

Improving Figures of Merit for Quantum Circuit Compilation

Patrick Hopf*, Nils Quetschlich*, Laura Schulz†, and Robert Wille*

*Chair for Design Automation, Technical University of Munich, Munich, Germany

†QCT Department, Leibniz Supercomputing Centre, Garching, Germany

patrick.hopf@tum.de, nils.quetschlich@tum.de, schulz@lrz.de, robert.wille@tum.de

www.cda.cit.tum.de/research/quantum

Abstract—Quantum computing is an emerging technology that has seen significant software and hardware improvements in recent years. Executing a quantum program requires the compilation of its quantum circuit for a target *Quantum Processing Unit* (QPU). Various methods for qubit mapping, gate synthesis, and optimization of quantum circuits have been proposed and implemented in *compilers*. These compilers try to generate a quantum circuit that leads to the best *execution quality*—a criterium which is usually approximated by *figures of merit* such as the number of (two-qubit) gates, the circuit depth, expected fidelity, or estimated success probability. However, it is often unclear how well these figures of merit represent the *actual execution quality* on a QPU.

In this work, we investigate the correlation between established figures of merit and actual execution quality on real machines—revealing that the correlation is weaker than anticipated and that more complex figures of merit are not necessarily more accurate. Motivated by this finding, we propose an improved figure of merit (based on a machine learning approach) that can be used to predict the expected execution quality of a quantum circuit for a chosen QPU without actually executing it. The employed machine learning model reveals the influence of various circuit features on generating high correlation scores. The proposed figure of merit demonstrates a strong correlation and outperforms all previous ones in a case study—achieving an average correlation improvement of 49%.

Index Terms—quantum computing, quantum circuit compilation, figures of merit, machine learning

I. INTRODUCTION

Quantum computing has made remarkable progress in recent years, with improvements in both the software and hardware used to run programs on quantum computers. A quantum program is typically represented as a quantum circuit, composed of a sequence of operations. For a quantum program to run on a specific *Quantum Processing Unit* (QPU), it needs to be translated into a form that the hardware can execute. This process is known as quantum circuit *compilation*.

The quality of a compiled quantum circuit is usually measured by so-called *figures of merit*. Established figures of merit include the number of gates in the circuit, its depth, or the expected fidelity and *Estimated Success Probability* (ESP, [1]). These figures of merit are intended to describe how well the circuit will perform on a target QPU. However, while these figures of merit provide an approximation of the circuit’s *execution quality*, they might not always give an accurate picture of how well the circuit will actually run on quantum hardware. QPUs are complex systems that face many challenges during execution, such as interference between signals applied to neighboring qubits, errors during gate or

measurement operations, and other hardware imperfections that can impact their performance. These effects are often difficult to capture with the simple figures of merit used today.

In this paper, we take a closer look at the established figures of merit and investigate how well they truly reflect the quality of a circuit’s execution on real QPUs. We find that the correlation between these metrics and real execution performance is often weaker than expected. In some cases, it even turns out that a more complex metric (like ESP) does not lead to a better approximation.

Furthermore, to address these weaknesses of established figures of merit, we propose a new way of evaluating circuit quality using machine learning techniques. This results in an improved figure of merit that takes into account a variety of quantum circuit characteristics without requiring QPU calibration data. The approach achieves an average correlation improvement of 49%, accurately predicting how well a circuit can be executed on a targeted QPU. By offering a simple yet more effective method to assess circuit quality, this work helps researchers and engineers to develop or adjust compilers, so that the generated circuits are better suited for a given target QPU.

This paper is structured in the following way: Section II offers a concise review of quantum circuit compilation including its primary tasks and the established figures of merit for assessing circuit quality. Section III discusses the limitations of current metrics (providing the motivation of this work) and presents the proposed approach for an improved evaluation of circuit execution quality. Section IV provides detailed insights into the implementation of the proposed method. The results of a study are presented in Section V, along with a comparison of the new approach to established figures of merit and a thorough discussion. Finally, Section VI summarizes the findings.

II. QUANTUM CIRCUIT COMPILATION

A quantum program is usually designed as a quantum circuit in order to execute it on quantum hardware. Such a circuit typically consists of multiple operations called quantum gates and measurements. During the execution of a program, these operations modify the state of quantum bits, so-called *qubits*—the fundamental computational elements of a QPU. There are various quantum computing hardware technologies that realize these operations and qubits in different ways (e.g., superconducting, trapped ions, neutral atoms, etc). Since quantum circuits are typically defined on a hardware-agnostic level, these need to be translated into machine-executable operations.

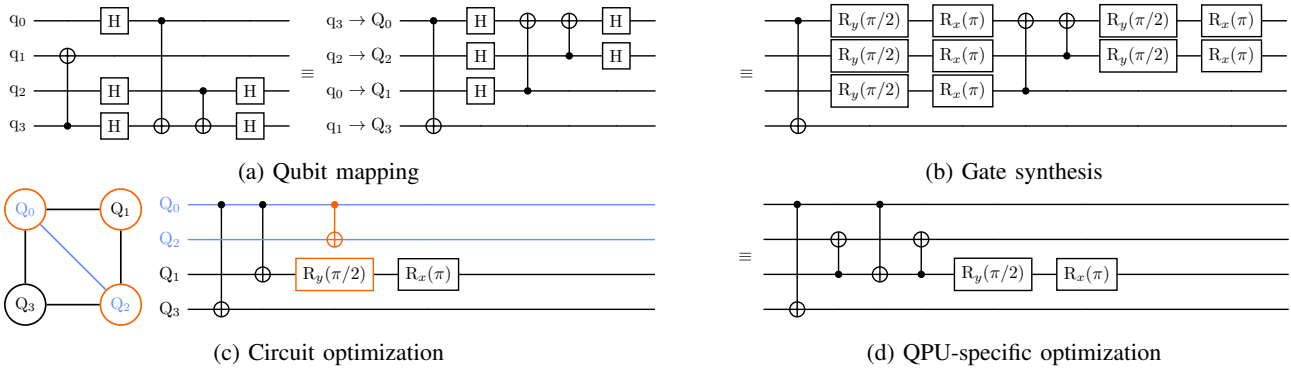


Fig. 1: Compilation of a quantum circuit demonstrating (a) mapping, (b) synthesis, and (c), (d) optimization passes for a four-qubit square layout (only missing a link between Q_1 and Q_3). The exemplary QPU is subject to crosstalk errors from parallel gate execution (orange) on neighboring qubits and provides only a low CNOT fidelity (blue) between distant qubits Q_0 and Q_2 .

Any QPU supports a specific set of executable operations. Consequently, quantum circuit compilation is necessary to convert any given quantum circuit into an equivalent one that utilizes only these supported operations. This section provides an overview of the main tasks involved in quantum circuit compilation, along with a review of how the quality of a compiled circuit is currently assessed using so-called figures of merit.

A. Compilation Tasks

Depending on the type of the QPU and its specific constraints, the compilation procedure usually involves a combination of the following tasks:

Qubit mapping: Quantum circuits typically contain multi-qubit gates acting on multiple arbitrary qubits. However, some hardware (like superconducting QPUs) only provides a limited set of qubits on which multi-qubit operations are possible. Similarly, for technologies (like trapped ions and neutral atoms) that do not have this limitation, it is often sensible to perform them on specific qubits only (e.g., to reduce shuttling operations). Executing the algorithm, therefore, usually requires a mapping between the logical (program) qubits and the physical (QPU) qubits.

Example 1. Fig. 1a shows how the qubits of an example circuit are mapped to an exemplary superconducting QPU architecture with a square qubit layout where almost all physical qubits Q_0 , Q_1 , Q_2 , and Q_3 are connected; only missing a direct link between Q_1 and Q_3 .

Gate synthesis: Quantum algorithms and their corresponding circuits are commonly designed using a wide set of gate and measurement operations. Since any QPU only supports a small number of natively executable operations, each non-native operation must be synthesized into one or more of these supported operations. This task is non-trivial, and doing it optimally is NP-complete [2].

Example 2. Fig. 1b illustrates how the Hadamard gates in the circuit of the previous example can be translated into the native R_x and R_y rotation gates provided by the exemplary superconducting QPU architecture.

Circuit optimization: It is possible to alter a circuit's gate composition without changing its original function. Following specific transformation rules, a general quantum circuit can be expressed through numerous combinations of distinct gates. Hence, a circuit can often be optimized with respect to a desired metric.

Example 3. Fig. 1c demonstrates how the four single-qubit rotation gates of the previously unoptimized circuit can be eliminated (according to circuit transformation rules [3]).

Various compilation methods have been proposed for qubit mapping [4]–[16], gate synthesis [2], [17]–[22], and circuit optimization [22]–[30]. These tasks are usually implemented by individual compilation passes that manipulate the circuit. Passes can be performed in any order and might be repeated multiple times. Hence, there are various pass sequences that lead to distinct compiled versions representing the same original circuit. Finding a suitable order of compilation passes that yield an efficient circuit is a non-trivial task. This raises the question: How can the quality of a chosen sequence of passes and its associated quantum circuit be assessed?

B. Figures of Merit

Besides imposing hardware-specific constraints, current QPUs additionally pose the risk of erroneous calculations. Quantum hardware is usually subject to environmental noise and suffers from imperfect gate and qubit realizations. Due to this, any quantum operation (even the identity that should leave a qubit unmodified) can only be performed with some probability of error. To obtain a circuit that produces high-quality execution results, its design must minimize the accumulation of errors. Figures of merit can act as a proxy for the result quality, enabling the assessment of a compilation run without the need to execute the resulting circuit. Thus far, the following figures of merit have been employed in state-of-the-art mapping, synthesis, and optimization methods such as those mentioned above.

- **Number of gates**, i.e., the integer gate count in the circuit. Often, only two-qubit gates are considered because of their

dominant error rates. A lower number indicates better performance.

- **Circuit depth**, i.e., the integer number of gates on the longest path in the circuit graph. A lower circuit depth usually indicates lower execution time and fewer gates, hence a higher circuit quality.
- **Expected fidelity**, i.e., the product of all decimal gate and measurement fidelities in the circuit. Since the fidelity is inversely proportional to the error, higher values suggest better execution quality.
- **Estimated Success Probability (ESP)**, i.e., the expected fidelity multiplied by the exponential decay factor $\exp[-t_{\text{idle}}^q / \min(T_1^q, T_2^q)]$, for all qubits, where t_{idle}^q is the total idle time of qubit q . This figure additionally requires each qubit's T_1 and T_2 relaxation times (which measure how long it can retain information) and is based on variants of ESP [1], [31], [32]. High values indicate good performance.

The first two figures of merit are hardware-agnostic metrics and, hence, independent of the executing QPU. The latter two require experimental data about the specific hardware—usually obtained during device calibration, a process that involves fine-tuning qubits, gates, and measurement fidelities. The intuition behind using these as proxies for a circuit's execution quality is that these numbers are expected to scale (directly or indirectly) proportional to the anticipated errors. Under this assumption, most quantum compilation flows optimize for one (sometimes multiple) of these figures of merit. However, it often remains unclear whether the resulting compiled circuit actually leads to the least error-affected execution and, therefore, the best solution.

III. MOTIVATION AND PROPOSED APPROACH

The compilation concepts reviewed before provide an easy-to-use and general approach to converting any algorithm encoded as a quantum circuit into an executable set of operations. However, the simplicity and generality can come at the cost of missing out on better circuit designs, as demonstrated in the following example.

Example 4. *The previously considered circuit, depicted in Fig. 1c, is compiled according to the established figures of merit reviewed before, i.e., is minimized with respect to the overall number of gates and, accordingly, the circuit depth. Depending on the actual gate fidelity and relaxation values, this solution will also maximize the expected fidelity and ESP.*

However, considering the QPU in Fig. 1(c), this circuit is unnecessarily prone to crosstalk on neighboring qubits Q_0 , Q_1 , and Q_2 —an error that occurs when gates are executed in parallel (highlighted in orange). This effect, along with a low CNOT fidelity between distant qubits Q_0 and Q_2 (highlighted in blue), can be avoided by using the functionally equivalent circuit shown in Fig. 1(d), where two CNOTs are rearranged and an additional one is added. This circuit has a higher number of gates and depth (and would, therefore, be rejected when applying established figures of merit) but still performs better when executed on the considered QPU. While expected fidelity and ESP can account for the low CNOT fidelity between distant qubits, they remain indifferent to the crosstalk effects.

The example illustrates how relying on the established figures of merit, which may not fully capture hardware-specific characteristics, can guide the compilation procedure to subpar solutions. Similar concerns have been raised before [33], [34], and were confirmed in a study demonstrating that calibration-based compilation strategies can achieve higher circuit fidelities compared to those that solely focus on minimizing the number of two-qubit gates [35]. Likewise, an individual assessment of ESP demonstrated its poor correlation with actual device performance [36]. This already indicates the need for a comprehensive investigation (and, eventually, improved figures of merit), but to the best of our knowledge, no comprehensive study that directly compares the correlation scores of various established figures of merit and circuit execution quality has been conducted yet.

At the same time, the investigation and development of alternative figures of merit are still in the early stages. New figures of merit have been introduced employing basic machine learning techniques [37], [38], where the underlying circuit representations scale with the depth of the input circuit—making these methods impractical for deep quantum circuits. This issue is also present in another approach utilizing the circuit graph representation, which was used by a transformer-based model to accurately predict the probability of successful trials [39]. Although more sophisticated, this work only considered circuits of up to seven qubits and (like ESP) requires accurate T_1 , T_2 , gate and measurement fidelity data, which is often outdated or not available.

In summary, there is a lack of comprehensive analysis of the established figures of merit, while emerging alternatives struggle with scalability and practical limitations. In this work, we address these gaps with the following contributions:

- 1) We conduct a comprehensive investigation to quantify the (weak) correlation between the established figures of merit (i.e., number of gates, circuit depth, expected fidelity, and ESP) and a circuit's actual execution quality. The study is designed with a focus on real-world applicability by executing circuits from practical quantum computing applications on real QPUs.
- 2) Based on these findings, we propose an interpretable machine-learning-based figure of merit as an improved representation of a circuit's execution quality. This model works with a depth-independent circuit representation and provides an individualized figure of merit for any QPU without requiring detailed calibration data.

IV. IMPLEMENTATION

This section provides details on the implementation of the contributions outlined above. First, we introduce the measure required to evaluate the execution quality of a quantum circuit and demonstrate its correlation with the previously introduced figures of merit. Based on this measure, a machine learning approach to generate an improved figure of merit is proposed, which offers a better correlation and, thus, provides a more accurate approximation of the execution quality for a given quantum circuit.

A. Investigating the Correlation Between Figures of Merit and Execution Quality

The measurement result of a quantum circuit is usually described in terms of a discrete probability distribution over all possible qubit states, i.e., combinations of zeroes and ones that can be illustrated in a histogram (see green and blue charts in Fig. 2). In order to understand how well a figure of merit represents the presumed execution quality of a quantum circuit, we evaluate the result quality of its execution on an actual QPU and compare it to its *true distribution*. The true (noiseless) distribution can be obtained, e.g., from a state vector simulation, whereas the (noisy) experimental distribution can be obtained from repeated circuit executions on a QPU. To quantify the execution quality and, accordingly, the (mis)alignment of the two histograms, the *Hellinger distance*

$$d(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=0}^{2^N-1} (\sqrt{p_{|i\rangle}} - \sqrt{q_{|i\rangle}})^2} \in [0, 1] \quad (1)$$

between the true distribution $P = \{p_{|0\rangle}, \dots, p_{|2^N-1\rangle}\}$ and its QPU counterpart $Q = \{q_{|0\rangle}, \dots, q_{|2^N-1\rangle}\}$ is used. If the measurement histograms of the true and experimental QPU distribution overlap completely, their distance is zero. Conversely, for highly distinct histograms, the distance approaches one.

In addition to assessing the Hellinger distance d , we investigate its correlation with any previously introduced figure of merit y on a set of M quantum circuits. For this task, the *Pearson correlation coefficient*

$$r = \frac{\sum_{j=1}^M (d_j - m_d)(y_j - m_y)}{\sqrt{\sum_{j=1}^M (d_j - m_d)^2 \sum_{j=1}^M (y_j - m_y)^2}} \in [-1, 1] \quad (2)$$

is calculated, where m_d and m_y are the mean (Hellinger distance d and figure of merit y) values over all circuits in the set. A perfect linear correlation is represented by $|r| = 1$, whereas $r = 0$ indicates no Pearson correlation at all¹.

This correlation (based on the Hellinger distance) can now be used to quantify how well any figure of merit actually approximates the execution quality. Furthermore, the Hellinger distance is additionally used to derive an improved (machine-learning-based) figure of merit that aims to capture it more accurately and, thus, can be used as a more precise figure of merit.

B. Proposed Figure of Merit

With insights from the Hellinger distance, it is possible to quantify the (mis)alignment between the results obtained from executing a circuit on a real QPU and the true distribution. While established figures of merit use indirect metrics to approximate this measure in order to guide the circuit compilation, it would be far more efficient to directly optimize for a reduction of the Hellinger distance. However, evaluating the distance for every possible circuit configuration during compilation would require an impractical amount of simulation and execution data.

¹There might be non-linear correlation measures that better capture the relationship between individual figures of merit and the Hellinger distance. However, any such measure must include a linear (or anti-linear) component that the Pearson correlation coefficient can capture.

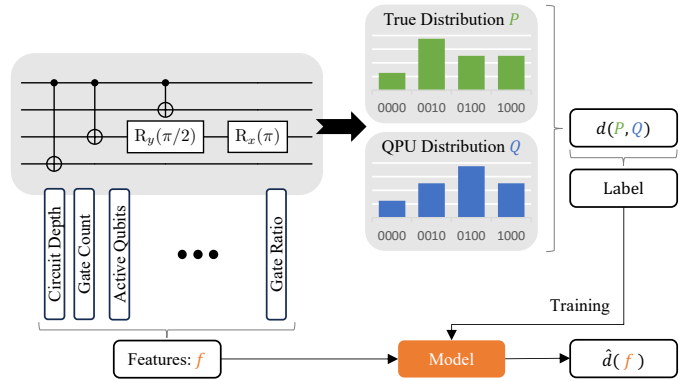


Fig. 2: Workflow for feature and label generation from a compiled quantum circuit. The Hellinger distance—representing the difference between the circuit’s true distribution and the QPU execution results—is used as label data for model training.

Hence, we instead propose to train a machine learning model on a representative set of practical algorithms labeled with experimentally obtained Hellinger distance values. The resulting model then acts as an estimator to predict the distance for any circuit during compilation, effectively serving as a figure of merit.

To this end, we employ the workflow depicted in Fig. 2. In order to train an estimator model (pictured in orange) for a specific QPU, a comprehensive set of feature and label data is required. Such an estimator receives as input a vectorized representation, called *feature vector* (shown in the bottom left), of all the quantum circuits. To this end, we utilize a revised version of the circuit encoding introduced in [40], whose size is independent of the circuit depth and, therefore, constant for any specific QPU. Among the basic features are the hardware-agnostic figures of merit, i.e., the circuit depth and its gate counts. More sophisticated features include circuit liveness, which captures how actively qubits are utilized; directed program communication, which quantifies the ratio between the actual and maximum possible average node degree of the circuit’s directed interaction graph; as well as parallelism (all based on [41]) and gate ratios, which reflect the circuit’s operational density. Notably, the feature vector does not require calibration data, as is required for calculating fidelity-related figures of merit.

In addition to extracting the feature representation, every single circuit must be associated with its Hellinger distance, which serves as the training label (shown in the top right). This requires evaluating the noiseless true result distribution and the noisy QPU distribution, as shown in the green and blue sample Histogram.

Given a representative set of such circuit features and label data, a model can be trained and then used to estimate the Hellinger distance for a given circuit, thereby allowing the compilation to aim directly at the reduction of Hellinger distance.

V. EXPERIMENTAL EVALUATION

The ideas and implementation details described above eventually lead to a framework that allows for (1) a comprehensive

investigation of the correlation between circuit execution quality and the established figures of merit and (2) an evaluation of the proposed (improved) figure of merit. In this section, we summarize the main results obtained by these investigations and evaluations. To this end, we first review the used setup. Afterwards, the obtained results are provided and discussed.

A. Setup

The following describes the experimental setup used in our investigations and evaluations. All of the presented methods and results were implemented in Python and are accessible through the MQT Predictor [40] as part of the *Munich Quantum Toolkit* [42]. The source code is publicly available on GitHub².

1) *Used Benchmarks*: For the comprehensive investigation of quantum circuit quality, we utilize all circuits provided by the MQT Bench collection, an open-source library frequently used to evaluate compilers, QPUs, and more [43]. This collection offers a variety of algorithms (like VQE, QAOA, QFT, etc.), which have been mapped, synthesized, and optimized for any number between 2 and 20 qubits using the Qiskit [44] transpiler module at optimization level three. Since circuits with a depth of more than 1000 are too much affected by noise when executed on current quantum computers (and, eventually, would not produce any meaningful results), we only considered circuits with a compiled depth smaller than 1000—leaving a total of 222 circuits.

2) *Used QPUs*: The resulting set of benchmark circuits has been executed on two superconducting IQM QPUs hosted at the German Leibniz Supercomputing Centre. Both devices are members of the 20-qubit series (labeled Q20-A and Q20-B) [45]. Their native gate set consists of a parameterized single-qubit rotation gate and the CZ gate on IQM’s crystal architecture (qubits located on a square grid). In addition to running them on both QPUs (and generating the corresponding true distributions), the full set of benchmark circuits has been simulated using the Qiskit Aear noiseless state vector simulator on a MacBook Pro (M2 chip), completing within a few hours.

3) *Machine Learning Model*: All circuits have been expressed through the numeric feature vector of size 30 and have been labeled with their associated Hellinger distance values. Then, a random forest regressor (consisting of multiple decision trees, implemented with scikit-learn [46]) was trained for each QPU on the same classical hardware in a few seconds. This was done using cross-validation over three training sets and an overall 80/20 train-test ratio. The Pearson correlation coefficient served as the model performance score during validation. A hyperparameter grid search to optimize, e.g., the number of decision trees, their maximum depth, and the minimum samples per leaf and split, could be completed in under a minute on the same classical hardware. Eventually, like any other figure of merit, the trained model was used to determine the quality of a compiled circuit and was evaluated on the (previously unseen) test set.

B. Investigation of Established Figures of Merit

After executing the entire benchmark set on both QPUs and generating the true distributions, the correlation between the established figures of merit and the actual circuit execution

TABLE I: Pearson correlation with Hellinger distance

Figure of merit / QPU	Q20-A	Q20-B	Combined
Number of gates	0.46	0.61	0.53
Circuit depth	0.46	0.62	0.54
Expected fidelity	0.66	0.80	0.73
ESP	0.59	0.70	0.64
Proposed approach	0.88	0.94	0.91

quality has been evaluated. The results are summarized in Table I, showing the Pearson coefficients for each investigated figure of merit. Values in the columns *Q20-A* and *Q20-B* correspond to the executing QPUs, whereas the values in the column *Combined* provide the correlation for all circuit executions on both QPUs. To enhance clarity, the table only shows the absolute correlation scores. Values closer to 1 indicate higher quality figures of merit.

The results provide some interesting insights (both expected as well as unexpected): First, they show that the number of gates and circuit depth have very similar correlation scores, which, considering their clear link between each other, is not really surprising. Furthermore, the expected fidelity and ESP obviously provide significantly higher correlation values than the other figures on both QPUs. Also, this is not surprising: The number of gates and the circuit depth are rather simple (albeit easy to use) figures of merit, while expected fidelity and ESP take hardware information into account. Hence, a better quality is expected from these figures of merits.

What surprises, though, is that, in some cases, a more complex metric does not necessarily lead to a better correlation. In fact, even though expected fidelity and ESP share the same fidelity term in their calculation, the former achieves a higher, i.e., better correlation score (0.66 vs. 0.59 on Q20-A and 0.80 vs. 0.78 on Q20-B). Since the only difference lies within the calibration-data-dependent relaxation term, this result points to possibly outdated T_1 , T_2 times.

Independently from those differences, the results confirm that *all* established figures of merits *do not* provide a fully effective correlation between estimated and real execution performance. Even though the hardware-specific figures of merit (i.e., expected fidelity and ESP) represent the actual circuit execution quality better than the target-agnostic figures of merit (i.e., number of gates and circuit depth), the combined correlation remains at 0.73 in the best case. This confirms the discussions and the motivation from Section III, highlighting that the established figures of merit indeed leave room for improvement.

C. Evaluation of Proposed Figure of Merit

The above investigation confirmed the weaknesses of the established figures of merit. Next, we evaluated whether the proposed figure of merit provides an improvement. To this end, the correlation of the trained machine learning model is assessed using the unseen circuit test set. Its Pearson correlation is presented in the last row of Table I, again for both individual QPUs and for the total set of all executed test circuits.

The results clearly confirm the improved correlation of the proposed figure of merit to the actual execution quality. In fact, on average, the correlation score increases by 62% and 38% for the Q20-A and Q20-B, respectively. Considering the

² <https://github.com/cda-tum/mqt-predictor>

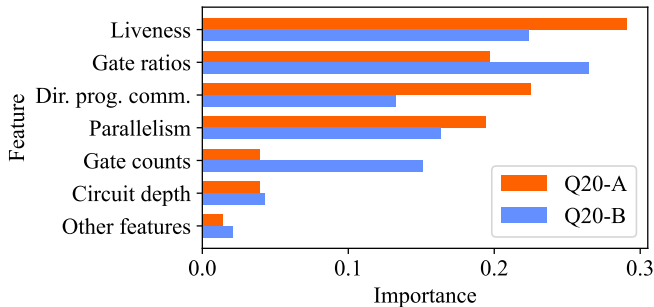


Fig. 3: Random forest model feature importance.

average correlation of all previous figures of merit over both QPUs (last column), the proposed figure of merit outperforms their correlation scores by 49%.

In order to understand how the proposed figure of merit managed to capture the execution quality so well, we investigate the model’s feature importance depicted in Fig. 3. It can be observed that the model’s prediction quality strongly depends on features designed to capture qubit activity, operational density, and qubit interactions—specifically, liveness, gate ratios, parallelism, and directed program communication. In contrast, the basic gate counts and circuit depth features show moderately low importance, aligning with the correlation values observed in the previous figure of merit analysis.

Overall, the proposed figure of merit offers a significant improvement over the established figures of merit. By leveraging the right combination of circuit features, it manages to capture the actual circuit execution quality much better.

D. Discussion

The findings presented above provide valuable insights into the characteristics of an effective figure of merit, enhancing our understanding of both hardware-agnostic and hardware-specific (calibration-data-based) approaches. The discrepancy between expected fidelity and ESP indicates that overly specific metrics can reduce the correlation when using poor calibration data. This result is consistent with work on error-aware compilation methods, which has also found decreased compilation performance on outdated calibration data [35].

Furthermore, the results showed that neither the number of gates nor the circuit depth alone serves as a reliable estimator of circuit execution quality. This finding is supported by their relatively low contribution to the random forest model accuracy. Accounting for individual qubit performance through idle times in the exponential decay factor (see ESP) did not improve correlation scores. However, incorporating their impact via liveness and parallelism features did. This underscores the importance of selecting and combining circuit features effectively, aligning with similar findings in related work [47], [48], and demonstrates that such a feature set can yield a far better figure of merit than any single measure alone.

Finally, unlike the expected fidelity and ESP, the proposed model does not directly rely on device-specific measurement data, which is highly valuable when this information is not (frequently) provided by a QPU provider. Importantly, the model was trained on real QPU data rather than simulations, which means it indirectly incorporates device-specific calibration information.

Overall, these findings show that it is crucial to find the right balance between incorporating an accurate hardware representation and abstracting device details. The high correlation scores obtained through the proposed figure of merit indicate that the model managed to achieve this performance requirement. Future work will focus on examining the model’s performance over time, comparing it to other QPU-specific figures of merit in the context of evolving QPU noise characteristics.

Lastly, in our experiments, we trained the model on circuits that can still be classically simulated. With improving hardware, this will become more challenging. However, there is evidence to suggest that the *Probability of Successful Trials* (PST) derived by appending a circuit’s inverse (hence, removing the need for simulation) can successfully represent its execution quality [39]. Future work will investigate to what extent the PST can be used to improve our proposed approach.

VI. CONCLUSION

This work investigated and improved upon the limitations of current figures of merit in representing the actual execution quality of quantum circuits. By analyzing the correlation between established figures of merit—such as the number of gates, circuit depth, expected fidelity, as well as ESP—and real-world execution results, we unveiled that these figures of merit often fall short of accurately representing quantum circuit performance on a QPU. This gap highlights the need for improved metrics that better align with execution quality.

Motivated by that, we introduced a machine-learning-based figure of merit designed to better correlate with actual circuit execution quality. The proposed model does work with a depth-independent circuit representation and provides a QPU-specific figure of merit. The method outperformed the traditional figures of merit, showing a 49% improvement in its correlation with execution quality.

Overall, the obtained findings underscore the significance of selecting and combining the right circuit characteristics to develop a figure of merit that closely aligns with actual circuit execution quality. This work demonstrates that the appropriate format and combination of circuit features can yield a far superior figure of merit than any individual measure alone. We have shown that, to this end, machine learning can significantly enhance quantum circuit compilation, providing a more effective approach to evaluating execution quality.

VII. ACKNOWLEDGEMENTS

P.H., N.Q., and R.W. acknowledge funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No. 101001318), the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, and has been supported by the BMWK on the basis of a decision by the German Bundestag through project QuaST, as well as by the BMK, BMDW, the State of Upper Austria in the frame of the COMET program, and the QuantumReady project within Quantum Austria (managed by the FFG). L.S. acknowledges funding by the German Federal Ministry for Education and Research under grants 13N15689 (DAQC), and 13N16063 (Q-Exa).

REFERENCES

- [1] P. Murali, D. C. McKay, M. Martonosi, and A. Javadi-Abhari, "Software mitigation of crosstalk on noisy intermediate-scale quantum computers," in *Proceedings of the Twenty-Fifth Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, 2020.
- [2] T. Peham, N. Brandl, R. Kueng, R. Wille, and L. Burgholzer, "Depth-optimal synthesis of Clifford circuits with SAT solvers," in *Proceedings of the 2023 IEEE Int'l Conf. on Quantum Computing and Engineering (QCE 2023)*, pp. 802–813, IEEE, 2023.
- [3] J. C. Garcia-Escartin and P. Chamorro-Posada, "Equivalent quantum circuits," 2011. arXiv 1110.2998.
- [4] W.-H. Lin, J. Kimko, B. Tan, N. Bjørner, and J. Cong, "Scalable optimal layout synthesis for NISQ quantum processors," in *Design Automation Conf.*, pp. 1–6, 2023.
- [5] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the IBM QX architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2019.
- [6] A. Matsuo and S. Yamashita, "An efficient method for quantum circuit placement problem on a 2-D grid," in *Int'l Conf. of Reversible Computation*, pp. 162–168, 2019.
- [7] B. Tan and J. Cong, "Optimal qubit mapping with simultaneous gate absorption," in *Int'l Conf. on CAD*, pp. 1–8, 2021.
- [8] R. Wille, L. Burgholzer, and A. Zulehner, "Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations," in *Design Automation Conf.*, 2019.
- [9] G. Li, Y. Ding, and Y. Xie, "Tackling the qubit mapping problem for NISQ-era quantum devices," in *Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, 2019.
- [10] T. Peham, L. Burgholzer, and R. Wille, "On Optimal Subarchitectures for Quantum Circuit Mapping," *ACM Transactions on Quantum Computing*, 2023.
- [11] S. Hillmich, A. Zulehner, and R. Wille, "Exploiting Quantum Teleportation in Quantum Circuit Mapping," in *Asia and South Pacific Design Automation Conf.*, 2021.
- [12] J. Liu, E. Younis, M. Weiden, P. Hovland, J. Kubiatowicz, and C. Iancu, "Tackling the qubit mapping problem with permutation-aware synthesis," in *Int'l Conf. on Quantum Computing and Engineering*, vol. 01, pp. 745–756, 2023.
- [13] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah, "On the Qubit Routing Problem," in *Conf. on the Theory of Quantum Computation, Communication and Cryptography (TQC)*, vol. 135 of *Leibniz Int'l Proceedings in Informatics (LIPIcs)*, pp. 5:1–5:32, 2019.
- [14] L. Schmid, S. Park, and R. Wille, "Hybrid Circuit Mapping: Leveraging the Full Spectrum of Computational Capabilities of Neutral Atom Quantum Computers," in *Design Automation Conf.*, 2024.
- [15] R. Wille and L. Burgholzer, "MQT QMAP: Efficient quantum circuit mapping," in *Int'l Symp. on Physical Design*, 2023.
- [16] N. Paraskevopoulos, F. Sebastiano, C. G. Almudever, and S. Feld, "Spinq: Compilation strategies for scalable spin-qubit architectures," *ACM Transactions on Quantum Computing*, vol. 5, no. 1, 2023.
- [17] B. Giles and P. Selinger, "Exact synthesis of multiqubit Clifford+T circuits," *Physical Review A*, vol. 87, no. 3, p. 032332, 2013.
- [18] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 32, no. 6, pp. 818–830, 2013.
- [19] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-control Toffoli gates," in *Int'l Symp. on Multi-Valued Logic*, 2011.
- [20] A. Zulehner and R. Wille, "One-pass design of reversible circuits: Combining embedding and synthesis for reversible logic," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 5, pp. 996–1008, 2018.
- [21] P. Niemann, R. Wille, and R. Drechsler, "Advanced exact synthesis of Clifford+T circuits," *Quantum Information Processing*, 2020.
- [22] E. Younis and C. Iancu, "Quantum circuit optimization and transpilation via parameterized circuit instantiation," in *Int'l Conf. on Quantum Computing and Engineering*, pp. 465–475, 2022.
- [23] W. Hattori and S. Yamashita, "Quantum circuit optimization by changing the gate order for 2D nearest neighbor architectures," in *Int'l Conf. of Reversible Computation*, pp. 228–243, 2018.
- [24] G. Vidal and C. M. Dawson, "Universal quantum circuit for two-qubit transformations with three controlled-NOT gates," *Physical Review A*, vol. 69, no. 1, p. 010301, 2004.
- [25] T. Itoko, R. Raymond, T. Imamichi, A. Matsuo, and A. W. Cross, "Quantum circuit compilers using gate commutation rules," in *Asia and South Pacific Design Automation Conf.*, pp. 191–196, 2019.
- [26] D. Maslov, G. Dueck, D. Miller, and C. Negrevergne, "Quantum circuit simplification and level compaction," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 436–444, 2008.
- [27] S. Niu, A. Hashim, C. Iancu, W. A. de Jong, and E. Younis, "Powerful quantum circuit resizing with resource efficient synthesis," 2023.
- [28] K. Staudacher, T. Guggemos, S. Grundner-Culemann, and W. Gehrke, "Reducing 2-qubit gate count for zx-calculus based quantum circuit optimization," *Electronic Proceedings in Theoretical Computer Science*, vol. 394, p. 29–45, 2023.
- [29] L. Sünkel, D. Martyniuk, D. Mattern, J. Jung, and A. Paschke, "Ga4qco: Genetic algorithm for quantum circuit optimization," 2023. arXiv 2302.01303.
- [30] D. Kremer, V. Villar, H. Paik, I. Duran, I. Faro, and J. Cruz-Benito, "Practical and efficient quantum circuit synthesis and transpiling with reinforcement learning," 2024. arXiv 2405.13196.
- [31] P. D. Nation and M. Treinish, "Suppressing quantum circuit errors due to system variability," *PRX Quantum*, vol. 4, p. 010327, 2023.
- [32] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille, "Computational capabilities and compiler development for neutral atom quantum processors—connecting tool developers and hardware experts," *Quantum Science and Technology*, vol. 9, no. 3, p. 033001, 2024.
- [33] T. Lubinski, J. J. Goings, K. Mayer, S. Johri, N. Reddy, A. Mehta, N. Bhatia, S. Rappaport, D. Mills, C. H. Baldwin, L. Zhao, A. Barbosa, S. Maity, and P. S. Mundada, "Quantum algorithm exploration using application-oriented performance benchmarks," 2024. arXiv 2402.08985.
- [34] D. Venturelli, M. Do, B. O'Gorman, J. Frank, E. Rieffel, K. E. C. Booth, T. Nguyen, P. Narayan, and S. Nanda, "Quantum circuit compilation: An emerging application for automated reasoning," in *Proceedings of the Scheduling and Planning Applications Workshop (SPARK)*, 2019.
- [35] H. Kurniawan, L. Rodríguez-Soriano, D. Cuomo, C. G. Almudever, and F. G. Herrero, "On the use of calibration data in error-aware compilation techniques for NISQ devices," 2024. arXiv 2407.21462.
- [36] S. Dangwal, G. S. Ravi, L. M. Seifert, and F. T. Chong, "Clifford assisted optimal pass selection for quantum transpilation," 2023. arXiv 2306.15020.
- [37] B. Mete, M. Schulz, and M. Ruefenacht, "Predicting the optimizability for workflow decisions," in *2022 IEEE/ACM Third Int'l Workshop on Quantum Computing Software (QCS)*, pp. 68–74, 2022.
- [38] A. Vadali, R. Kshirsagar, P. Shyamsundar, and G. N. Perdue, "Quantum circuit fidelity estimation using machine learning," *Quantum Machine Intelligence*, vol. 6, no. 1, 2023.
- [39] H. Wang *et al.*, "Torchquantum case study for robust quantum circuits," in *Proceedings of the 41st IEEE/ACM Int'l Conf. on Computer-Aided Design*, 2022.
- [40] N. Quetschlich, L. Burgholzer, and R. Wille, "MQT Predictor: Automatic Device Selection with Device-Specific Circuit Compilation for Quantum Computing," *ACM Transactions on Quantum Computing (TQC)*, 2024.
- [41] T. Tomesh *et al.*, "Supermarq: A scalable quantum benchmark suite," 2022. arXiv 2202.11045.
- [42] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schoenberger, Y. Stade, and L. Burgholzer, "The MQT handbook: A summary of design automation tools and software for quantum computing," in *IEEE International Conference on Quantum Software (QSW)*, 2024.
- [43] N. Quetschlich, L. Burgholzer, and R. Wille, "MQT Bench: Benchmarking software and design automation tools for quantum computing," *Quantum*, 2023.
- [44] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024. arXiv 2405.08810.
- [45] L. Abdurakhimov *et al.*, "Technology and performance benchmarks of iqm's 20-qubit quantum computer," 2024. arXiv 2408.12433.
- [46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [47] M. Bandic, P. le Henaff, A. Ovide, P. Escofet, S. B. Rached, S. Rodrigo, H. van Someren, S. Abadal, E. Alarcon, C. G. Almudever, and S. Feld, "Profiling quantum circuits for their efficient execution on single- and multi-core architectures," 2024. arXiv 2407.12640.
- [48] M. Bandic, C. G. Almudever, and S. Feld, "Interaction graph-based characterization of quantum benchmarks for improving quantum circuit mapping techniques," *Quantum Machine Intelligence*, vol. 5, no. 2, 2023.