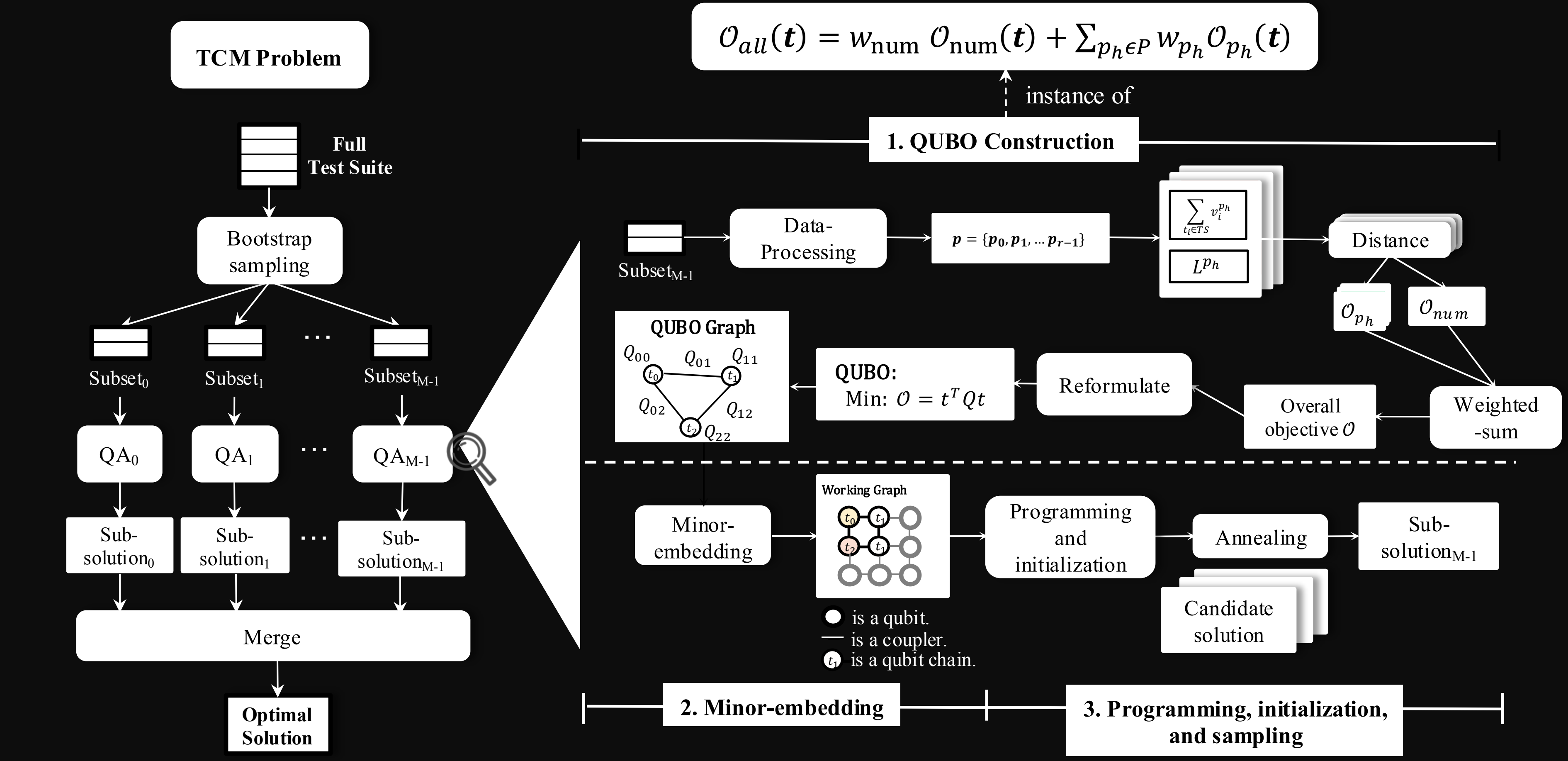


Example: One test case is mapped to one qubit.

Test Case Feature Values



A test case has code coverage of 10 statements



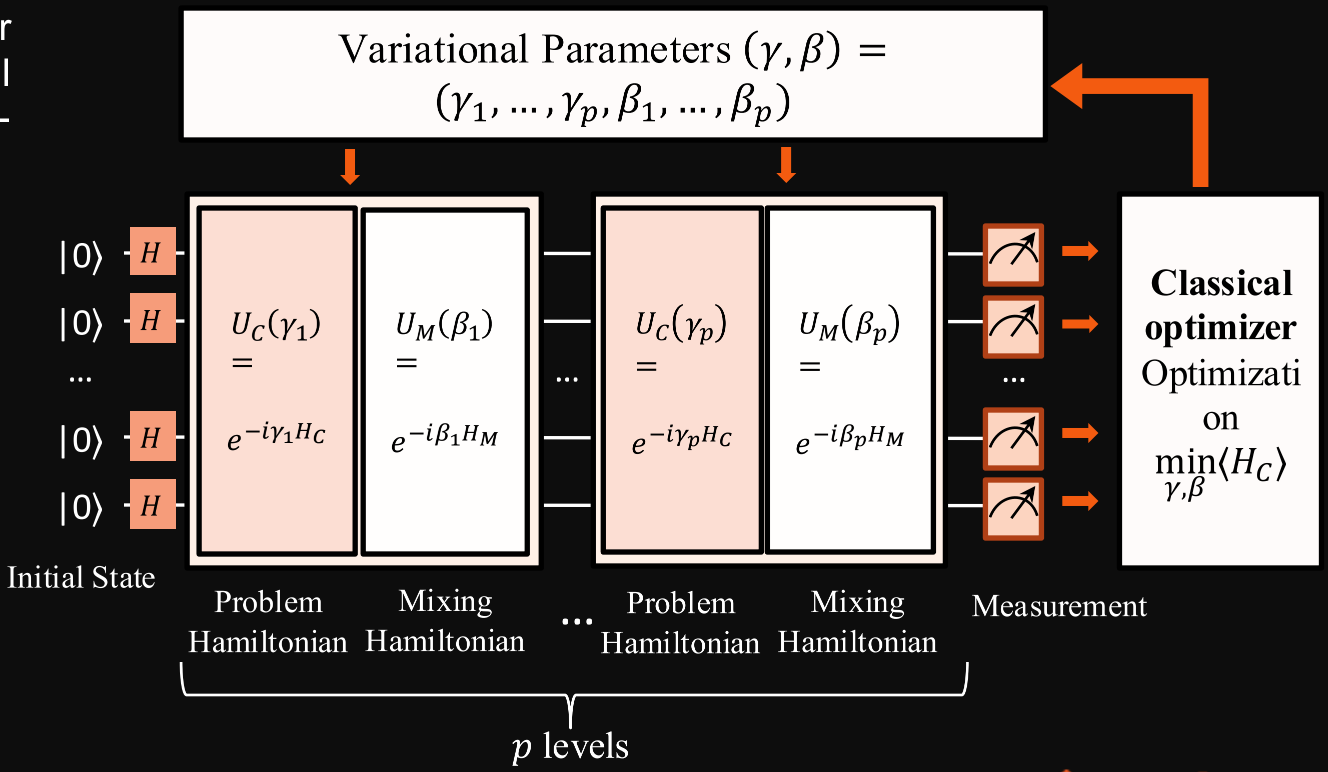
Test Case Minimization (TCM) with Quantum Annealing

Quantum Approximate Optimization Algorithm (QAOA)

Hybrid iterative algorithm for solving combinatorial optimization problems on gate-based quantum computers

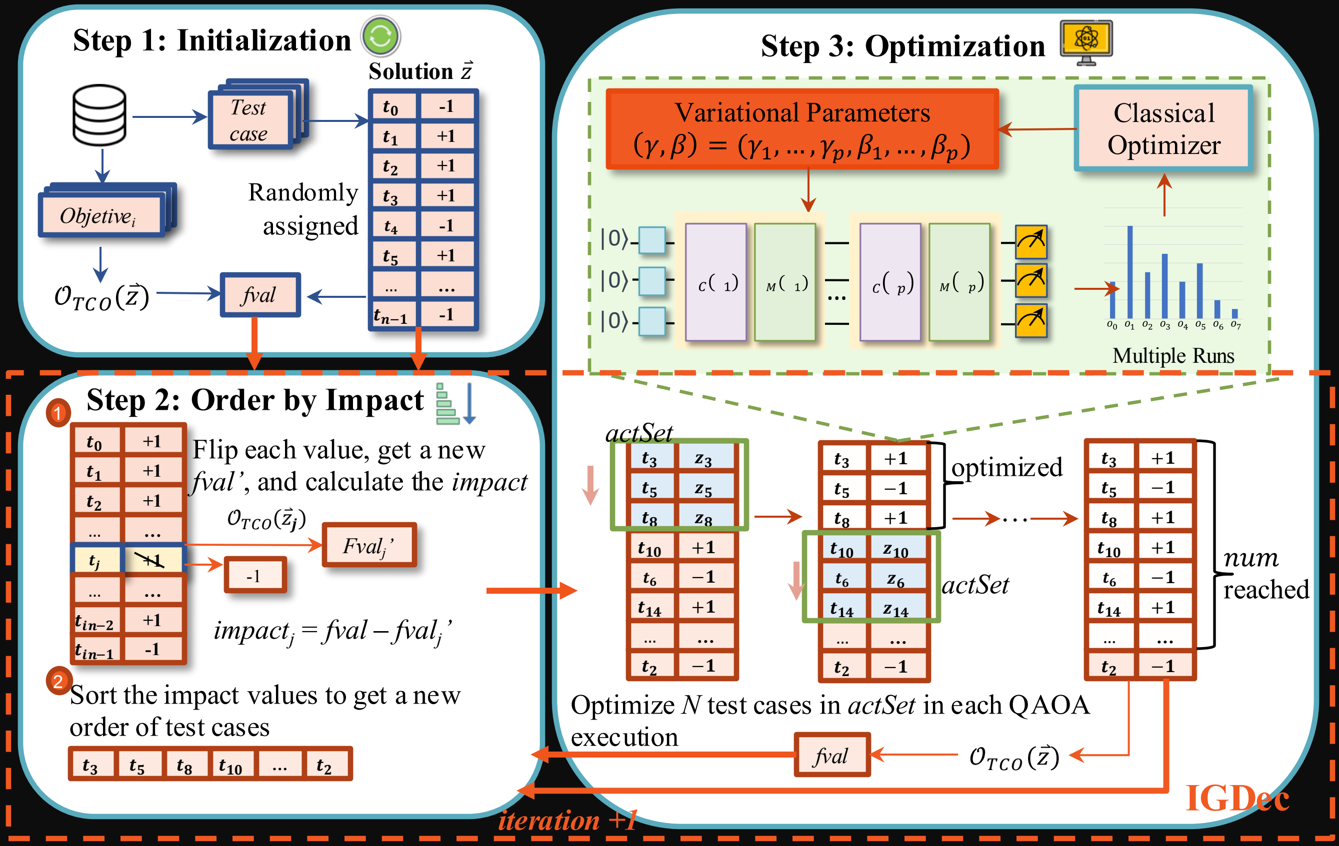
Problem Hamiltonian encodes objective function. Mixing Hamiltonian enables search space exploration.

Classical optimizer to optimize the parameters γ, β of the two Hamiltonian



simula

Test Case Optimization with QAOA



simula

Empirical Evaluation

- **Optimization Problems:** Test Case Minimization and Selection
- **Datasets:** (1) PaintControl and IOF/ROL from ABB Robotics Norway; (2) GSDTSR6 from Google; (3) Orona, Spain
- **Objectives (ABB, Google):** (1) Minimize the number of test cases; (2) Minimize execution time; (3) Maximize fault detection capability
- **Objectives (Orona):** (1) Minimize the cost; (2) Maximize input diversity; (3) Maximize passenger count; (4) Maximize travel distance

simula

Results

Quantum Annealing

Execution: D-Wave's Quantum Annealers

Contributions: (1) A generic problem formulation; (2) Employed *bootstrap sampling* to decompose a large problem; (3) Compared with three baseline approaches on three real-world datasets (ABB Robotics, Google)

Key Results: BootQA demonstrates similar effectiveness with simulated annealing but has the highest efficiency

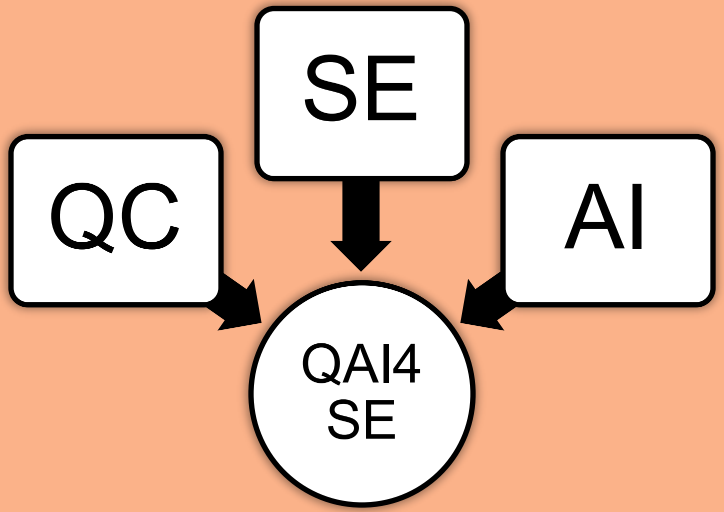
QAOA

Execution: IBM's gate-based quantum computer simulators and real quantum computers

Contributions: (1) A novel formulation for test case optimization; (2) Comparison with three baseline approaches on three industrial datasets (ABB, Google, and Orona)

Key Results: Better or at least the same effectiveness as the baselines

simula



Quantum Extreme Learning Machines for Testing Industrial Elevator Software

(QUELL)

X. Wang, S. Ali, A. Arrieta, P. Arcaini, M. Arratibel, Application of Quantum Extreme Learning Machines for QoS Prediction of Elevators' Software in an Industrial Context in 32nd ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2024), 2024

simula

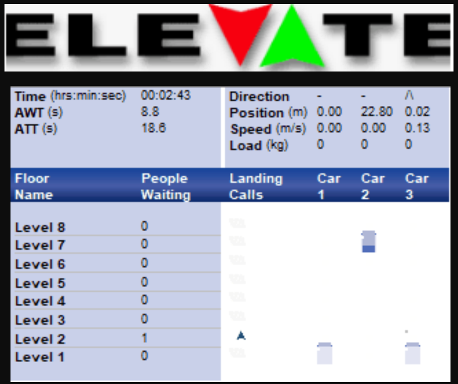
Industrial Context: Orona, Spain

Orona

- Develop and maintain elevators
- Software in the loop testing, hardware in the loop testing, etc

Elevate

- Performance analysis
- Simulation with visual display
- Building, elevator and passenger data



Elevator attributes

- Safety
- Quality of service (QoS)
- ...

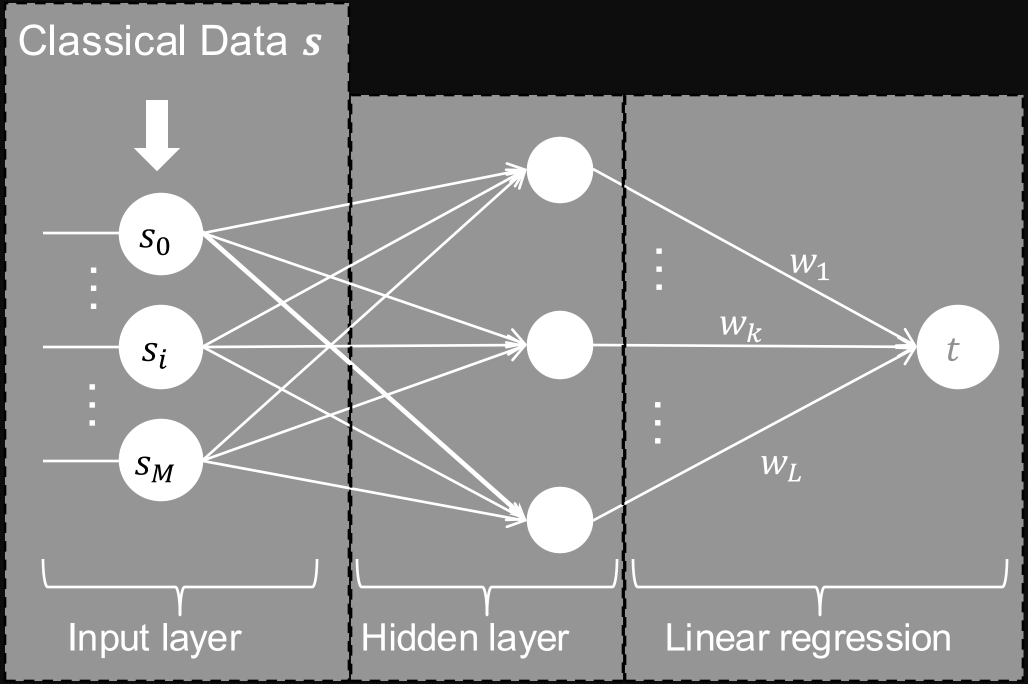
Application Context

- Elevator software testing
- Classical machine learning-based approach

Study quantum machine learning in this context

simula

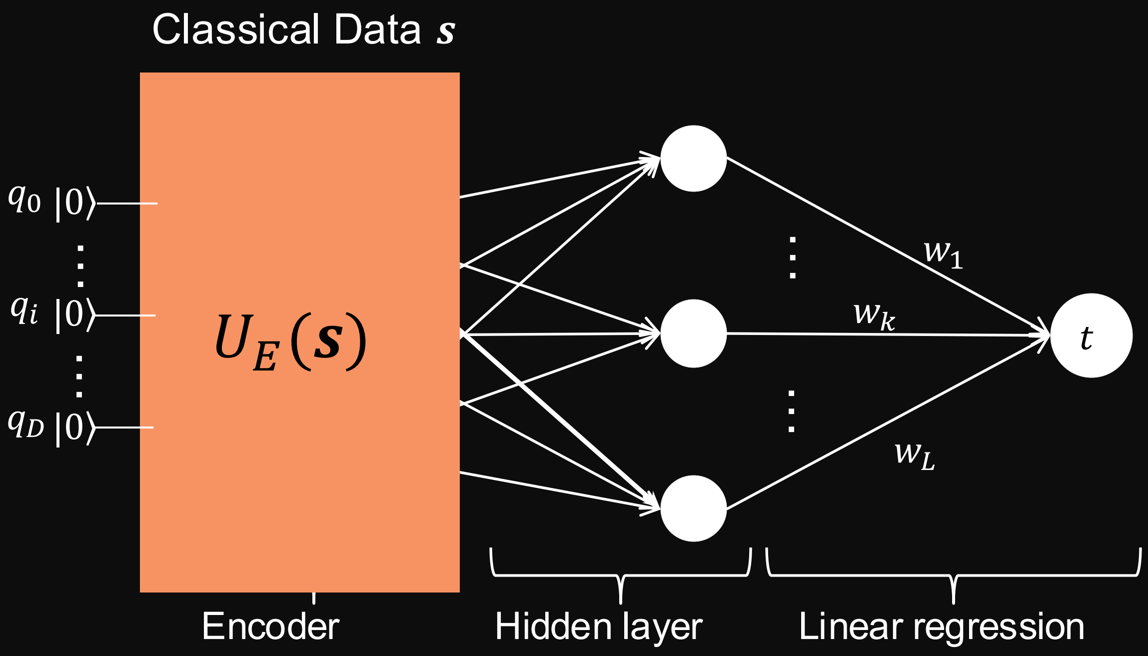
Classical Extreme Learning Machines



- Feedforward neural network
- Classical data into the input layer
- Hidden layer: **fixed** and **randomly** assigned weights and biases
- Train the **linear regression model** on the output layer's weights to predict the target value

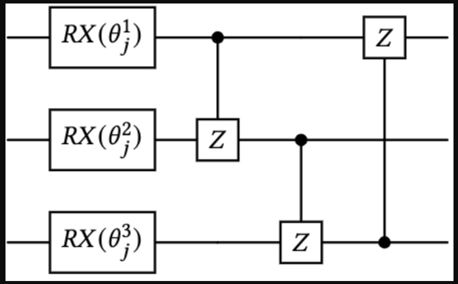
simula

Classical to Quantum Extreme Learning Machines



Encoder

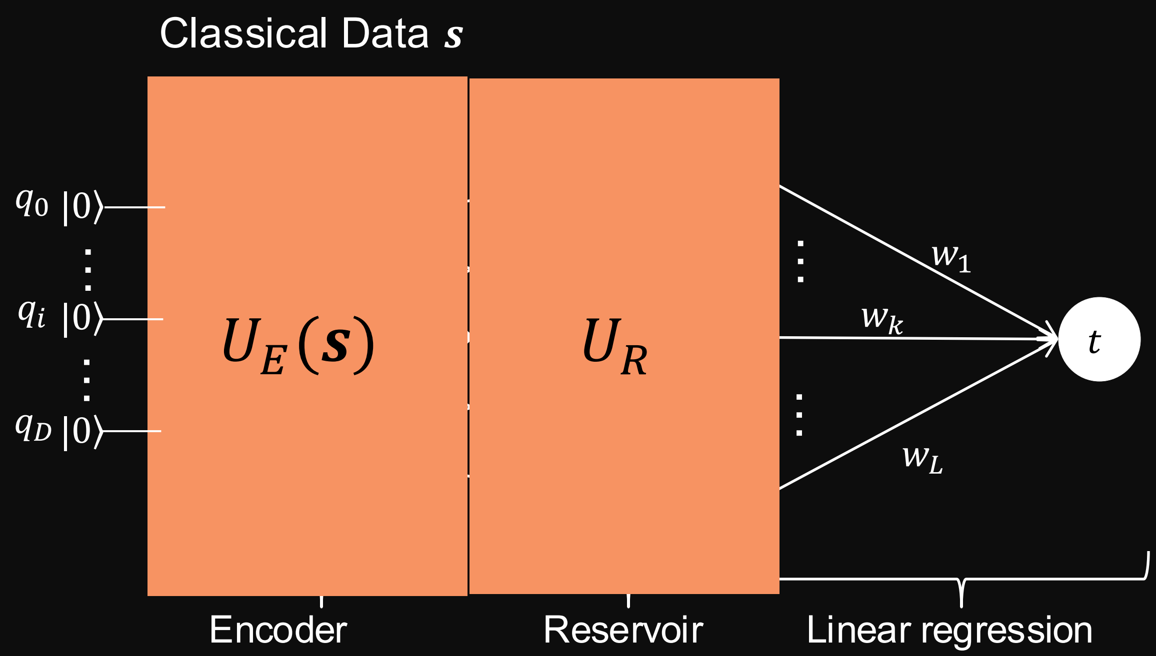
Maps classical data into high dimensional quantum states



Determined hardware efficient **encoder**

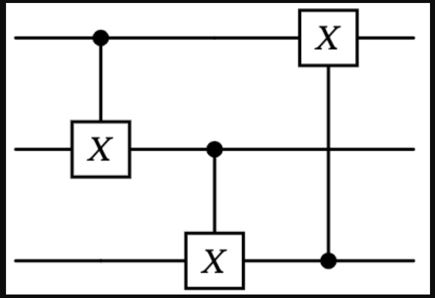
simula

Classical to Quantum Extreme Learning Machines



Quantum Reservoir

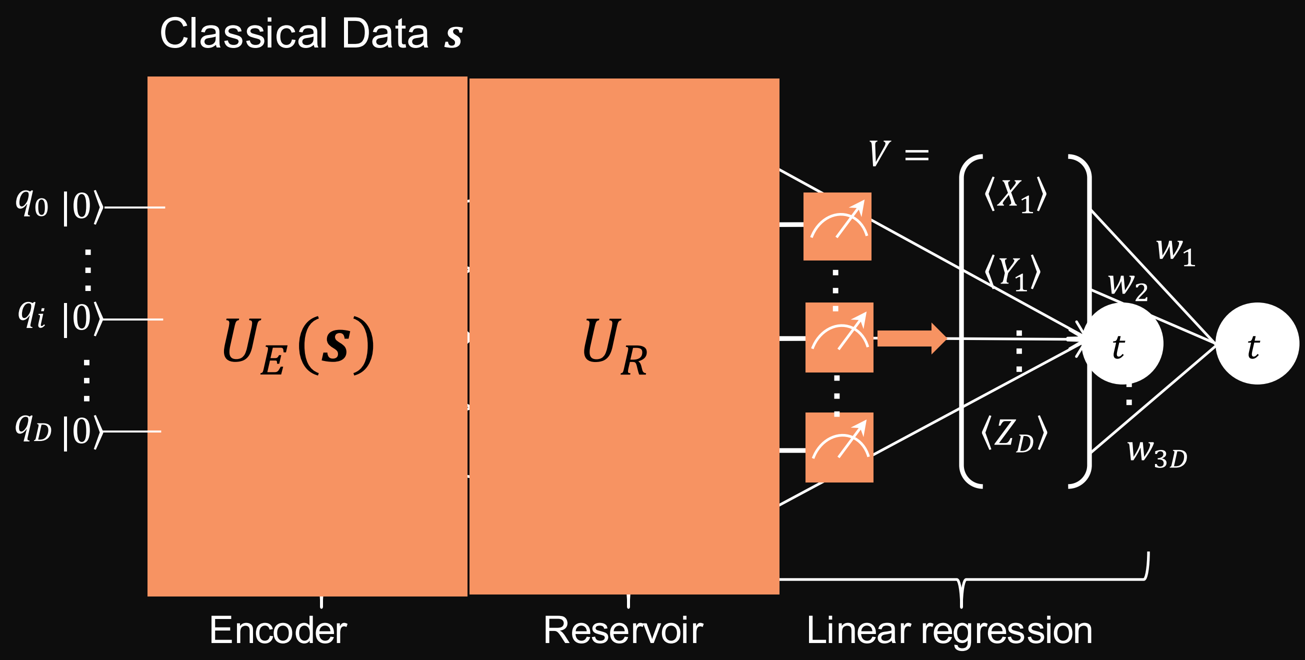
Processes the encoded quantum state



CNOT reservoir

simula

Classical to Quantum Extreme Learning Machines



Training

Read qubits

Apply linear regression

Benefits

Efficient linear regression training with fewer features while maintaining good prediction quality.

Enables applications that require predictions in real-time

simula

Key Results and Findings

Results

Evaluation with real operational data and existing classical machine learning techniques at Orona

QUELL with few features outperforms QUELL with the maximum number of features (10 in our context)

For the same prediction task in our industrial context, QUELL outperforms classical machine-learning approaches

Applications

Runtime predictions during elevators' real operation

Integration into existing digital twins of elevators

Future Directions

Dealing with noise

Build new quantum encoder and reservoir types

Conclusions and Way Forward

- Quantum software engineering (QSE) is immature; many research opportunities exist.
- Applying quantum search and optimization in classical software engineering (e.g., software design and development) is still being explored.
- Many opportunities exist to speed up classical search and optimization and AI techniques (e.g., quantum reservoir computing)
- The potential of applying classical search and optimization and AI techniques to quantum computing, including quantum circuit design and noise learning, is vast and promising.
- Applying quantum search and optimization to solve QSE problems is mostly untouched.

simula