Comparing and learning figures of merit for quantum circuit compilation

Harshdeep Singh*[1], Marvin Richter[1], Mats Granath[2], Anton Frisk Kockum[1]

[1]Department of Microtechnology and Nanoscience,
Chalmers University of Technology, 41296 Gothenburg, Sweden
[2]Department of Physics, University of Gothenburg, 41296 Gothenburg, Sweden
**harshdeep.singh@chalmers.se*

Table of Contents

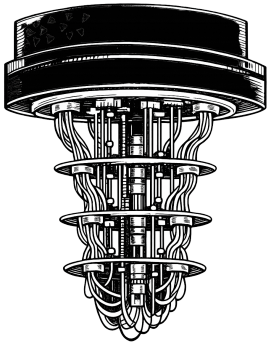
1. Quantum Compilation
2. Figures of Merit
3. wPST
4. Connecting wPST to a Quantum Compiler

Quantum Circuits on NISQ Hardware

Quantum circuit compilation: Making quantum circuits hardware–ready!

Quantum Circuits on NISQ Hardware

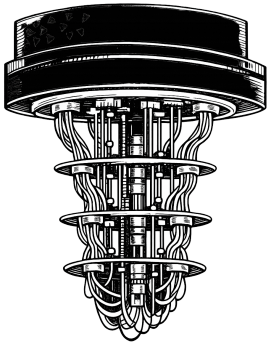
Quantum circuit compilation is needed because,



NISQ Hardware

Quantum circuit compilation is needed because,

- *(Not all gates are native!)*
  Native gate set: collection of quantum gates that a quantum computer can implement directly in hardware without decomposition.

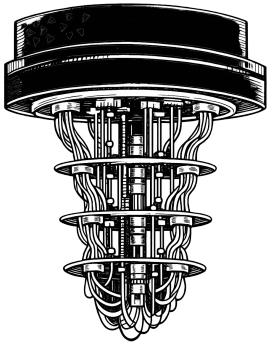


NISQ Hardware

Quantum circuit compilation is needed because,

- *(Not all gates are native!)*
  Native gate set: collection of quantum gates that a quantum computer can implement directly in hardware without decomposition.

- *(Not all qubits are connected!)*
  Coupling map: specifies which pairs of qubits can directly take part in two-qubit operations.



Quantum Circuit Compilation

The process of translating a high-level quantum algorithm into a sequence of physical operations (gates) that can be executed on a specific quantum hardware architecture.

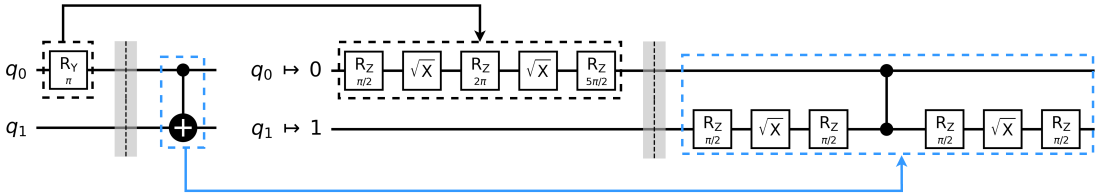


Figure: Running a quantum circuit on IBM Miami.

Quantum Circuit Compilation-Steps

- **Layout** – Maps logical qubits to physical qubits.

Quantum Circuit Compilation-Steps

- **Layout** – Maps logical qubits to physical qubits.
- **Routing** – Inserts SWAP operations to satisfy connectivity constraints.

Quantum Circuit Compilation-Steps

- **Layout** – Maps logical qubits to physical qubits.
- **Routing** – Inserts SWAP operations to satisfy connectivity constraints.
- **Translation** – Converts operations into native hardware gates.

Quantum Circuit Compilation-Steps

- **Layout** – Maps logical qubits to physical qubits.
- **Routing** – Inserts SWAP operations to satisfy connectivity constraints.
- **Translation** – Converts operations into native hardware gates.
- **Optimization** – Simplifies the circuit and reduces depth.

Quantum Circuit Routing

- Routing is hard, and we need an efficient one!

Quantum Circuit Routing

- Routing is hard, and we need an efficient one!

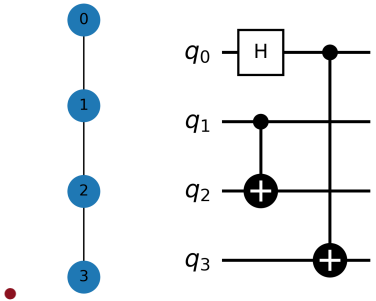
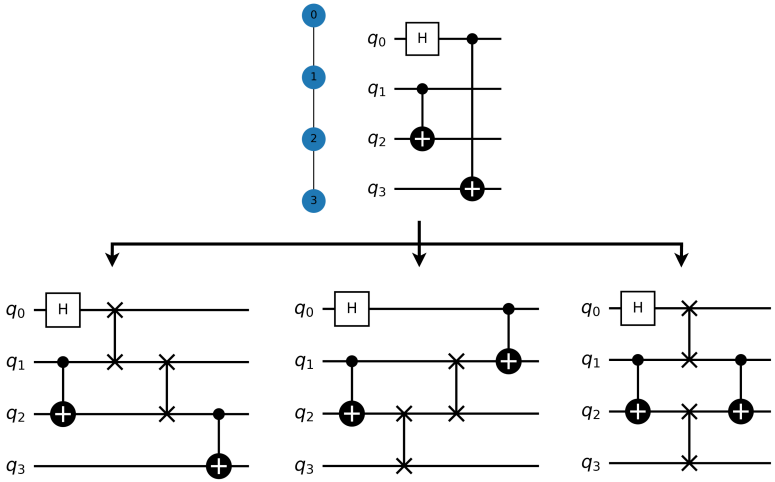


Figure: Circuit routing on a linear graph.

Quantum Circuit Routing



Quantum Circuit Routing

Therefore, the central question becomes how to score the placement of the swaps, or more generally, how to rank the different routed quantum circuits??

Figures of Merit

- **Figures of Merit (FoMs):** quantitative metrics that can quantify the quality of a compiled quantum circuit.

Figures of Merit

- **Figures of Merit (FoMs):** quantitative metrics that can quantify the quality of a compiled quantum circuit.
- FoMs should provide a systematic way to evaluate how well a given compilation performs under realistic hardware constraints.

Figures of Merit

- **Figures of Merit (FoMs):** quantitative metrics that can quantify the quality of a compiled quantum circuit.
- FoMs should provide a systematic way to evaluate how well a given compilation performs under realistic hardware constraints.
- A good FoM $\rightarrow$ A good compiled circuit $\rightarrow$ Better results.

A Good FoM

- Hardware-Aware

A Good FoM

- Hardware-Aware
- Scalable

A Good FoM

- Hardware-Aware
- Scalable
- Computationally efficient

A Good FoM

- Hardware-Aware
- Scalable
- Computationally efficient
- Good correlation with Fidelity

A Good FoM

- Hardware-Aware
- Scalable
- Computationally efficient
- Good correlation with Fidelity
- Distinguishability

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

Measurement-Based FoMs

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate

Measurement-Based FoMs

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate
- Circuit Depth

Measurement-Based FoMs

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate
- Circuit Depth
- Number of SWAP gates

Measurement-Based FoMs

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate
- Circuit Depth
- Number of SWAP gates
- Fail to capture the intricacies of the circuit and hardware[a]

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Measurement-Based FoMs

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate
- Circuit Depth
- Number of SWAP gates
- Fail to capture the intricacies of the circuit and hardware[a]

Measurement-Based FoMs

- Require circuit measurements

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate
- Circuit Depth
- Number of SWAP gates
- Fail to capture the intricacies of the circuit and hardware[a]

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Measurement-Based FoMs

- Require circuit measurements
- Distance-based FoMs

Different FoMs

FoMs can be broadly classified into:

Circuit-Based FoMs

- Often straightforward and easy to evaluate
- Circuit Depth
- Number of SWAP gates
- Fail to capture the intricacies of the circuit and hardware[a]

[a] Quetschlich, Burgholzer, and Wille 2025; Hopf et al. 2025.

Measurement-Based FoMs

- Require circuit measurements
- Distance-based FoMs
- Probabilities-based FoMs

Distance-based FoMs

- The Hellinger distance is a statistical measure of dissimilarity between two probability distributions[1].

[1] Hopf et al. 2025.

Distance-based FoMs

- The Hellinger distance is a statistical measure of dissimilarity between two probability distributions[1].
- Let $P = \{p_i\}_{i=1}^n$ and $Q = \{q_i\}_{i=1}^n$ be two discrete probability distributions,

$$p_i \geq 0, \quad q_i \geq 0, \quad \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1. \tag{1}$$

[1] Hopf et al. 2025.

Distance-based FoMs

- The Hellinger distance is a statistical measure of dissimilarity between two probability distributions[1].
- Let $P = \{p_i\}_{i=1}^n$ and $Q = \{q_i\}_{i=1}^n$ be two discrete probability distributions,

$$p_i \geq 0, \quad q_i \geq 0, \quad \sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1. \tag{1}$$

- The squared Hellinger distance between P and Q is defined as

$$H^2(P, Q) = \frac{1}{2} \sum_{i=1}^{n} \left(\sqrt{p_i} - \sqrt{q_i}\right)^2. \tag{2}$$

[1] Hopf et al. 2025.

Distance-based FoMs

- Similar to Hellinger distance, one can employ trace distance

Distance-based FoMs

- Similar to Hellinger distance, one can employ trace distance
- For two classical probability distributions P and Q, the trace distance is defined as

$$D_{\mathrm{tr}}(P, Q) = \frac{1}{2} \sum_{x} |P(x) - Q(x)| \,. \tag{3}$$

Distance-based FoMs

- Similar to Hellinger distance, one can employ trace distance
- For two classical probability distributions P and Q, the trace distance is defined as

$$D_{\mathrm{tr}}(P, Q) = \frac{1}{2} \sum_{x} |P(x) - Q(x)| \,. \tag{3}$$

- These distances can be used to rank compiled circuits according to their closeness to the ideal distribution.

Problems with Distance-based FoMs

- Computationally inefficient (exponential costs)

Problems with Distance-based FoMs

- Computationally inefficient (exponential costs)
- Not scalable

Problems with Distance-based FoMs

- Computationally inefficient (exponential costs)
- Not scalable
- In many cases, it has poor correlation with fidelity/noise strength

Probability of Successful Trials

- The probability of successful trials (PST) is a circuit-level figure of merit that quantifies how faithfully a quantum circuit is implemented on a given quantum device[2].

[2]Wang et al. 2022.
[3]Proctor et al. 2022.

Probability of Successful Trials

- The probability of successful trials (PST) is a circuit-level figure of merit that quantifies how faithfully a quantum circuit is implemented on a given quantum device[2].
- based on the idea of *circuit inversion*, or *mirror circuits*[3]

[2] Wang et al. 2022.
[3] Proctor et al. 2022.

Probability of Successful Trials

- Let U be the unitary operator implemented by an ideal quantum circuit acting on n qubits.

Probability of Successful Trials

- Let U be the unitary operator implemented by an ideal quantum circuit acting on n qubits.
- Consider the initial state

$$|\psi_0\rangle = |0\rangle^{\otimes n}. \tag{4}$$

Probability of Successful Trials

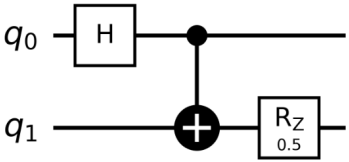
- Let U be the unitary operator implemented by an ideal quantum circuit acting on n qubits.
- Consider the initial state

$$|\psi_0\rangle = |0\rangle^{\otimes n}. \tag{4}$$

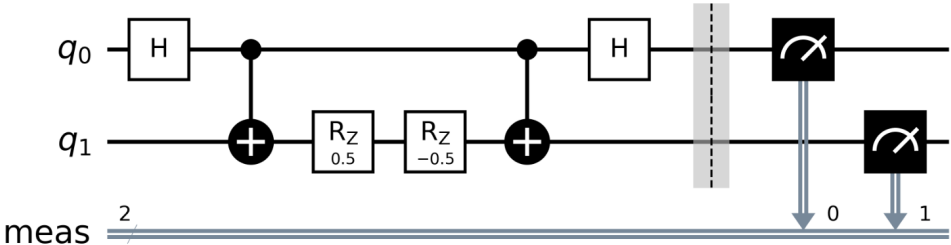
- In the absence of noise, applying U followed by its inverse $U^\dagger$ yields

$$U^\dagger U |\psi_0\rangle = |\psi_0\rangle. \tag{5}$$

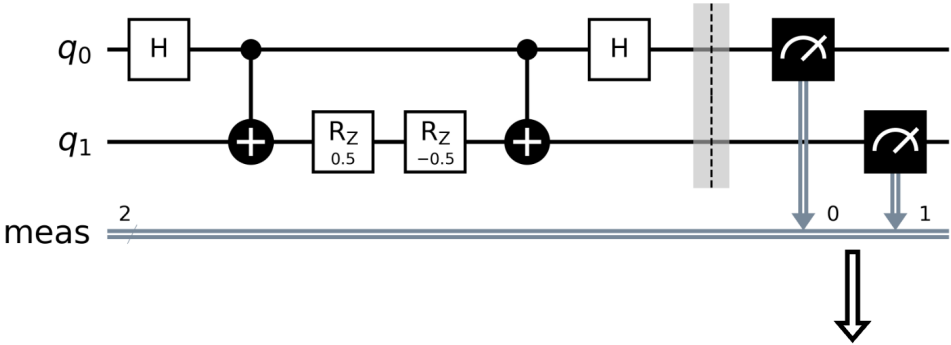
Probability of Successful Trials



Probability of Successful Trials

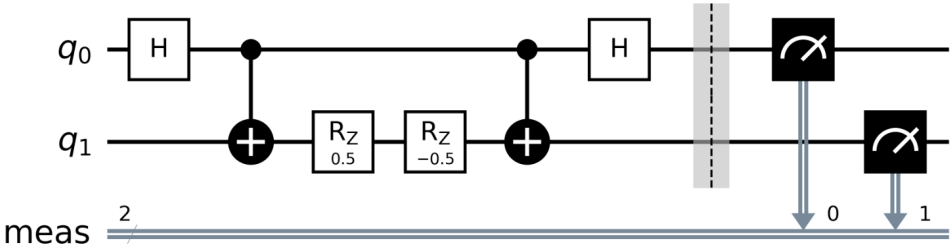


Probability of Successful Trials



In noisy environments, possible outcomes = {00, 01, 10, 11}

Probability of Successful Trials



PST: Probability of finding {00}

In noisy environments, possible outcomes = {00, 01, 10, 11}

Problems with PST

- highly susceptible to noise.

Problems with PST

- highly susceptible to noise.
- any minor error (especially in the larger qubit systems) leads to PST being zero.

Problems with PST

- highly susceptible to noise.
- any minor error (especially in the larger qubit systems) leads to PST being zero.
- cannot distinguish between the ‘extent’ of the effect of noise.

Weighted Probability of Successful Trials

- Consider a result for a quantum measurement, where the states $\{0000..0, 00..01, 00..10, ...\}$ were found with probabilities $\{p_1, p_2, p_3, ...\}$.

Weighted Probability of Successful Trials

- Consider a result for a quantum measurement, where the states $\{0000..0, 00..01, 00..10, ...\}$ were found with probabilities $\{p_1, p_2, p_3, ...\}$.
- The weights, in context to the wPST, are defined as the amount of non-flipped bits as compared to the total bits,

$$w = \frac{\#0s \quad \text{in the bit string}}{\text{Total number of bits in the string}} \tag{6}$$

wPST

- Using these weights, one can essentially quantify the amount of 'correct' result in the quantum measurement. The wPST can then be defined as

$$\mathrm{wPST} = \begin{cases} \sum_i p_i w_i, & w_i \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

wPST

- Using these weights, one can essentially quantify the amount of 'correct' result in the quantum measurement. The wPST can then be defined as

$$\mathrm{wPST} = \begin{cases} \sum_i p_i w_i, & w_i \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

- To ensure that we reward only the overall correct results, a threshold weight (w_{th}) is also introduced for wPST ($= 0.5$).

wPST

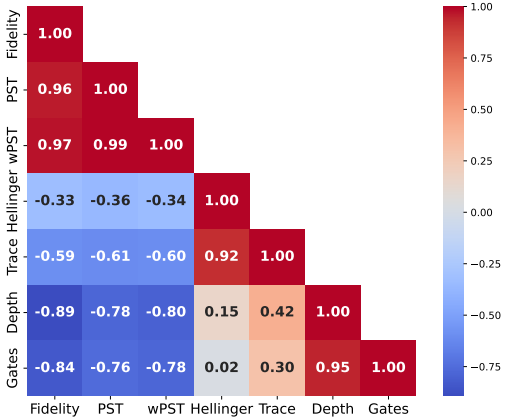


Figure: Correlation of different FoMs with fidelity for 500 random circuits in a noisy simulator.

So why wPST?

- Highly correlated with fidelity ✓

So why wPST?

- Highly correlated with fidelity ✓
- Self-sufficient ✓

So why wPST?

- Highly correlated with fidelity ✓
- Self-sufficient ✓
- Scalable ✓

So why wPST?

- Highly correlated with fidelity ✓
- Self-sufficient ✓
- Scalable ✓
- Hardware-aware ✓

So why wPST?

- Highly correlated with fidelity ✓
- Self-sufficient ✓
- Scalable ✓
- Hardware-aware ✓
- Distinguishable ?

So why wPST?

- Highly correlated with fidelity ✓
- Self-sufficient ✓
- Scalable ✓
- Hardware-aware ✓
- Distinguishable ?
- Computationally efficient ?

wPST–Distinguishable?

- Consider a circuit, which has a particular ideal result, say 0000.

wPST–Distinguishable?

- Consider a circuit, which has a particular ideal result, say 0000.
- Now, let's add bit errors into the final result successively and evaluate different FoMs.

wPST–Distinguishable?

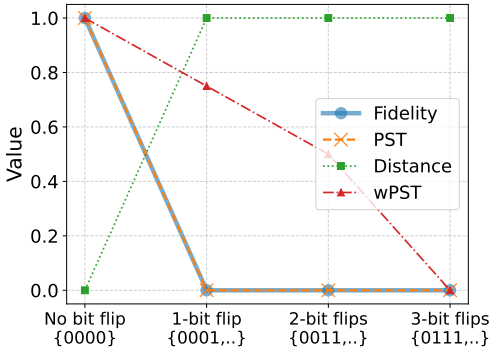


Figure: Values of fidelity, PST, distance-based metrics, and wPST with increasing number of bit flips (error).

Calculating wPST: Computationally efficient?

- Accurately evaluating wPST requires executing the quantum circuit a sufficiently large number of times.

Calculating wPST: Computationally efficient?

- Accurately evaluating wPST requires executing the quantum circuit a sufficiently large number of times.
- This process can become extremely costly in the context of quantum compilation workflows.

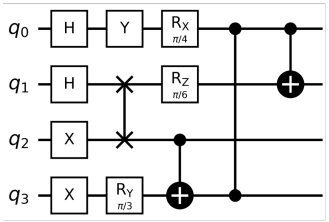
Calculating wPST: Computationally efficient?

- Accurately evaluating wPST requires executing the quantum circuit a sufficiently large number of times.
- This process can become extremely costly in the context of quantum compilation workflows.
- Consequently, direct computation of PST for each candidate circuit is not scalable.

Calculating wPST: Computationally efficient?

- Accurately evaluating wPST requires executing the quantum circuit a sufficiently large number of times.
- This process can become extremely costly in the context of quantum compilation workflows.
- Consequently, direct computation of PST for each candidate circuit is not scalable.
- This is why we turn to machine learning.

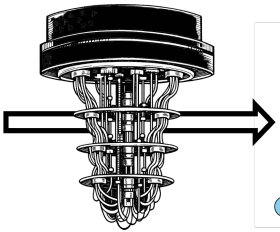
Calculating wPST: The ML Way



Quantum Circuit

#qubits,
#gates,
depth,
programme communication,
entanglement ratio,
.
.

Feature Vector



Hardware Data

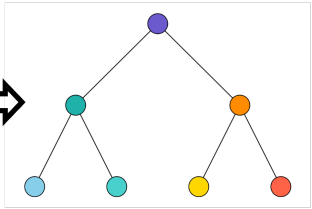


Figure of Merit Prediction

Calculating wPST: The ML Way

- The feature vector contains both the circuit data and hardware data via with the coherence times, T_1, and T_2.

[4] Chen and Guestrin 2016.
[5] Quetschlich, Burgholzer, and Wille 2023.

Calculating wPST: The ML Way

- The feature vector contains both the circuit data and hardware data via with the coherence times, T_1, and T_2.
- XGBoost[4] is used as the ML algorithm.

[4] Chen and Guestrin 2016.
[5] Quetschlich, Burgholzer, and Wille 2023.

Calculating wPST: The ML Way

- The feature vector contains both the circuit data and hardware data via with the coherence times, T_1, and T_2.
- XGBoost[4] is used as the ML algorithm.
- Training data includes random quantum circuits.

[4] Chen and Guestrin 2016.
[5] Quetschlich, Burgholzer, and Wille 2023.

Calculating wPST: The ML Way

- The feature vector contains both the circuit data and hardware data via with the coherence times, T_1, and T_2.
- XGBoost[4] is used as the ML algorithm.
- Training data includes random quantum circuits.
- System Size: 21 qubits (IBM Strasbourg), 21 qubits (IBM Torino), 20 qubits (IBM Miami), and 51 qubits (IBM Brussels).

[4] Chen and Guestrin 2016.
[5] Quetschlich, Burgholzer, and Wille 2023.

Calculating wPST: The ML Way

- The feature vector contains both the circuit data and hardware data via with the coherence times, T_1, and T_2.
- XGBoost[4] is used as the ML algorithm.
- Training data includes random quantum circuits.
- System Size: 21 qubits (IBM Strasbourg), 21 qubits (IBM Torino), 20 qubits (IBM Miami), and 51 qubits (IBM Brussels).
- Testing is done on random and MQT benchmark circuits[5](M).

[4] Chen and Guestrin 2016.
[5] Quetschlich, Burgholzer, and Wille 2023.

Results: The PST Model

FoM	IBM Strasbourg	IBM Torino	IBM Miami	IBM Brussels	Average
Circuit Depth	0.26	0.43	0.36	0.20	0.31
# Total Gates	0.37	0.47	0.54	0.20	0.39
# 2Q-Gates	0.44	0.47	0.48	0.20	0.40
Predicted PST	**0.95**	**0.97**	**0.94**	**0.95**	**0.95**

Pearson Correlation Coefficients for True PST across four IBM devices and their total average.

Results: The wPST Model

FoM	IBM Strasbourg	IBM Torino	IBM Miami	IBM Brussels	Average
# 2Q-Gates	0.46	0.53	0.61	0.29	0.47
# Total Gates	0.47	0.56	0.66	0.29	0.49
Circuit Depth	0.51	0.63	0.69	0.29	0.53
Predicted wPST	**0.88**	**0.94**	**0.87**	**0.85**	**0.89**

Pearson Correlation Coefficients for True wPST across four IBM devices and their total average.

Connecting the wPST to a Quantum Compiler

- The results were for circuits that were already transpiled for a particular quantum device.

Connecting the wPST to a Quantum Compiler

- The results were for circuits that were already transpiled for a particular quantum device.
- For a FoM to be useful in the quantum compilation process, it should be able to figure out a way to quantify the non-transpiled quantum circuits.

Connecting the wPST to a Quantum Compiler

- The results were for circuits that were already transpiled for a particular quantum device.
- For a FoM to be useful in the quantum compilation process, it should be able to figure out a way to quantify the non-transpiled quantum circuits.
- So we devised a two-step process to connect the wPST to a quantum compiler.

The Two-Step Process

- The first step involves predicting the feature vector of the transpiled quantum circuit to be executed on a given quantum hardware.

The Two-Step Process

- The first step involves predicting the feature vector of the transpiled quantum circuit to be executed on a given quantum hardware.
- This feature vector is then used to get a final prediction on the figure of merit.

The Two-Step Process

- The first step involves predicting the feature vector of the transpiled quantum circuit to be executed on a given quantum hardware.
- This feature vector is then used to get a final prediction on the figure of merit.
- This allows us to approximately introduce the impact of both the coupling maps and the quantum hardware into the model.

The Two-Step Process

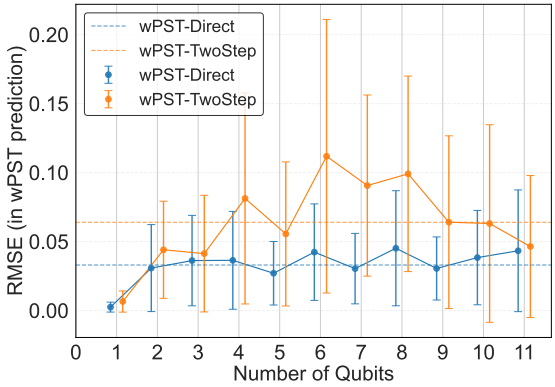


Figure: RMSE for predicted wPST using direct and two-step schemes in IBM Torino device.

In Summary

- wPST can be viable FoM for quantum circuit compilation.

In Summary

- wPST can be viable FoM for quantum circuit compilation.
- We formulated an ML method to predict the wPST.

In Summary

- wPST can be viable FoM for quantum circuit compilation.
- We formulated an ML method to predict the wPST.
- The predictions can get us really close to the true wPST, much better than any other FoM.

In Summary

- wPST can be viable FoM for quantum circuit compilation.
- We formulated an ML method to predict the wPST.
- The predictions can get us really close to the true wPST, much better than any other FoM.
- We have a two-step process that can extend the method to non-transpiled circuits.

Thank You!