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

Abstract

Qubit mapping/routing is a critical compilation stage for both near-term and fault-tolerant quantum computers, yet existing scalable methods typically impose several times the routing overhead in terms of circuit depth or duration. This performance gap stems from a fundamental disconnect: compilers relies on the simplified routing model (e.g., three-CX-unrolled SWAP insertion), which fails to exploit the intrinsic properties of hardware-native quantum gates.

Recent hardware breakthroughs have also enabled high-precision implementations of diverse instruction set architectures (ISAs) beyond standard CX-based gates. Advanced ISAs such as $\sqrt{\text{iSWAP}}$ and $\text{ZZ}(\theta)$ basis gates offer superior circuit synthesis capabilities and inherent noise resilience. However, the absence of systematic compiler optimization strategies tailored to these advanced ISAs has prevented the community from leveraging their full capabilities.

To address this, we propose Canopus, a unified qubit mapping/routing framework across diverse quantum ISAs. Built upon the canonical representation of two-qubit gates, Canopus centers on qubit routing to perform deep co-optimization in an ISA-aware approach. Canopus leverages the two-qubit canonical representation and monodromy polytope theory to model the synthesis cost for more intelligent SWAP search during routing. Commutation relations between two-qubit gates can be formalized through the canonical form in our findings, providing a generalized approach to commutative optimizations. Experiments show that Canopus consistently reduces routing overhead by 30%-40% 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. 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.

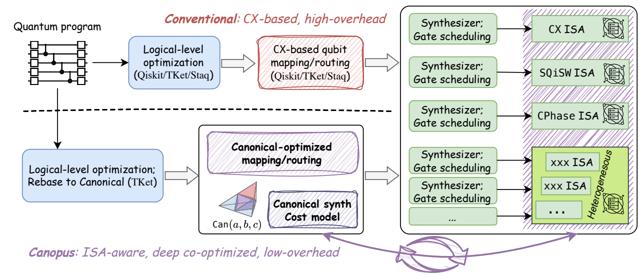


Figure 1. Compilation workflows by means of conventional approaches (top) and Canopus (bottom) targeting diverse quantum ISAs. Canopus integrates the synthesis cost model (monodromy polytopes within the Weyl chamber) by taking backend ISAs' properties into account. Canopus routing operates in the 2Q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

1 Introduction

Quantum computing is a revolutionary computational paradigm leveraging quantum mechanics principles like superposition and entanglement of qubit states [41]. It has been rapidly growing in recent decades due to its potential speedups in task such as integer factorization [52], solving linear equations [23], and microscale system simulation [38].

The holistic benchmarks of quantum computers such as quantum volume [16] are predicated on concurrent advancements in both hardware and software. Recently numerous systematic techniques regarding compiler optimization and architectural support have been presented to push the limit of hardware performance. Quantum compiler plays a pivotal role in this process, translating high-level programs into executable single-qubit (1Q) and two-qubit (2Q) gates on realistic quantum hardware. This typically involves several critical stages: (1) compiling programs into basic quantum gates, (2) performing hardware-agnostic (logical-level) circuit optimization, (3) resolve backend topology constraints via qubit placement and routing, and (4) converting circuits to native gates for final optimization and scheduling. The primary goal of compiler optimization is to lower the 2Q gate count and circuit depth other than resolving backend constraints, as 2Q gates have significantly longer duration and higher error rate than 1Q gates.

For mainstream quantum platforms like superconducting qubits [31], 2Q gates can only operate between the near-neighbor physical qubit pairs. Consequently, qubit placement and routing is crucial for resolving this connectivity constraint (e.g., Google's devices with 2D square topology [3], IBM's devices with 2D heavy-hex topology [10]), through dynamically remapping logical qubits to physical ones via inserting SWAP gates acting on adjacent physical qubit pairs. Such induced routing overhead typically increase the gate count and circuit depth by a factor of 2x-4x relative to the pre-mapping circuits when using state-of-the-art scalable routing methods [35, 37, 67, 71]. Therefore, mitigating this routing overhead remains a central and long-standing challenge in compiler optimization.

Most studies on qubit routing rely on a simplified routing model, where circuit cost is quantified by the CX-based gate count and circuit depth while each SWAP gate is unrolled into three CX gates according to the textbook pattern $\mathrm{SWAP}_{q_0,q_1} = \mathrm{CX}_{q_0,q_1}\mathrm{CX}_{q_1,q_0}\mathrm{CX}_{q_0,q_1}$. However, this CX-centric view is misaligned with the physical reality of modern quantum systems. Although quantum algorithms are typically expressed in terms of CX gates, the underlying hardware may not execute native CX-equivalent gates, nor does this gate/circuit cost quantification method accurately reflect the true operational cost. Indeed, beyond the native support for CX-equivalent gates (e.g., CZ [31], Cross-Resonance [50], Mølmer-Sørensen [7]), modern quantum hardwares increasingly feature diverse native 2Q basis gates in recent several years. These alternative basis gates, or, the abstracted instruction set architectures (ISAs) in a narrow sense, could be more powerful than CX-equivalent gates in terms of synthesis capability and noise resilience, such as $\sqrt{\mathrm{iSWAP}}$ [26], fractional iSWAP-family or CX-family gates [28, 40], and heterogeneous/combinatorial basis gates [40, 45]. With such ISAs, SWAP can be implemented with lower cost than three CX gates or even be natively realized in high fidelity [12, 63]. Therefore, the gap between the simplified routing model and properties of diverse quantum ISAs severely constrains the potential of compiler optimization, thereby limiting practical circuit execution performance. Conversely, the absence of systematic compiler optimization methods tailored for these diverse ISAs has prevented the community from fully exploiting their power and exploring the rich software-hardware co-design space they enable.

gap: at least, SWAP may not as costly as three CX gates

has prevented the community from leveraging their full capabilities and exploring cross-ISA hardware-software co-design

Consequently,

However, there are neither systematic compiler optimization strategies tailored to these advanced ISAs nor comprehensive cross-ISA evaluation to unlock their potential.

Canopus (**Can**onical-**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 [11] and $\sqrt{\mathrm{iSWAP}}$ [26], which are adaptive to versatile hardware architectures. Canopus is designed to optimize the placement of qubits and the 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 main contributions are summarized as follows:

- Canonical ... gate synthesis ... ... synthesis cost model ... monodromy polytopes
- We systematically ... unified qubit routing framework ... utilizing advanced ISAs ... demonstrate the practical superiority of advanced quantum ISAs

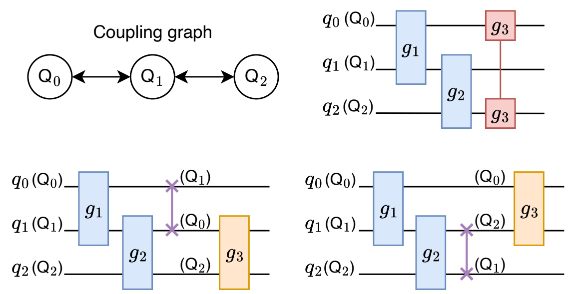


Figure 2. Mapping/routing to resolve physical-qubit topology constraints via SWAP insertion.

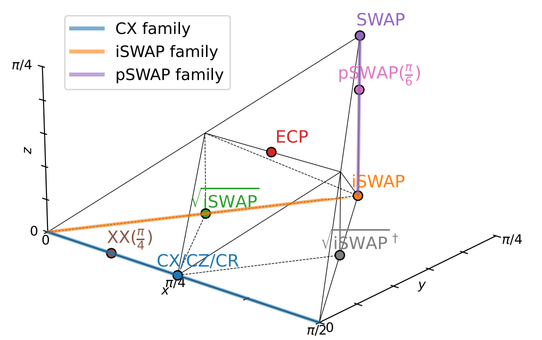


Figure 3. Geometric illustration of canonical gates confined to the Weyl chamber. [ZY: Consider $\mathrm{XX}(\pi/6)$ instead of $\mathrm{XX}(\pi/4)$.]

- Generalized commutative optimization by means of canonical representation, ... canonical gate commutation relations in our findings
- Evaluation cross-ISA, co-exploration ...
- Case studies (QFT, QEC) .... real quantum computer experiment ... FTQC

LLVM-style optimization strategy

Our framework can be extended to integrate more fine-grain hardware information such as qubit-specific basis gate fidelities.

Source code and data are available via the Anynomous Github link [4].

2 Background

[1]

2.1 Qubit mapping/routing

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

[1] For convenient visualization

2.2 Canonical description of 2Q gates

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

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

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

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 .....

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

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

2.3 Gate realization cost on hardware

Conventional iSWAP better than CZ ... however now CZ is dedicatedly implemented with better fidelity than iSWAP

SQiSW ...

Fractional ZZ gate ... IBM's Heron QPU ...

IonQ QPU ... different realization costs

The AshN gate offers optimal performance ... flux-tunable transmons ...

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

- ISA - gate duration - depth driven

Limitations of the conventional qubit routing models.

Co-optimization as the key to unlocking superiority of advanced ISAs.

[ZY: How about the SWiSQ-based compilation?]

Babel Tower dilemma for utilizing diverse ISAs.

- Formal description of 2Q gates to capture synthesis overhead / properties ?? - Formal description of synthesis cost model (monodromy polytopes)

(Coherent) Cross-ISA, topology, program pattern exploration. Vivamus vehicula leo a justo. Quisque nec augue. Morbi mauris wisi, aliquet vitae, dignissim eget, sollicitudin molestie, ligula. In dictum enim sit amet risus. Curabitur vitae velit eu diam rhoncus hendrerit. Vivamus ut elit. Praesent mattis ipsum quis turpis. Curabitur rhoncus neque eu dui. Etiam vitae magna. Nam ullamcorper. Praesent interdum bibendum magna. Quisque auctor aliquam dolor. Morbi eu lorem et est porttitor fermentum. Nunc egestas arcu at tortor varius viverra. Fusce eu nulla ut nulla interdum consectetuer. Vestibulum gravida. Morbi mattis libero sed est.

[ZY: Routing overhead is not as costly as you think]

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:

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

which involves the basic (left term) and lookahead (right term) components. In practice, there is a w_{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:

$$\begin{aligned} H =& w_g\, c_g + w_d\, \Delta_{\mathrm{depth}} \\ &+ (\Delta_{\mathrm{Avg}\{\mathrm{dist}[i,j]\}_F} + k_E\, \Delta_{\mathrm{Avg}\{\mathrm{dist}[i,j]\}_E})c_{\mathrm{swap}} \end{aligned} \quad (5)$$

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

[ZY: Now the decay component is not need any more.]

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* $\mathrm{Can}(a,b,c)_{q_0,q_1}$ *and* $\mathrm{Can}(a',b',c')_{q_1,q_2}$ *denote canonical gates acting on qubits* (q_0,q_1) *and* (q_1,q_2) *respectively, with an overlapping qubit* q_1.

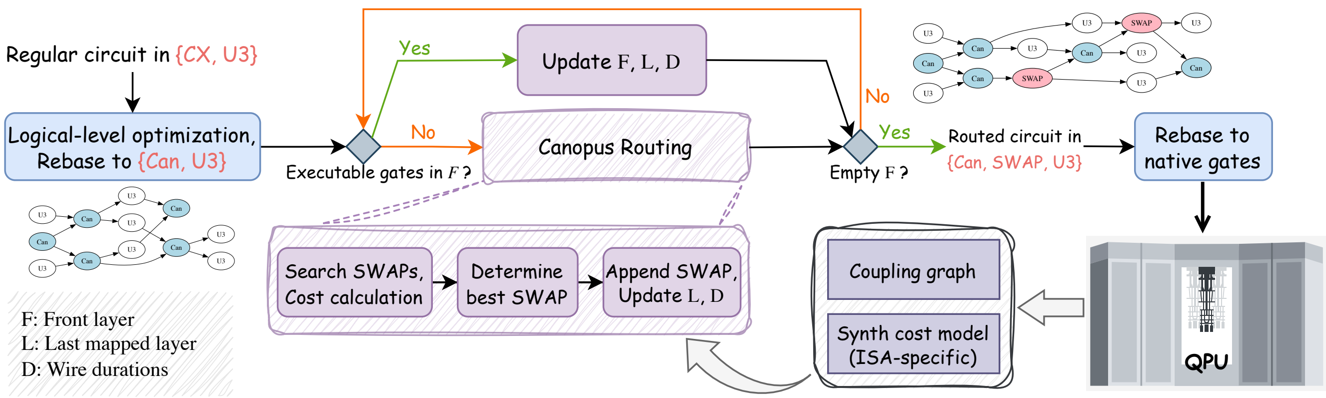
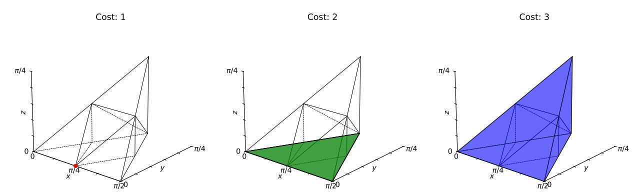
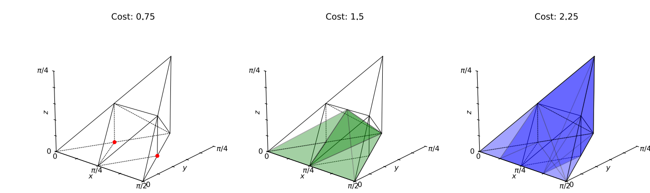


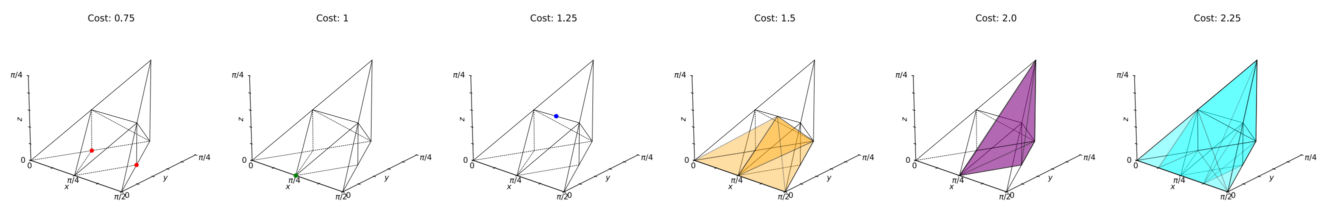
Figure 4. Overview of the Canopus framework. ...



(a) Coverage set for CX ISA.



(b) Coverage set for SQiSW ISA.



(c) Coverage set for SQiSW_ ISA.

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

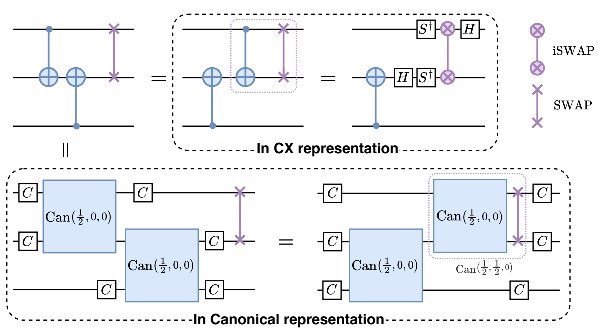


Figure 6. Canonical gate representation enables easily capturing commutative relations within real-world circuits.

They are commutative if and only if

$$b = b' = c = c' = 0, \tag{6}$$

that is, when both consist solely of XX *rotations.*

[ZY: Proposition?? Theorem?]

5 Implementation

5.1 Core functionalities

5.2 Extensions

5.3 Scalability

6 Case Studies

6.1 QFT kernel

6.2 QEC stabilizer circuit

Hardware implementation of quantum LDPC (QLDPC) codes [6, 42] remains highly challenging due to the frequent long-range interactions between qubits [5, 58]. Although emerging platforms such as neutral-atom [36, 43, 56, 57, 62] and ion-trap [7, 60] have shown greater potential for realizing

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

Input : G' (Routed DAG), π (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 */
g' ← G'.pushBack(g, π[g.q0], π[g.q1]); // g'.qi = π[g.qi]
d ← max(D[g'.q0], D[g'.q1]) + synthCost(g);
D[g'.q0] ← d; D[g'.q1] ← d;
for pred ∈ G'.predecessors(g') do
    if is2QGate(pred) then
        if isCommutativeCanonicalPair(g', pred) then
            C[(pred.q0, pred.q1)] ← (g'.q0, g'.q1);
        else
            L.pop((pred.q0, pred.q1), None);
            C.pop((pred.q0, pred.q1), None);
    else
        /* pred_pred must be None or a 2Q gate */
        pred_pred ← next(G'.predecessors(pred));
        if pred_pred ≠ None then
            L.pop((pred_pred.q0, pred_pred.q1), None);
            C.pop((pred_pred.q0, pred_pred.q1), None);
L[(g'.q0, g'.q1)] ← g';
```

Algorithm 2: Update D when adding a SWAP gate

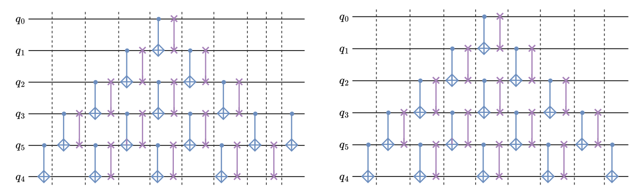
Input : swap (encountered SWAP gate), can (canonical gate within L on the same qubits as swap), D, C
Output: Updated D

```
if (swap.q0, swap.q1) ∈ C then
    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} */
    D[qi] ← D[q'j] + synthCost(can);
    D[the other swap qubit] ← D[qi];
d ← max(D[swap.q0], D[swap.q1]) +
    synthCost(can.mirror()) − synthCost(can);
D[swap.q0] ← d; D[swap.q1] ← d;
```

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	66	21
2D Square	TOQM	21	13	100	39
	Canopus	15	9	75 (±10%)	33 (±10%)

QLDPC codes with less routing overhead, Canopus shows that the long-range interaction overhead can be suppressed



(a) Mapping qft_6 by TOQM. (b) Mapping qft_6 by Canopus.

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.

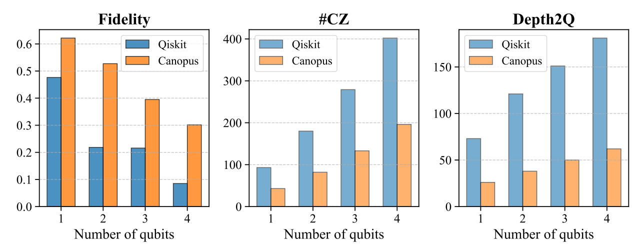


Figure 8. QFT kernel fidelity comparison benchmarked on IBM® Quantum Platform (ibm_torino).

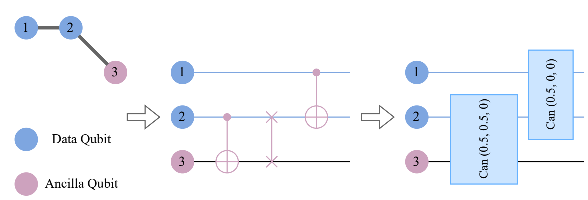
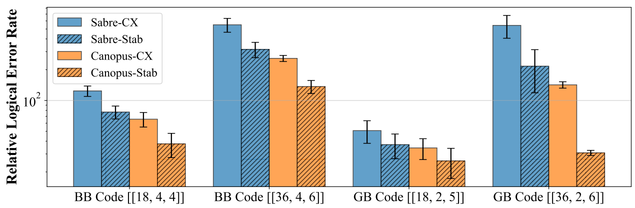


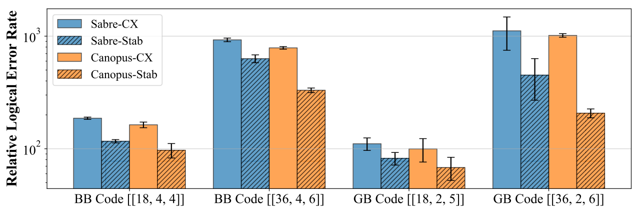
Figure 9. Stabilizer check circuit with less routing overhead.

significantly when combining the iSWAP and CX gates together — our stabilizer ISA (Stab-ISA), marking an initial attempt at realizing QLDPC codes on superconducting platforms subject to topological constraints. A similar observation was also employed in [69] to handle defects, while it relied heavily on manual design and experience.

Figure 9 shows why the Stab-ISA can provide optimization space for FTQC circuit execution: One ancilla qubit (grey vertex) needs to interact with multiple data qubits (blue vertices) and be measured in the end to get the syndrome information. Performing an iSWAP gate makes it possible to conduct one CX and SWAP gate together, since the iSWAP gate can be decomposed into a CX, a SWAP, and some single-qubit rotations. This allows us to "piggyback" a SWAP on a CX without incurring an extra two-qubit gate, enabling ancilla movement across the lattice at no additional routing



(a) Relative Logical Error Rate - Square Topo



(b) Relative Logical Error Rate - Hhex Topo

Figure 10. Logical error rate comparison for stabilizer circuits mapped on square (a) and heavy-hex (b) topologies.

cost. Therefore, *qubit 3* in Figure 9 can be switched to *position 2* "freely" without an additional SWAP gate, thus reducing the circuit depth from 3 to 2.

We also build up a fully end-to-end evaluation pipeline using `stim` [20] and `qLDPC` [44] to construct complete QLDPC code memory circuits under the circuit level noise model [1, 18] and then decode the error syndrome using the BPOSD decoder [25, 42, 51] to acquire the logical accuracy. We benchmark the circuit duration and logical error rate for QLDPC code examples chosen from [42, 58] in Figure **?? ??** 10(a) 10(b), with different kinds of ISAs including `CX`, `Stab` and hardware topologies including `Square`, `Hhex`. Note that Canopus with `Stab-ISA` achieves the lowest circuit duration and logical error rate, among all tested examples. The reductions of circuit duration and logical error further highlight the effectiveness of native compilation of iSWAP and CX gates in mitigating mapping and routing overhead for experimental QLDPC demonstrations.

7 Evaluation

We further holistically evaluate Canopus compared to other scalable SOTA methods, across representative quantum ISAs and hardware topologies. The evaluation not only provides cross-compiler but also cross-ISA comparisons under the coherent settings of basis gate cost and routing overhead metric.

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` [26]

Table 2. Selected quantum ISAs.

ISA	2Q basis gates	Description
`CX`	$\{\mathrm{CX}\}$	Conventional CX gate
`ZZPhase`	$\left\{\mathrm{ZZ}_{\frac{\pi}{6}}, \mathrm{ZZ}_{\frac{\pi}{4}}, \mathrm{ZZ}_{\frac{\pi}{2}}\right\}$	Discrete CX-family gates, i.e., $\left\{\sqrt[3]{\mathrm{CX}}, \sqrt{\mathrm{CX}}, \mathrm{CX}\right\}$ [45]
`SQiSW`	$\left\{\sqrt{\mathrm{iSWAP}}, \mathrm{iSWAP}\right\}$	Half evolution of iSWAP and iSWAP [26]
`ZZPhase_`	`ZZPhase` + $\left\{\mathrm{pSWAP}_{\frac{\pi}{6},\frac{\pi}{4},\frac{\pi}{2}}\right\}$	`ZZPhase` ISA with the mirror gates
`SQiSW_`	`SQiSW` + $\{\mathrm{ECP}, \mathrm{CX}\}$	`SQiSW` ISA with the mirror gates [40]
`Het`	`ZZPhase` + `SQiSW`	Heterogeneous CX-family and iSWAP-family gates

proves to a more powerful ISA option and has been adopted by recent software projects [22, 40]. `ZZPhase` ISA containing three fractional $\mathrm{ZZ}(\theta)$ rotation gates is adopted by Qiskit's latest synthesis functionalities [28, 45]. Mirorr .... [ZY: TODO: find the initial paper about mirror gate]

[40]

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:

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

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

[ZY: Explain why we assume such a basis gate cost settings]

7.1.2 Benchmarks. We select a set of twelve medium-size benchmarks from QASMBench [34] and MQTBench [49] spanning various categories of quantum programs. These benchmarks first go through logical-level optimization by TKet [53] and 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 C_{count} and C_{depth} denote costs of the total gate count and circuit duration, respectively, 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 Suppression of routing overhead

7.3 Mirroring and combination effects for ISA design

7.4 Co-exploration of routing and ISA selection

The real-machine experiment in Section 6.1 showcases how our method can help achieve superior compilation results and thus higher program fidelities for QFT kernels using the standard `CX` ISA (CZ on `ibm_torino`). However, extending this validation to alternative ISAs is currently challenging due to the scarcity of quantum processors equipped with well-calibrated heterogeneous gate sets. For instance, while IBM has proposed fractional gates, i.e., the continuous $\mathrm{ZZ}(\theta)$

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 (C_{count} and C_{depth}) is calculated in CX ISA.

Program	#Qubit	#Can	Depth2Q	C_{count}	C_{depth}
bigadder [34]	18	114	79	130.0	88.0
bv [34]	19	18	18	18.0	18.0
ising [34]	26	25	2	50.0	4.0
knn [34]	25	72	50	84.0	62.0
multiplier [34]	15	198	122	222.0	133.0
qec9xz [34]	17	32	12	32.0	12.0
qft [49]	18	153	33	306.0	66.0
qpeexact [49]	16	127	43	260.0	86.0
qram [34]	20	110	70	130.0	78.0
sat [34]	11	210	182	252.0	204.0
swap_test [34]	25	72	50	84.0	62.0
wstate [34]	27	52	28	52.0	28.0

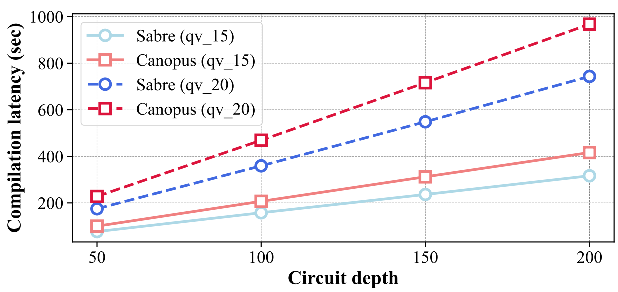


Figure 11. Compilation latency comparison.

gate set [27], their implementation details and calibration procedures are not publicly disclosed. To our knowledge, the ZZ(θ) gates have the same duration as CZ on IBM's Heron QPUs regardless of the rotation angle, and their error rates are consistently 1x-3x that of CZ. This performance is far from the ideal assumptions of ZZPhase. Fortunately, a path forward is emerging with the recently proposed AshN gate scheme [11] that enables directly implementing any basis gates with the optimal gate durations. It is also experimentally evinced on transmon qubits by Chen et al. [12], where multiple basis gates are calibrated in high fidelities that aligns with our cost model as well. This development may enable comprehensive, real-machine co-exploration of programs, ISAs, and hardware topologies in the near future.

7.5 Breakdown analysis

In this section, we analyze individual factors in the improvement brought by CANOPUS, mainly about the commutative optimization mechanism.

Note that the ...

7.6 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 volume [16] circuits generated by QISKIT for scalability benchmarking, which represents the end of the spectrum with respect to canonical-form circuits [ZY: Why the end of the spectrum]. 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. We select quantum volume circuits with two different widths (number of qubits), 15 and 20. We vary the depth of these circuits (`qv_15` and `qv_20`) 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 11 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 (±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_15` (`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.

7.7 Diverse-ISA compilation paradigms

hete-ISA

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 [14, 24] and register allocation [9, 47] 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. [71] 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. [35] also utilizes the circuit DAG layering thought to tackle the qubit mapping problem and proposes the bidirectional routing

Table 4. Routing overhead C_{count} for different compilers across different topologies and quantum ISAs.

Chain	CX ISA				ZZPhase ISA				SQiSW ISA				ZZPhase_ ISA				SQiSW_ ISA				Het ISA			
Bench	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop
bigadd	2.53	2.44	1.59	1.92	2.35	2.26	1.53	1.97	2.39	2.23	1.42	1.98	1.90	1.81	1.57	1.59	1.94	1.85	1.49	1.57	1.95	1.85	1.31	1.77
bv	2.67	4.06	10.94	2.00	2.67	4.06	10.94	2.33	2.38	3.12	8.12	1.88	2.09	3.01	8.12	1.63	2.03	2.89	7.22	1.61	2.12	3.07	7.76	1.79
ising	1.00	1.00	1.00	1.00	0.38	0.38	0.38	0.38	0.75	0.75	0.75	0.75	0.38	0.38	0.38	0.38	0.75	0.75	0.75	0.75	0.38	0.38	0.38	0.38
knn	2.60	4.02	1.48	1.29	2.40	3.93	1.22	1.23	2.43	3.46	1.21	1.39	1.93	3.01	1.13	1.04	1.98	2.90	1.06	1.07	1.98	3.11	1.01	1.08
multi	2.32	4.97	2.53	2.68	2.18	4.83	2.28	2.49	2.26	4.17	1.99	2.38	1.79	3.67	1.84	2.02	1.81	3.56	1.73	2.04	1.83	3.78	1.68	2.01
qec9	4.44	12.34	6.88	3.56	4.44	12.34	5.33	3.53	3.89	9.52	3.47	2.84	3.43	9.05	5.23	2.77	3.25	8.45	3.98	2.56	3.52	9.34	4.27	2.77
qft	1.74	1.50	1.78	1.49	1.51	1.45	2.02	1.45	1.31	1.12	1.53	1.12	1.12	1.05	1.50	1.05	1.19	1.00	1.32	1.00	1.16	1.10	1.41	1.10
qpe	2.77	3.32	3.15	2.86	2.46	3.09	2.89	2.75	2.08	2.50	2.23	2.13	1.82	2.27	2.07	1.99	1.89	2.24	2.13	1.88	1.89	2.36	2.35	2.04
qram	2.94	5.37	2.75	3.23	2.75	5.21	2.73	3.02	2.63	4.44	2.53	2.80	2.16	3.93	2.29	2.45	2.19	3.80	2.22	2.37	2.22	4.06	2.26	2.43
sat	2.44	2.66	1.88	2.29	2.23	2.43	1.38	2.13	2.24	2.36	1.42	2.03	1.79	1.92	1.39	1.67	1.85	1.99	1.32	1.76	1.83	1.96	1.13	1.73
swapt	2.87	4.02	1.43	1.29	2.67	3.93	1.22	1.23	2.66	3.46	1.21	1.39	2.13	3.01	1.02	1.10	2.17	2.90	1.07	1.07	2.19	3.11	1.00	1.08
wstate	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.50	1.50	1.47	1.50	1.00	1.00	0.99	1.00	1.00	1.00	0.99	1.00	1.00	1.00	0.99	1.00
Avg.	2.26	3.07	2.27	1.88	1.97	2.75	1.92	1.7	2.06	2.63	1.85	1.73	1.61	2.18	1.69	1.39	1.72	2.25	1.68	1.45	1.65	2.23	1.58	1.43

HHex	CX ISA				ZZPhase ISA				SQiSW ISA				ZZPhase_ ISA				SQiSW_ ISA				Het ISA			
Bench	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop
bigadd	2.17	2.10	1.90	1.89	2.00	1.93	2.35	1.92	2.14	2.02	1.86	1.84	1.65	1.58	1.24	1.45	1.70	1.63	1.57	1.45	1.69	1.61	1.33	1.53
bv	3.28	2.22	7.22	1.94	3.28	2.22	7.23	1.83	3.00	2.12	5.83	1.50	2.58	1.80	5.51	1.51	2.47	1.75	4.94	1.42	2.64	1.82	5.35	1.68
ising	1.72	3.20	1.64	1.42	1.10	2.77	0.83	0.83	1.29	2.50	1.17	1.11	0.90	2.12	0.84	0.67	1.23	2.24	1.15	1.04	0.92	2.18	0.86	0.58
knn	2.18	2.57	2.17	1.49	1.98	2.43	1.81	1.39	2.12	2.33	1.91	1.54	1.63	1.92	1.61	1.16	1.70	1.93	1.66	1.25	1.66	1.97	1.75	1.17
multi	2.23	3.48	2.11	2.24	2.09	3.35	1.69	2.10	2.19	3.07	1.95	2.00	1.72	2.61	1.64	1.64	1.75	2.57	1.70	1.75	1.75	2.68	1.53	1.67
qec9	3.16	4.78	4.84	3.16	3.16	4.78	4.82	3.19	2.91	4.03	4.20	2.84	2.49	3.64	3.81	2.43	2.39	3.45	4.94	2.39	2.55	3.73	5.23	2.53
qft	1.91	2.62	2.44	1.67	1.60	2.35	1.83	1.52	1.43	1.97	1.61	1.27	1.19	1.73	1.50	1.12	1.31	1.78	1.46	1.12	1.24	1.80	1.59	1.16
qpe	2.58	2.90	2.89	2.43	2.15	2.59	2.44	2.10	1.94	2.18	2.30	1.86	1.61	1.92	1.96	1.62	1.77	1.97	1.86	1.68	1.67	1.99	1.82	1.67
qram	2.52	4.32	3.03	2.42	2.32	4.15	3.23	2.31	2.35	3.68	2.52	2.19	1.87	3.17	2.58	1.86	1.92	3.11	2.18	1.85	1.91	3.27	1.94	1.87
sat	2.27	2.29	1.60	2.00	2.05	2.06	1.28	1.81	2.10	2.10	1.35	1.83	1.65	1.66	1.33	1.44	1.74	1.74	1.26	1.52	1.69	1.69	1.03	1.49
swapt	2.24	2.57	2.06	1.50	2.04	2.43	1.78	1.42	2.18	2.33	1.98	1.56	1.68	1.92	1.63	1.14	1.74	1.93	2.20	1.21	1.71	1.97	1.82	1.18
wstate	2.65	2.04	2.60	1.69	2.65	2.04	2.47	1.46	2.71	2.28	2.21	1.67	2.19	1.75	2.07	1.50	2.10	1.69	1.78	1.35	2.23	1.78	1.97	1.60
Avg.	2.37	2.82	2.59	1.93	2.12	2.65	2.25	1.74	2.14	2.48	2.17	1.72	1.7	2.08	1.88	1.4	1.78	2.09	1.98	1.46	1.74	2.13	1.86	1.43

Square	CX ISA				ZZPhase ISA				SQiSW ISA				ZZPhase_ ISA				SQiSW_ ISA				Het ISA			
Bench	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop
bigadd	1.62	1.89	1.41	1.38	1.44	1.71	1.02	1.18	1.75	1.92	1.42	1.43	1.26	1.44	1.17	1.04	1.34	1.51	1.23	1.14	1.27	1.46	1.10	1.02
bv	2.72	2.22	4.39	1.50	2.72	2.22	4.39	1.50	2.75	2.38	3.67	1.50	2.23	1.87	3.44	1.22	2.14	1.81	2.94	1.31	2.28	1.90	3.17	1.22
ising	1.00	1.00	1.78	1.00	0.38	0.38	1.16	0.38	0.75	0.75	1.33	0.75	0.38	0.38	0.90	0.38	0.75	0.75	1.27	0.75	0.38	0.38	0.96	0.38
knn	1.79	2.64	1.88	1.29	1.57	2.43	1.52	1.23	1.88	2.46	1.59	1.42	1.35	1.96	1.40	1.04	1.45	2.01	1.46	1.07	1.38	2.00	1.28	1.05
multi	1.84	2.81	1.99	1.56	1.68	2.67	1.49	1.42	1.95	2.64	1.64	1.58	1.44	2.14	1.32	1.24	1.50	2.14	1.64	1.28	1.47	2.19	1.21	1.26
qec9	2.06	4.44	3.69	1.78	2.06	4.44	3.80	1.72	2.20	3.89	2.88	1.71	1.74	3.43	2.30	1.44	1.69	3.25	2.34	1.50	1.77	3.52	2.42	1.46
qft	1.41	2.30	2.09	1.35	1.03	1.88	1.41	1.05	1.06	1.75	1.50	1.00	0.79	1.42	1.08	0.80	0.99	1.60	1.37	0.94	0.82	1.47	1.06	0.81
qpe	1.68	2.50	2.06	1.46	1.22	2.00	1.42	1.18	1.26	1.91	1.47	1.11	0.95	1.53	1.08	0.93	1.18	1.74	1.48	0.99	0.98	1.58	1.26	0.90
qram	1.88	2.67	2.40	1.60	1.65	2.48	1.78	1.44	1.88	2.48	2.00	1.55	1.39	1.99	1.70	1.22	1.50	2.02	1.74	1.27	1.41	2.04	1.47	1.29
sat	1.65	2.09	1.44	1.54	1.42	1.87	1.22	1.34	1.70	2.06	1.25	1.45	1.22	1.55	1.11	1.11	1.34	1.64	1.14	1.21	1.24	1.58	0.94	1.12
swapt	1.75	2.64	2.06	1.29	1.54	2.43	1.14	1.18	1.85	2.46	1.46	1.44	1.33	1.96	1.38	1.04	1.43	2.01	1.47	1.11	1.35	2.00	1.33	1.07
wstate	1.00	1.00	1.25	1.00	1.00	1.00	1.39	1.00	1.50	1.50	1.82	1.50	1.00	1.00	1.53	1.00	1.00	1.00	1.31	1.00	1.00	1.00	1.32	1.00
Avg.	1.64	2.18	2.06	1.38	1.35	1.87	1.61	1.16	1.63	2.05	1.74	1.34	1.16	1.55	1.43	0.99	1.31	1.69	1.56	1.11	1.18	1.58	1.36	1.0

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 [2, 33, 70], or attempt to minimize circuit depth via graph matching [13]. Zhang et al. [67] 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 [54]. However, the optimality of qubit routing is a complex task. There are rarely theoretical studies that claims the holistic optimality of some SWAP insertion strategy 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 [67] 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 [27], ion-trapped partial entangling gates [29, 63], and the AshN gates [11, 12], some works began exploring how to efficiently utilize these ISAs to make compiler optimizations closer to hardware characteristics. McKinney et al. [40] investigates the practical performance of SQiSW ISA proposed by Huang et al. [26] and the synthesis capability when incorporating the basis gates' mirrors into the ISA. Their modified Sabre algorithm 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 [65] and the series of works behind it [17, 32, 61, 66] 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

Table 5. Routing overhead C_{depth} for different compilers across different topologies and quantum ISAs.

Chain	CX ISA				ZZPhase ISA				SQiSW ISA				ZZPhase_ ISA				SQiSW_ ISA				Het ISA			
Bench	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop
bigadd	2.82	1.95	1.45	1.73	2.66	1.81	1.27	1.81	2.65	1.88	1.40	1.88	2.14	1.48	1.16	1.48	2.15	1.53	1.30	1.48	2.19	1.51	1.05	1.46
bv	2.83	2.72	5.06	2.11	2.83	2.72	5.06	2.39	2.62	2.12	4.12	2.12	2.26	2.01	3.73	1.74	2.19	1.89	3.58	1.72	2.29	2.07	3.99	1.85
ising	1.00	1.00	1.00	1.00	0.46	0.46	0.46	0.46	0.75	0.75	0.75	0.75	0.46	0.46	0.46	0.46	0.75	0.75	0.75	0.75	0.46	0.46	0.46	0.46
knn	3.16	2.66	1.65	1.39	2.90	2.53	1.45	1.31	2.76	2.32	1.35	1.35	2.27	1.95	1.32	1.05	2.32	1.95	1.17	1.10	2.33	2.03	1.12	1.11
multi	2.33	3.49	2.45	2.17	2.23	3.41	2.14	1.93	2.28	3.06	1.88	1.96	1.82	2.64	1.89	1.64	1.82	2.56	1.57	1.59	1.86	2.71	1.58	1.58
qec9	5.33	7.33	6.00	4.08	5.33	7.33	5.58	3.83	4.38	5.75	4.00	3.38	4.00	5.40	4.70	3.36	3.75	5.04	4.33	2.79	4.12	5.58	4.75	3.29
qft	2.85	1.50	2.50	1.50	1.95	1.43	3.02	1.42	2.14	1.12	2.24	1.12	1.53	1.04	2.27	1.03	2.01	1.01	2.02	1.01	1.58	1.08	2.29	1.08
qpe	4.00	2.97	3.66	2.83	3.32	2.69	3.67	2.77	3.00	2.23	2.73	2.12	2.49	1.99	2.62	1.83	2.75	2.01	2.64	1.90	2.58	2.06	2.70	1.71
qram	2.94	3.79	2.45	2.50	2.80	3.73	2.24	2.24	2.64	3.21	2.16	2.28	2.21	2.84	2.03	1.90	2.19	2.72	2.03	1.91	2.27	2.93	1.76	1.94
sat	2.28	2.00	1.77	1.88	2.12	1.85	1.25	1.64	2.19	1.87	1.30	1.75	1.73	1.50	1.36	1.37	1.77	1.54	1.20	1.48	1.77	1.52	1.06	1.38
swapt	3.42	2.66	1.58	1.39	3.15	2.53	1.44	1.31	2.98	2.32	1.37	1.35	2.45	1.95	1.12	1.13	2.50	1.95	1.16	1.10	2.52	2.03	1.11	1.11
wstate	1.00	1.00	1.04	1.00	1.00	1.00	1.02	1.00	1.50	1.50	1.50	1.50	1.00	1.00	1.02	1.00	1.00	1.00	1.02	1.00	1.00	1.00	1.01	1.00
Avg.	2.57	2.38	2.18	1.81	2.22	2.15	1.91	1.63	2.32	2.08	1.84	1.68	1.82	1.72	1.66	1.35	1.95	1.76	1.66	1.4	1.86	1.76	1.56	1.36

HHex	CX ISA				ZZPhase ISA				SQiSW ISA				ZZPhase_ ISA				SQiSW_ ISA				Het ISA			
Bench	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop
bigadd	2.55	1.77	1.81	1.80	2.39	1.62	2.12	1.80	2.46	1.78	1.59	1.69	1.95	1.37	1.32	1.32	1.97	1.43	1.56	1.35	1.99	1.38	1.19	1.34
bv	3.06	2.11	3.28	2.00	3.06	2.11	3.28	1.89	2.79	1.96	2.79	1.58	2.41	1.69	2.51	1.56	2.31	1.64	2.33	1.53	2.46	1.71	2.52	1.85
ising	4.50	6.25	4.25	3.00	3.58	5.92	2.71	2.12	3.38	5.25	2.44	2.62	2.75	4.54	2.42	1.60	3.12	4.50	2.88	2.62	2.83	4.67	2.44	1.19
knn	2.52	2.19	1.94	1.48	2.24	2.01	1.68	1.35	2.30	2.02	1.90	1.56	1.80	1.60	1.49	1.09	1.90	1.66	1.69	1.30	1.84	1.65	1.56	1.19
multi	2.14	2.76	1.88	1.92	2.05	2.69	1.50	1.80	2.11	2.51	1.82	1.93	1.68	2.12	1.57	1.54	1.68	2.06	1.59	1.75	1.71	2.17	1.41	1.57
qec9	4.50	3.83	4.25	3.25	4.50	3.83	4.29	3.42	4.06	3.38	5.25	3.06	3.51	2.96	3.29	2.64	3.33	2.83	5.38	2.75	3.60	3.02	5.77	3.08
qft	3.56	3.62	3.80	2.42	2.87	3.18	2.78	2.34	2.67	2.73	2.44	1.78	2.16	2.36	2.34	1.84	2.46	2.48	2.43	1.51	2.23	2.44	2.55	1.85
qpe	4.53	2.98	4.13	2.97	3.67	2.64	3.37	2.52	3.41	2.26	3.43	2.48	2.77	1.97	2.76	1.83	3.14	2.03	2.49	2.38	2.87	2.04	2.29	1.90
qram	2.86	3.01	2.87	2.17	2.67	2.84	2.44	1.92	2.66	2.71	2.16	2.04	2.15	2.25	2.26	1.74	2.17	2.24	2.02	1.62	2.20	2.30	2.02	1.80
sat	2.18	1.84	1.48	1.77	2.03	1.69	1.17	1.62	2.09	1.81	1.28	1.68	1.65	1.40	1.27	1.33	1.70	1.46	1.18	1.39	1.69	1.42	0.95	1.36
swapt	2.47	2.19	2.00	1.53	2.19	2.01	1.64	1.34	2.25	2.02	1.79	1.58	1.76	1.60	1.51	1.13	1.86	1.66	1.77	1.17	1.80	1.65	1.68	1.21
wstate	3.07	1.93	2.43	1.57	3.07	1.93	2.02	1.57	3.03	2.25	2.41	2.04	2.49	1.69	1.86	1.65	2.38	1.64	1.89	1.36	2.54	1.71	2.07	1.42
Avg.	3.05	2.68	2.66	2.08	2.77	2.52	2.26	1.91	2.71	2.43	2.28	1.96	2.2	2.0	1.96	1.56	2.27	2.02	2.1	1.66	2.25	2.05	1.98	1.58

Square	CX ISA				ZZPhase ISA				SQiSW ISA				ZZPhase_ ISA				SQiSW_ ISA				Het ISA			
Bench	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop	sabre	toqm	bqskit	canop
bigadd	1.90	1.48	1.28	1.33	1.74	1.32	1.04	1.16	2.00	1.60	1.18	1.46	1.49	1.16	1.01	1.04	1.55	1.23	1.14	1.09	1.51	1.17	0.98	1.01
bv	2.44	1.89	3.00	1.56	2.44	1.89	3.00	1.61	2.46	2.04	2.83	1.50	2.00	1.58	2.42	1.22	1.92	1.53	2.22	1.25	2.04	1.61	2.36	1.22
ising	1.00	1.00	8.00	1.00	0.46	0.46	6.75	0.46	0.75	0.75	4.31	0.75	0.46	0.46	2.67	0.46	0.75	0.75	3.62	0.75	0.46	0.46	3.56	0.46
knn	1.98	2.11	1.90	1.39	1.69	1.85	1.48	1.16	1.92	1.96	1.62	1.32	1.41	1.50	1.36	1.01	1.55	1.62	1.52	1.10	1.44	1.53	1.27	1.02
multi	1.86	1.98	1.94	1.57	1.73	1.92	1.48	1.43	1.98	1.99	1.52	1.71	1.48	1.58	1.30	1.28	1.52	1.56	1.55	1.23	1.51	1.61	1.18	1.23
qec9	3.00	3.58	3.67	1.92	3.00	3.58	3.67	1.83	3.06	3.38	3.31	1.81	2.47	2.85	2.71	1.51	2.38	2.71	2.75	1.50	2.52	2.92	2.87	1.62
qft	3.09	2.88	4.41	2.15	2.22	2.32	2.98	1.54	2.32	2.20	3.14	1.57	1.72	1.75	2.11	1.45	2.17	2.02	3.05	1.91	1.78	1.81	2.03	1.37
qpe	2.80	2.50	3.02	2.22	2.10	1.95	2.04	1.61	2.10	1.90	2.16	1.49	1.62	1.49	1.58	1.18	1.95	1.76	2.08	1.46	1.67	1.54	1.62	1.24
qram	2.06	1.83	2.35	1.35	1.88	1.71	1.58	1.28	2.03	1.86	1.83	1.55	1.56	1.42	1.56	1.28	1.63	1.46	1.71	1.21	1.59	1.45	1.34	1.23
sat	1.57	1.55	1.44	1.50	1.41	1.42	1.16	1.32	1.68	1.68	1.22	1.45	1.22	1.22	1.09	1.06	1.30	1.28	1.11	1.20	1.24	1.24	0.86	1.12
swapt	1.97	2.11	1.98	1.39	1.68	1.85	1.23	1.24	1.94	1.96	1.49	1.31	1.41	1.50	1.27	1.05	1.55	1.62	1.54	1.08	1.44	1.53	1.22	1.00
wstate	1.00	1.00	1.36	1.00	1.00	1.00	1.23	1.00	1.50	1.50	1.79	1.50	1.00	1.00	1.85	1.00	1.00	1.00	1.29	1.00	1.00	1.00	1.32	1.00
Avg.	1.94	1.87	2.47	1.49	1.63	1.61	1.94	1.24	1.89	1.81	2.02	1.42	1.39	1.36	1.65	1.09	1.54	1.47	1.83	1.2	1.41	1.38	1.56	1.09

synthesis by BQSKit does not ensure optimal schemes for two-qubit and multi-qubit circuit synthesis. In addition, due to the lack of native compilation strategies and rational synthesis cost model, Kalloor et al. [30] claims that alternative ISAs are hardly comparable to CX when evaluating quantum hardware roofline by BQSKit. As for applicability of expanded ISAs to QEC, Google's latest theoretical [39] and experimental [19] works demonstrate the CX-iSWAP combination ISA could benefits suppressing fault-tolerant threshold. Zhou et al. [69] proposes a routing-based method enhanced by CX-iSWAP for overcoming ancilla defects among surface code blocks while preserving encoded logical information, while relying on manual design and experience.

9 Conclusion

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

References

[1] Rajeev Acharya, I. Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham T. Asfaw, Juan Atalaya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, João Marcos Vensi Basso, Andreas Bengtsson, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, Leon Brill, Mick Broughton, Bob B Buckley, David A. Buell, Tim Burger, Brian Burkett, Nicholas Bushnell, Yu Chen, Zijun Chen, Benjamin Chiaro, J. Zachery Cogan, Roberto Collins, P. N. Conner, William Courtney, Alexander L. Crook, B Curtin, Dripto M. Debroy, A Del Toro Barba, Sean Demura, Andrew Dunsworth, Daniel Eppens, Catherine Erickson, Lara Faoro, Edward Farhi, Reza Fatemi, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, William Giang, Craig Gidney, Dar Gilboa, Marissa Giustina, Alejandro Grajales Dau, Jonathan A. Gross, S. Habegger, Michael C. Hamilton, Matthew P. Harrigan, Sean D. Harrington, Oscar Higgott, Jeremy P. Hilton, Michael J. Hoffmann, Sabrina Hong, Trent Huang, Ashley Huff, William J. Huggins, L. B. Ioffe, Sergei V. Isakov, Justin Iveland, Evan Jeffrey, Zhang Jiang, Cody Jones, Pavol Juhás, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Tanuj Khattar, Mostafa Khezri, M'aria Kieferov'a, Seon Kim, Alexei Kitaev, Paul V. Klimov, Andrey R. Klots, Alexander N. Korotkov, Fedor Kostritsa, John Mark Kreikebaum, David Landhuis, Pavel Laptev, Kim Ming Lau, Lily Laws, Joonho Lee, Kenny Lee, Brian J. Lester, Alexander T Lill, Wayne Liu, Aditya

Table 6. Routing overhead improvement analysis for Canopus relative to the routing process without commutative optimization (no_comm). *Avg.* in the table indicates the relative reduction of geometric-mean C_{count} or C_{depth} across all benchmarks; *Max.* indicates the maximum reduction achieved on one of benchmarks.

C_{count} improv. v.s. no_comm	Chain		HHex		Square	
	Avg.	*Max.*	*Avg.*	*Max.*	*Avg.*	*Max.*
CX	-10.56%	-37.57%	-0.77%	-12.35%	-4.1%	-20.59%
ZZPhase	-4.31%	-34.81%	-8.44%	-35.29%	-2.51%	-15.62%
SQiSW	-5.81%	-30.97%	-6.13%	-42.86%	-4.82%	-20.0%
ZZPhase_	0.04%	-5.38%	-5.44%	-26.58%	-2.56%	-8.0%
SQiSW_	-2.88%	-12.12%	-5.9%	-27.14%	-2.86%	-11.86%
Het	-3.59%	-26.67%	-8.74%	-47.92%	-3.59%	-18.52%

C_{depth} improv. v.s. no_comm	Chain		HHex		Square	
	Avg.	*Max.*	*Avg.*	*Max.*	*Avg.*	*Max.*
CX	-9.15%	-38.57%	-1.99%	-20.0%	-1.88%	-10.13%
ZZPhase	-4.76%	-40.44%	-10.61%	-31.08%	0.16%	-12.89%
SQiSW	-3.26%	-31.71%	1.14%	-29.63%	-2.16%	-13.75%
ZZPhase_	0.94%	-6.4%	-4.04%	-25.96%	-2.81%	-27.81%
SQiSW_	-1.5%	-11.43%	-1.45%	-17.91%	-1.82%	-7.69%
Het	-5.12%	-32.43%	-4.84%	-48.65%	-1.61%	-14.87%

Locharla, Erik Lucero, Fionn D. Malone, Jeffrey Marshall, Orion Martin, Jarrod R. McClean, Trevor McCourt, Matthew J. McEwen, Anthony Megrant, Bernardo Meurer Costa, Xiao Mi, Kevin C. Miao, Masoud Mohseni, Shirin Montazeri, Alexis Morvan, Emily Mount, Wojciech Mruczkiewicz, Ofer Naaman, Matthew Neeley, Charles J. Neill, Ani Nersisyan, Hartmut Neven, Michael Newman, Jiun How Ng, Anthony Nguyen, Murray L. Nguyen, Murphy Yuezhen Niu, Thomas E. O'Brien, Alexander Opremcak, J. Platt, Andre Petukhov, Rebecca Potter, Leonid P. Pryadko, Chris Quintana, Pedram Roushan, Nicholas C. Rubin, Negar Saei, Daniel Thomas Sank, K Sankaragomathi, Kevin J. Satzinger, Henry F. Schurkus, C. Schuster, Michael Shearn, Aaron Shorter, Vladimir Shvarts, Jindra Skruzny, Vadim N. Smelyanskiy, William C. Smith, George Sterling, Doug Strain, Yuan Su, Marco Szalay, Alfredo Torres, Guifre Vidal, Benjamin Villalonga, Catherine Vollgraff Heidweiller, Theodore White, Chen Xing, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Grayson Young, Adam Zalcman, Yaxing Zhang, and Ningfeng Zhu. 2022. Suppressing quantum errors by scaling a surface code logical qubit. *Nature* 614 (2022), 676 – 681.

[2] 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*.

[3] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[4] Anonymous authors. 2025. Canopus GitHub repo. https://anonymous.4open.science/r/canopus-asplos2026-8B43.

[5] Sergey Bravyi, Andrew W Cross, Jay M Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J Yoder. 2024. High-threshold and low-overhead fault-tolerant quantum memory. *Nature* 627, 8005 (2024), 778–782.

[6] Nikolas P Breuckmann and Jens Niklas Eberhardt. 2021. Quantum low-density parity-check codes. *PRX Quantum* 2, 4 (2021), 040101.

[7] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. 2019. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews* 6, 2 (2019), 021314.

[8] 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.

[9] Gregory J Chaitin. 1982. Register allocation & spilling via graph coloring. *ACM Sigplan Notices* 17, 6 (1982), 98–101.

[10] Christopher Chamberland, Guanyu Zhu, Theodore J Yoder, Jared B Hertzberg, and Andrew W Cross. 2020. Topological and subsystem codes on low-degree graphs with flag qubits. *Physical Review X* 10, 1 (2020), 011022.

[11] 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.

[12] Zhen Chen, Weiyang Liu, Yanjun Ma, Weijie Sun, Ruixia Wang, He Wang, Huikai Xu, Guangming Xue, Haisheng Yan, Zhen Yang, Jiayu Ding, Yang Gao, Feiyu Li, Yujia Zhang, Zikang Zhang, Yirong Jin, Haifeng Yu, Jianxin Chen, and Fei Yan. 2025. Efficient implementation of arbitrary two-qubit gates using unified control. *Nature Physics* (15 Aug 2025). doi:10.1038/s41567-025-02990-x

[13] Andrew M Childs, Eddie Schoute, and Cem M Unsal. 2019. Circuit transformations for quantum architectures. *arXiv preprint arXiv:1902.09102* (2019).

[14] 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.

[15] Gavin E Crooks. 2020. Gates, states, and circuits. Available at https://threeplusone.com/pubs/on-gates-v0-5/.

[16] 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.

[17] 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.

[18] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. 2002. Topological quantum memory. *J. Math. Phys.* 43, 9 (2002), 4452–4505.

[19] 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 Buchea, 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 Kobrin, 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, Eliott Rosenberg, Elizabeth Rossi, Kannan Sankaragomathi, Henry F. Schurkus, Michael J. Shearn, Aaron Shorter, Noah Shutty, 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 Vdovichev, 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).

[20] Craig Gidney. 2021. Stim: a fast stabilizer circuit simulator. *Quantum* 5 (2021), 497. https://api.semanticscholar.org/CorpusID:232104816

[21] 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.

[22] Google Quantum AI. 2025. Cirq API. https://quantumai.google/reference/python/cirq/two_qubit_matrix_to_sqrt_iswap_operations.

[23] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. 2009. Quantum algorithm for linear systems of equations. *Physical review letters* 103, 15 (2009), 150502.

[24] 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.

[25] Timo Hillmann, Lucas Berent, Armanda O Quintavalle, Jens Eisert, Robert Wille, and Joschka Roffe. 2024. Localized statistics decoding: A parallel decoding algorithm for quantum low-density parity-check codes. *arXiv preprint arXiv:2406.18655* (2024).

[26] 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

[27] 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.

[28] IBM Quantum. 2025. Qiskit API. https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.synthesis.XXDecomposer.

[29] IonQ. 2023. Getting started with IonQ's hardware-native gateset. https://docs.ionq.com/guides/getting-started-with-native-gates.

[30] 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.

[31] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. 2019. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews* 6, 2 (2019), 021318.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] Wan-Hsuan Lin, Daniel Bochen Tan, and Jason Cong. 2025. Reuse-aware compilation for zoned quantum architectures based on neutral atoms. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 127–142.

[37] Ji Liu, Ed Younis, Mathias Weiden, Paul Hovland, John Kubiatowicz, and Costin Iancu. 2023. Tackling the qubit mapping problem with permutation-aware synthesis. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1. IEEE, 745–756.

[38] Seth Lloyd. 1996. Universal quantum simulators. *Science* 273, 5278 (1996), 1073–1078.

[39] Matt McEwen, Dave Bacon, and Craig Gidney. 2023. Relaxing hardware requirements for surface code circuits using time-dynamics. *Quantum* 7 (2023), 1172.

[40] 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.

[41] Michael A Nielsen and Isaac L Chuang. 2010. *Quantum computation and quantum information*. Cambridge university press.

[42] Pavel Panteleev and Gleb Kalachev. 2021. Degenerate quantum LDPC codes with good finite length performance. *Quantum* 5 (2021), 585.

[43] Laura Pecorari, Sven Jandura, Gavin K Brennen, and Guido Pupillo. 2025. High-rate quantum LDPC codes for long-range-connected neutral atom registers. *Nature Communications* 16, 1 (2025), 1111.

[44] Michael A. Perlin. 2023. qLDPC. https://github.com/qLDPCOrg/qLDPC.

[45] Eric C Peterson, Lev S Bishop, and Ali Javadi-Abhari. 2022. Optimal synthesis into fixed xx interactions. *Quantum* 6 (2022), 696.

[46] Eric C Peterson, Gavin E Crooks, and Robert S Smith. 2020. Fixed-depth two-qubit circuits and the monodromy polytope. *Quantum* 4

(2020), 247.

[47] Massimiliano Poletto and Vivek Sarkar. 1999. Linear scan register allocation. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 21, 5 (1999), 895–913.

[48] Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. 2022. Measuring the capabilities of quantum computers. *Nature Physics* 18, 1 (2022), 75–79.

[49] Nils Quetschlich, Lukas Burgholzer, and Robert Wille. 2023. MQT Bench: Benchmarking software and design automation tools for quantum computing. *Quantum* 7 (2023), 1062.

[50] Chad Rigetti and Michel Devoret. 2010. Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies. *Physical Review B—Condensed Matter and Materials Physics* 81, 13 (2010), 134507.

[51] Joschka Roffe, David R White, Simon Burton, and Earl Campbell. 2020. Decoding across the quantum low-density parity-check code landscape. *Physical Review Research* 2, 4 (2020), 043423.

[52] Peter W Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 124–134.

[53] 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.

[54] 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.

[55] Robert R Tucci. 2005. An introduction to Cartan's KAK decomposition for QC programmers. arXiv preprint quant-ph/0507171.

[56] Joshua Viszlai, Willers Yang, Sophia Fuhui Lin, Junyu Liu, Natalia Nottingham, Jonathan M Baker, and Frederic T Chong. 2023. Matching generalized-bicycle codes to neutral atoms for low-overhead fault-tolerance. *arXiv preprint arXiv:2311.16980* (2023).

[57] Hanrui Wang, Daniel Bochen Tan, Pengyu Liu, Yilian Liu, Jiaqi Gu, Jason Cong, and Song Han. 2024. Q-pilot: Field programmable qubit array compilation with flying ancillas. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*. 1–6.

[58] Ke Wang, Zhide Lu, Chuanyu Zhang, Gongyu Liu, Jiachen Chen, Yanzhe Wang, Yaozu Wu, Shibo Xu, Xuhao Zhu, Feitong Jin, et al. 2025. Demonstration of low-overhead quantum error correction codes. *arXiv preprint arXiv:2505.09684* (2025).

[59] Ken Xuan Wei, Isaac Lauer, Emily Pritchett, William Shanks, David C McKay, and Ali Javadi-Abhari. 2024. Native two-qubit gates in fixed-coupling, fixed-frequency transmons beyond cross-resonance interaction. *PRX Quantum* 5, 2 (2024), 020338.

[60] Xian Wu, Chenghong Zhu, Jingbo Wang, and Xin Wang. 2025. BOSS: Blocking algorithm for optimizing shuttling scheduling in Ion Trap. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 290–303.

[61] 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).

[62] Qian Xu, J Pablo Bonilla Ataides, Christopher A Pattison, Nithin Raveendran, Dolev Bluvstein, Jonathan Wurtz, Bane Vasić, Mikhail D Lukin, Liang Jiang, and Hengyun Zhou. 2024. Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays. *Nature Physics* 20, 7 (2024), 1084–1090.

[63] 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).

[64] Christopher G Yale, Rich Rines, Victory Omole, Bharath Thotakura, Ashlyn D Burch, Matthew NH Chow, Megan Ivory, Daniel Lobser, Brian K McFarland, Melissa C Revelle, et al. 2024. Noise-Aware Circuit Compilations for a Continuously Parameterized Two-Qubit Gateset. *arXiv preprint arXiv:2411.01094* (2024).

[65] 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

[66] 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.

[67] 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.

[68] 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.

[69] 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).

[70] 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).

[71] 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.

[72] 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.

A Canonical gate and 2Q circuit synthesis

In this section we show the basic mathematical properties the its canonical form of 2Q unitary and then discuss the synthesis capability of some 2Q basis gates.

A.1 Canonical decomposition

$\mathbf{SU}(N)$ is a real manifold with dimension $N^2 - 1$, within which any element is a *special unitary* matrix with determinant equal to 1. Since the global phase does not affect quantum computation processes, it is sufficient to focus on the mathematical properties of special unitaries in the area of circuit synthesis. A generic 2Q gate, despite having 15 real parameters, can have its nonlocal behavior fully characterized by only 3 real parameters. This method, known as *Canonical decomposition* or *KAK decomposition* from Lie algebra theory, is widely adopted in quantum computing [8, 55, 68, 72]. Specifically, for any $U \in \mathbf{SU}(4)$, there exists a unique $\vec{\eta} = (x, y, z) \in W \subseteq \mathbb{R}^3$, along with $V_1, V_2, V_3, V_4 \in \mathbf{SU}(2)$ and a global phase, such that

$$U = g \cdot (V_1 \otimes V_2) e^{-i\vec{\eta}\cdot\vec{\Sigma}} (V_3 \otimes V_4),\ g \in \{1, i\} \tag{8}$$

where $\vec{\Sigma} \equiv (XX, YY, ZZ)$ [55]. The set

$$W := \left\{(x, y, z) \in \mathbb{R}^3 \mid \frac{\pi}{4} \geq x \geq y \geq |z|,\ z \geq 0 \text{ if } x = \frac{\pi}{4}\right\} \tag{9}$$

is known as the *Weyl chamber* [68], and $\vec{\eta} \in W$ is known as the *Weyl coordinate* of U. We also refer to a gate of the form

$$\mathrm{Can}(a, b, c) := e^{-i\frac{\pi}{2}(aXX + bYY + cZZ)} = \begin{pmatrix} e^{-i\frac{c\pi}{2}} \cos\frac{(a-b)\pi}{2} & 0 & 0 & -ie^{-i\frac{c\pi}{2}} \sin\frac{(a-b)\pi}{2} \\ 0 & e^{i\frac{c\pi}{2}} \cos\frac{(a+b)\pi}{2} & -ie^{i\frac{c\pi}{2}} \sin\frac{(a+b)\pi}{2} & 0 \\ 0 & -ie^{i\frac{c\pi}{2}} \sin\frac{(a+b)\pi}{2} & e^{i\frac{c\pi}{2}} \cos\frac{(a+b)\pi}{2} & 0 \\ -ie^{-i\frac{c\pi}{2}} \sin\frac{(a-b)\pi}{2} & 0 & 0 & e^{-i\frac{c\pi}{2}} \cos\frac{(a-b)\pi}{2} \end{pmatrix} \tag{10}$$

as a *canonical* gate. Two 2Q gates U and V are considered *locally equivalent* if they differ only by 1Q gates, meaning their canonical coefficients can be transformed into one another via the equivalence rules [15]:

1. $(a, b, c) \sim (b, a, c)$ or $(a, b, c) \sim (c, b, a)$, i.e., any permutation of the coefficients;
2. $(a, b, c) \sim (-a, -b, c)$;
3. $(a, b, c) \sim (a - 1, b, c)$;
4. $(1/2, b, c) \sim (1/2, b, -c)$.

Note that we align the conventional that canonical coefficient (a, b, c) differs from Weyl coordinate (x, y, z) by a $\frac{\pi}{2}$ factor. Unless otherwise specified, the canonical coefficients of gates in quantum ISAs and circuits are confined to $\frac{1}{2} \geq a \geq b \geq |c|$. While for the Weyl chamber visualization by means of `weylchamber` [21], we assume the Weyl coordinates are confined to $\left\{\frac{\pi}{4} \geq x \geq y \geq z \geq 0\right\} \cup \left\{\frac{\pi}{4} \geq \frac{\pi}{2} - x \geq y \geq z \geq 0\right\}$, as illustrated by Figure 3. Conversion of Weyl coordinates for different conventions is not simple according to the equivalence rules above.

A.2 Quantum ISA and the synthesis capability

A quantum ISA typically includes qubit initialization, a universal gate set, and measurement. It serves as an interface between software and hardware by mapping high-level semantics of quantum programs to low-level native quantum operations or pulse sequences on hardware. The universal gate set, especially specified by its 2Q basis gates, is the key component of a quantum ISA that dominates its hardware-implementation accuracy and cost, as well as software-expressivity sufficiency.

CX or CNOT is the most popular basis gate provides by hardware vendors and considered by various quantum compiler optimization methods. The superconducting Cross-Resonance gate [50] and ion-trapped Mølmer-Sørensen gate [7] are both CX-equivalent gates with the same canonical form $\mathrm{Can}(\frac{1}{2}, 0, 0)$. In the superconducting platforms with XY-coupled Hamiltonian like Google's Sycamore [3], iSWAP $\sim \mathrm{Can}(\frac{1}{2}, \frac{1}{2}, 0)$ is another representative native 2Q basis gate and could be less sensitive to

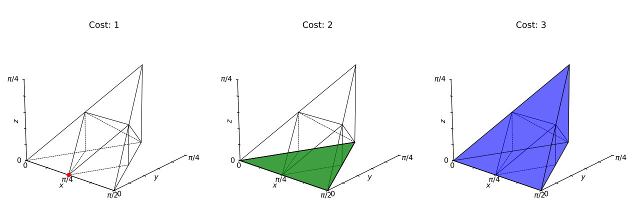


Figure 12. Coverage set for `CX` ISA.

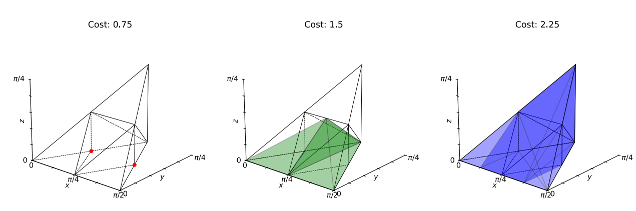


Figure 13. Coverage set for `SQiSW` ISA.

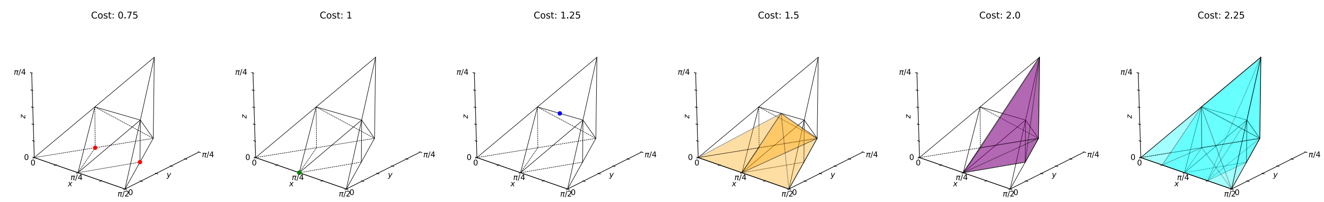


Figure 14. Coverage set for `SQiSW_` ISA.

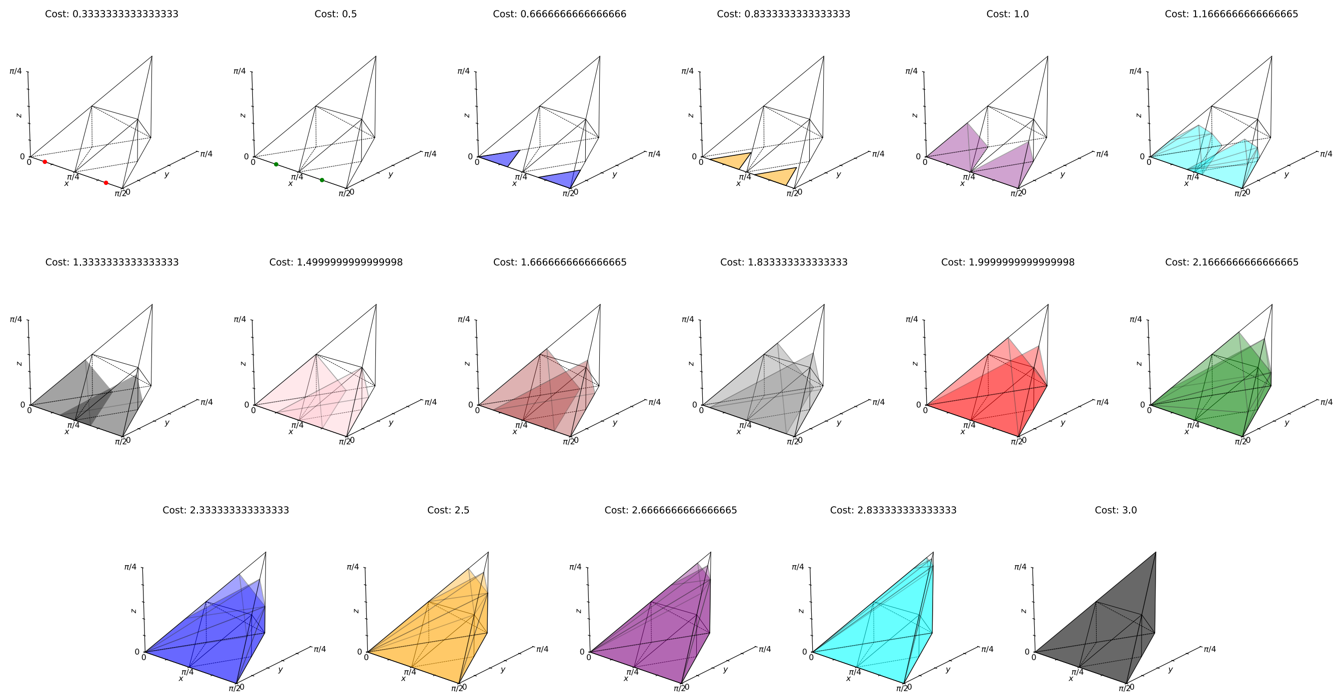


Figure 15. Coverage set for `ZZPhase` ISA.

leakage error than the native CZ gate. Recent experimental advances demonstrate that more basis gates could be implemented natively and calibrated in high precision [12, 59, 63]. Particularly, some basis gates like $\sqrt{\text{iSWAP}} \sim \text{Can}\big(\frac{1}{4}, \frac{1}{4}, 0\big)$ and fractional $\text{ZZ}(\theta) \sim \text{Can}\big(a, 0, 0\big)$ gates offers more promising ISA selections as they exhibit shorter gate duration, higher gate accuracy, and stronger synthesis capability.

The synthesis capability or computational power of basis gates can be geometrically illustrated by monodrome polytopes within the Weyl chamber. The coverage set for `CX` depicted in Figure 12 implies that

1. One CX gate is required to synthesize 2Q gates $\sim \text{Can}\big(\frac{1}{2}, 0, 0\big)$, i.e., CX-equivalent gates $(V_1 \otimes V_2)\text{CX}(V_3 \otimes V_4)$;
2. Two CX gates are required to synthesize 2Q gates $\sim \text{Can}(a, b, 0)$, i.e., $(V_1 \otimes V_2)\text{CX}(V_3 \otimes V_4)\text{CX}(V_5 \otimes V_6)$;
3. Three CX gates are required to synthesize 2Q gates $\sim \text{Can}(a, b, c)$, i.e., $(V_1 \otimes V_2)\text{CX}(V_3 \otimes V_4)\text{CX}(V_5 \otimes V_6)\text{CX}(V_7 \otimes V_8)$.

We assume the cost of one CX gate is 1.0, polytopes in different colors denotes the minimal circuit cost (duration) for the coverage set if synthesized by CX and arbitrary 1Q gates. That is, on average, the number of CX gates required to synthesize arbitrary 2Q gates is 3. In contrast, the number for `SQiSW` ISA is 2.21 [26].

Monodromy polytope theory [46] provides a framework for determining the synthesis coverage set and circuit cost (in 2Q depth) for any set of basis gates with specified costs, while the specific gate decomposition process is left to the synthesizer to complete. For the selected ISAs in Table 2 with the basis gate costs assumed in Equation (7), Figures 12 to 17 describes their coverage sets, respectively. With the enrichment of quantum ISA (e.g., combining gate families, involving mirror gates) and heterogeneous basis gate cost settings, the coverage set reveals a richer variety of convex polyhedra. That implies more optimization effects for the ISA-ware routing mechanism in Canopus.

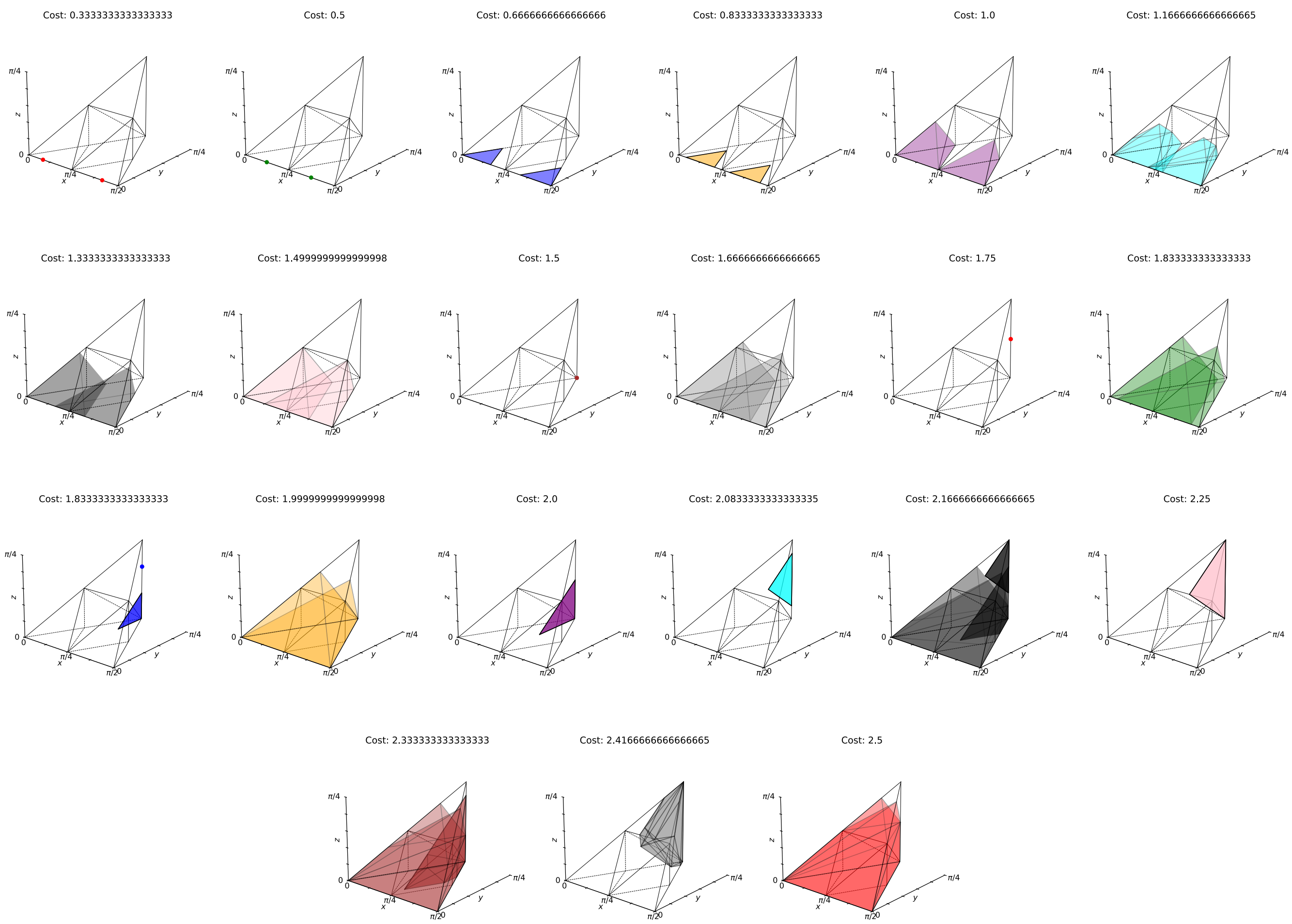
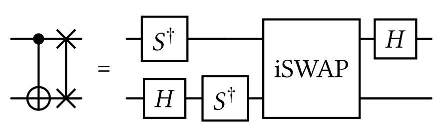


Figure 16. Coverage set for ZZPhase_ ISA.

A.3 2Q gate mirroring

The mirror symmetry of a 2Q gate U is defined as the composition of the original gate and a SWAP gate [48], i.e., SWAP $\cdot\, U$. For example, CX and iSWAP is a typical pair of mirror gates as shown below.



In general, the mirroring rule for Canonical coefficients is described as

$$\mathrm{SWAP}\cdot\mathrm{Can}(a,b,c) \sim \left(a+\frac{1}{2}, b+\frac{1}{2}, c+\frac{1}{2}\right) \quad \sim \left(a+\frac{1}{2}-1, b+\frac{1}{2}-1, c+\frac{1}{2}-1\right) \sim \begin{cases} \left(\frac{1}{2}-c, \frac{1}{2}-b, a-\frac{1}{2}\right), & \text{if } c \geq 0 \\ \left(\frac{1}{2}+c, \frac{1}{2}-b, \frac{1}{2}-a\right), & \text{if } c < 0 \end{cases}. \tag{11}$$

The mirror pair of CX and iSWAP is a specical case implying that a CX-iSWAP combination ISA could result in lower overhead in routing-synthesis collaborative optimization. Yale et al. [64] once considers inserting SWAP gates to get mirrored gates with lower synthesis overhead compared to the original gates, given the all-to-all topology and continuous ZZ(θ) gate set on ion-trapped hardware. McKinney et al. [40] discusses that integrating $\sqrt{\text{iSWAP}}$'s mirror gate, i.e, ECP $\sim$ Can$\left(\frac{1}{4},\frac{1}{4},0\right)$ gate, into the powerful SQiSW ISA, could further improve the ISA's synthesis capability and end-to-end routing-synthesis co-optimization on limited topologies.

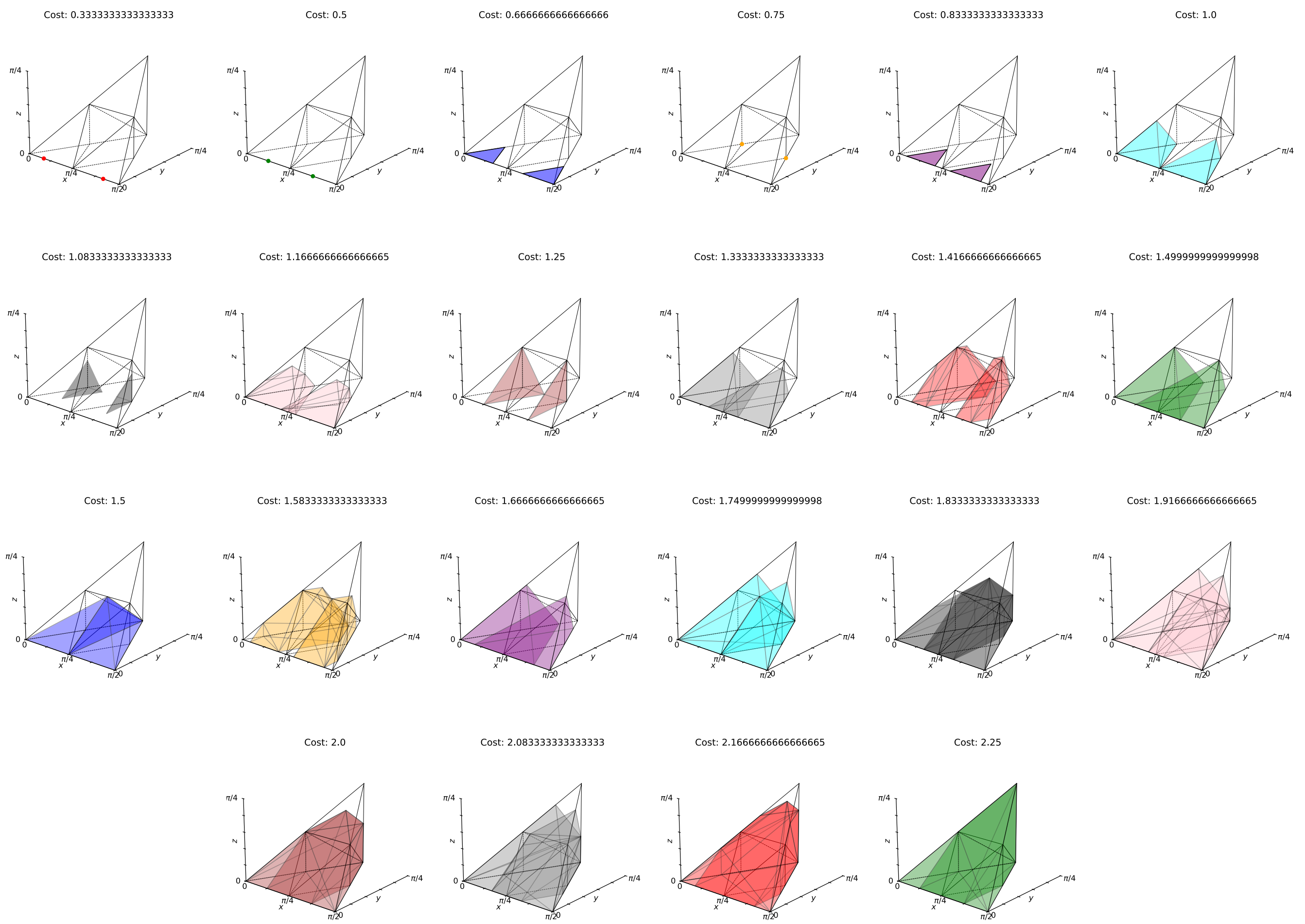


Figure 17. Coverage set for `Het` ISA.

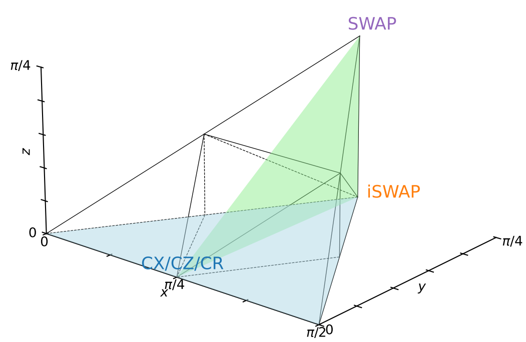


Figure 18. Morrir symmetry for $\mathrm{Can}(a, b, 0)$ and $\mathrm{Can}(\frac{1}{2}, b', c')$ gate families.

B Commutative relation of canonical gates

Herein we present detailed proof for Theorem 1. The *if* direction is trivial, and hence we justify the *only if* direction, relying on the following two lemmas.

Lemma 1. *Let A, B be two Hermitian matrices with eigenvalues in the range $[-2, 2)$. If $[e^{-i\frac{\pi}{2}A}, e^{-i\frac{\pi}{2}B}] = 0$ then $[A, B] = 0$.*

Proof. This follows from the fact that compatible observables (commuting operators) can be simultaneously diagonalized. In this case, the respective unitary matrix $e^{-i\frac{\pi}{2}A}$ commutes with $e^{-i\frac{\pi}{2}B}$. Denote by A_λ the eigenspace corresponding to the eigenvalue λ of $e^{-i\frac{\pi}{2}A}$, i.e. $e^{-i\frac{\pi}{2}A} = \oplus_\lambda \lambda A_\lambda$. Then we have

$$\forall \vec{v} \in A_\lambda,\ e^{-i\frac{\pi}{2}B}e^{-i\frac{\pi}{2}A}\vec{v} = e^{-i\frac{\pi}{2}B}\lambda\vec{v} = \lambda e^{-i\frac{\pi}{2}B}\vec{v} = e^{-i\frac{\pi}{2}A}e^{-i\frac{\pi}{2}B}\vec{v}, \tag{12}$$

and thus $e^{-i\frac{\pi}{2}B}\vec{v} \in A_\lambda$. Thus A_λ is $e^{-i\frac{\pi}{2}B}$-invariant and the restriction $e^{-i\frac{\pi}{2}B}\big|_{A_\lambda}$ of $e^{-i\frac{\pi}{2}B}$ to A_λ is still unitary since it preserves inner products. Hence it is diagonalizable and we can find an orthonormal basis $w_{\lambda_1}, w_{\lambda_2}, \ldots, w_{\lambda_k}$ consisting of eigenvectors of $e^{-i\frac{\pi}{2}B}\big|_{A_\lambda}$. Note that these are also eigenvectors of $e^{-i\frac{\pi}{2}A}$ (with eigenvalue λ). Following the same token as above, for each eigenspace E_{λ_i} of $e^{-i\frac{\pi}{2}A}$, we can construct an orthonormal basis β_i for it consisting of eigenvectors of $e^{-i\frac{\pi}{2}B}$. Finally since the eigenspaces of different eigenvalues of $e^{-i\frac{\pi}{2}A}$ are orthogonal to each other, $\beta = \cup_i \beta_i$ forms an orthonormal basis of the entire Hilbert space $\mathcal{H}_n$ consisting of the coeigenvectors of both $e^{-i\frac{\pi}{2}A}$ and $e^{-i\frac{\pi}{2}B}$.

Now let U be a unitary matrix with the vectors in β being its columns, then

$$\begin{aligned} U^\dagger e^{-i\frac{\pi}{2}A}U &= D_A \\ U^\dagger e^{-i\frac{\pi}{2}B}U &= D_B \end{aligned} \tag{13}$$

In general, an eigenvector of $e^{-i\frac{\pi}{2}A}$ need *not* be that of A. However, since A has its eigenvalues in the range $[-2, 2)$, the map

$$f : [-2, 2) \longrightarrow U(1), a \longrightarrow e^{-i\frac{\pi}{2}a} \tag{14}$$

is injective. Consequently different eigenvalues of A correspond to different eigenvalues of $e^{-i\frac{\pi}{2}A}$, and hence the eigenspaces of $e^{-i\frac{\pi}{2}A}$ and A coincide. Therefore, we have that

$$\begin{aligned} U^\dagger AU &= \Sigma_A \\ U^\dagger BU &= \Sigma_B \end{aligned} \tag{15}$$

and since $[\Sigma_A, \Sigma_B] = 0$ as they are diagonal, $[A, B] = 0$. We obtain the desired result.

□

Lemma 2. *Let* $P_1 = (a_1X_1X_2 + b_1Y_1Y_2 + c_1Z_1Z_2)I_3$, $P_2 = I_1(a_2X_2X_3 + b_2Y_2Y_3 + c_2Z_2Z_3)$ *with* $|c_1| \le b_1 \le a_1 \le \frac{1}{2}$, $|c_2| \le b_2 \le a_2 \le \frac{1}{2}$. *If* $[P_1, P_2] = 0$ *and* $P_1, P_2 \ne 0$, *then* $b_1 = b_2 = c_1 = c_2 = 0$.

Proof. Consider the product P_1P_2. We assume for the sake of contradiction that $b_1 \ne 0$. Using $[X, Y] = 2iZ$, $[Y, Z] = 2iX$, $[Z, X] = 2iY$, we expand

$$[P_1, P_2] = 2i\big(a_1b_2\, X_1Z_2Y_3 - b_1a_2\, Y_1Z_2X_3 + b_1c_2\, Y_1X_2Z_3\big) - 2i\big(a_1c_2\, X_1Y_2Z_3 + c_1a_2\, Z_1Y_2X_3 + c_1b_2\, Z_1X_2Y_3\big).$$

Since the each Pauli string is linearly independent in the 8×8 operator basis, e.g. term $Y_1Z_2X_3$ cannot be canceled out by any other terms, contradictory to the fact that $[P_1, P_2] = 0$. Hence, vanishing of $[P_1, P_2]$ requires

$$a_1b_2 = a_1c_2 = b_1c_2 = b_1a_2 = c_1a_2 = c_1b_2 = 0.$$

Since $P_1, P_2 \ne 0$, at least a_1, a_2 is nonzero, leading to $b_1 = b_2 = c_1 = c_2 = 0$. □

Using Lemma 1 and Lemma 2 above, it is straightforward to prove Theorem 1. We see that $\|P_1\| \le \|a_1X_1X_2I_3\| + \|b_1Y_1Y_2I_3\| + \|c_1Z_1Z_2I_3\| \le |a_1| + |b_1| + |c_1| \le \frac{3}{2}$, where $\|\cdot\|$ is the operator norm. Hence, eigenvalues of P_1 are in range of $[-2, 2)$. Same as the eigenvalues of P_2. Now if $[e^{-i\frac{\pi}{2}P_1}, e^{-i\frac{\pi}{2}P_2}] = 0$, then we have that $[P_1, P_2] = 0$ according to Lemma 1, and thus $b_1 = b_2 = c_1 = c_2 = 0$ according to Lemma 2, which proves the *only if* direction.