

# Qubit Mapping and Routing tailored to Advanced Quantum ISAs: Not as Costly as You Think

## Abstract

Qubit mapping/routing is critical for executing quantum circuits NISQ and fault-tolerant hardware. Existing scalable methods typically impose several times the routing overhead, especially in circuit depth or duration, due to the disconnect between simplified routing models (e.g., three-CX-unrolled SWAP insertion) and hardware-native gate properties. This gap severely limits compiler optimization potential and thus practical circuit execution performance.

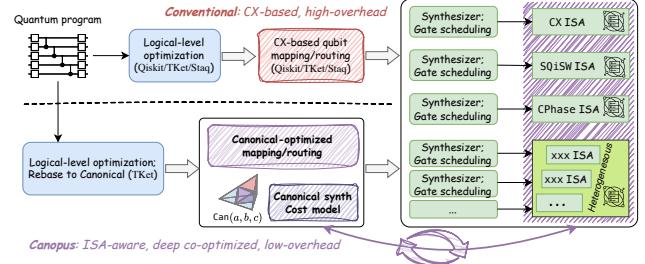
Recent advances in quantum hardware have enabled high-precision implementations of diverse instruction set architectures (ISAs) beyond standard CX-based gates. Advanced ISAs such as  $\sqrt{i}$ SWAP and ZZ( $\theta$ ) offer superior circuit synthesis capabilities while retaining hardware-native noise resilience. However, there are neither systematic compiler optimization strategies tailored to these advanced ISAs nor comprehensive cross-ISA evaluation to unlock their potential.

To address this, we propose CANOPUS, a unified qubit mapping/routing framework tailored to diverse quantum ISAs. With the canonical representation of two-qubit gates, CANOPUS revolves around qubit routing to perform deep co-optimization in an ISA-aware approach. CANOPUS utilizes the canonical forms of arbitrary basis gates and monodromy polytope theory to model the synthesis cost for better SWAP search during routing. Commutation relations between two-qubit gates can be formalized through the canonical representation in our findings, providing a generalized approach to capturing commutative optimizations. Experiments show that CANOPUS consistently results in 30%-40% reduction of routing overhead compared to state-of-the-art methods across versatile ISAs and topologies. Our work also presents a coherent method for co-exploration of program patterns, quantum ISAs, and hardware topologies. Our framework can be extended to integrate more fine-grain hardware information such as qubit-specific basis gate fidelities. We have for the first time demonstrated that advanced quantum ISAs can be efficiently utilized within a unified routing framework, paving the way for more effective co-design of quantum software and hardware.

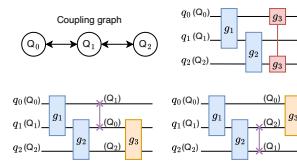
## 1 Introduction

Advanced ISAs—Can [4],  $\sqrt{i}$ SWAP [14]

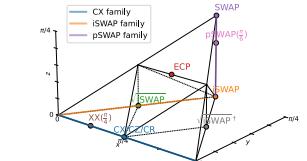
CANOPUS (**C**anonical-**O**ptimized **P**lacement **U**tility **S**uite) is a qubit mapping and routing framework that is tailored to advanced quantum ISAs, such as Can [4] and  $\sqrt{i}$ SWAP [14], which are adaptive to versatile hardware architectures. CANOPUS is designed to optimize the placement of qubits and the



**Figure 1.** Compilation workflows by means of conventional approaches (top) and CANOPUS (bottom) targeting diverse quantum ISAs. CANOPUS integrates the synthesis cost model (monodromy polytopes within the Weyl chamber) by taking backend ISAs’ synthesis properties into account. CANOPUS routing operates in the 2Q canonical representation while the specific synthesis is completed by backend synthesizer.



**Figure 2.** Mapping/routing to resolve physical-qubit topology constraints via SWAP insertion.



**Figure 3.** Geometric illustration of canonical gates confined to the Weyl chamber.

routing of quantum gates, taking into account the specific requirements of these advanced ISAs.

Our work addresses the “Babel Tower dilemma” in quantum compilation by establishing a canonical language for diverse two-qubit gates, enabling unified optimization across heterogeneous quantum ISAs. Our key contributions are summarized as follows:

- ....
- ....
- ....
- ....

## 2 Background

1

### 2.1 Qubit mapping/routing

Qubit placement and routing ... for connectivity-limited devices

<sup>1</sup>For convenient visualization

## 2.2 Quantum gates realization in diverse ISAs

**Definition 1** (Canonical gate). Any 2Q gate  $U \in \text{SU}(4)$  can be expressed by the composition of its unique canonical form

$$\text{Can}(a, b, c) := e^{-i\frac{\pi}{2}(aXX+bYY+cZZ)}, \frac{1}{2} \geq a \geq b \geq |c| \quad (1)$$

sandwiched by local 1Q gates such that we call  $U$  is locally equivalent to the canonical form  $\text{Can}(a, b, c)$ .

Canonical representation is ubiquitous as an effective ...

[ZY: It is ubiquitously used in many quantum computing task ...]

Although there are other conventions .... This definition aligns with the TK2 operation definition in TKET, ...

Figure 3 .....

$$\text{pSWAP}(\theta) \sim \text{Can}\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2} - \frac{\theta}{\pi}\right) \quad (2)$$

$$XX(\theta) = \text{Can}\left(\frac{\theta}{\pi}, 0, 0\right) \sim YY(\theta) \sim ZZ(\theta) \quad (3)$$

## 3 Motivation

Two-fold motivations:

1. The scalable qubit routing effects (2x-4x) is still a critical challenge in practical quantum computing systems
2. How to utilize the emerging advanced ISAs (hardware breakthroughs); across all phases of compilation, routing is the bottleneck and is the most easily handled for co-optimization

[ZY: Use a “optimal routing benchmark” to illustrate the OVERHEAD of existing methods]

[ZY: There should be many takeaways]

- Previous routing overhead is not precise for hardware execution
- Previous routing is costly and also not precise for hardware execution
- SWAP can be implemented in low overhead (gate duration) with the recent breakthrough gate schemes for advanced ISAs
- How to efficiently capture the rich commutation relations when performing co-optimization during qubit routing and gate scheduling

## 4 CANOPUS framework

### 4.1 Overview

### 4.2 2Q synthesis cost modeling

### 4.3 Routing in canonical form

In contrast to the regular heuristic cost function used in SABRE:

$$\begin{aligned} H &= \frac{1}{|F|} \sum_{(i,j) \in F} \text{dist}[i,j] + \frac{k_E}{|E|} \sum_{(i,j) \in E} \text{dist}[i,j] \\ &= \text{Avg}\{\text{dist}[i,j]\}_F + k_E \text{Avg}\{\text{dist}[i,j]\}_E \end{aligned} \quad (4)$$

---

### Algorithm 1: Update $L$ when adding a new 2Q gate

---

```

Input :  $G'$  (Routed DAG),  $\pi$  (current logic-to-physical mapping),  $L$  (last mapped layer),  $D$  (wire durations for each qubit),  $C$  (commutative pairs within  $L$ )
Output: Updated  $G'$ ,  $L$ ,  $D$ ,  $C$ 

/*  $g$ : resolved logical gate;  $g'$ : routed gate */
1  $g' \leftarrow G'.\text{PUSHBACK}(g, \pi[g.q_0], \pi[g.q_1]); // g'.q_i = \pi[g.q_i]$ ;
2  $d \leftarrow \max(D[g'.q_0], D[g'.q_1]) + \text{SYNTHCOST}(g)$ ;
3  $D[g'.q_0] \leftarrow d$ ;  $D[g'.q_1] \leftarrow d$ ;
4 for pred  $\in G'.\text{PREDECESSORS}(g')$  do
5   if  $\text{IS2QGATE}(\text{pred})$  then
6     if  $\text{ISCOMMUTATIVECANONICALPAIR}(g', \text{pred})$  then
7        $C[(\text{pred}.q_0, \text{pred}.q_1)] \leftarrow (g'.q_0, g'.q_1)$ ;
8     else
9        $L.\text{POP}((\text{pred}.q_0, \text{pred}.q_1), \text{None})$ ;
10       $C.\text{POP}((\text{pred}.q_0, \text{pred}.q_1), \text{None})$ ;
11    else
12      /* pred_pred must be None or a 2Q gate */
13      pred_pred  $\leftarrow \text{NEXT}(G'.\text{PREDECESSORS}(\text{pred}))$ ;
14      if pred_pred  $\neq \text{None}$  then
15         $L.\text{POP}((\text{pred\_pred}.q_0, \text{pred\_pred}.q_1), \text{None})$ ;
16         $C.\text{POP}((\text{pred\_pred}.q_0, \text{pred\_pred}.q_1), \text{None})$ ;
16  $L[(g'.q_0, g'.q_1)] \leftarrow g'$ ;

```

---

which involves the basic (left term) and lookahead (right term) components. In practice, there is a  $w_{\text{decay}}$  decay factor applied to  $H$ , which is not shown as it does not affect the composition of  $H$ .

The heuristic cost function in CANOPUS is defined as:

$$H = w_d \Delta_{\text{depth}} + c_0 (\Delta_{\text{Avg}\{\text{dist}[i,j]\}_F} + k_E \Delta_{\text{Avg}\{\text{dist}[i,j]\}_E}) \quad (5)$$

- Unified and highly-effective qubit routing approach in canonical form, with properties of quantum ISAs tailored to the routing process

### 4.4 Enhanced optimization via commutation

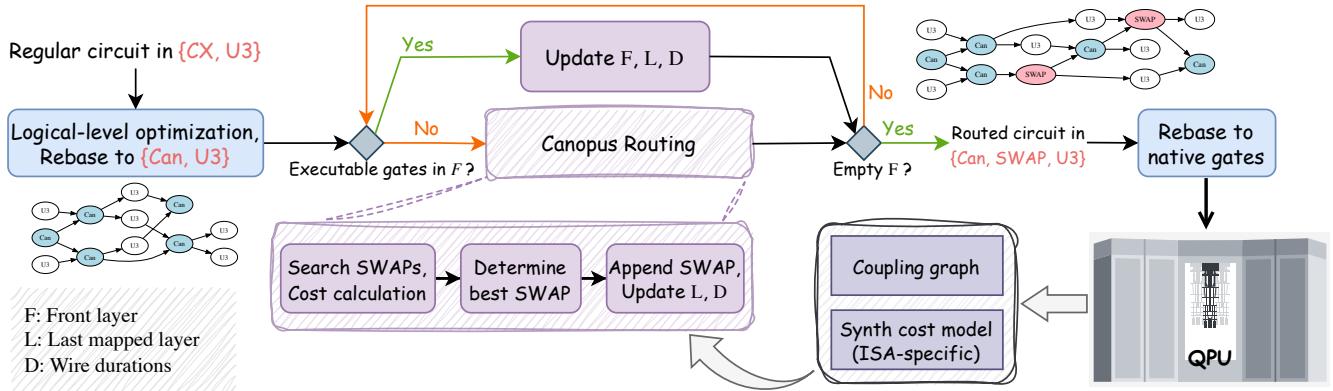
- Capture optimization opportunities exposed by gate commutation; while commutation relations can be uniformly described in canonical form

**Theorem 1** (Canonical gate commutation). Let  $\text{Can}(a, b, c)_{q_0, q_1}$  and  $\text{Can}(a', b', c')_{q_1, q_2}$  denote canonical gates acting on qubits  $(q_0, q_1)$  and  $(q_1, q_2)$  respectively, with an overlapping qubit  $q_1$ . They are commutative if and only if

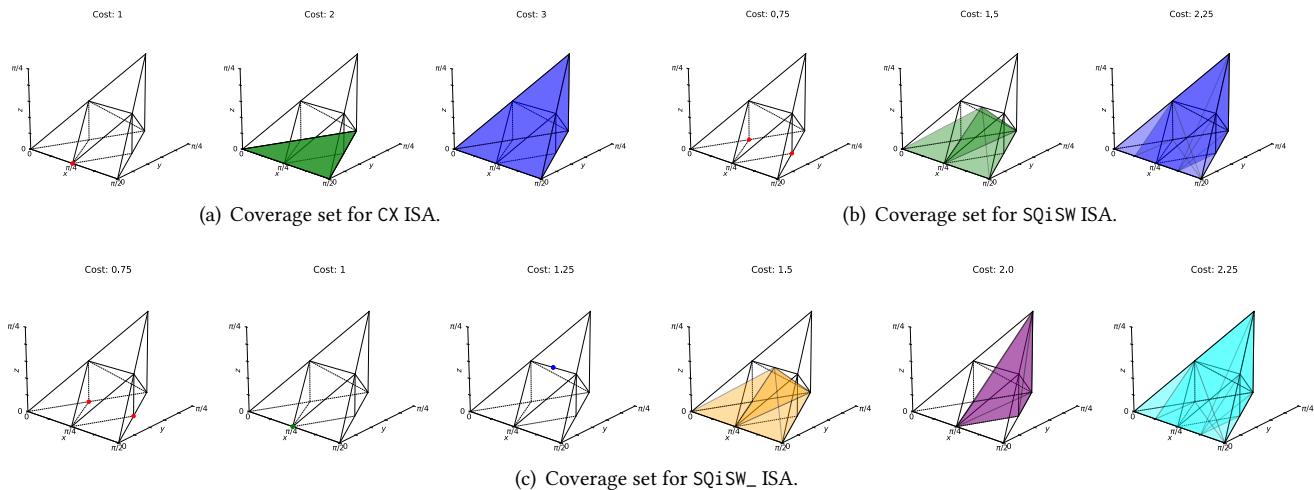
$$b = b' = c = c' = 0, \quad (6)$$

that is, when both consist solely of XX rotations.

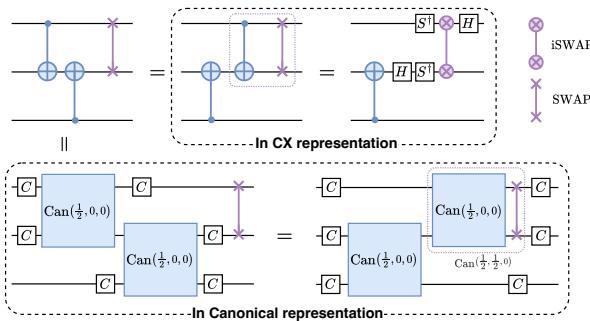
[ZY: Proposition?? Theorem?]



**Figure 4.** Overview of the CANOPUS framework. ...



**Figure 5.** Coverage set examples. CX ISA: {CX, U3} gate set; SQiSW ISA:  $\{\sqrt{iSWAP}, iSWAP, U3\}$  gate set; SQiSW\_ISA:  $\{\sqrt{iSWAP}, iSWAP, ECP, CX, U3\}$  gate set. Costs of Basis 2Q gates are set as CX  $\sim 1$ ,  $\sqrt{iSWAP} \sim 0.75$ , iSWAP  $\sim 1.5$ , ECP  $\sim 1.25$ .



**Figure 6.** Canonical gate representation enables easily capturing commutative relations within real-world circuits.

#### Algorithm 2: Update D when adding a SWAP gate

---

**Input :** swap (encountered SWAP gate), can (canonical gate within L on the same qubits as swap), D, C  
**Output:** Updated D

```

1 if (swap.q0, swap.q1) ∈ C then
2   q'0, q'1 ← C[(swap.q0, swap.q1)];
   /* Adjust D by finding matched qubits
      qi ∈ {swap.q0, swap.q1} and q'j ∈ {q'0, q'1} */
3   D[qi] ← D[q'j] + SYNTHCOST(can);
4   D[the other swap qubit] ← D[qi];
5   d ← MAX(D[swap.q0], D[swap.q1]) +
      SYNTHCOST(can.MIRROR()) - SYNTHCOST(can);
6   D[swap.q0] ← d; D[swap.q1] ← d;

```

---

## 5 Implementation

### 5.1 Core functionalities

### 5.2 Extensions

### 5.3 Scalability

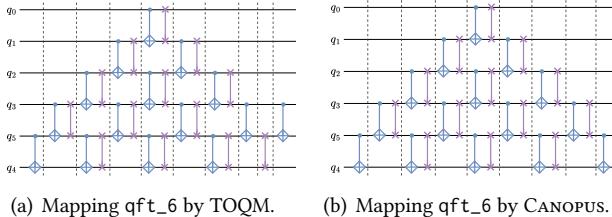
## 6 Case Studies

### 6.1 QFT kernel

### 6.2 Co-exploration of routing and ISA selection

**Table 1.** Qubit routing comparison for the QFT kernel.

QFT kernel		qft_6		qft_12	
Topology	Method	#Can	Depth2Q	#Can	Depth2Q
1D Chain	Optimal	15	9	66	21
	TOQM	16	10	67	22
	CANOPUS	15	9	67	21
2D Square	TOQM	21	13	100	39
	CANOPUS	15	9	80 ( $\pm 10\%$ )	30 ( $\pm 10\%$ )



**Figure 7.** [ZY: Re-draw this figure] Mapping/routing comparison for the QFT kernel. For convenient visualization, only CPhase and SWAP gates are shown. (a) TOQM generates a sub-optimal mapping scheme, with 2Q depth of 10. (b) CANOPUS generates the optimal scheme in a perfect butterfly structure, with 2Q depth of 9.

provides cross-compiler but also cross-ISA comparisons under the coherent basis gate cost and routing overhead settings.

## 7.1 Experimental settings

**7.1.1 ISAs and basis gate costs.** We consider six different ISAs (including the conventional CX ISA) listed in Table 2. These mainly cover a wide range of powerful basis gates from CX-family and iSWAP-family gates. Particularly, SQiSW [14] proves to a more powerful ISA option and has been adopted by recent software projects [12, 24]. ZZPhase ISA containing three fractional  $ZZ(\theta)$  rotation gates is adopted by QISKit’s latest synthesis functionalities [16, 25]. Mirorr ..... [ZY: TODO: find the initial paper about mirror gate]

[24]

We also involve the Het ISA that is the composition of ZZPhase and SQiSW.

The unit costs for the involved basis gates are set as:

$$\left\{ \begin{array}{l} CX : 1, ZZ(\pi/t) : 2/t, \sqrt{iSWAP} : 0.75, \\ iSWAP : 1.5, ECP : 1.25, pSWAP(\pi/t) : 2 - 1/t \end{array} \right\} \quad (7)$$

[ZY: Plot a weyl chamber to illustrate the cost settings]

**7.1.2 Benchmarks.** We select a set of medium-size benchmarks from QASMBench [21] and MQTBench [28] spanning various categories of quantum programs. These benchmarks first go through logical-level optimization by TKET [29] and

**Table 2.** Selected quantum ISAs.

ISA	2Q basis gates	Description
CX	{CX}	Conventional CX gate
ZZPhase	$\{ZZ_{\frac{\pi}{6}}, ZZ_{\frac{\pi}{4}}, ZZ_{\frac{\pi}{2}}\}$	Discrete CX-family gates, i.e., $\{\sqrt[3]{CX}, \sqrt{CX}, CX\}$ [25]
SQiSW	$\{\sqrt{iSWAP}, iSWAP\}$	Half evolution of iSWAP and iSWAP [14]
ZZPhase_	ZZPhase + $\{pSWAP_{\frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{2}}\}$	ZZPhase ISA with the mirror gates
SQiSW_	SQiSW + {ECP, CX}	SQiSW ISA with the mirror gates [24]
Het	ZZPhase + SQiSW	Heterogeneous CX-family and iSWAP-family gates

**Table 3.** Benchmarks information. These metrics are collected from the circuits after logical-level optimization by TKET, thus including only Can and U3 gates. Circuit cost (Duration  $T$ ) is calculated in CX ISA.

Program	#Qubit	#Can	Depth2Q	Circuit cost
bigadder [21]	18	114	79	88.0
bv [21]	19	18	18	18.0
gcm [21]	13	377	376	510.0
ising [21]	26	25	2	4.0
knn [21]	25	72	50	62.0
multiplier [21]	15	198	122	133.0
qec9xz [21]	17	32	12	12.0
qft [28]	18	153	33	66.0
qpeexact [28]	16	127	43	86.0
qram [21]	20	110	70	78.0
sat [21]	11	210	182	204.0
swap_test [21]	25	72	50	62.0
wstate [21]	27	52	28	28.0

are rebased to {Can, U3} as the input of qubit routing compilers. Information of benchmarks after logical-level optimization are summarized in Table 3, where “Circuit cost” denotes the circuit duration assuming each canonical gate will be finally rebased to CX ISA and the duration (cost) of each CX is set to 1.

## 7.1.3 Baselines.

### 7.2 Circuit duration reduction

### 7.3 Co-exploration of routing and ISA selection

### 7.4 Breakdown analysis

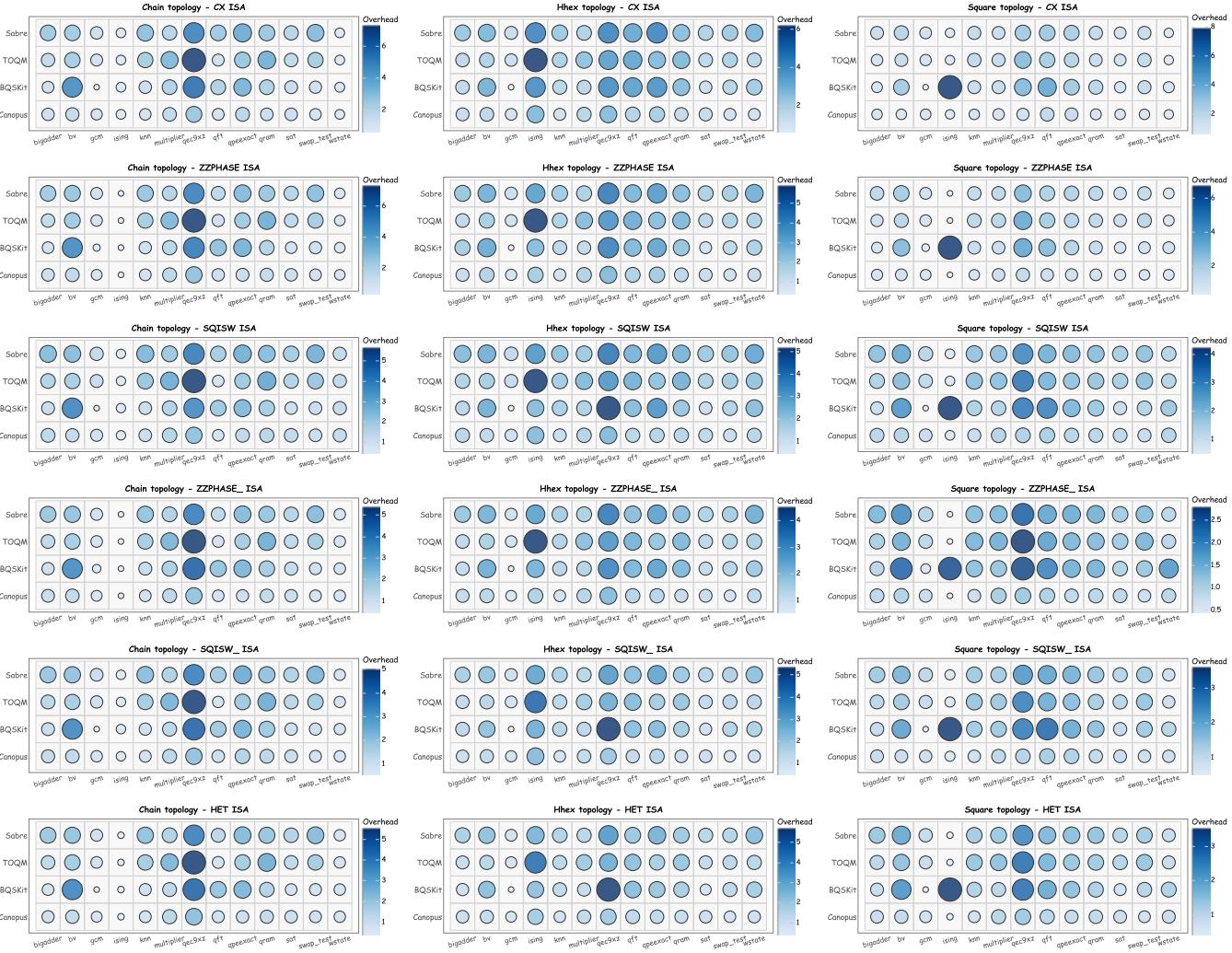
**7.4.1 Commutative optimization.** In this section, we analyze individual factors in the improvement brought by CANOPUS, mainly about the commutative optimization mechanism and the heuristic depth-cost weight factor.

**7.4.2 Depth weight factor.** Note that the ...

[ZY: about comm\_opt effect]

### 7.5 Real-system experiments

[ZY: Fractional gate IBM]



**Figure 8.** Detailed benchmarking results for different compilers (SABRE, TOQM, BQSKit, CANOPUS) across diverse topologies (columns from left to right: 1D Chain, 2D HHex, 2D Square) and quantum ISAs (rows from top to bottom: CX, ZZPhase, SQiSW, ZZPhase<sub>\_</sub>, SQiSW<sub>\_</sub>, Het). Sizes and colors of the bubbles represent the values for routing overhead (multiples of the routed and rebased circuits compared to that of prior-routing ones).

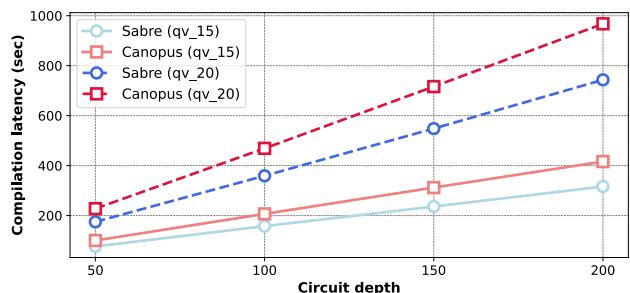
[ZY: QFT or QV test?]

## 7.6 For specific gate scheme

[ZY: For AshN gate scheme]

## 7.7 Runtime analysis

In our field tests for the set of benchmarks above, CANOPUS exhibits around 1x-2x compilation latency than SABRE, both of which are implemented by QISKit framework. This result aligns with the complexity analysis in Section 5.3. Herein we specifically demonstrate the end-to-end compilation latency for larger-scale quantum circuits. We use random quantum



**Figure 9.** Compilation latency comparison.

**Table 4.** Summarized results for the routing overhead with average (geometric-mean) values emphasized.

Topology	Routing overhead (CX)				Routing overhead (ZZPhase)			
	SABRE	TOQM	BQSKIT	CANOPUS	SABRE	TOQM	BQSKIT	CANOPUS
Chain	2.93x	2.67x	2.34x	<b>1.73x</b>	2.63x	2.51x	2.17x	<b>1.61x</b>
HHex	2.90x	2.56x	2.66x	<b>1.81x</b>	2.62x	2.37x	2.26x	<b>1.66x</b>
Square	2.14x	2.04x	2.47x	<b>1.42x</b>	1.87x	1.82x	1.94x	<b>1.19x</b>
Routing overhead (SQiSW)				Routing overhead (ZZPhase_)				
Topology	SABRE	TOQM	BQSKIT	CANOPUS	SABRE	TOQM	BQSKIT	CANOPUS
Chain	2.57x	2.27x	2.00x	<b>1.55x</b>	2.08x	1.94x	1.87x	<b>1.26x</b>
HHex	2.59x	2.33x	2.28x	<b>1.64x</b>	2.08x	1.90x	1.96x	<b>1.33x</b>
Square	2.05x	1.94x	2.02x	<b>1.39x</b>	1.54x	1.49x	1.65x	<b>1.00x</b>
Routing overhead (SQiSW_)				Routing overhead (Het)				
Topology	SABRE	TOQM	BQSKIT	CANOPUS	SABRE	TOQM	BQSKIT	CANOPUS
Chain	2.17x	1.93x	1.78x	<b>1.38x</b>	2.13x	2.00x	1.74x	<b>1.29x</b>
HHex	2.17x	1.93x	2.10x	<b>1.42x</b>	2.13x	1.94x	1.98x	<b>1.37x</b>
Square	1.67x	1.58x	1.83x	<b>1.15x</b>	1.58x	1.52x	1.56x	<b>1.02x</b>

volume [8] circuits generated by QISKit for scalability benchmarking, which represents the end of the spectrum with respect to canonical-form circuits. Each canonical gate within the quantum volume circuit contains unique canonical parameters as each 2Q unitary is randomly generated, thus there is no cached synthesis cost calculation for performance improvement in one pass. In the cases of circuit width (number of qubits) set to 15 and 20, we vary the depth of benchmarked circuits from 50 to 200. Quantum volume circuits consists of dense 2Q gates and the largest size for benchmarking is up to thousands of 2Q gates. Figure 9 illustrates the end-to-end compilation latencies, where each data point is tested with the same trial setting (`max_iterations` is 5, both `trials` and `layout_trials` are 10). For each benchmarked circuit, CANOPUS leads to on average 1.31x ( $\pm 1\%$ ) latency than SABRE. Both compilers’ latency scales linearly with circuit depth and width. If we compares the curve slopes, CANOPUS leads to 1.32x (1.30x) latency scaling than SABRE in terms of circuit depth for `qv_10` (`qv_20`) circuits. Overall, although CANOPUS involves sophisticated data structures and calculation mechanisms, its practical compilation scalability is comparable to the industrial-level SABRE algorithm.

## 8 Related Works

Qubit mapping/routing is one the the most well-explored topic of quantum compiler research, as it shares the similar methodologies with instruction scheduling [6, 13] and register allocation [3, 27] in classical computing. Conventional methods focus on the simplified routing model, that is, #SWAP-minimal insertion, three-CX-unrolled SWAP gate, and CX-based latency metric. That brings a gap between quantum hardware performance and its ceiling, which is particularly evident with the progress of underlying instruction models for modern quantum hardware.

Zulehner et al. [40] introduces an A\*-based algorithm to minimize SWAP gate overhead for concurrent CNOT gate layers. The approach partitions the circuit into layers and solves the mapping problem subsequently. Li et al. [22] also utilizes the circuit DAG layering thought to tackle the qubit mapping problem and proposes the bidirectional routing procedure to acquire a better initial mapping desired to result in #SWAP inserted minimization as expected. It also briefly discusses the trade-off between the inserted SWAP count and the circuit depth but does not prioritize optimizing circuit depth. Some other works leverage algorithmic procedures similar to SABRE to improve parallelism among inserted SWAPs and other 2Q gates [1, 20, 39], or attempt to minimize circuit depth via graph matching [5]. Zhang et al. [36] systematically investigates the time (circuit depth) optimality of qubit mapping and proposed an A\*-based method TOQM that results in better results than the SOTA solver-based depth-driven algorithm [30]. However, the optimality of qubit routing is a complex task. There are rarely theoretical studies that claims the holistic optimality of some SWAP insertion schemes provided the quantum ISAs, device topologies, and synthesis cost models. In our field tests, TOQM does not lead to time-optimal results compared to our heuristic CANOPUS, and the optimal mapping scheme for specific patterns such as QFT kernel analyzed in [36] are not indeed optimal, according to our case study in Section 6.1.

With the recent development of advanced quantum ISAs such as superconducting fractional gates [15], ion-trapped partial entangling gates [17, 33], and the AshN scheme [4], some works began exploring how to efficiently utilize these ISAs to make compiler optimizations closer to hardware characteristics. McKinney et al. [24] investigates the practical performance of SQiSW ISA proposed by Huang et al. [14] and the synthesis capability when incorporating the basis gates’ mirrors into the ISA. The modified SABRE algorithm

in [24] provides an attempt of the collaborative gate decomposition and qubit routing approach, while the optimization opportunities considered therein are limited and the algorithmic techniques are not sophisticated. BQSKit [34] and the series of works behind [9, 19, 32, 35] provides a toolkit to rebase arbitrary 2Q unitaries to specific ISAs through approximate synthesis (structural search and numerical optimization) that is not computational efficient. Approximate synthesis by BQSKit does not ensure an optimal schemes for two-qubit and multi-qubit synthesis cases. In addition, due to the lack of native compilation strategies and rational synthesis cost model, Kalloor et al. [18] claims that alternative ISAs are hard to be comparable to CX when evaluating quantum hardware roofline by BQSKit. As for applicability of expanded ISAs to QEC, Google’s latest theoretical [23] and experimental [10] works demonstrate the CX-iSWAP combination ISA could benefits suppressing fault-tolerant threshold. Zhou et al. [38] proposes a routing-based method enhanced by CX-iSWAP for overcoming ancilla defects among surface code blocks while preserving encoded logical information.

## 9 Conclusion

It is promising to explore novel Clifford circuit optimization techniques drawing on the canonical gate representation.

## References

- [1] Alessandro Annechini, Marco Venere, Donatella Sciuto, and Marco Santambrogio. 2025. DDRoute: a Novel Depth-Driven Approach to the Qubit Routing Problem. In *Proceedings of the 62st ACM/IEEE Design Automation Conference*.
- [2] Stephen S Bullock and Igor L Markov. 2003. An arbitrary two-qubit computation in 23 elementary gates or less. In *Proceedings of the 40th Annual Design Automation Conference*. IEEE, Anaheim, CA, USA, 324–329.
- [3] Gregory J Chaitin. 1982. Register allocation & spilling via graph coloring. *ACM Sigplan Notices* 17, 6 (1982), 98–101.
- [4] Jianxin Chen, Dawei Ding, Weiyuan Gong, Cupjin Huang, and Qi Ye. 2024. One Gate Scheme to Rule Them All: Introducing a Complex Yet Reduced Instruction Set for Quantum Computing. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. ACM, La Jolla, CA, USA, 779–796.
- [5] Andrew M Childs, Eddie Schoute, and Cem M Unsal. 2019. Circuit transformations for quantum architectures. *arXiv preprint arXiv:1902.09102* (2019).
- [6] Josep M Codina, Jesús Sánchez, and Antonio González. 2001. A unified modulo scheduling and register allocation technique for clustered processors. In *Proceedings 2001 International Conference on Parallel Architectures and Compilation Techniques*. IEEE, 175–184.
- [7] Gavin E Crooks. 2020. Gates, states, and circuits. Available at <https://threplusone.com/pubs/on-gates-v0-5/>.
- [8] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. 2019. Validating quantum computers using randomized model circuits. *Physical Review A* 100, 3 (2019), 032328.
- [9] Marc Grau Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. 2019. Heuristics for quantum compiling with a continuous gate set. 12 pages. *arXiv preprint arXiv:1912.02727*.
- [10] Alec Eickbusch, Matt McEwen, Volodymyr Sivak, Alexandre Bourassa, Juan Atalaya, Jahan Claes, Dvir Kafri, Craig Gidney, Christopher W. Warren, Jonathan Gross, Alex Opremcak, Nicholas Zobrist, Kevin C. Miao, Gabrielle Roberts, Kevin J. Satzinger, Andreas Bengtsson, Matthew Neeley, William P. Livingston, Alex Greene, Rajeev Acharya, Laleh Aghababaie Beni, Georg Aigeldinger, Ross Alcaraz, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Ryan Babbush, Brian Ballard, Joseph C. Bardin, Alexander Bilmes, Jenna Bovaird, Dylan Bowers, Leon Brill, Michael Broughton, David A. Browne, Brett Bueche, Bob B. Buckley, Tim Burger, Brian Burkett, Nicholas Bushnell, Anthony Cabrera, Juan Campero, Hung-Shen Chang, Ben Chiaro, Liang-Ying Chih, Agnetta Y. Cleland, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Sayan Das, Alexander Del Toro Barba, Sean Demura, Laura De Lorenzo, Agustin Di Paolo, Paul Donohoe, Ilya K. Drozdov, Andrew Dunsworth, Aviv Moshe Elbag, Mahmoud Elzouka, Catherine Erickson, Vinicius S. Ferreira, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, Suhas Ganjam, Gonzalo Garcia, Robert Gasca, Élie Genois, William Giang, Dar Gilboa, Raja Gosula, Alejandro Grajales Dau, Dietrich Graumann, Tan Ha, Steve Habegger, Monica Hansen, Matthew P. Harrigan, Sean D. Harrington, Stephen Heslin, Paula Heu, Oscar Higgott, Reno Hiltermann, Jeremy Hilton, Hsin-Yuan Huang, Ashley Huff, William J. Huggins, Evan Jeffrey, Zhang Jiang, Xiaoxuan Jin, Cody Jones, Chaitali Joshi, Pavol Juhas, Andreas Kabel, Hui Kang, Amir H. Karamlou, Kostyantyn Kechedzhi, Trupti Khaire, Tanuj Khattar, Mostafa Khezri, Seon Kim, Bryce Kobilin, Alexander N. Korotkov, Fedor Kostritsa, John Mark Kreikebaum, Vladislav D. Kurilovich, David Landhuis, Tiano Lange-Dei, Brandon W. Langley, Kim-Ming Lau, Justin Ledford, Kenny Lee, Brian J. Lester, Loïck Le Guevel, Wing Yan Li, Alexander T. Lill, Aditya Locharla, Erik Lucero, Daniel Lundahl, Aaron Lunt, Sid Madhuk, Ashley Maloney, Salvatore Mandrà, Leigh S. Martin, Orion Martin, Cameron Maxfield, Jarrod R. McClean, Seneca Meeks, Anthony Megrant, Reza Molavi, Sebastian Molina, Shirin Montazeri, Ramis Movassagh, Michael Newman, Anthony Nguyen, Murray Nguyen, Chia-Hung Ni, Logan Oas, Raymond Orosco, Kristoffer Ottosson, Alex Pizzuto, Rebecca Potter, Orion Pritchard, Chris Quintana, Ganesh Ramachandran, Matthew J. Reagor, David M. Rhodes, Elliott Rosenberg, Elizabeth Rossi, Kannan Sankaragomathi, Henry F. Schurkus, Michael J. Shearn, Aaron Shorter, Noah Shatty, Vladimir Shvarts, Spencer Small, W. Clarke Smith, Sofia Springer, George Sterling, Jordan Suchard, Aaron Szasz, Alex Sztein, Douglas Thor, Eifu Tomita, Alfredo Torres, M. Mert Torunbalci, Abeer Vaishnav, Justin Vargas, Sergey Vдовичев, Guifre Vidal, Catherine Vollgraff Heidweiller, Steven Waltman, Jonathan Waltz, Shannon X. Wang, Brayden Ware, Travis Weidel, Theodore White, Kristi Wong, Bryan W. K. Woo, Maddy Woodson, Cheng Xing, Z. Jamie Yao, Ping Yeh, Bicheng Ying, Juhwan Yoo, Noureldin Yosri, Grayson Young, Adam Zalcman, Yaxing Zhang, Ningfeng Zhu, Sergio Boixo, Julian Kelly, Vadim Smelyanskiy, Hartmut Neven, Dave Bacon, Zijun Chen, Paul V. Klimov, Pedram Roushan, Charles Neill, Yu Chen, and Alexis Morvan. 2024. Demonstrating dynamic surface codes. *arXiv preprint arXiv:2412.14360* (2024).
- [11] Michael Goerz and Evan McKinney. 2024. weylchamber: Python package for analyzing two-qubit gates in the Weyl chamber. <https://pypi.org/project/weylchamber/>. Python package.
- [12] Google Quantum AI. 2025. Cirq API. [https://quantumai.google/reference/python/cirq/two\\_qubit\\_matrix\\_to\\_sqrt\\_iswap\\_operations](https://quantumai.google/reference/python/cirq/two_qubit_matrix_to_sqrt_iswap_operations).
- [13] John L Hennessy and Thomas Gross. 1983. Postpass code optimization of pipeline constraints. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 5, 3 (1983), 422–448.
- [14] Cupjin Huang, Tenghui Wang, Feng Wu, Dawei Ding, Qi Ye, Linghang Kong, Fang Zhang, Xiaotong Ni, Zhijun Song, Yaoyun Shi, Hui-Hai Zhao, Chunqing Deng, and Jianxin Chen. 2023. Quantum Instruction Set Design for Performance. *Physical Review Letters* 130 (Feb 2023), 070601. Issue 7. doi:10.1103/PhysRevLett.130.070601

- [15] IBM Quantum. 2024. New fractional gates reduce circuit depth for utility-scale workloads. <https://www.ibm.com/quantum/blog/fractional-gates>. Accessed: Nov. 18, 2024.
- [16] IBM Quantum. 2025. Qiskit API. <https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.synthesis.XXDecomposer>.
- [17] IonQ. 2023. Getting started with IonQ's hardware-native gateset. <https://docs.ionq.com/guides/getting-started-with-native-gates>.
- [18] Justin Kalloor, Mathias Weiden, Ed Younis, John Kubiatowicz, Bert De Jong, and Costin Iancu. 2024. Quantum hardware roofline: Evaluating the impact of gate expressivity on quantum processor design. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1. IEEE, 805–816.
- [19] Alon Kukliansky, Ed Younis, Lukasz Cincio, and Costin Iancu. 2023. QFactor: A Domain-Specific Optimizer for Quantum Circuit Instantiation. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1. IEEE, 814–824.
- [20] Lingling Lao, Hans Van Someren, Imran Ashraf, and Carmen G Almudever. 2021. Timing and resource-aware mapping of quantum circuits to superconducting processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 41, 2 (2021), 359–371.
- [21] Ang Li, Samuel Stein, Sriram Krishnamoorthy, and James Ang. 2023. Qasmbench: A low-level quantum benchmark suite for nisq evaluation and simulation. *ACM Transactions on Quantum Computing* 4, 2 (2023), 1–26.
- [22] Gushu Li, Yufei Ding, and Yuan Xie. 2019. Tackling the qubit mapping problem for NISQ-era quantum devices. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*. 1001–1014.
- [23] Matt McEwen, Dave Bacon, and Craig Gidney. 2023. Relaxing hardware requirements for surface code circuits using time-dynamics. *Quantum* 7 (2023), 1172.
- [24] Evan McKinney, Michael Hatridge, and Alex K Jones. 2024. Mirage: Quantum circuit decomposition and routing collaborative design using mirror gates. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 704–718.
- [25] Eric C Peterson, Lev S Bishop, and Ali Javadi-Abhari. 2022. Optimal synthesis into fixed xx interactions. *Quantum* 6 (2022), 696.
- [26] Eric C Peterson, Gavin E Crooks, and Robert S Smith. 2020. Fixed-depth two-qubit circuits and the monodromy polytope. *Quantum* 4 (2020), 247.
- [27] Massimiliano Poletto and Vivek Sarkar. 1999. Linear scan register allocation. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 21, 5 (1999), 895–913.
- [28] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. 2023. MQT Bench: Benchmarking software and design automation tools for quantum computing. *Quantum* 7 (2023), 1062.
- [29] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. 2020.  $t|ket\rangle$ : a retargetable compiler for NISQ devices. *Quantum Science and Technology* 6, 1 (2020), 014003.
- [30] Bochen Tan and Jason Cong. 2020. Optimal layout synthesis for quantum computing. In *Proceedings of the 39th International Conference on Computer-Aided Design*. 1–9.
- [31] Robert R Tucci. 2005. An introduction to Cartan's KAK decomposition for QC programmers. arXiv preprint quant-ph/0507171.
- [32] Xin-Chuan Wu, Marc Grau Davis, Frederic T Chong, and Costin Iancu. 2020. QGo: Scalable quantum circuit optimization using automated synthesis. *arXiv preprint arXiv:2012.09835* (2020).
- [33] Christopher G Yale, Ashlyn D Burch, Matthew NH Chow, Brandon P Ruzic, Daniel S Lobser, Brian K McFarland, Melissa C Revelle, and Susan M Clark. 2025. Realization and calibration of continuously parameterized two-qubit gates on a trapped-ion quantum processor. *arXiv preprint arXiv:2504.06259* (2025).
- [34] Ed Younis, Costin C Iancu, Wim Lavrijsen, Marc Davis, and Ethan Smith. 2021. Berkeley Quantum Synthesis Toolkit (BQSKit). GitHub. doi:10.11578/dc.20210603.2
- [35] Ed Younis, Koushik Sen, Katherine Yelick, and Costin Iancu. 2021. Qfast: Conflating search and numerical optimization for scalable quantum circuit synthesis. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 232–243.
- [36] Chi Zhang, Ari B Hayes, Longfei Qiu, Yuwei Jin, Yanhao Chen, and Eddy Z Zhang. 2021. Time-optimal qubit mapping. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 360–374.
- [37] Jun Zhang, Jiri Vala, Shankar Sastry, and K Birgitta Whaley. 2003. Geometric theory of nonlocal two-qubit operations. *Physical Review A* 67, 4 (2003), 042313.
- [38] Runshi Zhou, Fang Zhang, Linghang Kong, and Jianxin Chen. 2024. Halma: a routing-based technique for defect mitigation in quantum error correction. *arXiv preprint arXiv:2412.21000* (2024).
- [39] Henry Zou, Matthew Treinish, Kevin Hartman, Alexander Ivrii, and Jake Lishman. 2024. Lightsabre: A lightweight and enhanced sabre algorithm. *arXiv preprint arXiv:2409.08368* (2024).
- [40] Alwin Zulehner, Alexandru Paler, and Robert Wille. 2018. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38, 7 (2018), 1226–1236.
- [41] Alwin Zulehner and Robert Wille. 2019. Compiling SU (4) quantum circuits to IBM QX architectures. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*. ACM New York, NY, USA, Tokyo, Japan, 185–190.