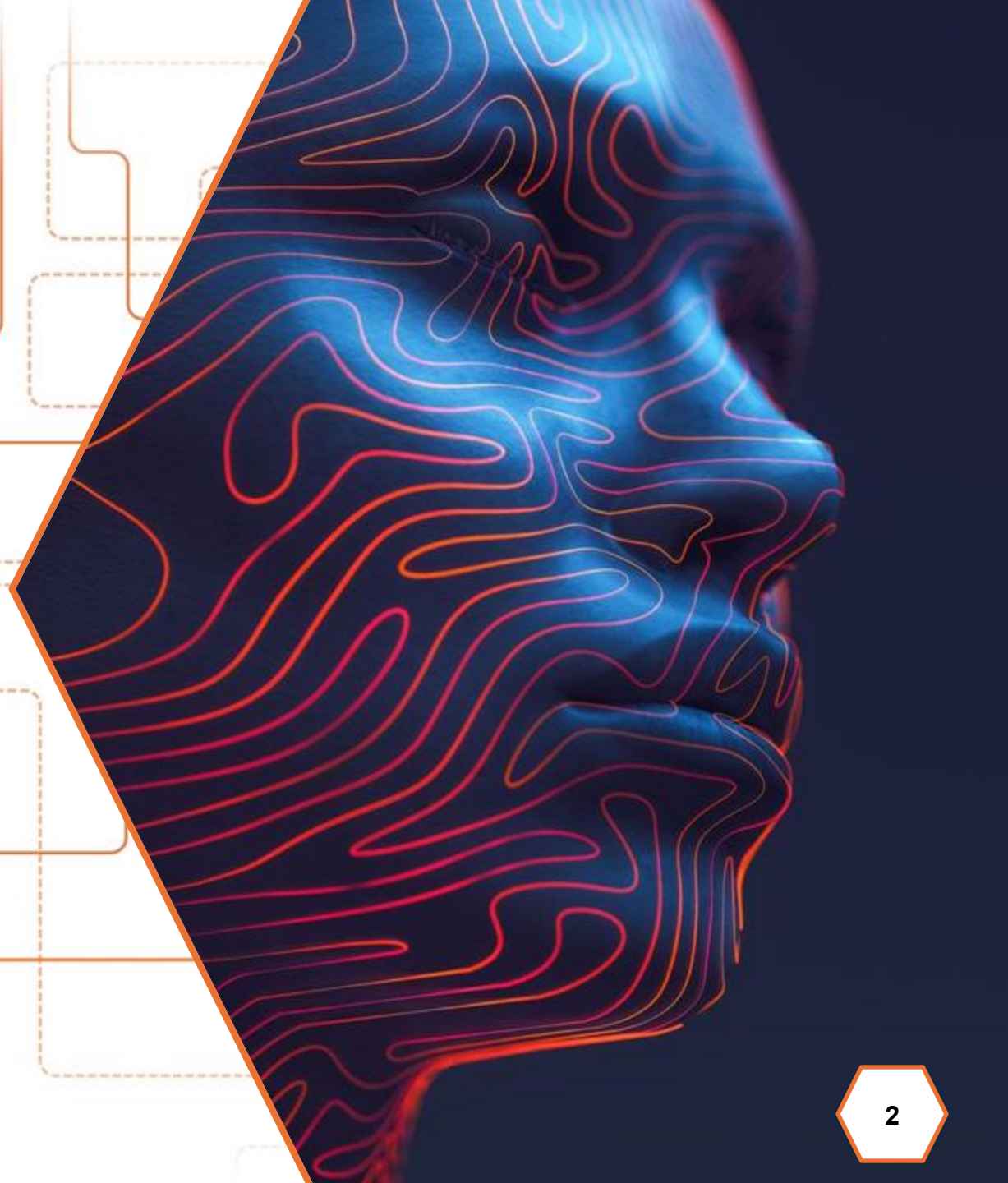
Toward Practical Quantum Computing Systems with Electronic Design Automation

SPONSORED BY CEDA sigda

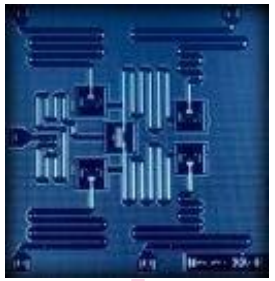
Agenda

- **Motivation**
- Background
- Problem Formulation
- Qplacer – Global Placement
- qGDP – Legalization & Detailed Placement

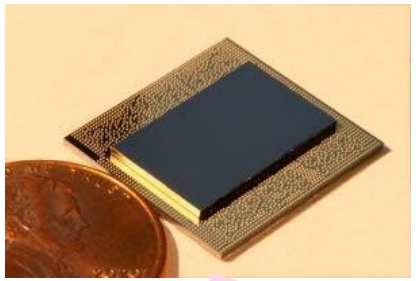


Quantum Computer Evolution

5 Qubits

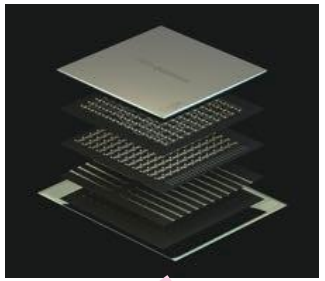


27 Qubits



65 Qubits

127 Qubits

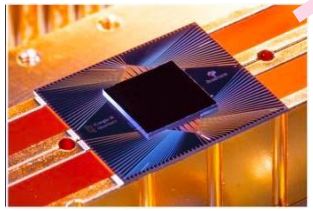


433 Qubits

1121 Qubits

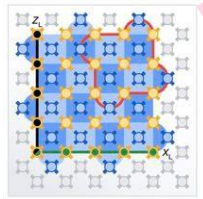
... 2017 2018 2019 2020 2021 2022 2023 2024

Cloud Access



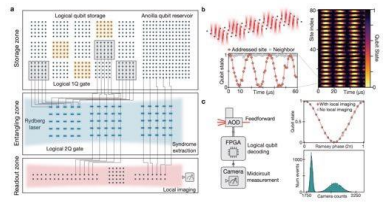
Quantum Supremacy

Local Installation @ Cleveland Clinic



Google Error Correction Demonstration

Modular Quantum

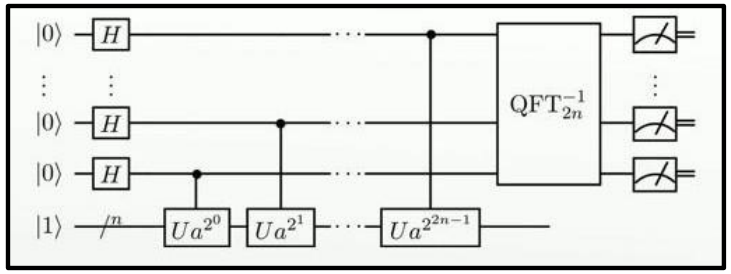


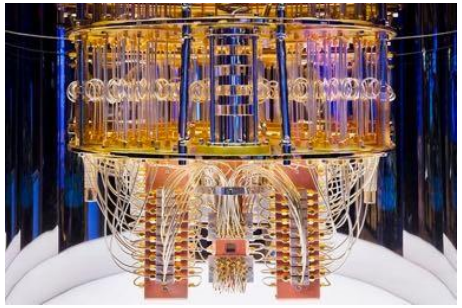
48 Logical Qubits

IBM Quantum, Google Quantum AI, Cleveland Clinic, Bluvstein, Hanrui Wang *et al*

Challenge One: Limited Size

Shor's Factoring Algorithm ~ 1 Million Qubits

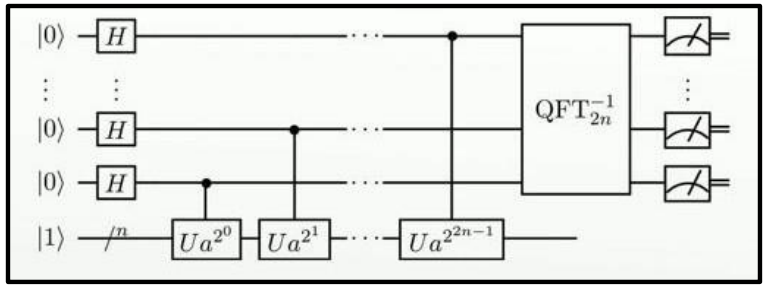




1000 Qubits

Challenge Two: Low Reliability

Technology	Gate Error Rates
Superconducting	$10^{-3} \sim 10^{-4}$
Ion, Atom	$10^{-3} - 10^{-4}$
Electron Spin	10^{-3}
Nuclear Spin	10^{-3}



Shor's Factoring Algorithm

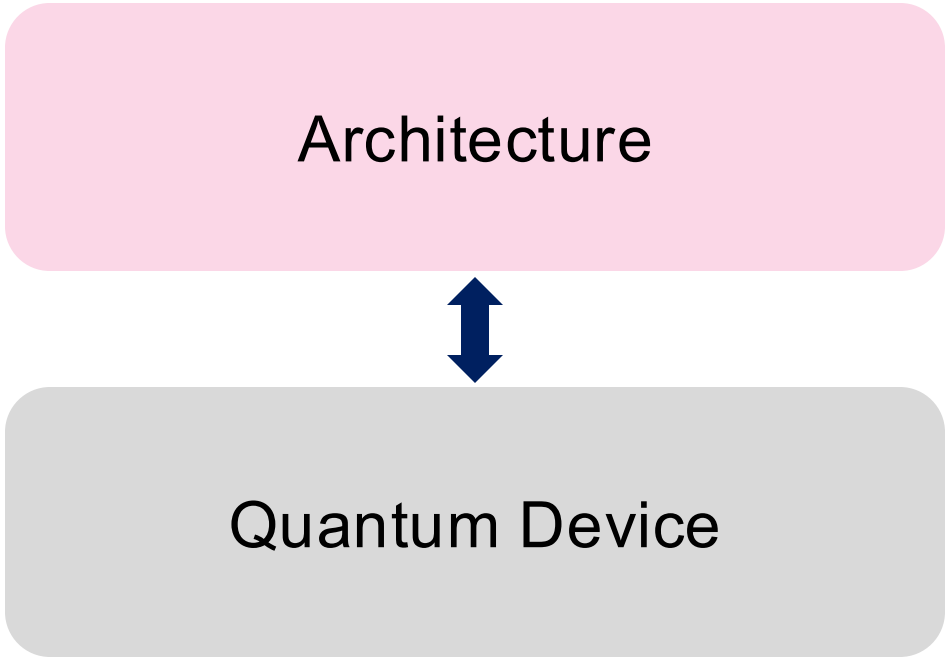
10^{-10}

Ding, Chong, "Quantum Computer Systems: Research for Noisy Intermediate-Scale Quantum Computers"

Motivation

- Tackle the Challenges

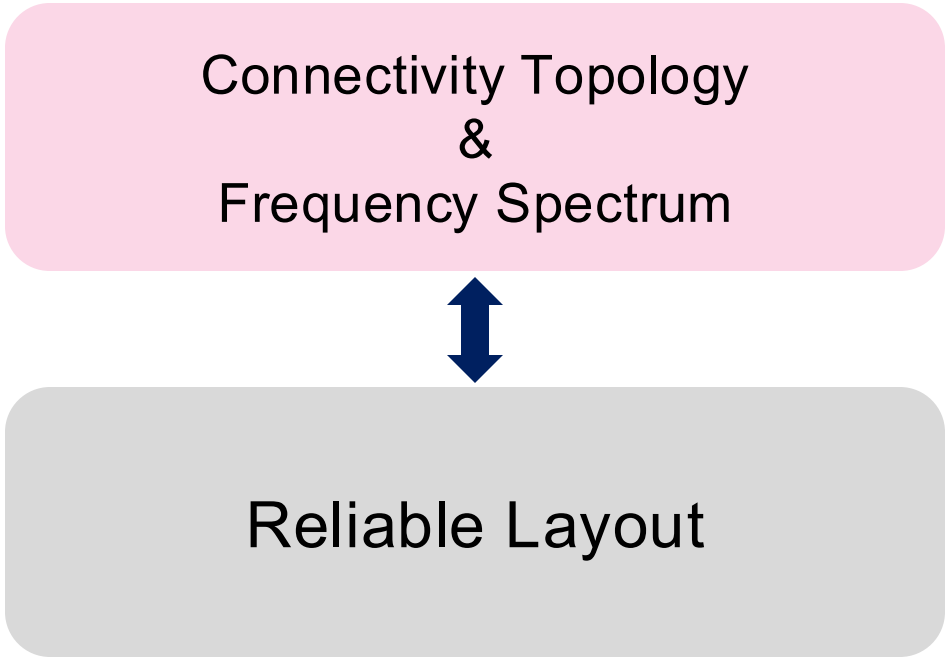
Bridge
Architecture to devices



Motivation

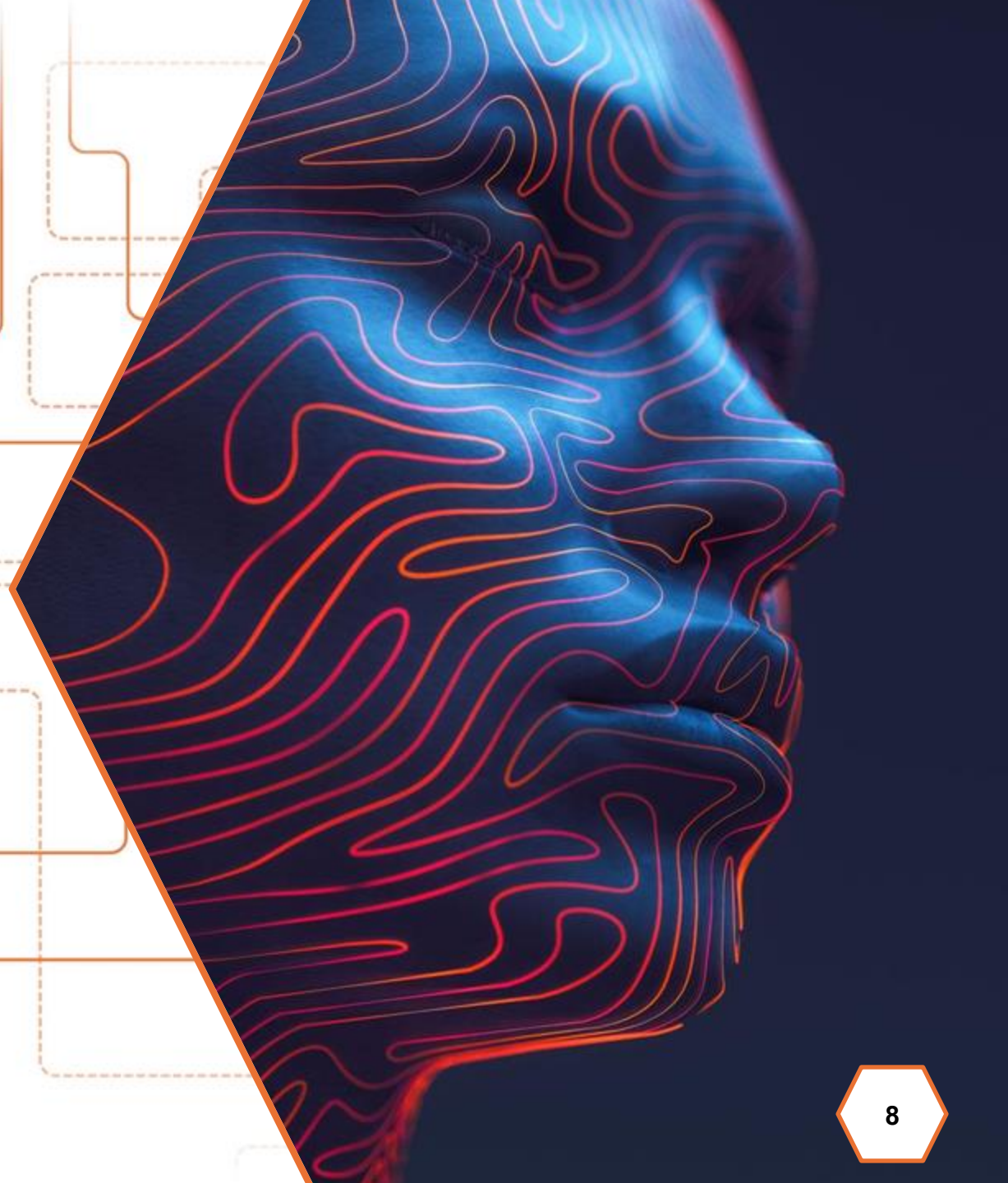
- Tackle the Challenges

Placement & Routing (PNR) tailored for superconducting quantum computers



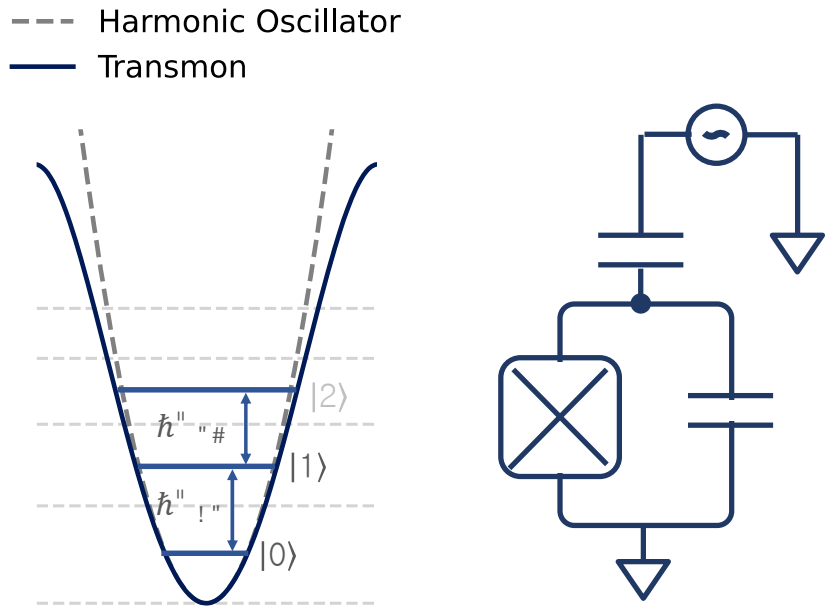
Agenda

- Motivation
- **Background**
- Problem Formulation
- Qplacer – Global Placement
- qGDP – Legalization & Detailed Placement



Background

- Physical Superconducting Qubits
  - Mimic the behavior of atom using superconducting circuit (Artificial Qubit)

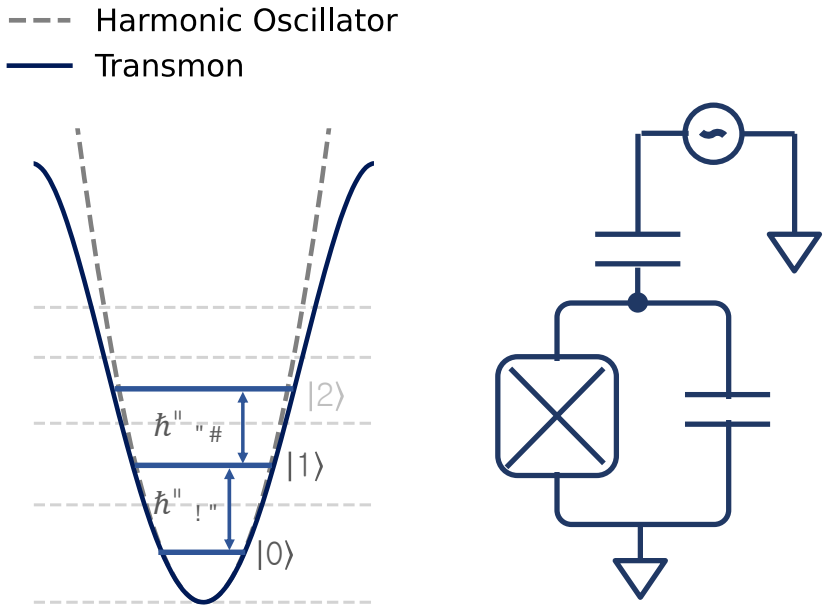


Energy level of Qubits & Circuit diagram

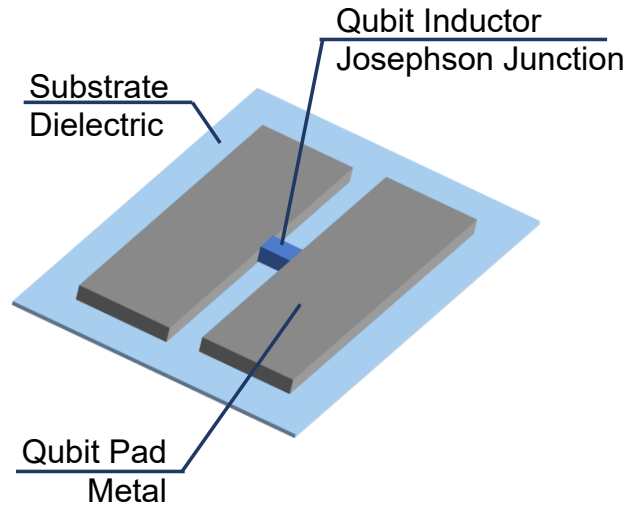
Blais, Alexandre, et al. "Circuit quantum electrodynamics." (Reviews of Modern Physics 2021)

Background

- Physical Superconducting Qubits
  - Mimic the behavior of atom using superconducting circuit (Artificial Qubit)



Energy level of Qubits & Circuit diagram

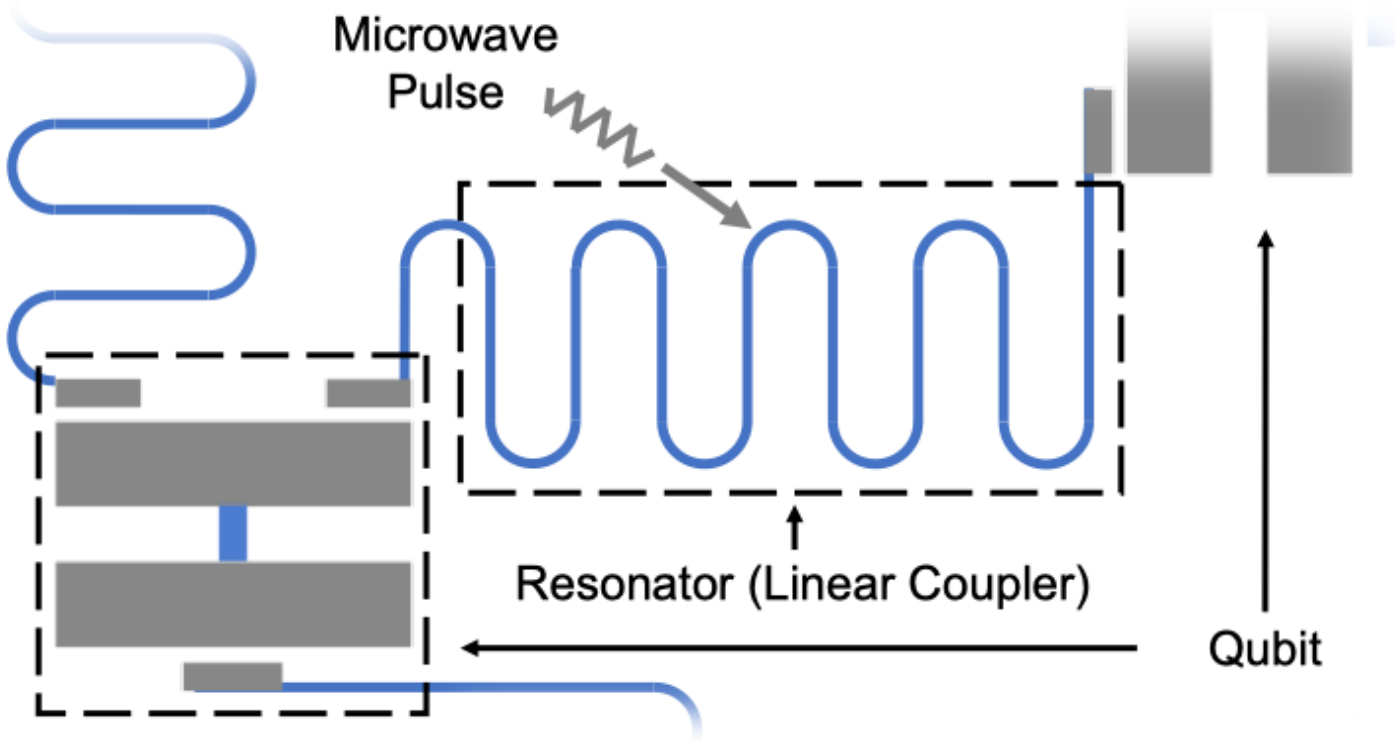


Physical Layout

Blais, Alexandre, et al. "Circuit quantum electrodynamics." (Reviews of Modern Physics 2021)

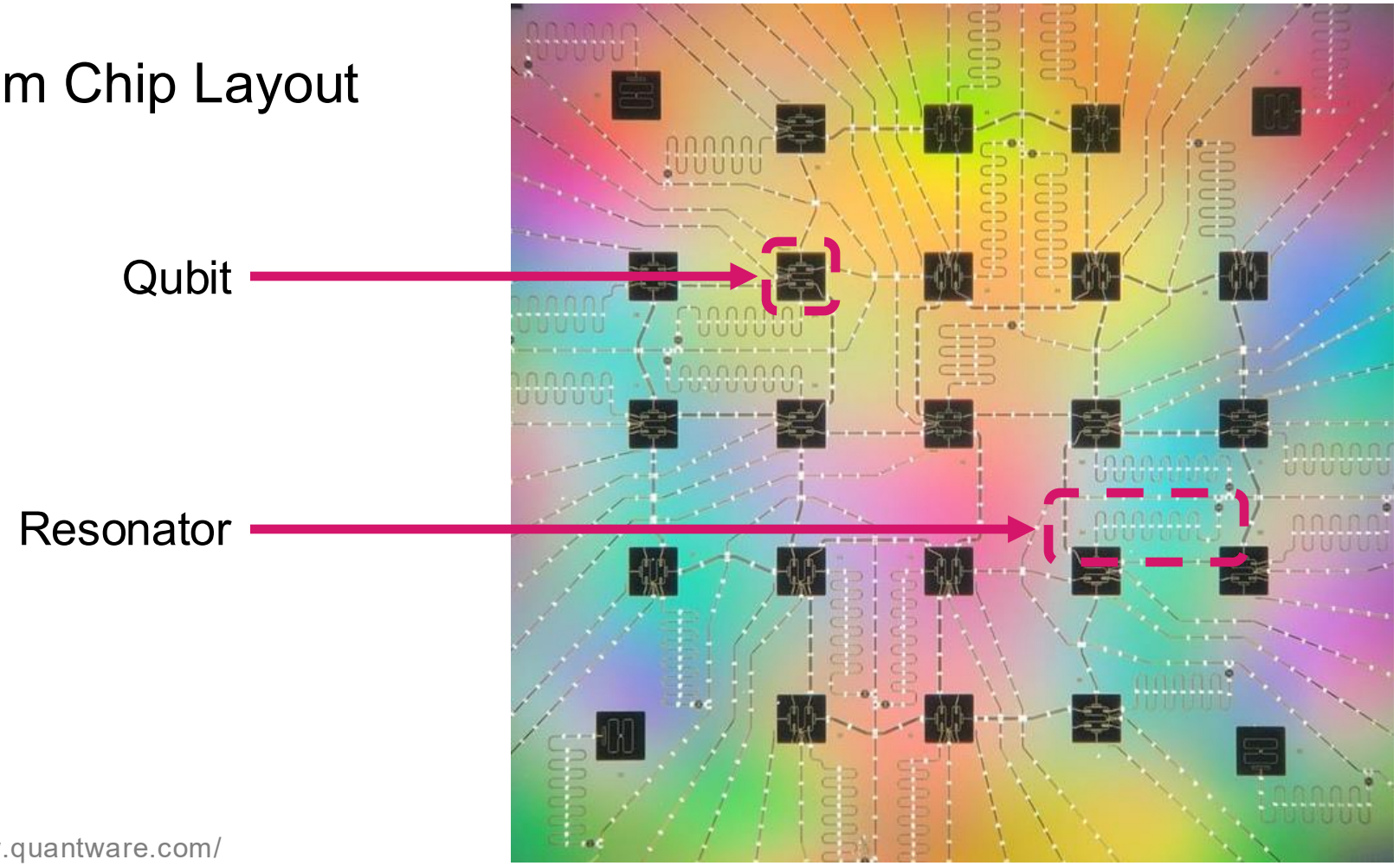
Background

- Resonator



Background

- Sample Quantum Chip Layout



Quantware, https://www.quantware.com/

Background

- Analytical placement
  - Determines physical locations of logic gates (cells) in layout
  - Significantly impacts chip quality (Power, Timing)
  - Key steps:
    - **Global Placement (GP):** Distributes cells across the layout, minimizing target cost
    - **Legalization (LG):** Removes overlaps and aligns cells to designated sites
    - **Detailed Placement (DP):** Refines layout for improved quality

Background

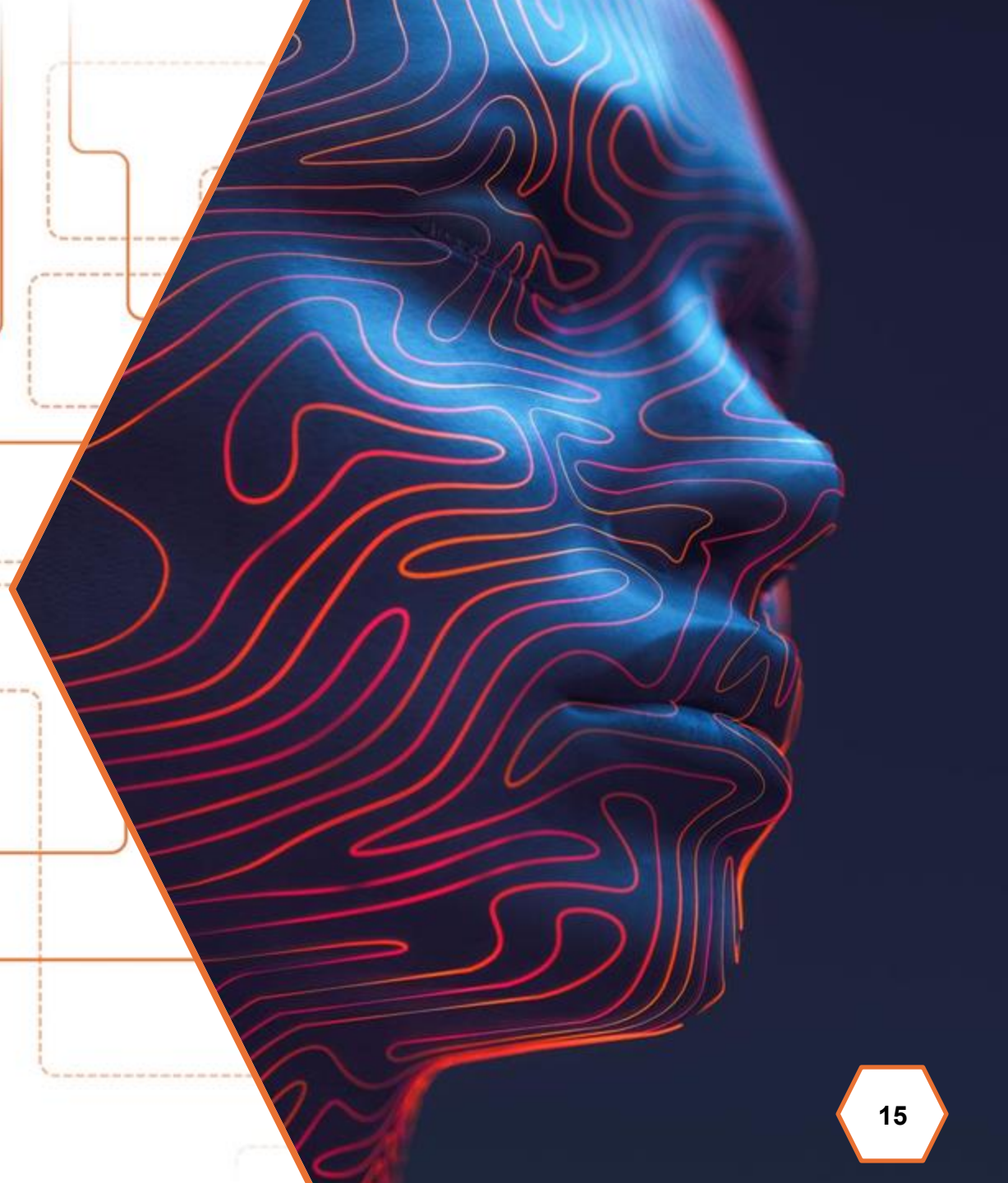
- Analytical Placement in Quantum Domain
  - Determine the physical locations of quantum components (Qubit, Resonators, etc)

  - Goal:
    - **Optimize Substrate Area Utilization**

    - **Ensure High-fidelity Layouts**: resolve quantum-specific issues

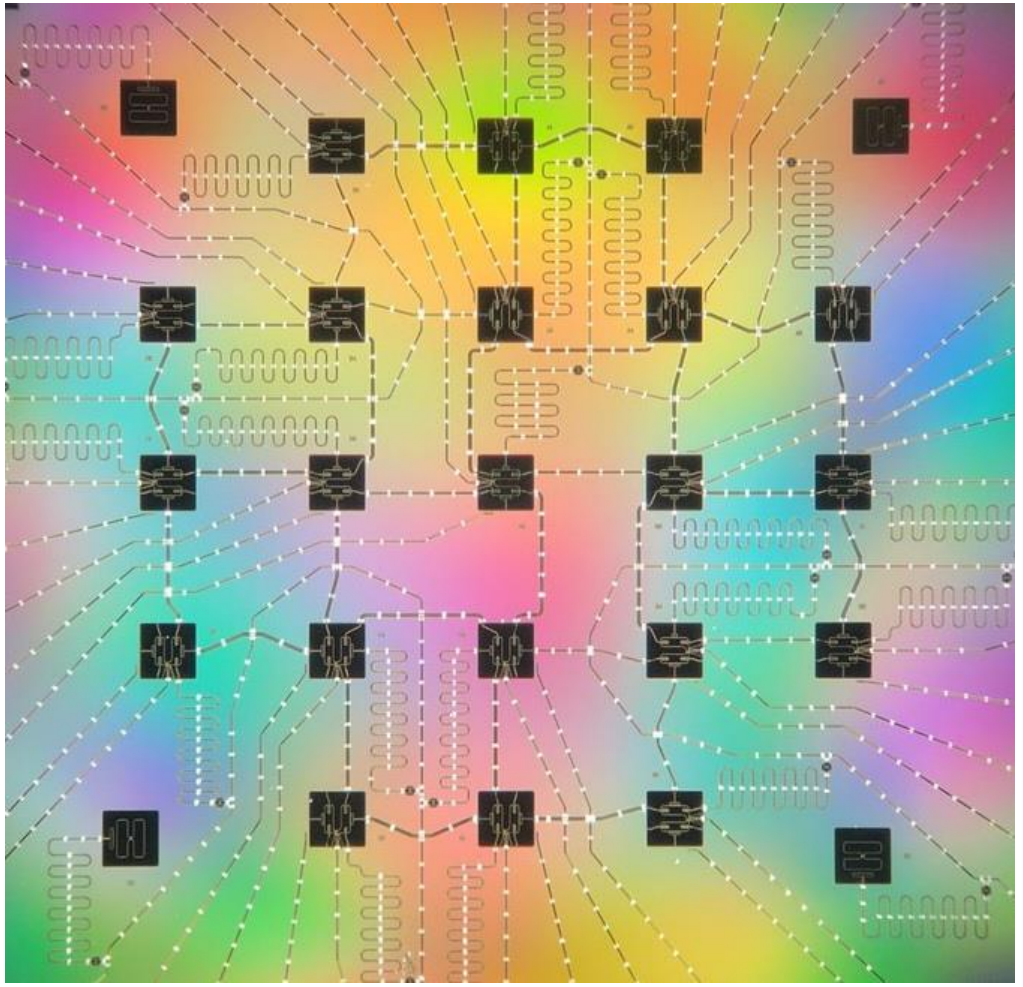
Agenda

- Motivation
- Background
- **Problem Formulation**
- Qplacer – Global Placement
- qGDP – Legalization & Detailed Placement



Background

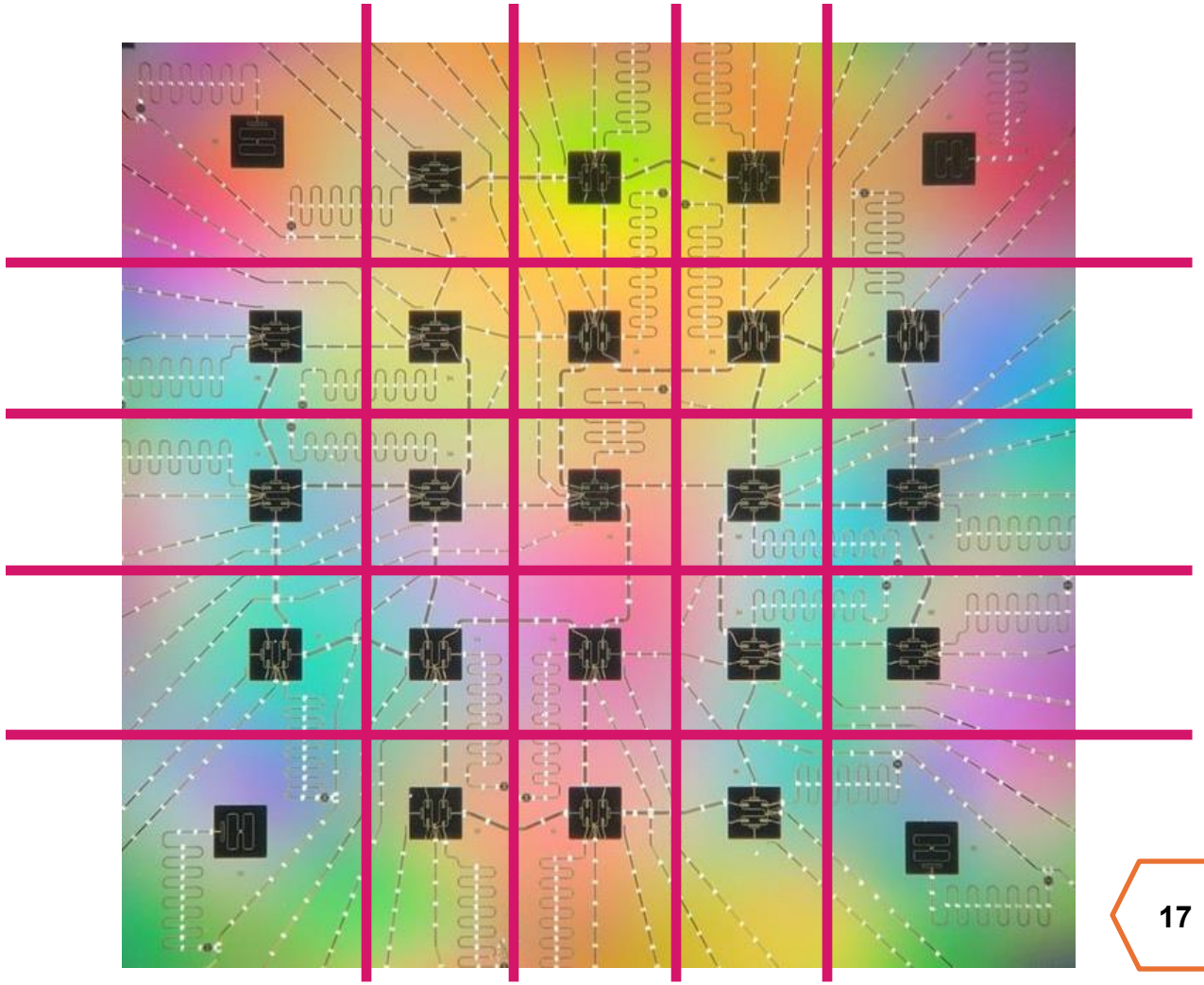
- Flexibility



Quantware, https://www.quantware.com/

Background

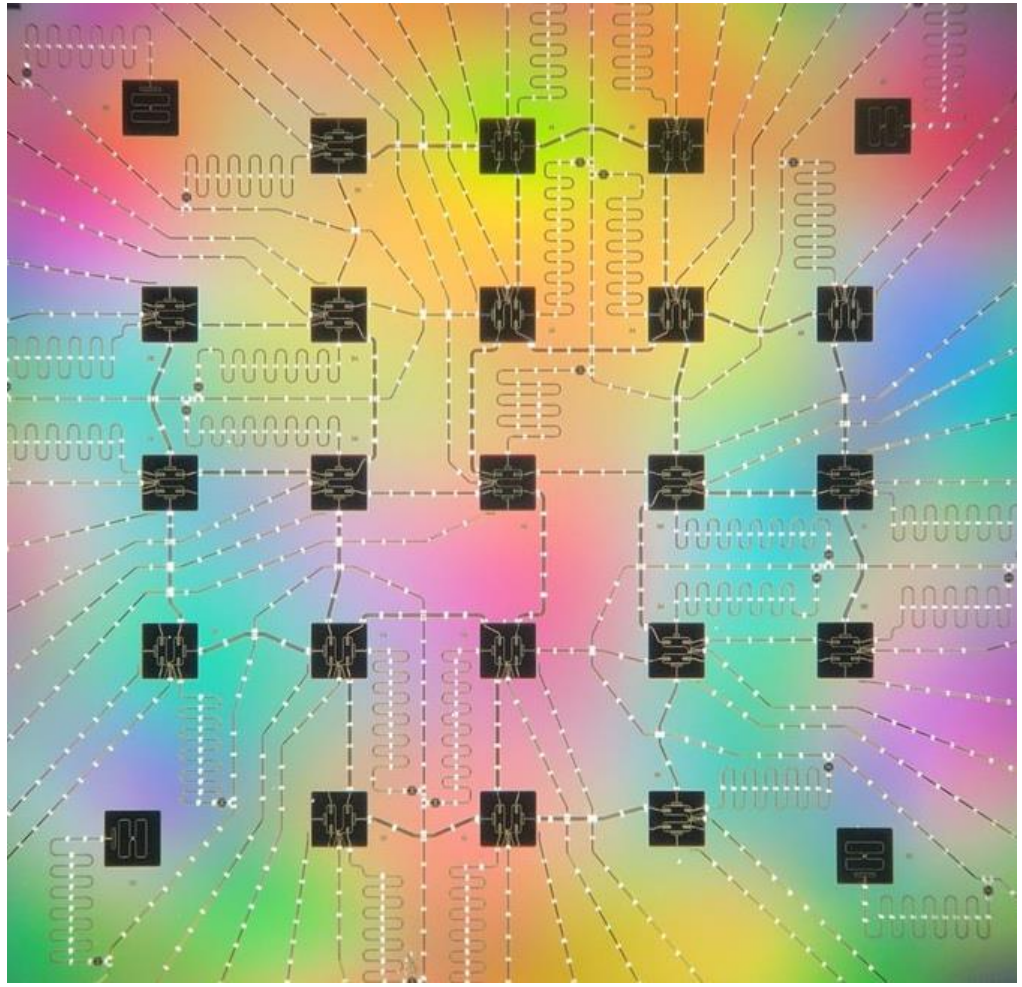
- Flexibility
  - 2D Grid



Quantware, https://www.quantware.com/

Background

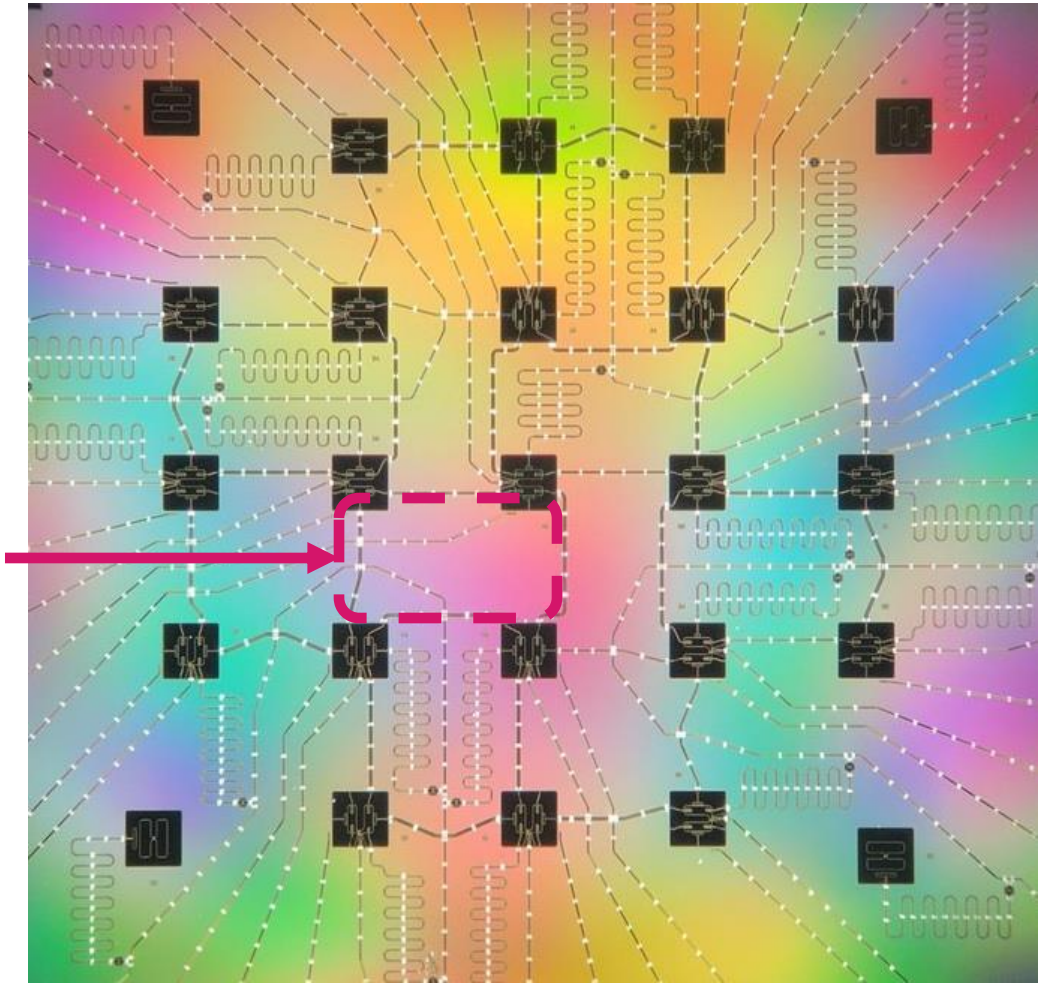
- 2D Grid Placement
  - Limited Flexibility
- Space Utilization



Quantware, https://www.quantware.com/

Background

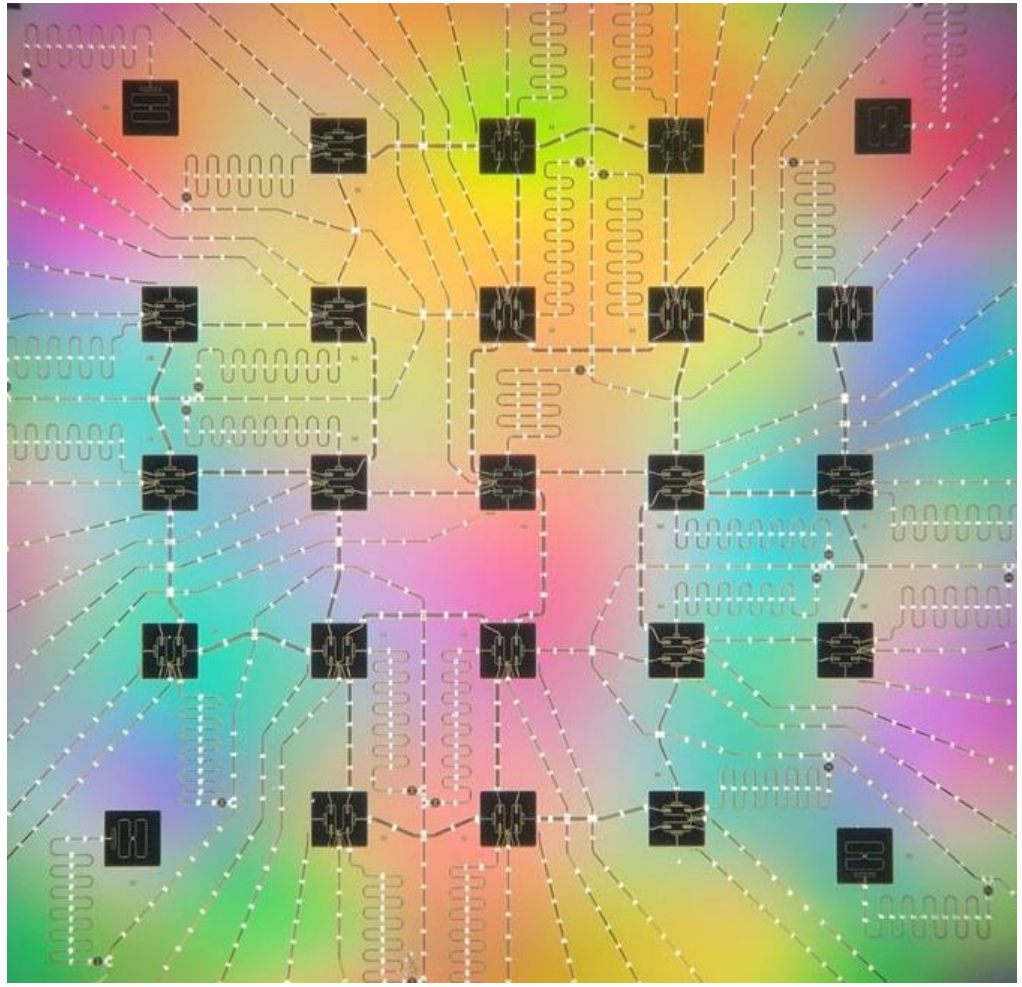
- 2D Grid Placement
  - Limited Flexibility
- Space Utilization
  - Low



Quantware, https://www.quantware.com/

Background

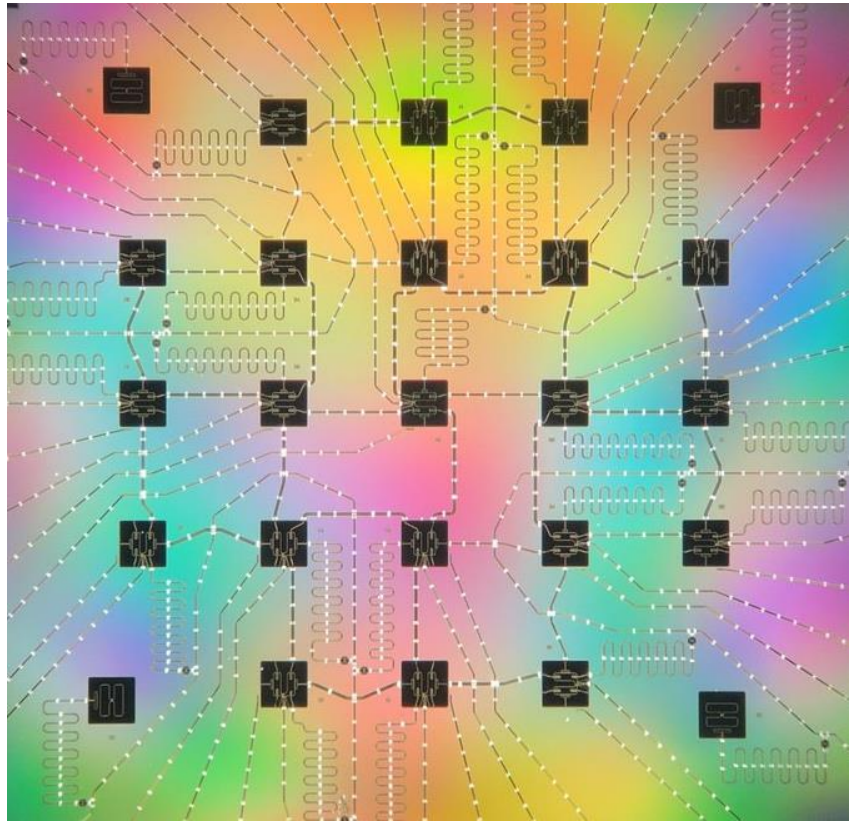
- 2D Grid Placement
  - Limited Flexibility
- Space Utilization
  - Low
- Scalability



Quantware, https://www.quantware.com/

Background

- 2D Grid Placement
  - Limited Flexibility
- Space Utilization
  - Low
- Scalability
  - High Labor Requirements



Quantware, https://www.quantware.com/

Why Placement Problem is Hard to Resolve?

- Lower Bound Constraints
  - Inter-Component Crosstalk
  - Design Rules (Overlap)
- Upper Bound Constraints
  - Spurious Mode
  - Chip Cost

Lower Bound Constraint: Crosstalk

- Definition in Quantum
  - Unintended interactions between quantum components, caused by resonating frequencies of element are connected or positioned in close proximity

Lower Bound Constraint: Crosstalk

- Definition in Quantum
  - Unintended interactions between quantum components, caused by resonating frequencies of element are connected or positioned in close proximity

Frequency Domain **Spatial Domain**

Lower Bound Constraint: Crosstalk

- Definition in Quantum
  - Unintended interactions between quantum components, caused by resonating frequencies of element are connected or positioned in close proximity

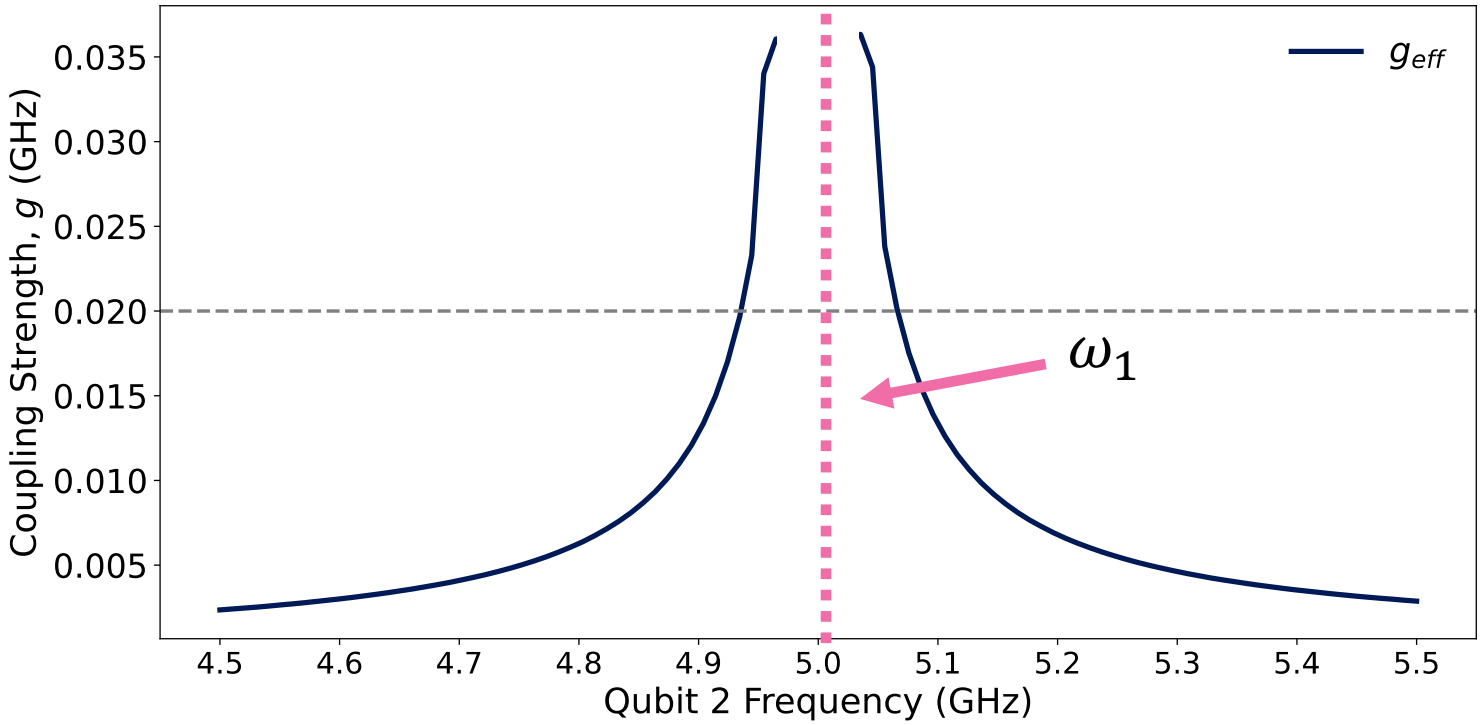
Frequency Domain **Spatial Domain**

- Mathematical Representation
  - Coupling strength (g) in Hamiltonian model

- Source of Crosstalk
  - Inter-Qubits interactions
  - Inter-Resonators interactions

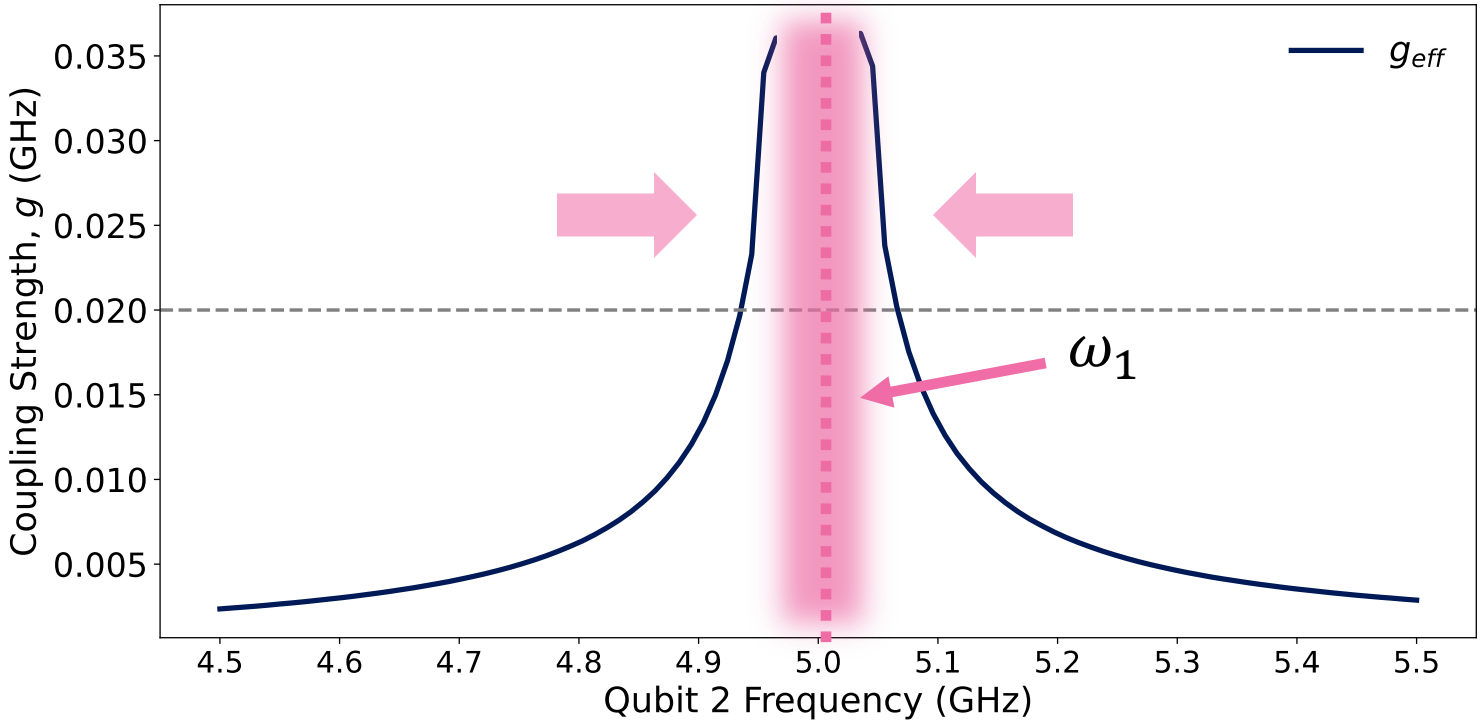
Lower Bound Constraint: Crosstalk

- Inter-Qubit Crosstalk (Frequency Domain)
  - Coupling strength between two directly connected qubits (Frequency: ω_1, ω_2)



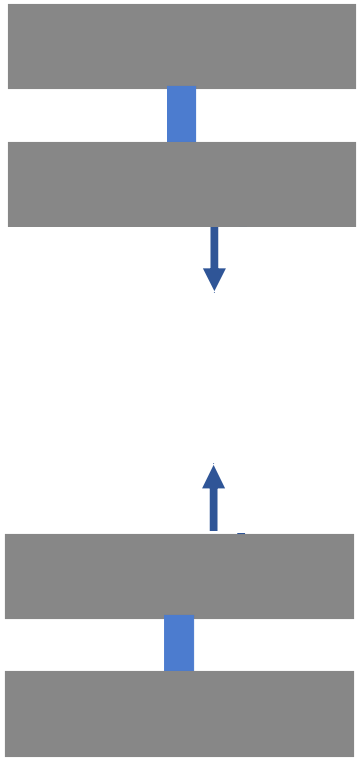
Lower Bound Constraint: Crosstalk

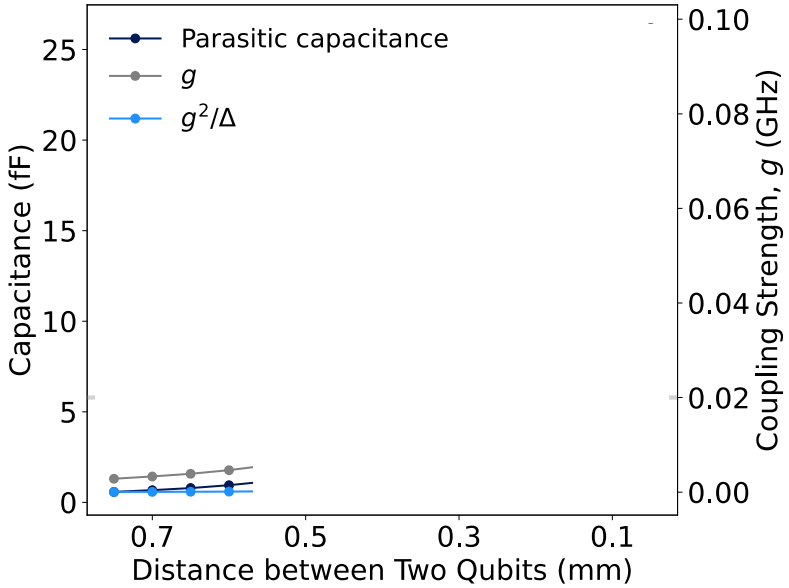
- Inter-Qubit Crosstalk (Frequency Domain)
  - Coupling strength between two directly connected qubits (Frequency: ω_1, ω_2)
  - **Peak** coupling strength occurs at $\omega_1 = \omega_2$



Lower Bound Constraint: Crosstalk

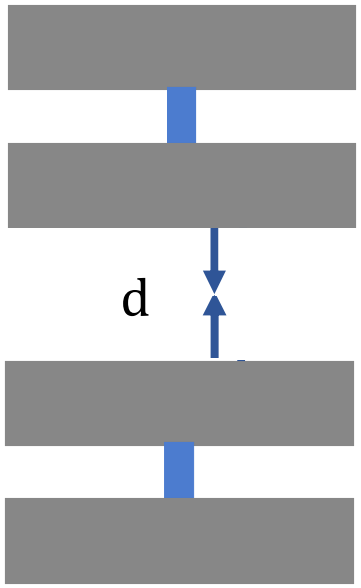
- Inter-Qubit Crosstalk (Spatial Domain)
  - Separation distance between qubits: $d\downarrow$

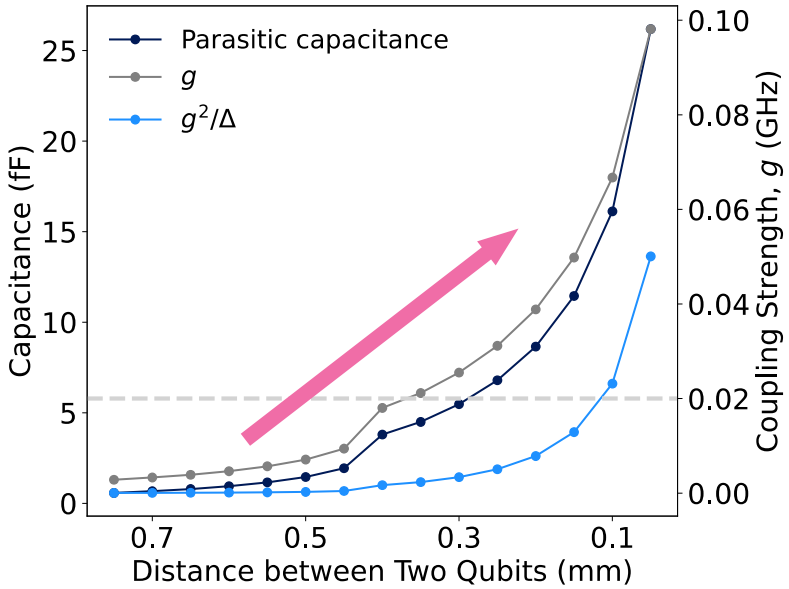




Lower Bound Constraint: Crosstalk

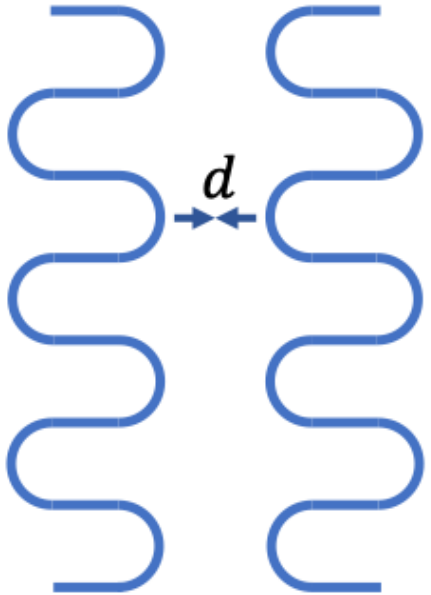
- Inter-Qubit Crosstalk (Spatial Domain)
  - Separation distance between qubits: $d\downarrow$
  - Parasitic capacitance: $C_p\uparrow$
  - Effective coupling strength: $g_{\mathrm{eff}}\uparrow$

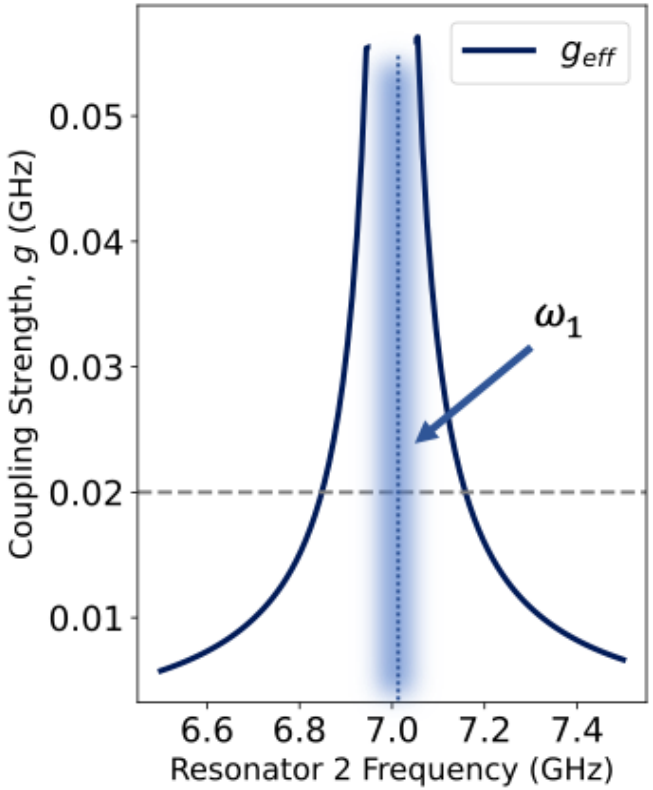


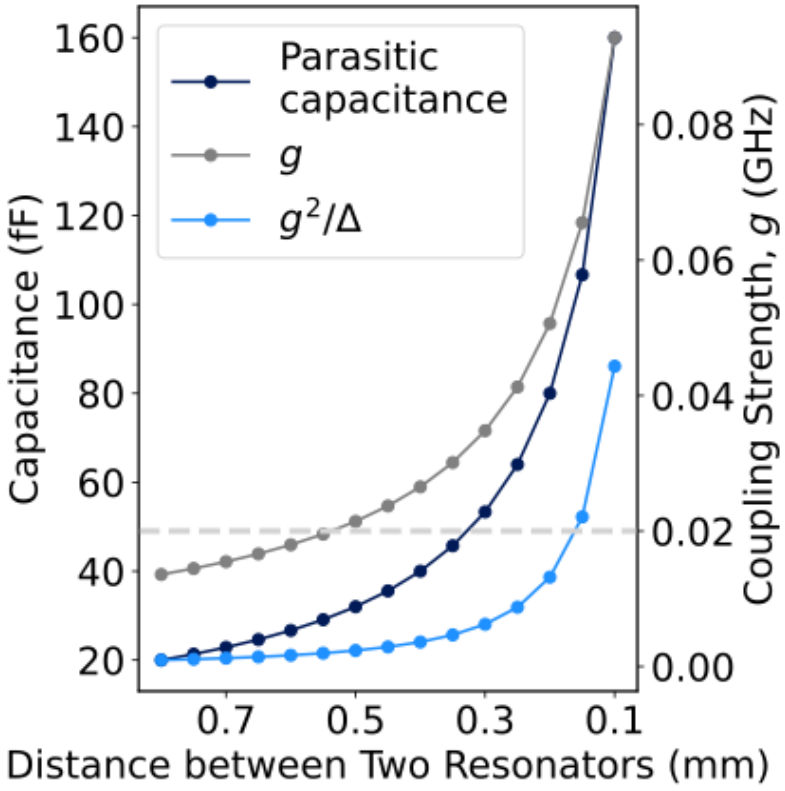


Lower Bound Constraint: Crosstalk

- Resonator Crosstalk (Frequency & Spatial)







Upper Bound Constraints

- Spurious mode
  - Area of substrate ↑, its frequency ↓
  - Low substrate frequency compromises quantum chip
- Substrate Cost with Larger Area
  - High-cost cooling systems
  - Increased manufacturing expenses

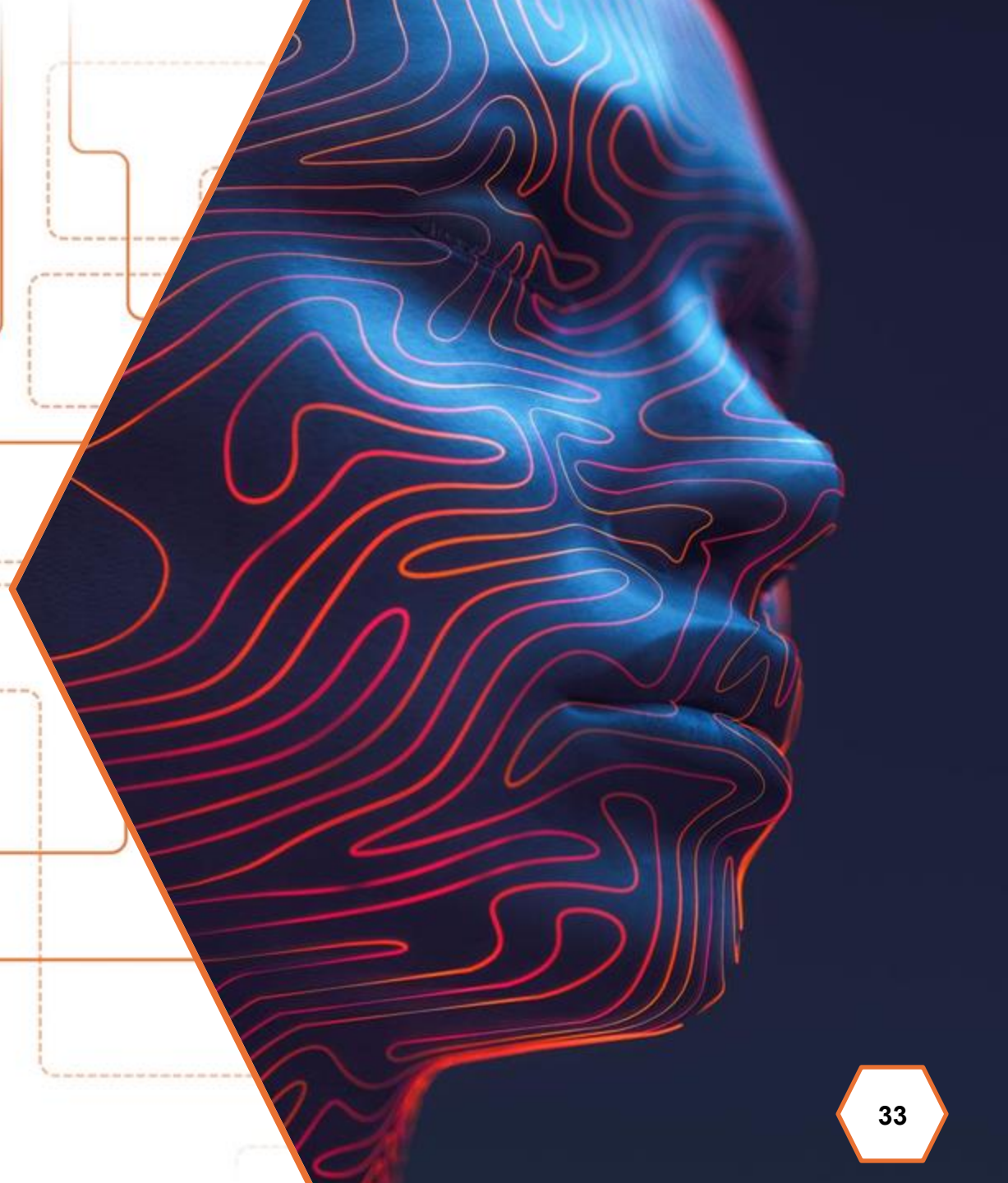
Upper Bound Constraints

- Spurious mode
  - Area of substrate ↑, its frequency ↓
  - Low substrate frequency compromises quantum chip
- Substrate Cost with Larger Area
  - High-cost cooling systems
  - Increased manufacturing expenses

Enhancing Area Utilization

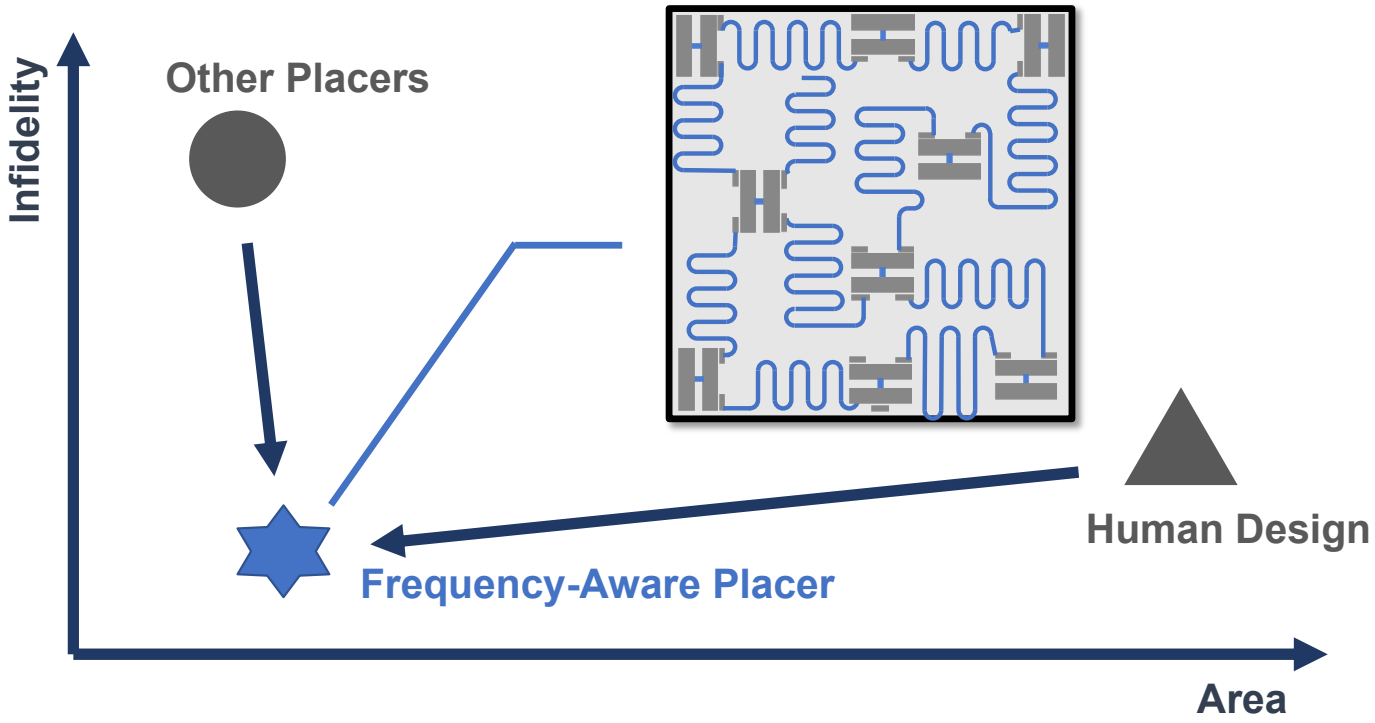
Agenda

- Motivation
- Background
- Problem Formulation
- **Qplacer – Global Placement**
- qGDP – Legalization & Detailed Placement



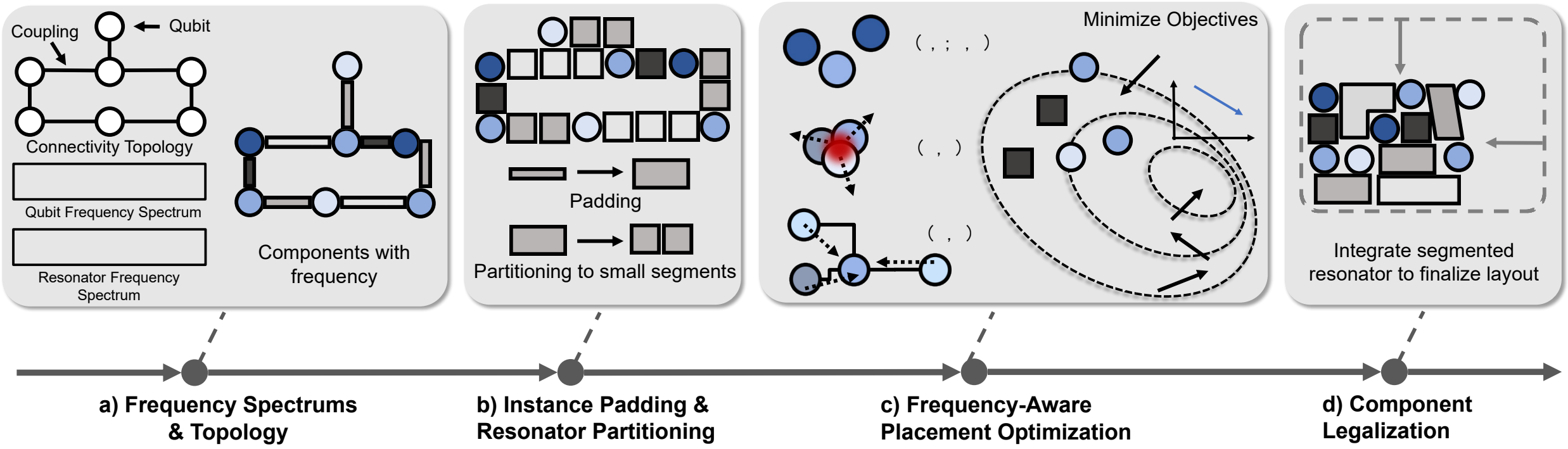
Placement Solution

- Qplacer[1], a **frequency-aware electrostatic-based placement framework**
- Resolves crosstalk in **spatial** and **frequency** domains with **compact substrate**



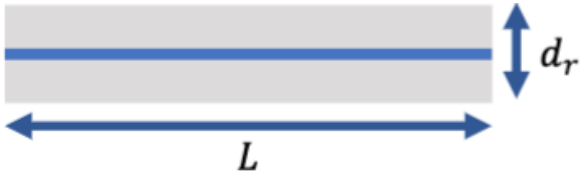
[1] Zhang, Junyao, et al. “Qplacer: Frequency-Aware Component Placement for Superconducting Quantum Computers.” (ISCA 2025)

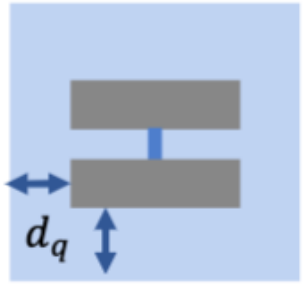
QPlacer Methodology



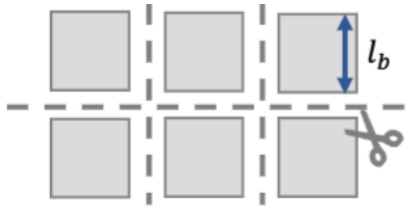
Placement Component Preprocessing

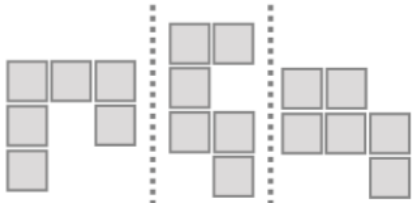
- Padding
  - Padding both resonators and qubits for minimum spacing

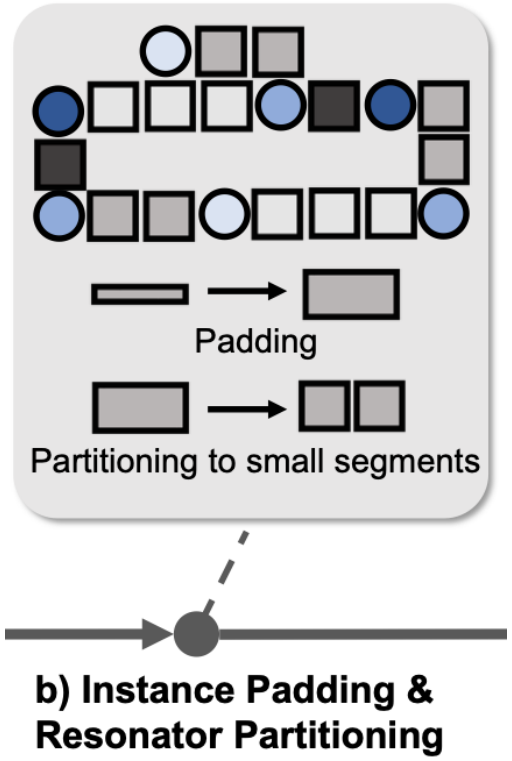




- Resonator Space Partitioning
  - For strong flexibility in placing resonators
  - Resonator space is partitioned rather than resonator itself



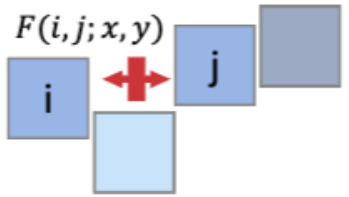


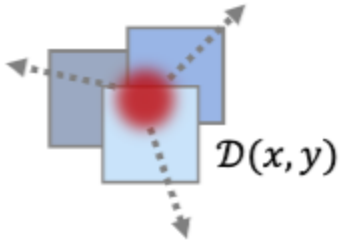


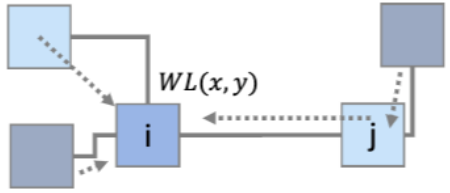
b) Instance Padding & Resonator Partitioning

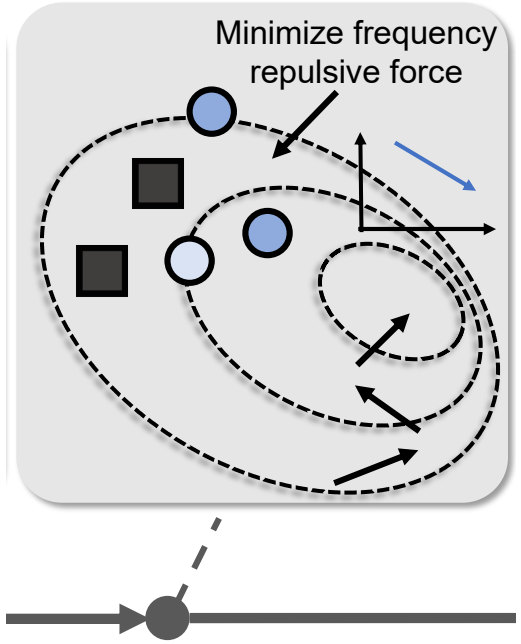
Placer Optimization Objective Design

- Frequency Force
  - Resolve crosstalk in spatial domain
- Components Density
  - Spread out the components to the whole substrate
- Wirelength
  - Draw components closer to minimize the substrate area (increase the area utilization)







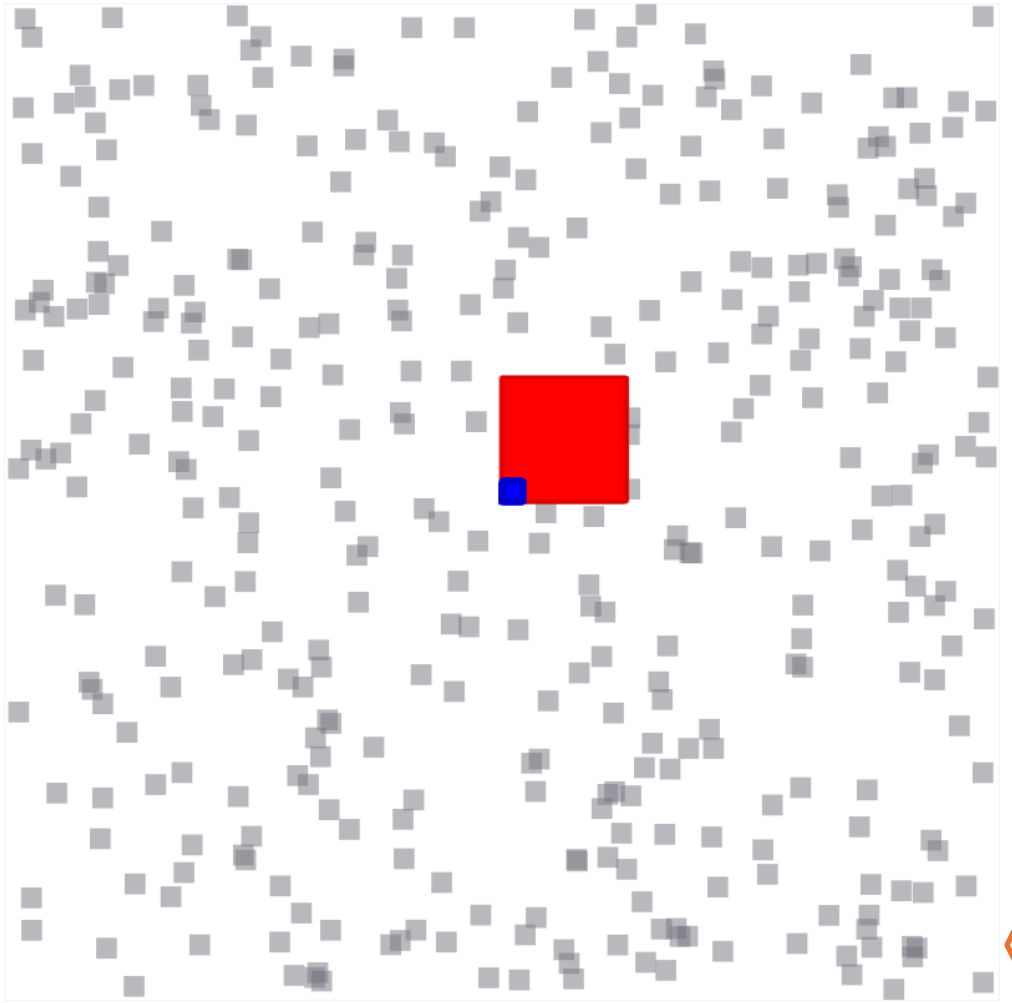


c) Frequency-Aware Placement Optimization

$$\min_{x,y} \sum_{i,j\in I;e\in E} \mathrm{WL}(e;x,y) + \lambda\mathcal{D}(x,y) + \lambda_f F(i,j;x,y)$$

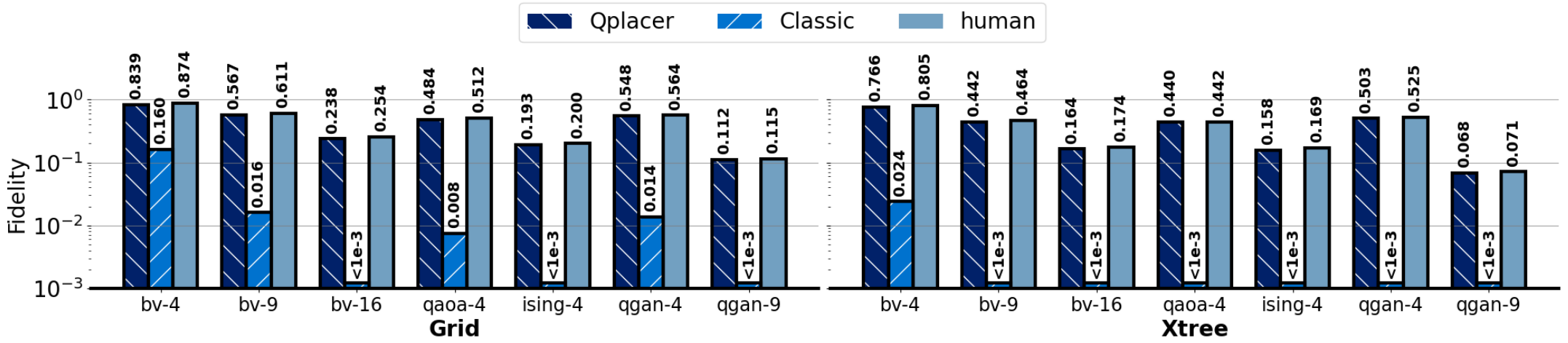
Optimization Demo

- **Red**: Qubits
- **Blue**: Partitioned resonator segments
- **Gray**: Fillers (balanced the potential in electrostatic field)



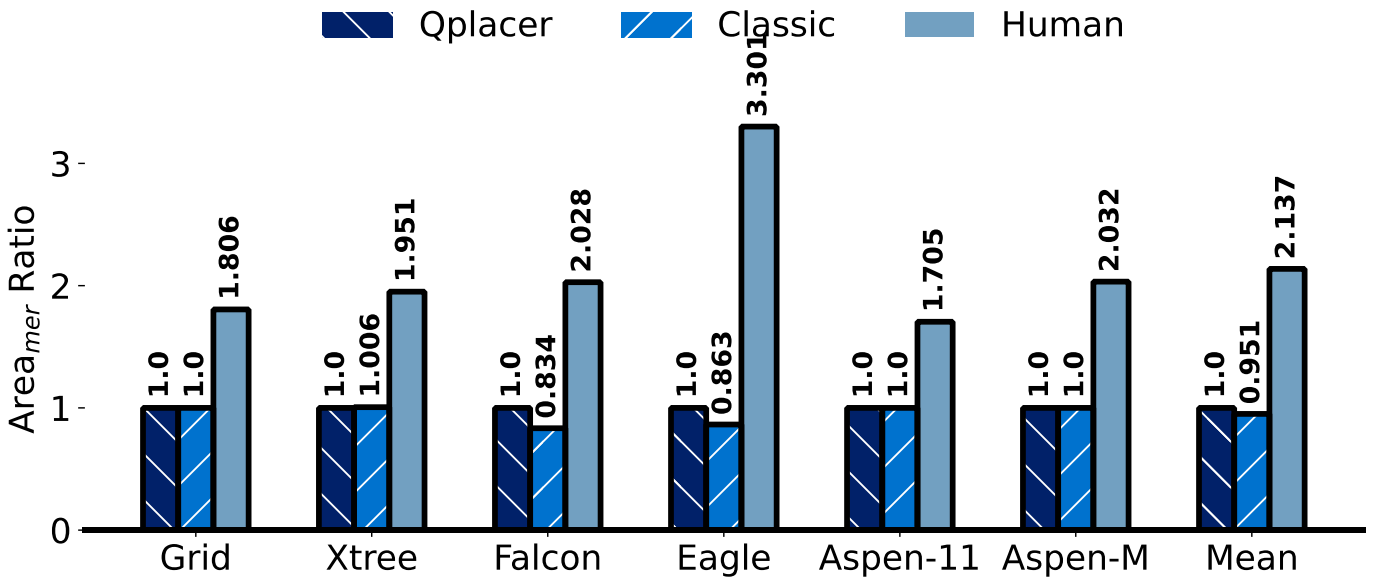
Substrate Fidelity Estimation Results

- Program Successful Rate
  - Higher value indicates better performance
  - 50 different Mappings (subset of physical qubits) to evaluate overall substrate
  - Qiskit Transpiler (minimize software-level impacts, focus on substrate evaluation)



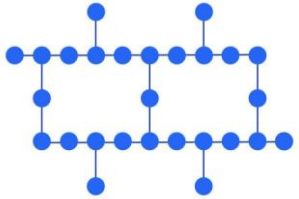
Area Optimization Results

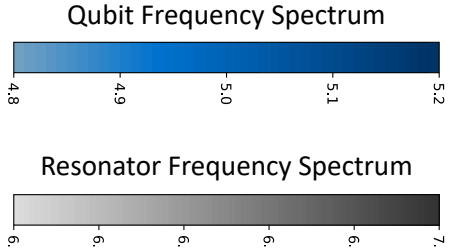
- Minimum enclosing rectangle area of layout
  - Smaller area is preferred.
  - QPlacer and Classical placer achieve similar results (same hyperparameter)
  - QPlacer outperforms Human designs in area efficiency



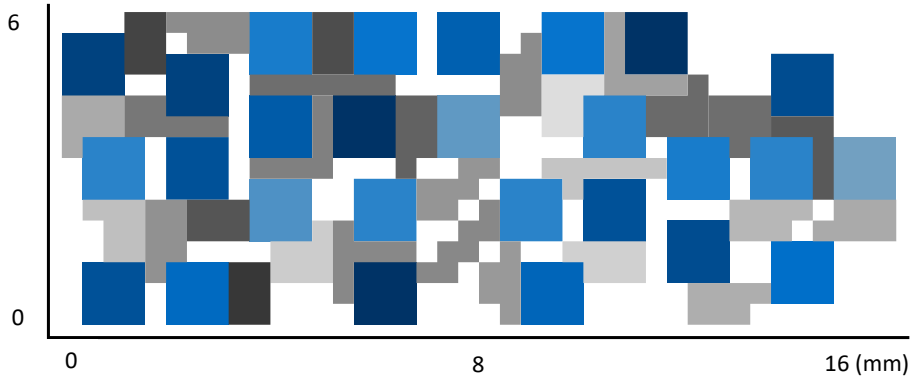
Layout Demo

- Connectivity Topology & Component Spectrum



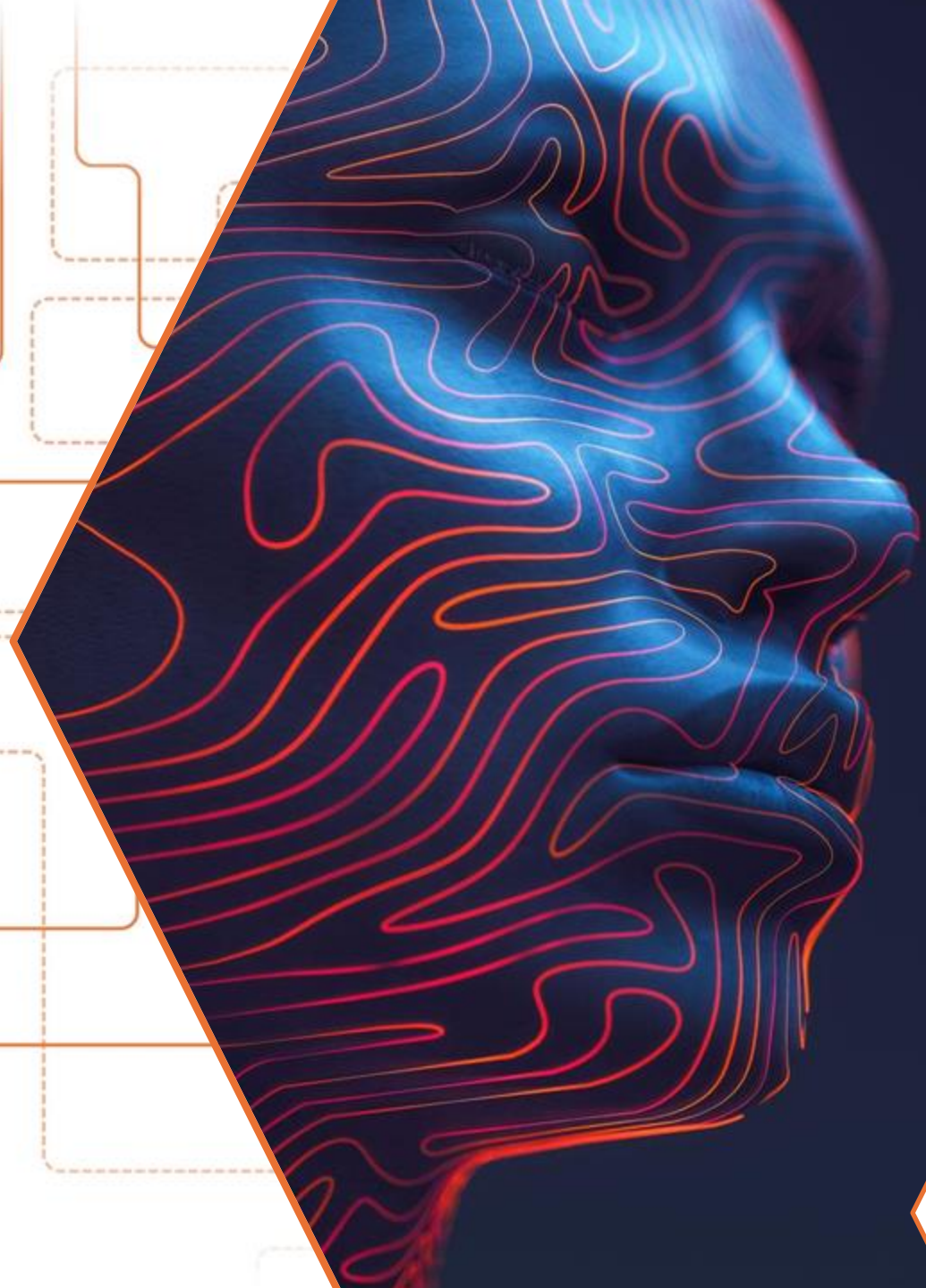


- Optimized Layout by QPlacer & **Generated GDS file based on Layout**



Agenda

- Motivation
- Background
- Problem Formulation
- Qplacer – Global Placement
- **qGDP – Legalization & Detailed Placement**



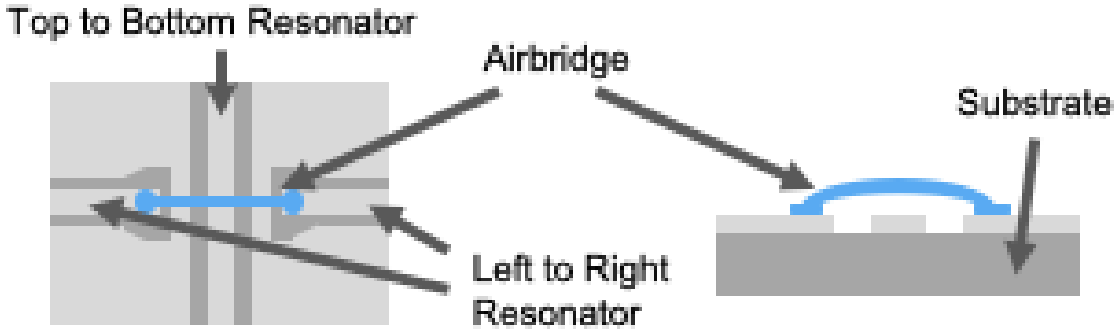
Quantum Legalization & Detailed Placement

- Quantum Legalization[2]
  - Resolves design rule violations from global placement (e.g., overlap, boundary checks)
  - Addresses resonator integration
- Quantum Detail Placement[2]
  - Refine post-global-placement configurations to enhance layout quality.

[2] Zhang, et, al, “qGDP: Quantum Legalization and Detailed Placement for Superconducting Quantum Computers” DATE 2025

Background

- Resonator Crossing
  - Utilizes airbridges to achieve crossings between resonators
  - Airbridges introduce **Additional Crosstalk** between resonators



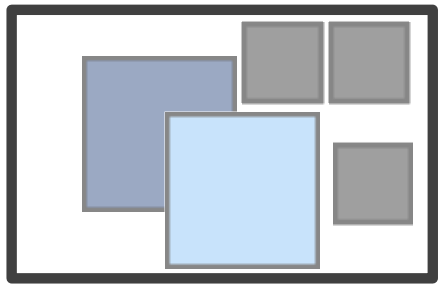
- Problem Formulation
  - **Minimize crossing** (airbridge usage) and **reduce Crosstalk** (frequency hotspots)

$$\text{Minimize}(\sum_{e \in E} |C_e|)$$

$$\text{Minimize } (P_h = \frac{\sum_{i,j \in G} (p_i \cap p_j) \cdot d_c(p_i, p_j) \cdot \tau(\omega_i, \omega_j, \Delta_c)}{\sum_{n \in G} w_n * h_n})$$

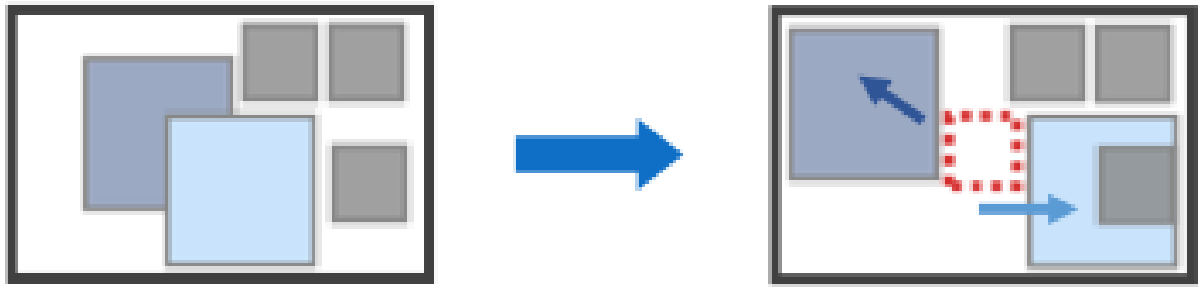
qGDP Methodology - Legalization

- Qubit Legalization
  - Qubits are treated as macros, resonator blocks are treated as standard cells



qGDP Methodology - Legalization

- Qubit Legalization
  - Qubits are treated as macros, resonator blocks are treated as standard cells
  - Minimum Space Injection (at least one standard cell space between qubits)

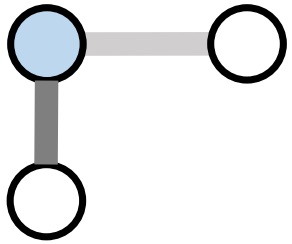


  - Minimum Qubit displacement to preserve GP layout result

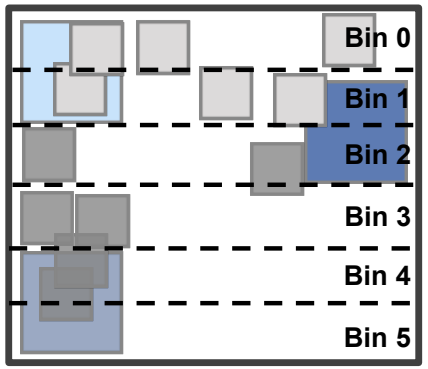
$$\text{Min}(\sum_{i})$$

qGDP Methodology - Legalization

- Resonator Legalization - Integration-aware
  - Qubit Connectivity



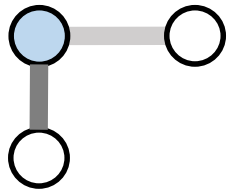
- Adopt Bin-aided Indexing to reduce query complexity[1] $O(n) \rightarrow O(\log n)$

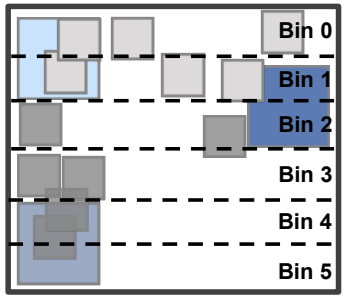


[1] H. Yang, et, al, “Mixed-cell-height legalization on cpu-gpu heterogeneous systems,” DATE 2022

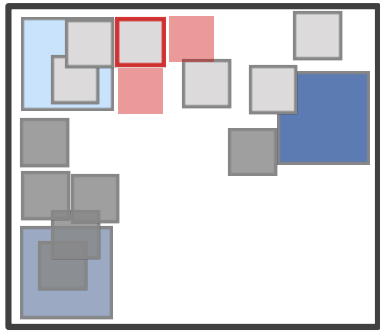
qGDP Methodology - Legalization

- Resonator Legalization - Integration-aware
  - Qubit Connectivity and Initial layout



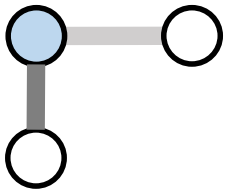


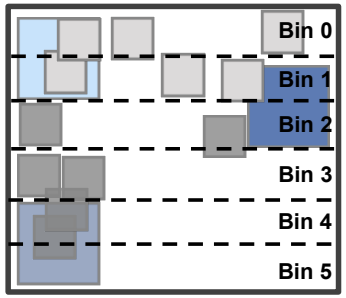
- Legalization Process (Gray resonator)



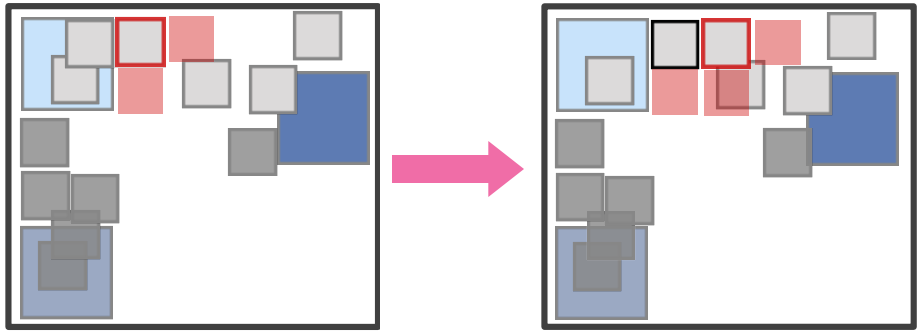
qGDP Methodology - Legalization

- Resonator Legalization - Integration-aware
  - Qubit Connectivity and Initial layout



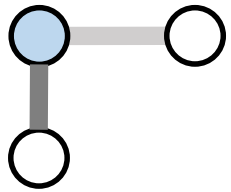


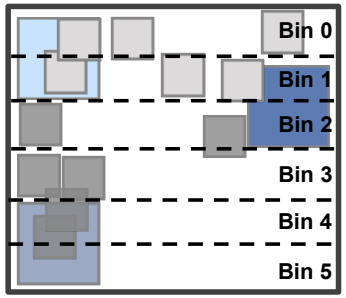
- Legalization Process (Gray resonator)



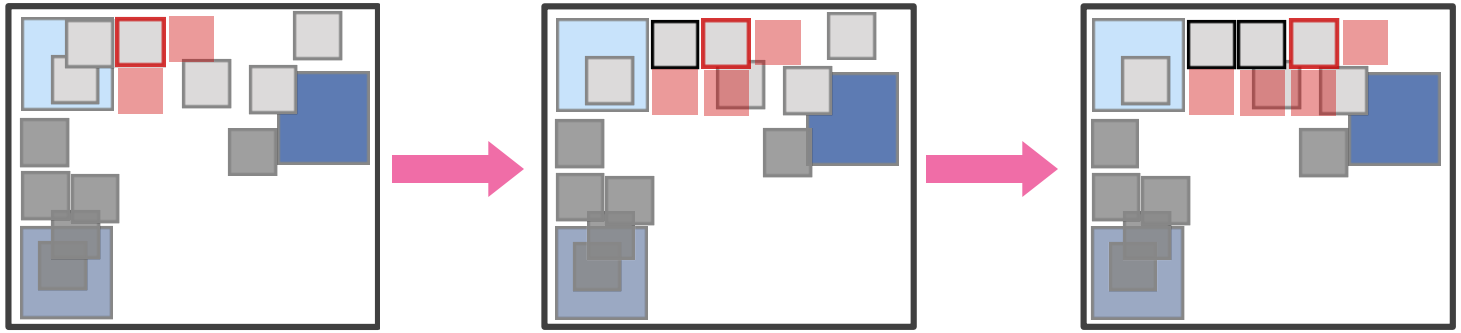
qGDP Methodology - Legalization

- Resonator Legalization - Integration-aware
  - Qubit Connectivity and Initial layout



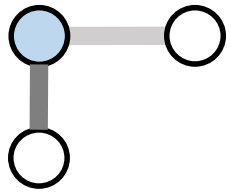


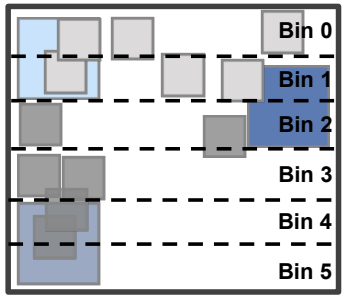
- Legalization Process (Gray resonator)



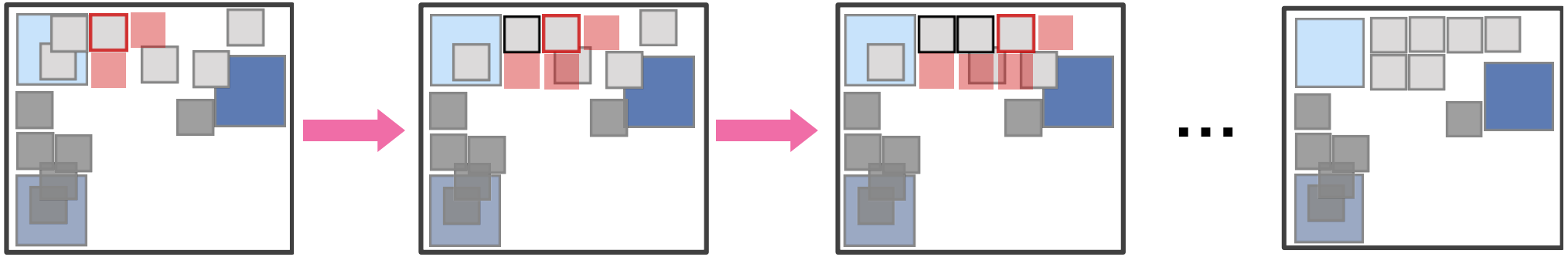
qGDP Methodology - Legalization

- Resonator Legalization - Integration-aware
  - Qubit Connectivity and Initial layout





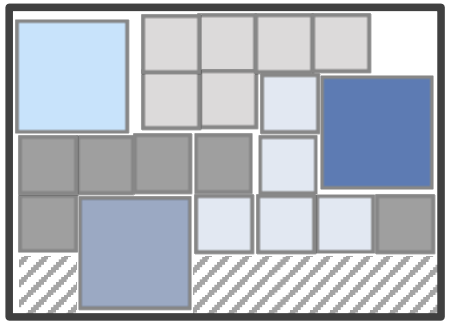
- Legalization Process (Gray resonator)



qGDP Methodology – Detailed Placement

- Detailed Placement
  - Sliding window

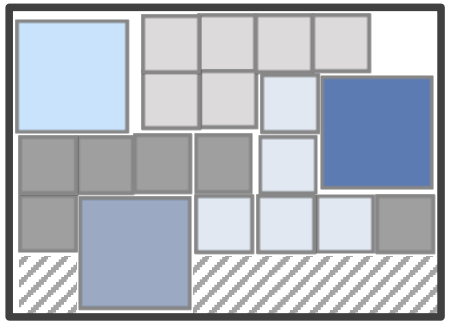
Legalized Layout



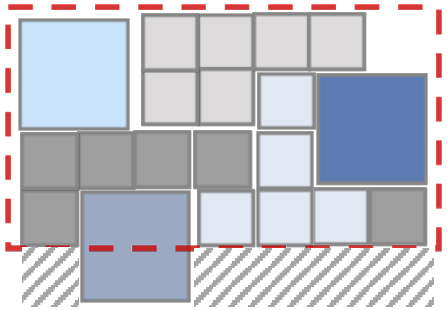
qGDP Methodology – Detailed Placement

- Detailed Placement
  - Sliding window

Legalized Layout

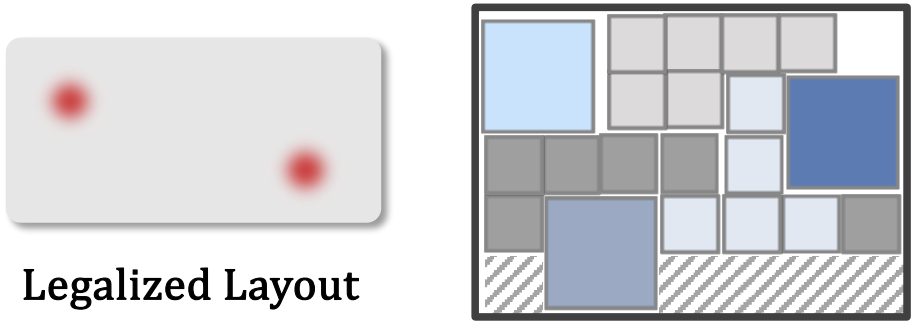


- Extract all the resonator blocks for rerouting

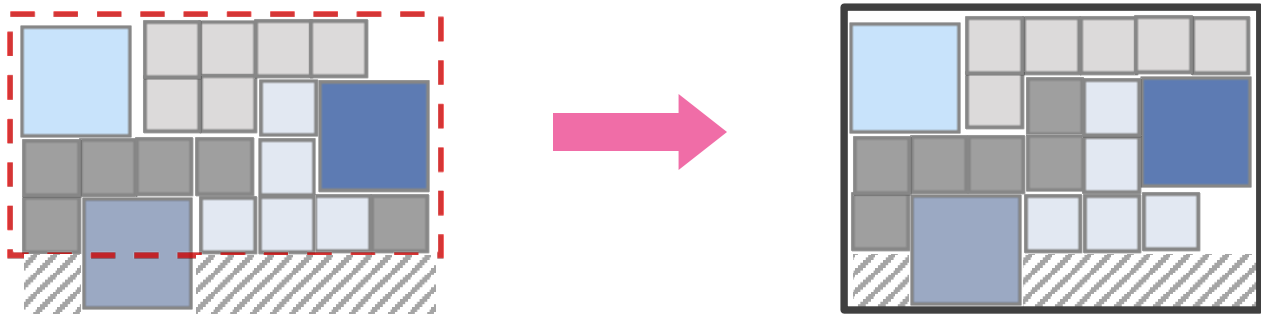


qGDP Methodology – Detailed Placement

- Detailed Placement
  - Sliding window

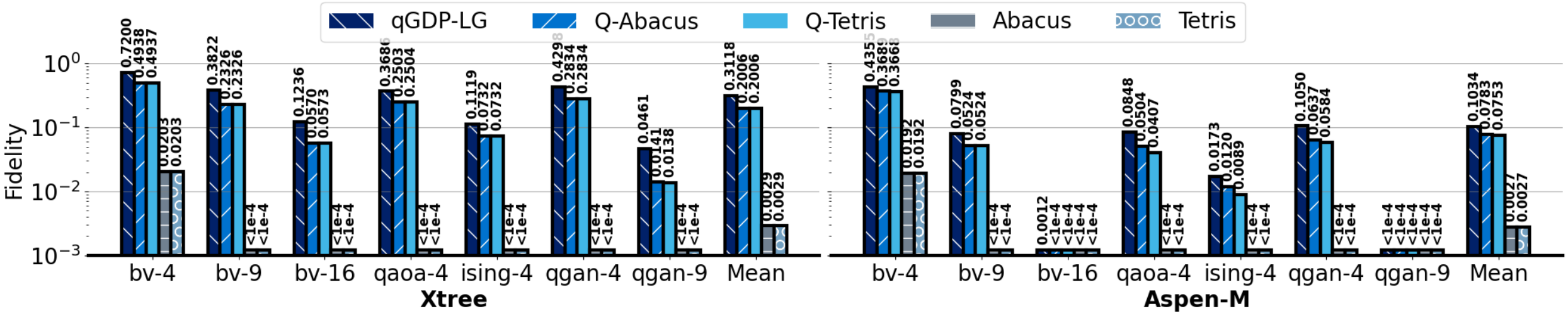


- Extract all the resonator blocks for rerouting



qGDP Substrate Fidelity Estimation Results

- Program Successful Rate
  - Higher value is better



LG & DP Comparison

- Comparison between Legalization and Detailed Placement
  - qGDP-DP further improves the quality of layout from multiple metrics

TABLE III: Detailed Placement Evaluation, I_{edge} is the number of unified resonators over total resonators (higher the better), X for resonator crossings, $P_h(\%)$ for the proportion of frequency hotspots, and H_Q, the number of qubits affected by hotspots, with lower values preferred for the last three metrics.

Topology	#Cells	qGDP-LG				qGDP-DP			
		I_{edge}	X	$P_h(\%)$	H_Q	I_{edge}	X	$P_h(\%)$	H_Q
Grid	490	37/40	3	1.38	11	37/40	3	0.81	5
Xtree	660	47/52	5	1.37	20	52/52	0	0.34	10
Falcon	354	28/28	0	0.92	8	28/28	0	0	0
Eagle	1801	142/144	2	1.27	68	143/144	1	0.32	15
Aspen-11	598	46/48	2	0.91	20	48/48	0	0.66	9
Aspen-M	1310	98/106	8	2.71	50	103/106	3	0.76	14

Insights

- Placement problem as a critical yet previously overlooked issue within the quantum computing community.
- Innovative placement strategy not only resolves crosstalk but also has the potential to mitigate other challenges (e.g. decoherence, leakage)
- Other chip components can be considered (airbridges, vias, and control lines).
- Resonator Routing is still an area requiring developments.

AI

Security

Systems

EDA

Design

62 THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA® IEEE Council on Electronic Design Automation

sigda acm

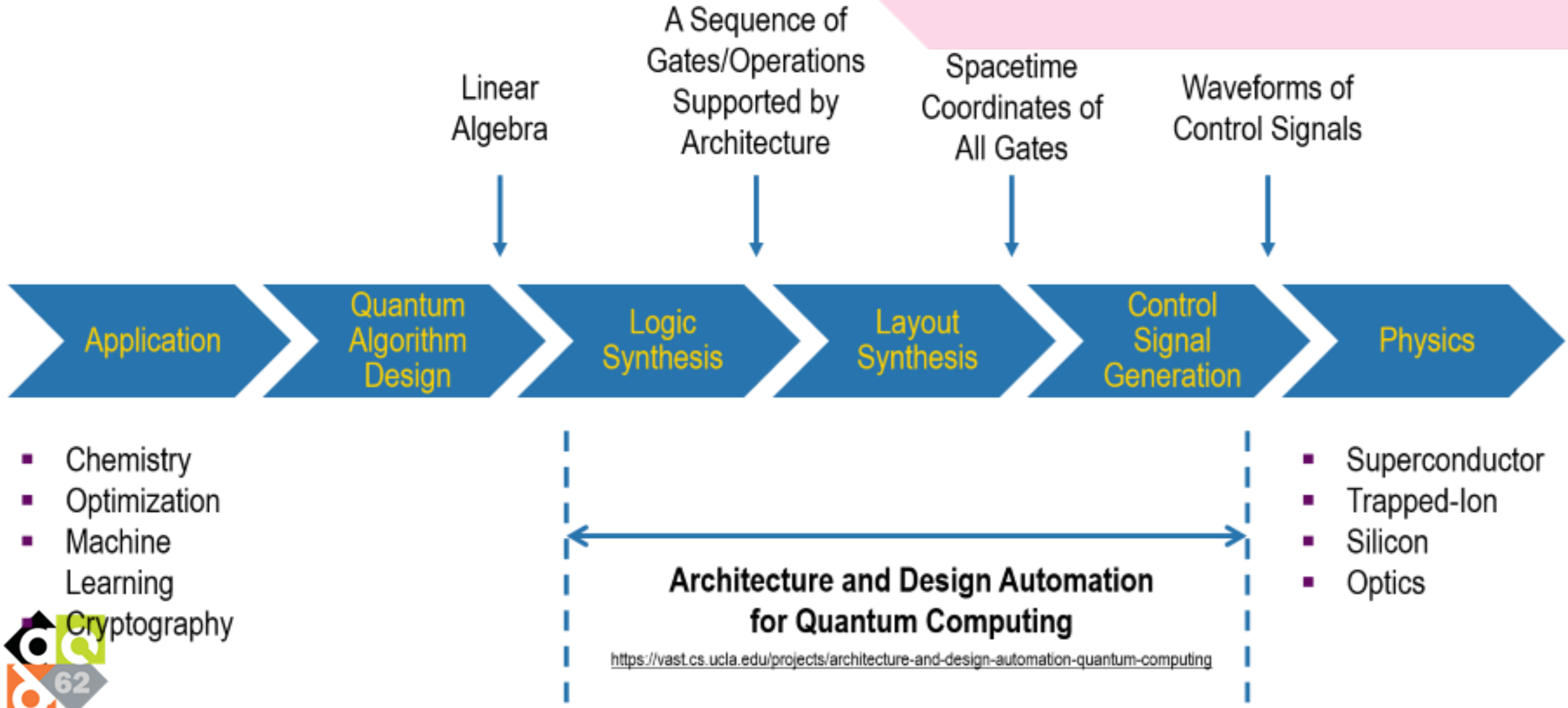
State Preparation Circuit Compilation

Daniel Tan, 6/22/25

THE CHIPS TO SYSTEMS CONFERENCE

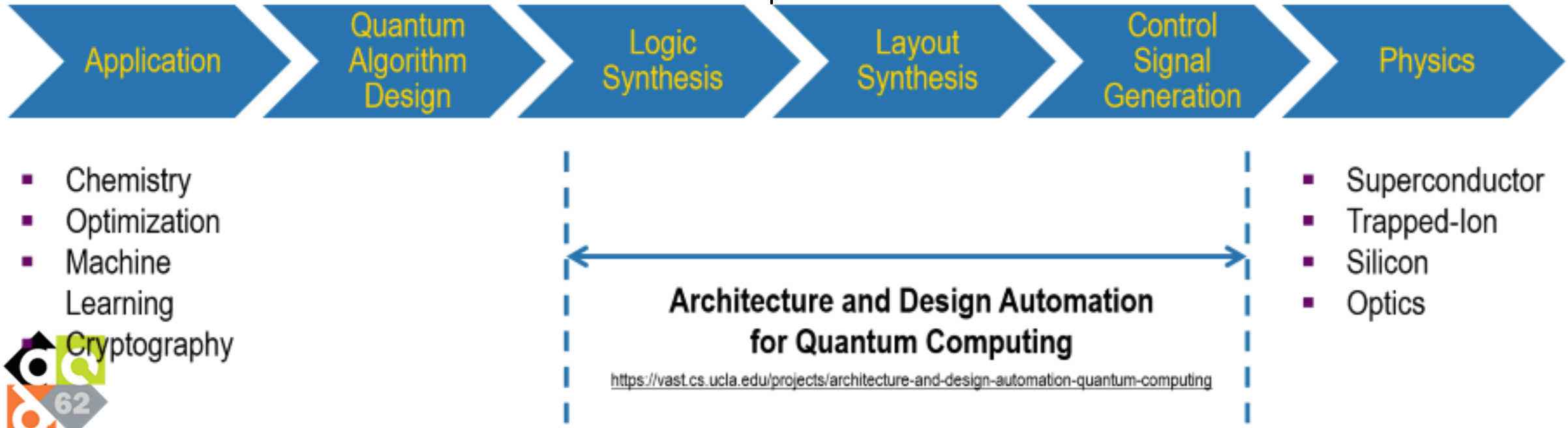
SPONSORED BY CEDA sigda

Compilation Flow for Quantum Computing



Compilation Flow for Quantum Computing

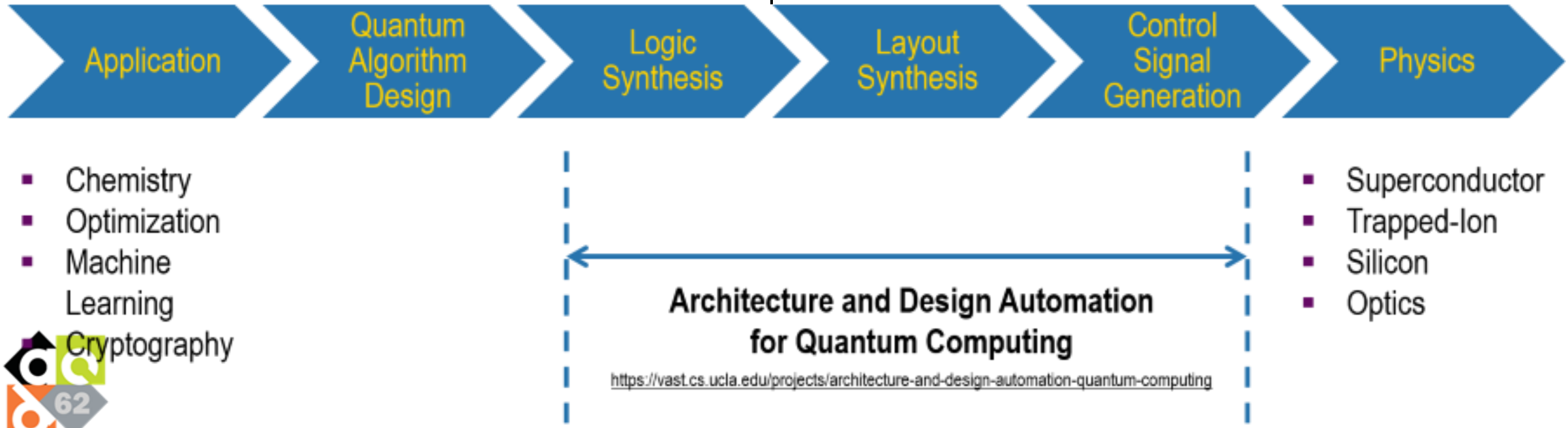
OLSQ [Tan+, ICCAD'20]
QLS-DPQA [Tan+, ICCAD'22]
Scalable QLS [Lin+, DAC'23]
Enola [Tan+, ASPDAC'25]
ML-QLS [Lin+, ISPD'25]
ZAC [Lin+, HPCA'25]
QUBIKOS [Ping+, DAC'25]



Compilation Flow for Quantum Computing

QSP Synthesis [Wang+, DATE'24]
QSP Resynthesis [Wang+, ICCAD'24]

OLSQ [Tan+, ICCAD'20]
QLS-DPQA [Tan+, ICCAD'22]
Scalable QLS [Lin+, DAC'23]
Enola [Tan+, ASPDAC'25]
ML-QLS [Lin+, ISPD'25]
ZAC [Lin+, HPCA'25]
QUBIKOS [Ping+, DAC'25]

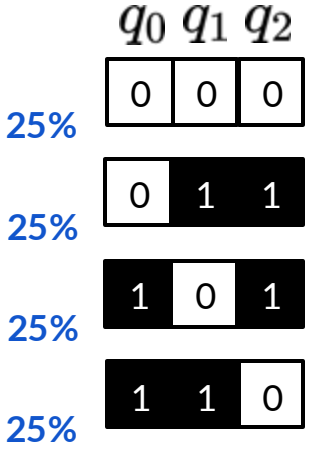


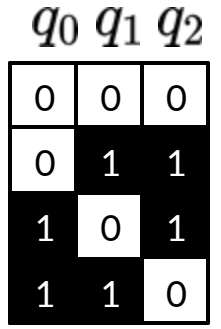
Quantum State Preparation (QSP)

QSP **initializes the registers** in quantum computers

Classical state

$$\begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}^T \cdot \begin{pmatrix} |000\rangle \\ |011\rangle \\ |101\rangle \\ |110\rangle \end{pmatrix} =$$





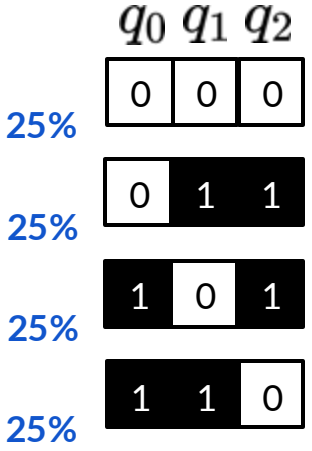
Quantum state

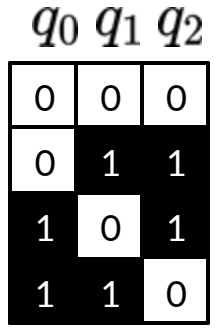
Quantum State Preparation (QSP)

QSP **initializes the registers** in quantum computers

Classical state

$$\begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}^T \cdot \begin{pmatrix} |000\rangle \\ |011\rangle \\ |101\rangle \\ |110\rangle \end{pmatrix} =$$

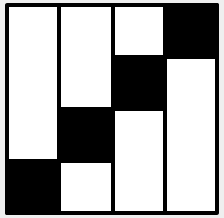




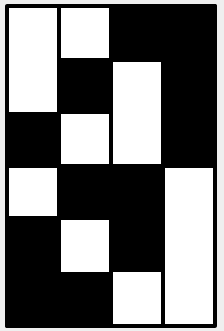
Quantum state

The initial states could be very **different**. Design **automation** is necessary.

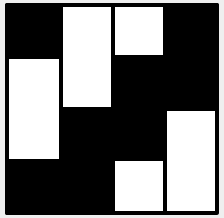
GHZ states
[Greenberger+, 89]



W states
[Dür+, PRA'00]



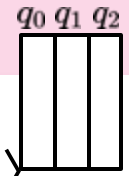
Dicke states
[Dicke+, PR'54]



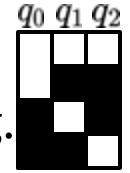
Cyclic states
[Kimbo+, Nature'08]

QSP Circuit Design Automation

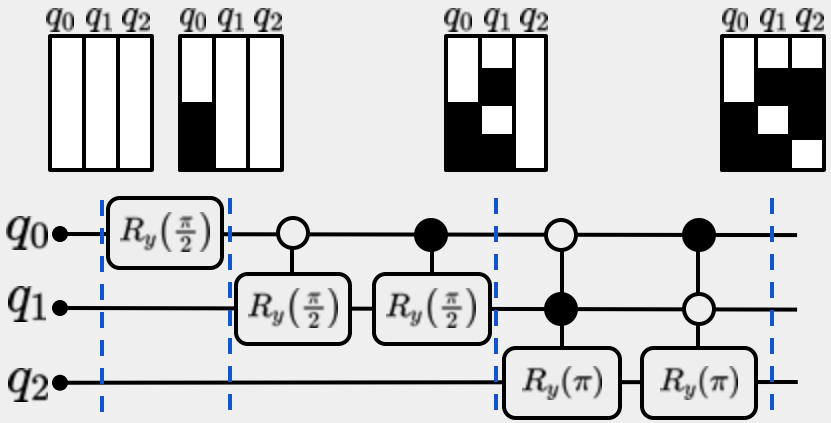
Problem formulation:



- Initially, qubits (e.g., q_0, q_1, and q_2) are cooled down to the **ground state** $|0\rangle$.
- Our goal is to find the simplest **physical process** to prepare a target state $|\psi\rangle$. E.g.



- …such that **CNOT cost** is minimized.

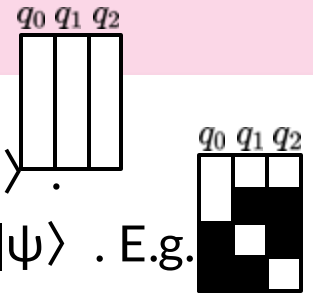


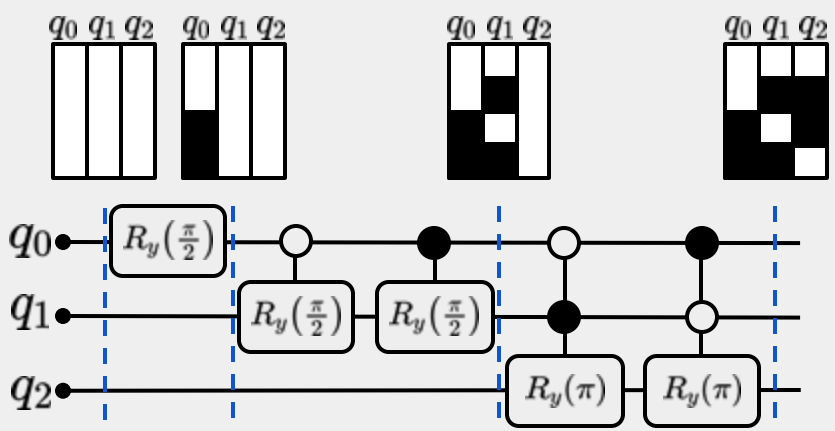
Qubit-reduction method [Mozafari+, 19]

QSP Circuit Design Automation

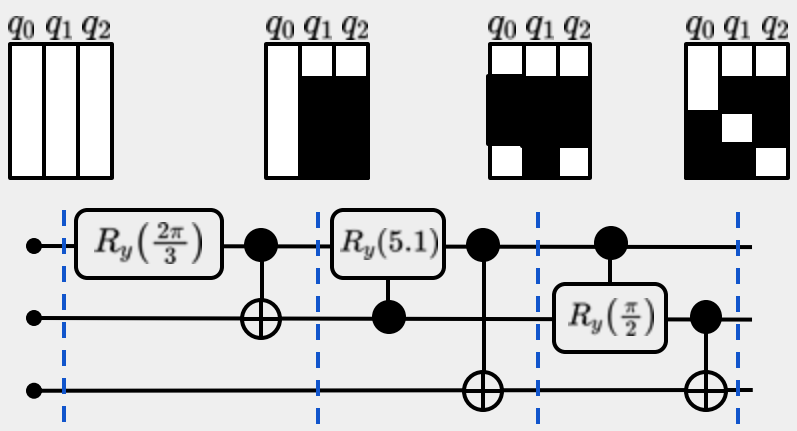
Problem formulation:

- Initially, qubits (e.g., q_0, q_1, and q_2) are cooled down to the **ground state** $|0\rangle$.
- Our goal is to find the simplest **physical process** to prepare a target state $|\psi\rangle$. E.g.
- …such that **CNOT cost** is minimized.





Qubit-reduction method [Mozafari+, 19]

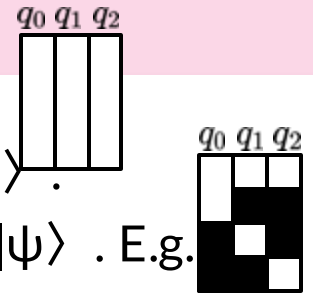


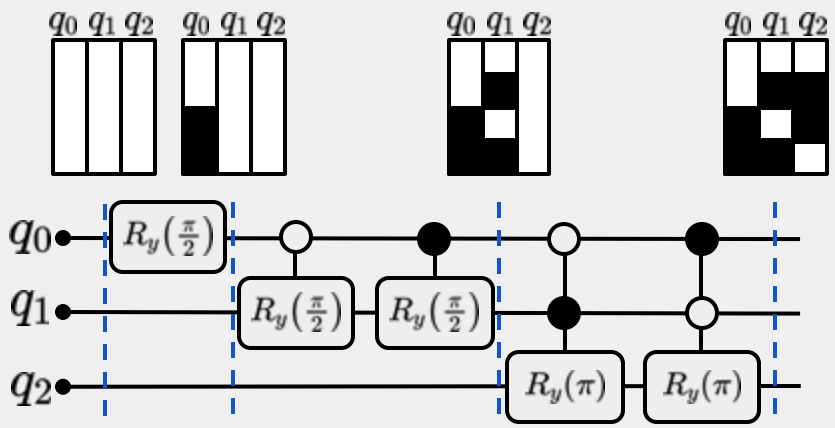
Cardinality-reduction method [Gleinig+, 21]

QSP Circuit Design Automation

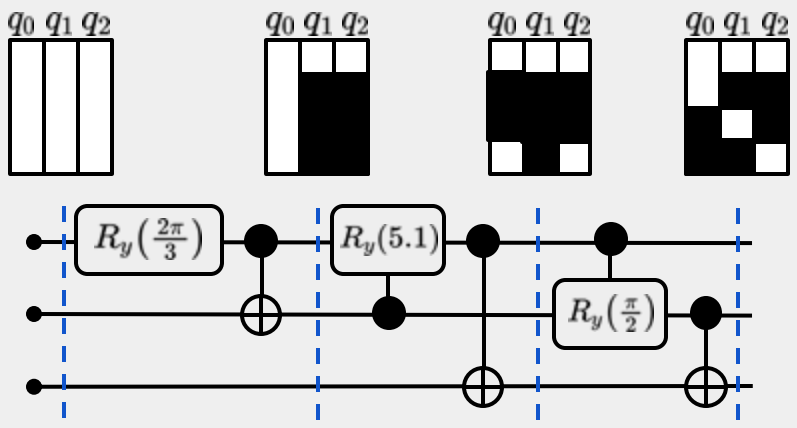
Problem formulation:

- Initially, qubits (e.g., q_0, q_1, and q_2) are cooled down to the **ground state** $|0\rangle$.
- Our goal is to find the simplest **physical process** to prepare a target state $|\psi\rangle$. E.g.
- …such that **CNOT cost** is minimized.

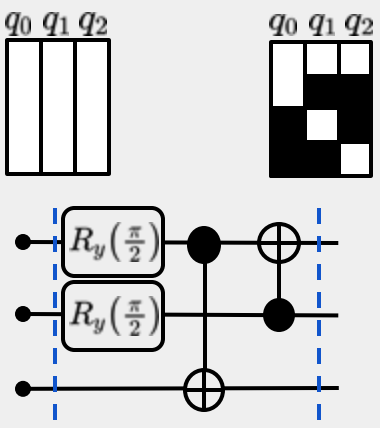




Qubit-reduction method [Mozafari+, 19]



Cardinality-reduction method [Gleinig+, 21]

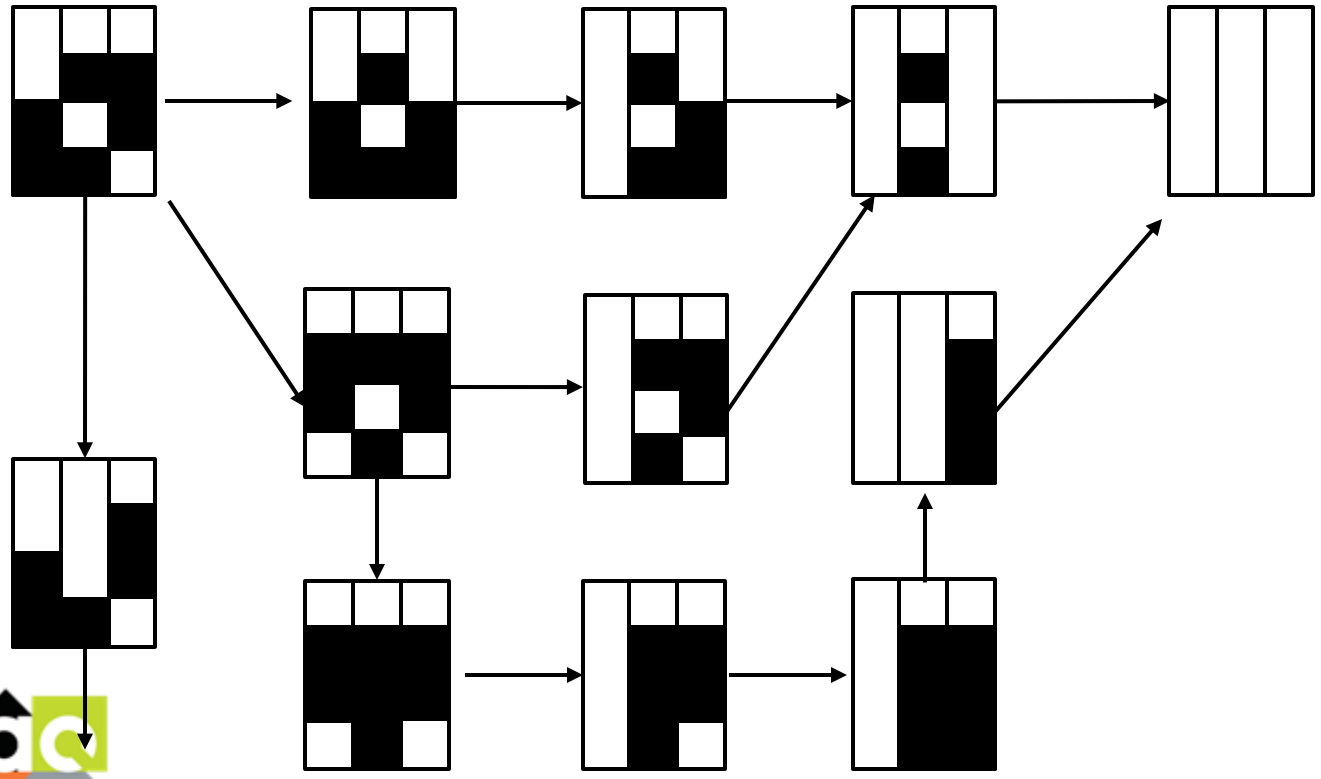


Exact CNOT Syn.[Wang+, 24]

Exact CNOT Synthesis [Wang+, DATE'24]

Key idea: formulate QSP problem on a graph, where:

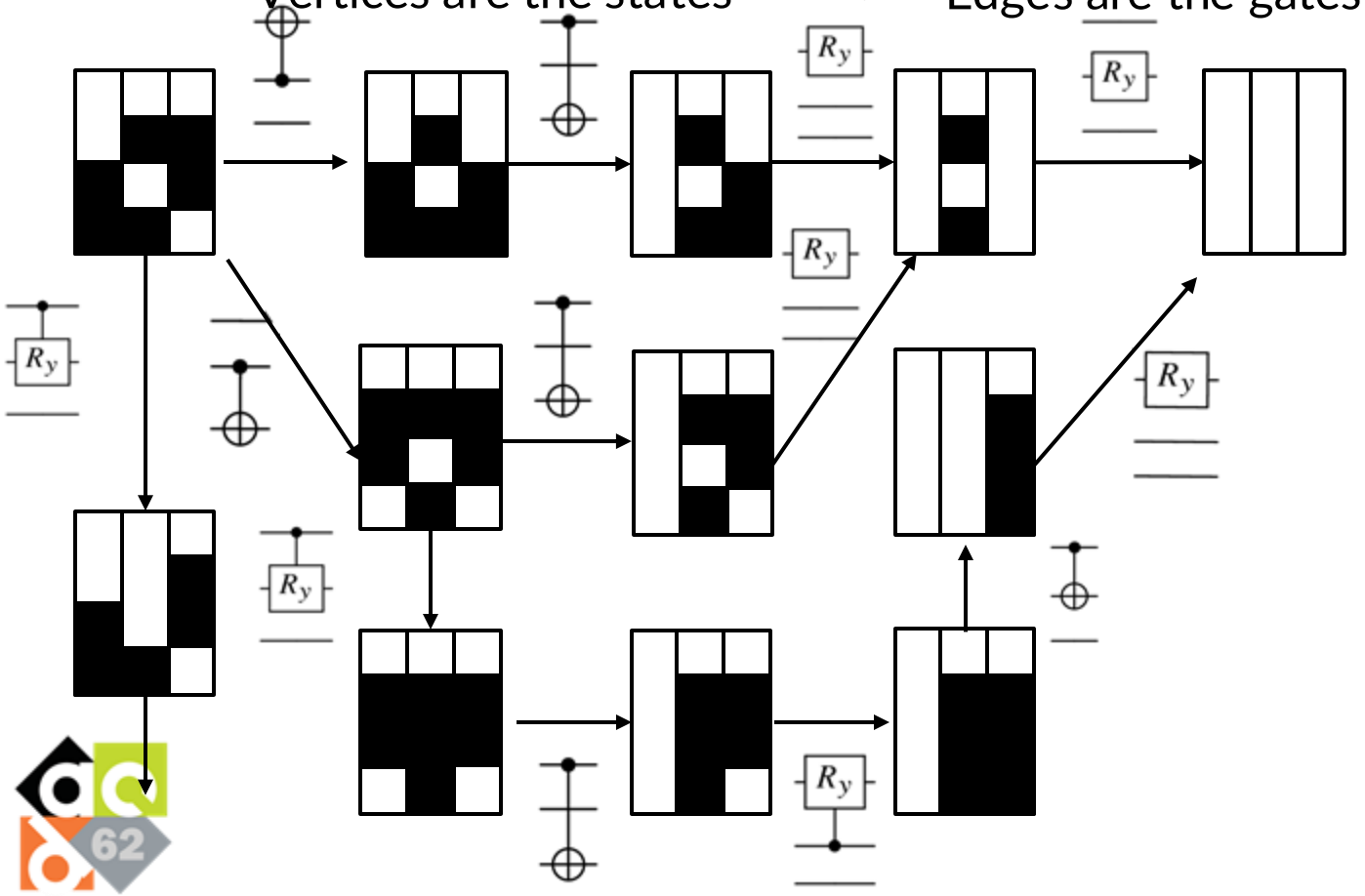
- Vertices are the states



Exact CNOT Synthesis [Wang+, DATE'24]

Key idea: formulate QSP problem on a graph, where:

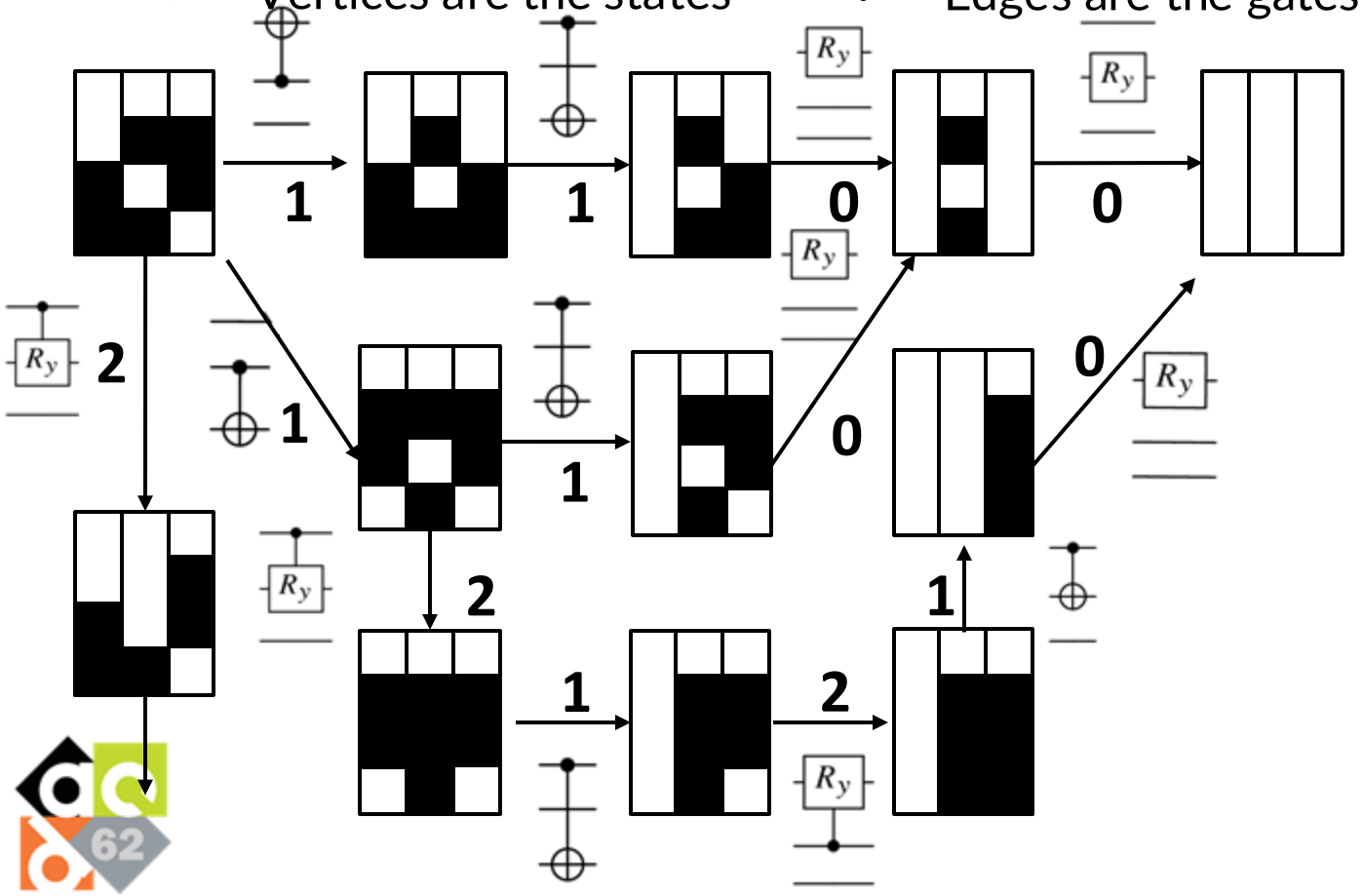
- Vertices are the states
- Edges are the gates



Exact CNOT Synthesis [Wang+, DATE'24]

Key idea: formulate QSP problem on a graph, where:

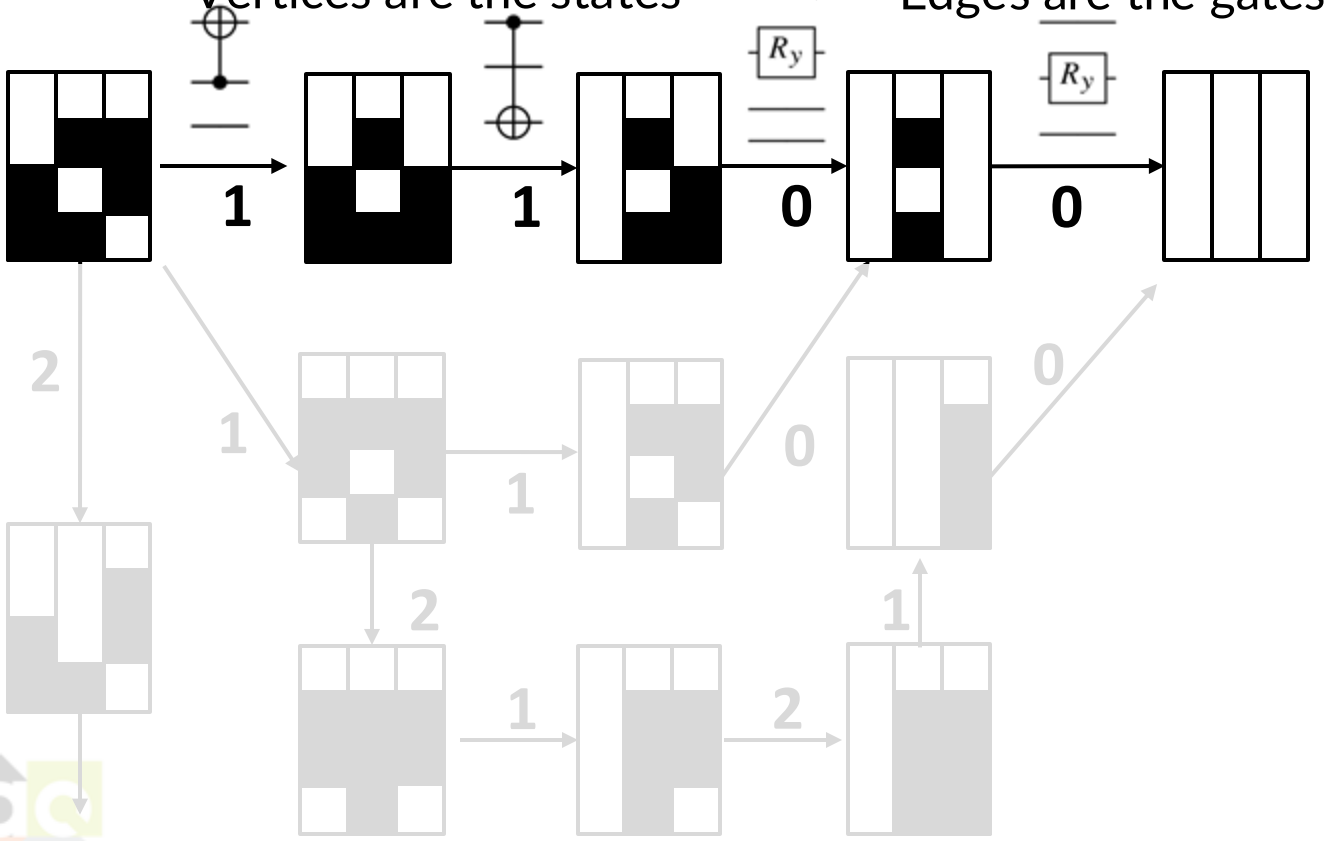
- Vertices are the states
- Edges are the gates
- Distance is the CNOT cost

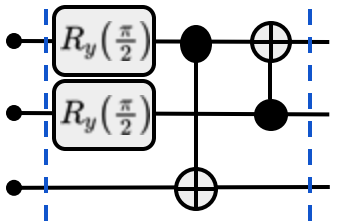


Exact CNOT Synthesis [Wang+, DATE'24]

Key idea: formulate QSP problem on a graph, where:

- Vertices are the states
- Edges are the gates
- Distance is the CNOT cost



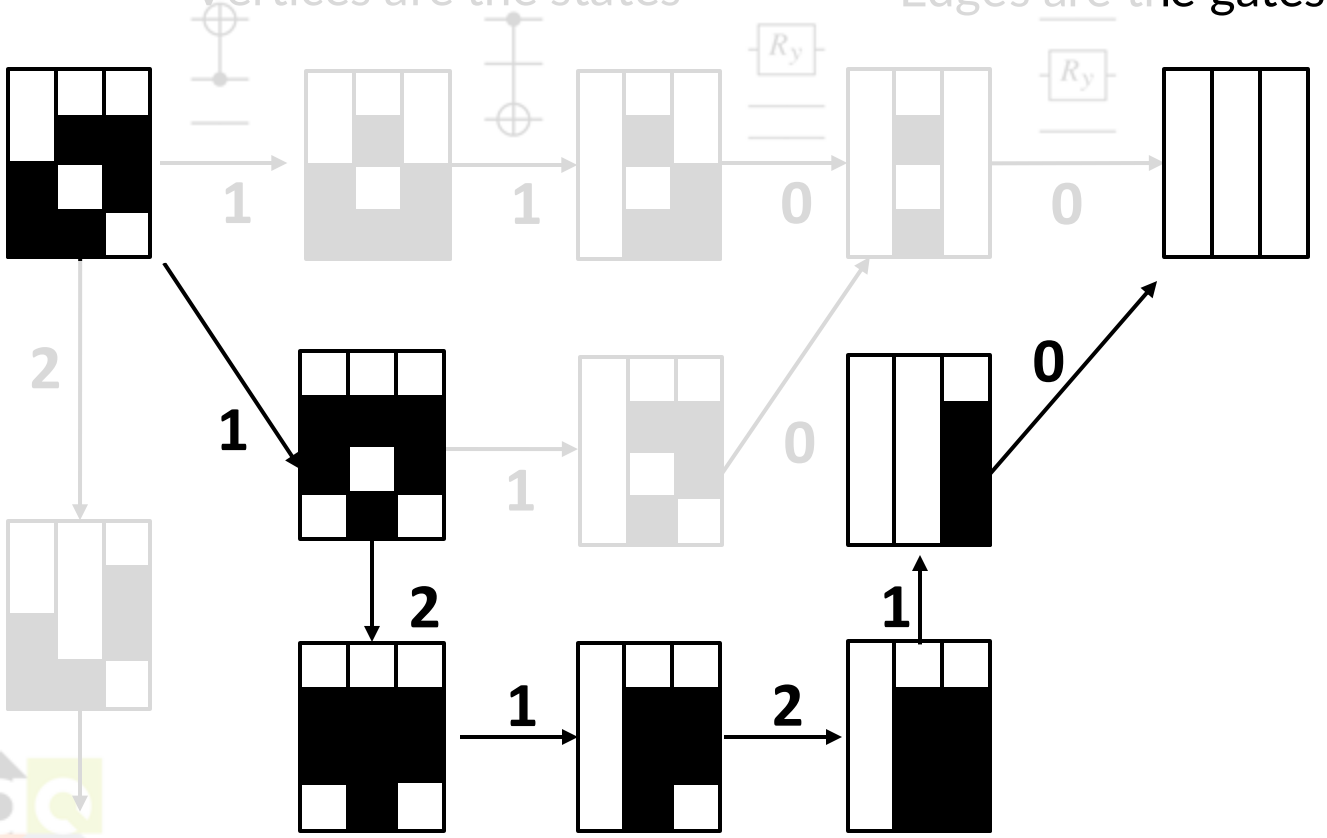


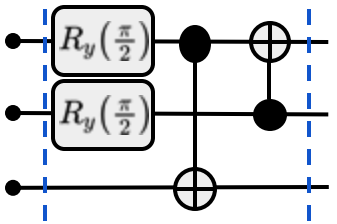
Path 1: CNOT cost = 2

Exact CNOT Synthesis [Wang+, DATE'24]

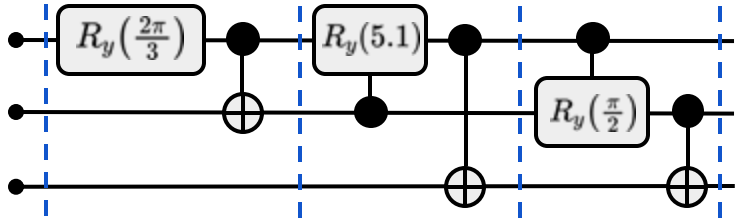
Key idea: formulate QSP problem on a graph, where:

- Vertices are the states
- Edges are the gates
- Distance is the CNOT cost



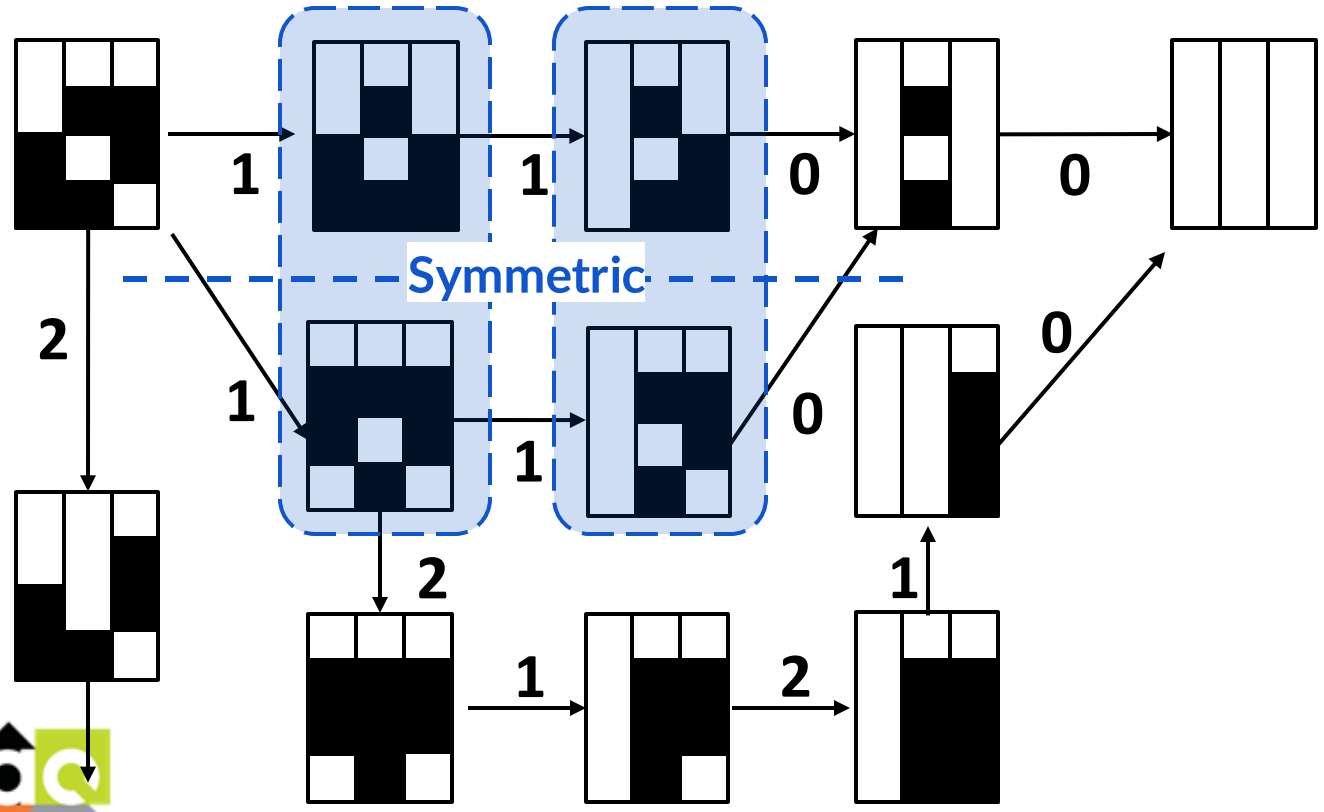


Path 1: CNOT cost = 2



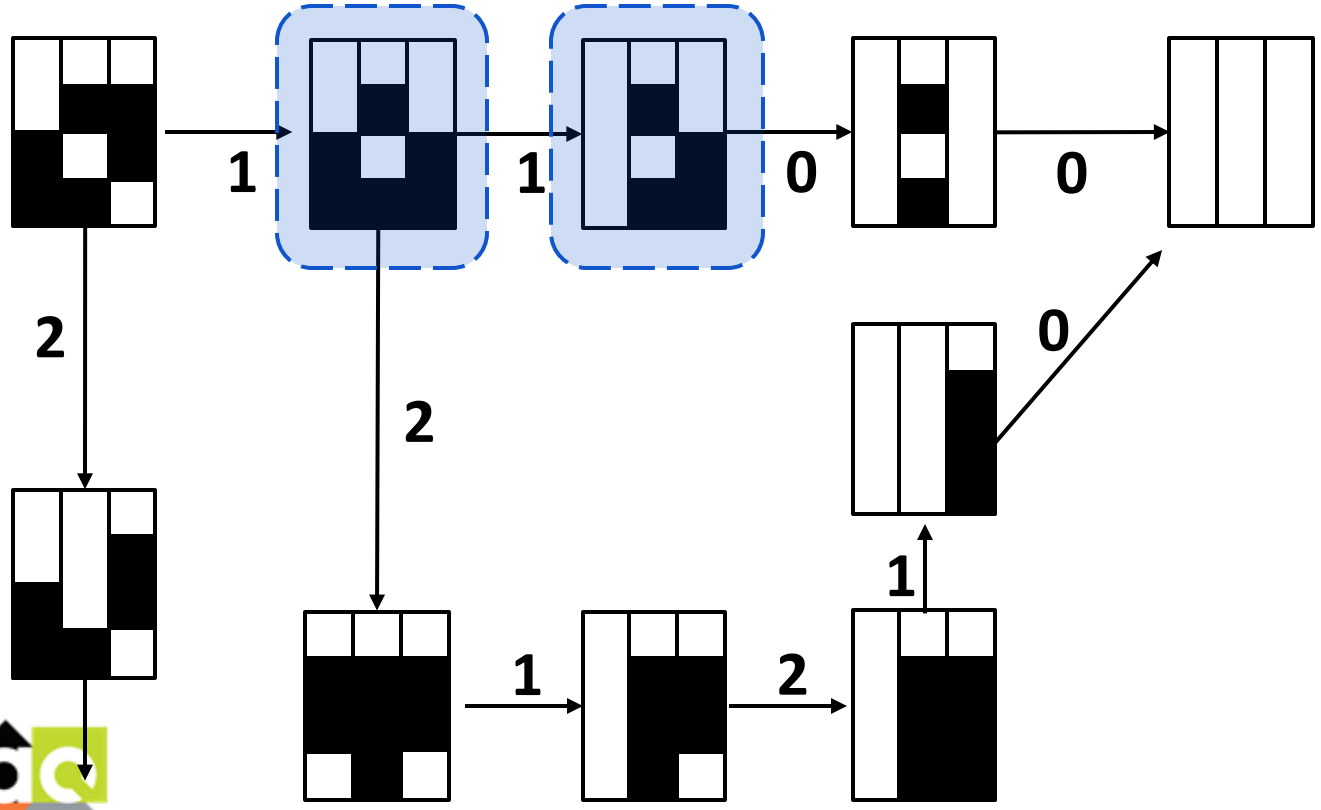
Path 2: CNOT cost = 7

Exact CNOT Synthesis [Wang+, DATE'24]



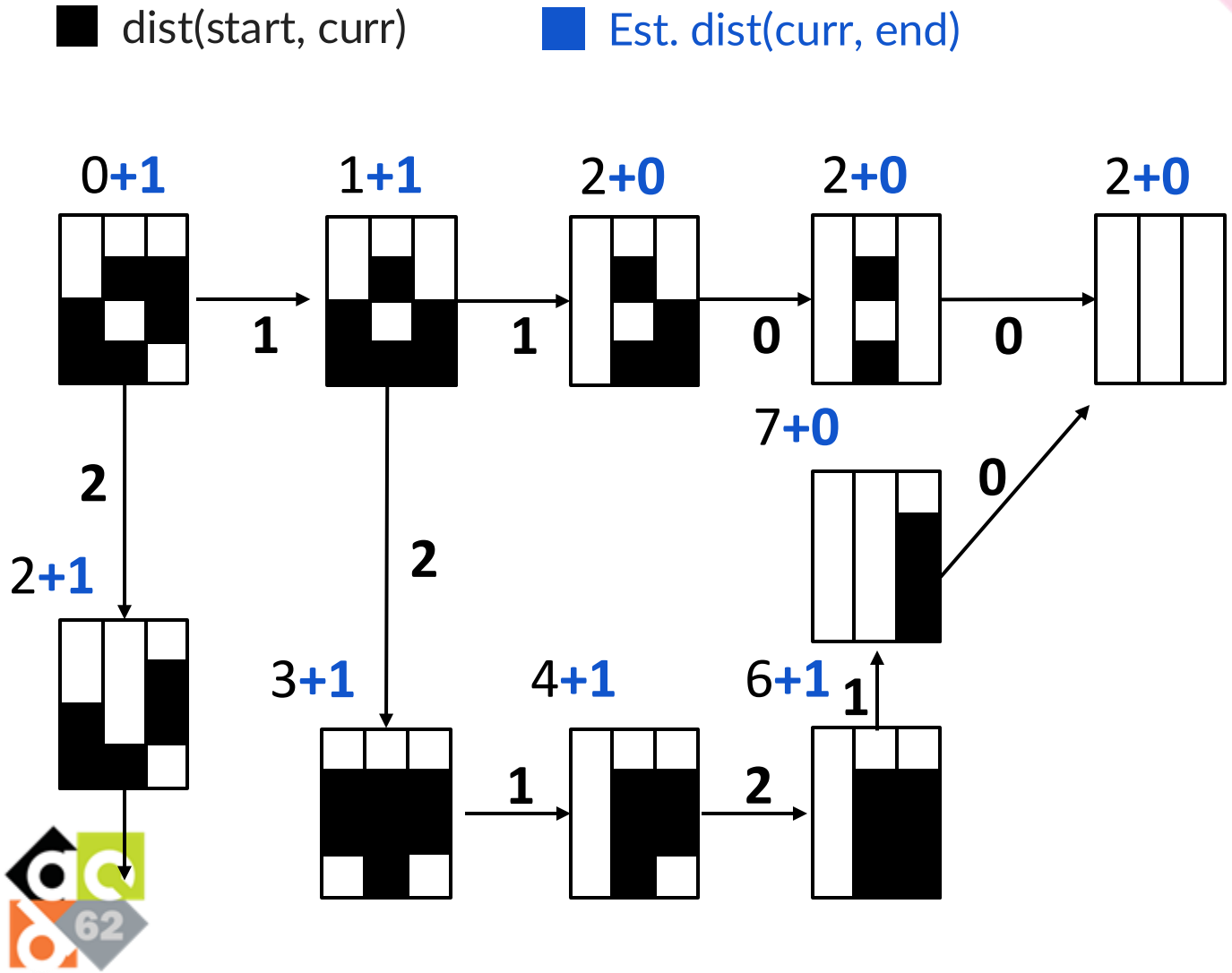
State compression:
identify and merge the **symmetric** states.

Exact CNOT Synthesis [Wang+, DATE'24]



State compression:
identify and merge the **symmetric** states.

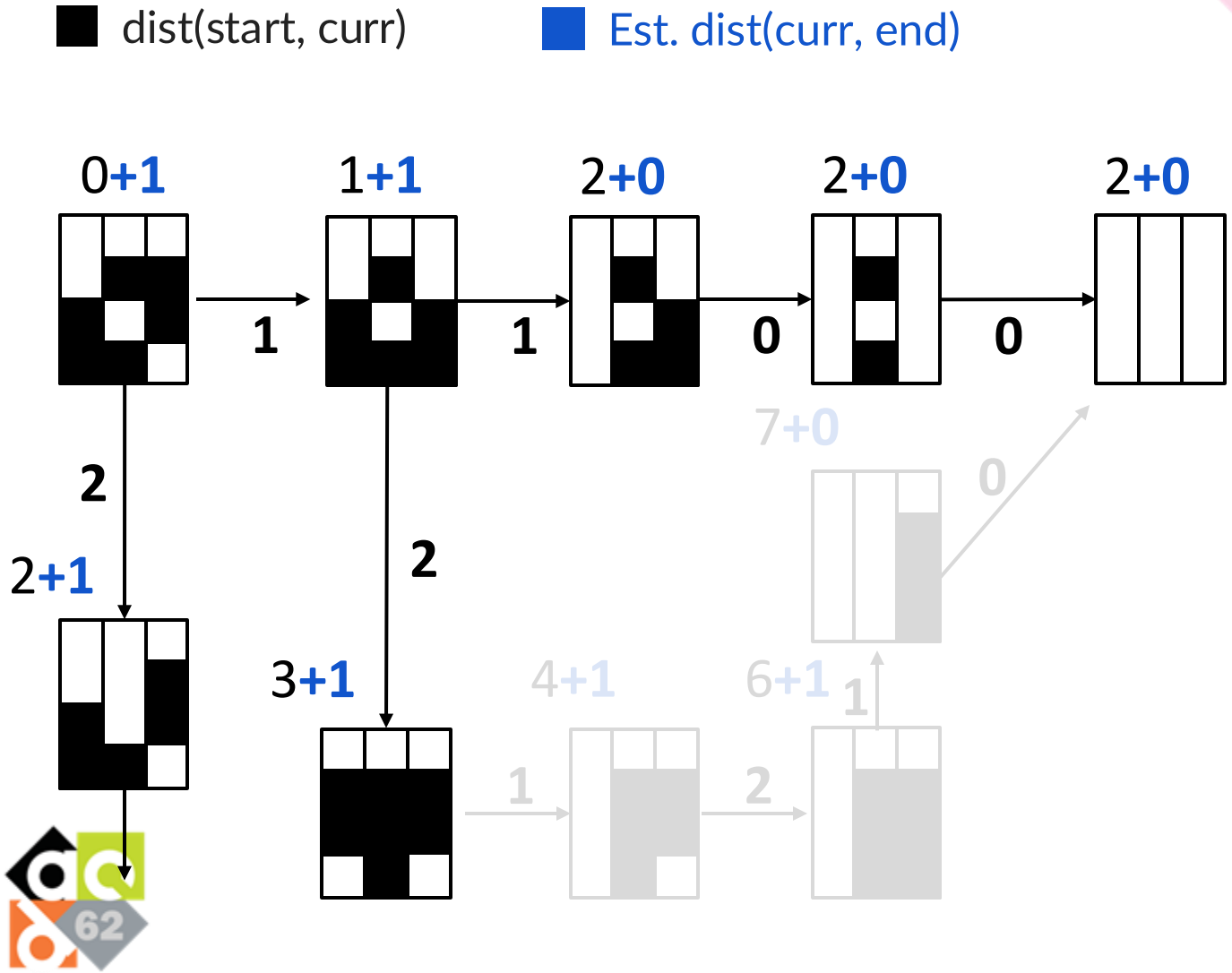
Exact CNOT Synthesis [Wang+, DATE'24]



State compression:
identify and merge the **symmetric** states.

A* algorithm:
Estimate the distance to the final state (the ground state).

Exact CNOT Synthesis [Wang+, DATE'24]

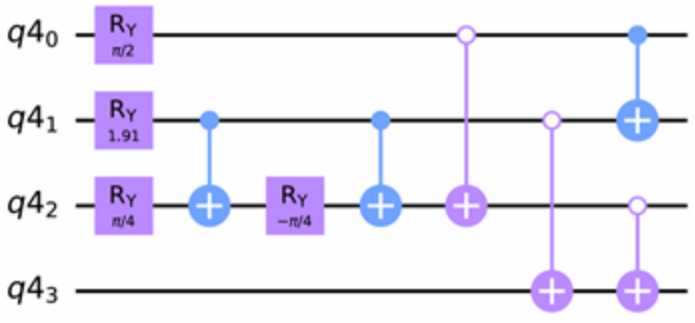


State compression:
identify and merge the **symmetric** states.

A* algorithm:
Estimate the distance to the final state (the ground state).
Prioritize more promising states.

Exact CNOT Synthesis [Wang+, DATE'24]

Contribution 1:
Discovered high-quality QSP circuits (<5 qubits).

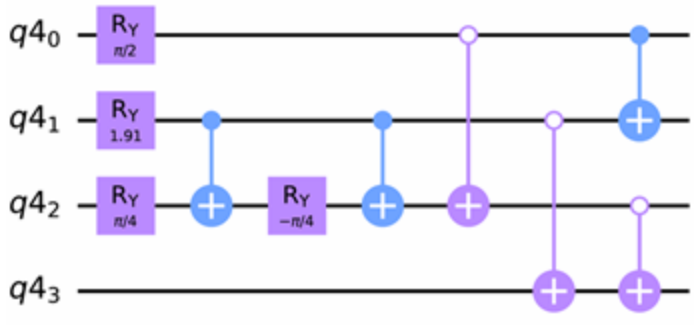


Quantum circuit to prepare the Dicke state D(4,2) using 4 single-qubit gates and 6 CNOTs

…even better than manual designs (12 CNOTs)

Exact CNOT Synthesis [Wang+, DATE'24]

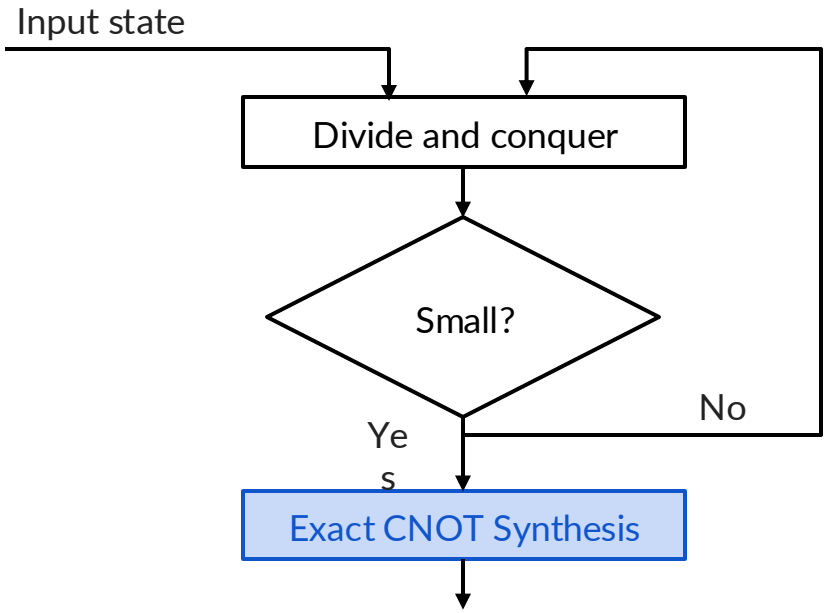
Contribution 1:
Discovered high-quality QSP circuits (<5 qubits).



Quantum circuit to prepare the Dicke state D(4,2) using 4 single-qubit gates and 6 CNOTs

...even better than manual designs (12 CNOTs)

Contribution 2:
Integration into a scalable workflow.



32% CNOT reduction for 5- to 20-qubit states.

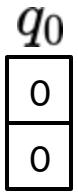
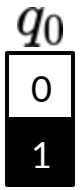
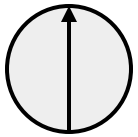
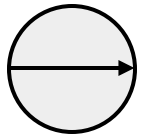
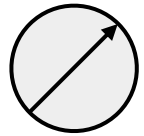
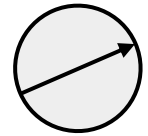
What we've done:

Quantum state preparation circuit…

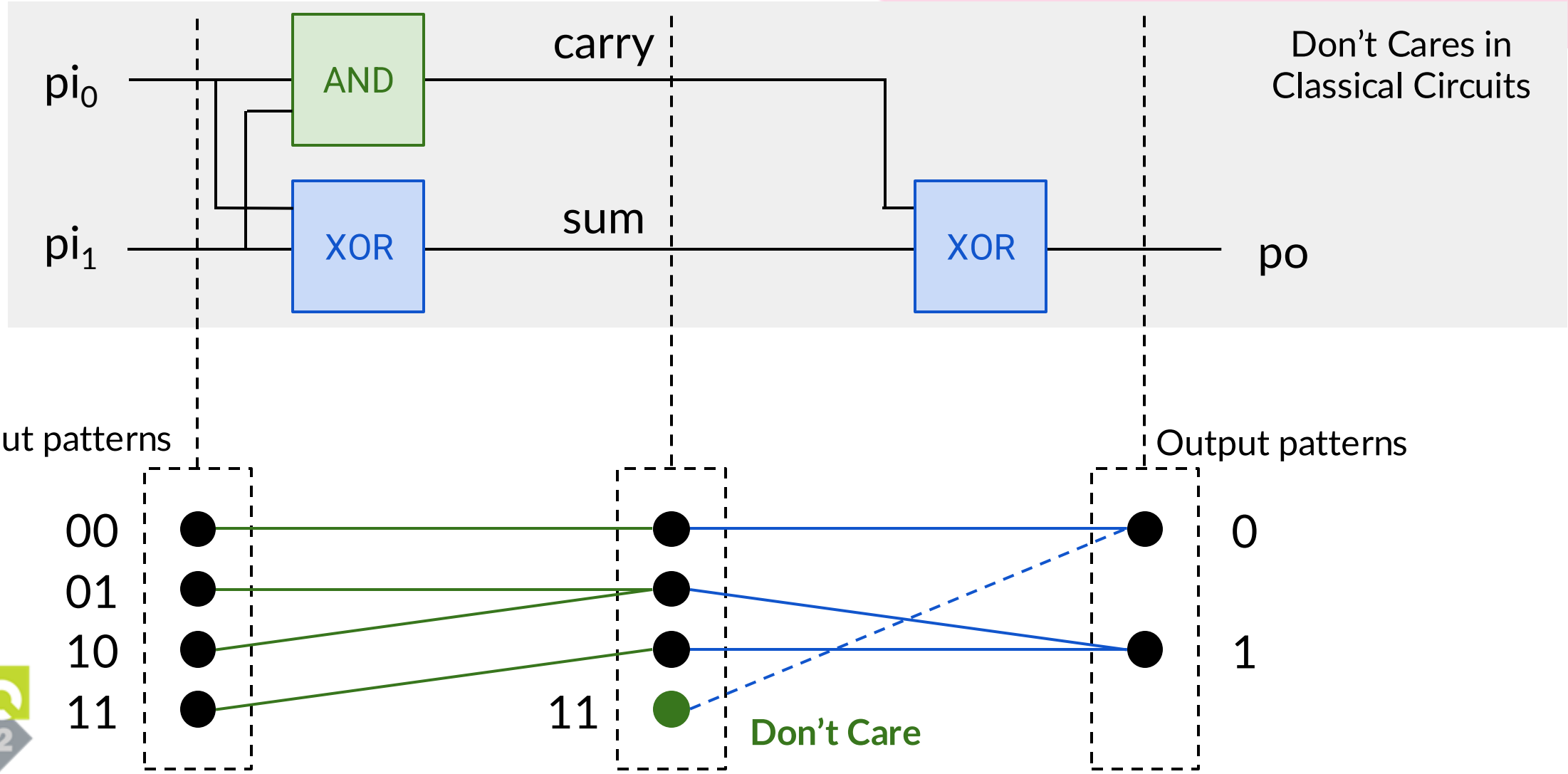
- … **synthesis** using an exact CNOT synthesis formulation
  - Boolean method
  - Generate circuits given initial **states**

- … **resynthesis** exploiting Don't Cares
  - Algebraic method
  - Optimize circuits given initial **circuits**

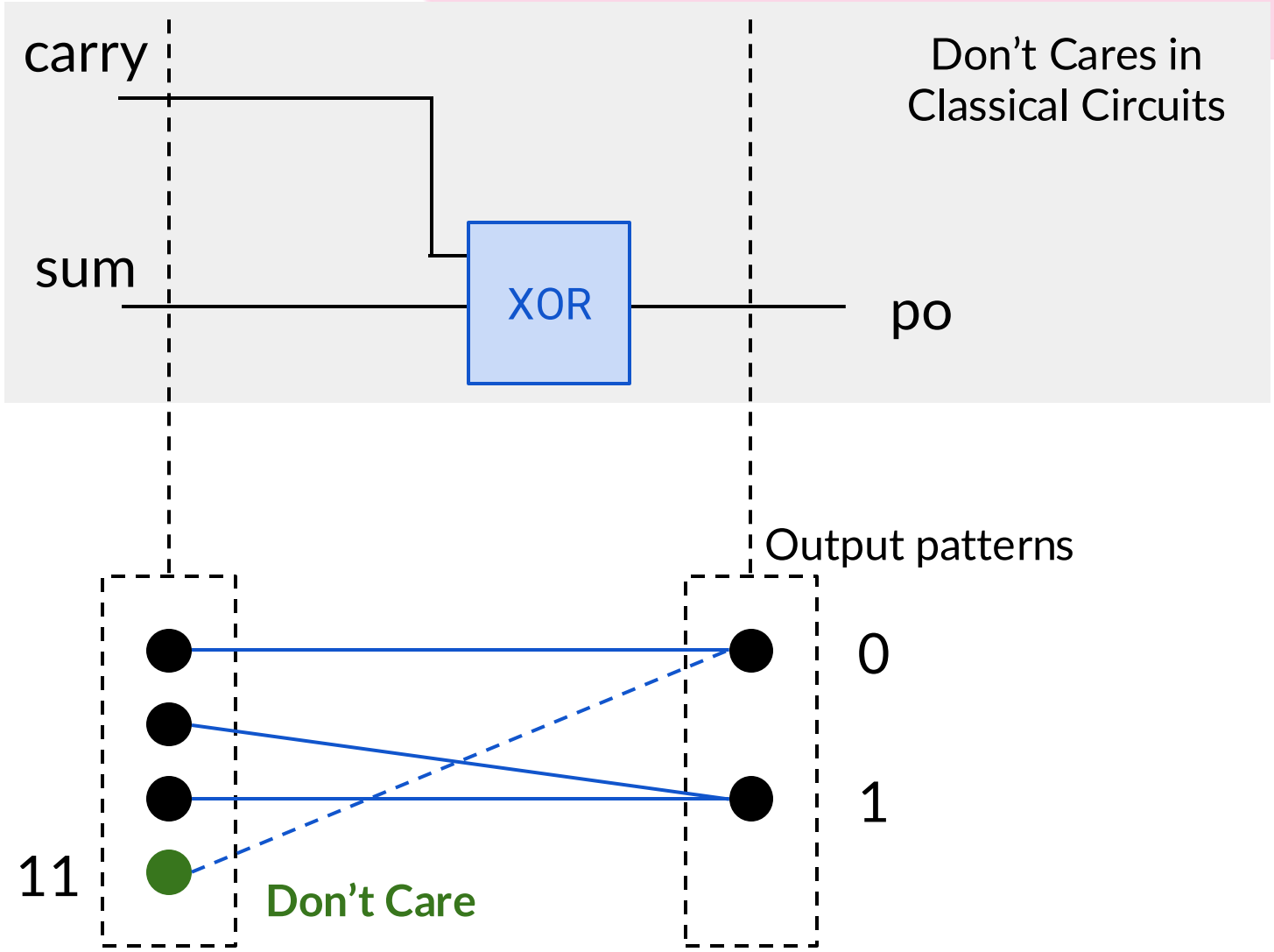
Boolean vs. Algebraic Methods

	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\sqrt{0.6}\|0\rangle + \sqrt{0.8}\|1\rangle$
Boolean methods: • Discretized the solution space • Enabled state canonicalization				N/A
Algebraic methods: • Rotation angle representations • Continuous solution space	 0°	 90°	 45°	 53°

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

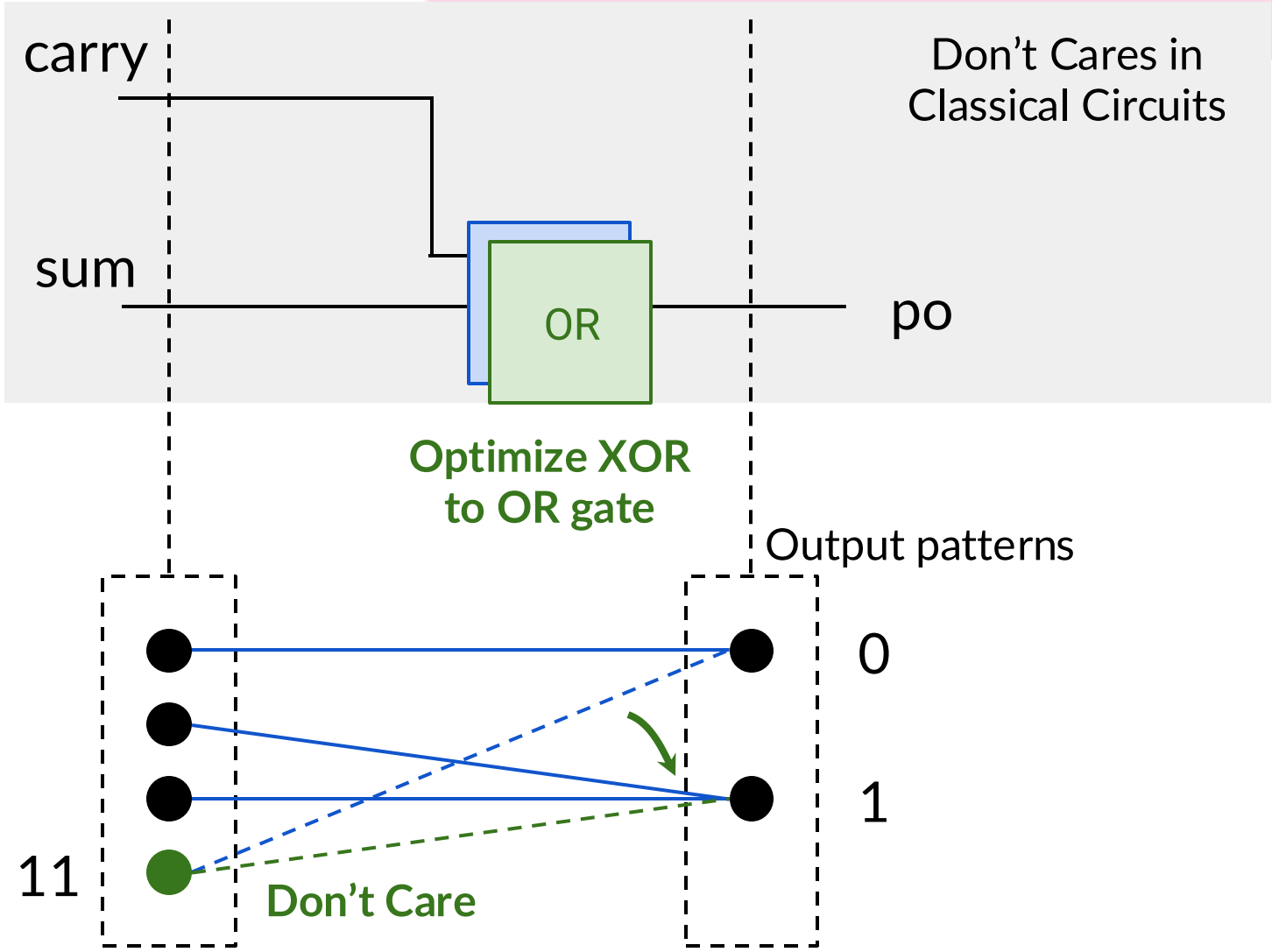


QSP Exploiting Don’t Cares [Wang+, ICCAD’24]



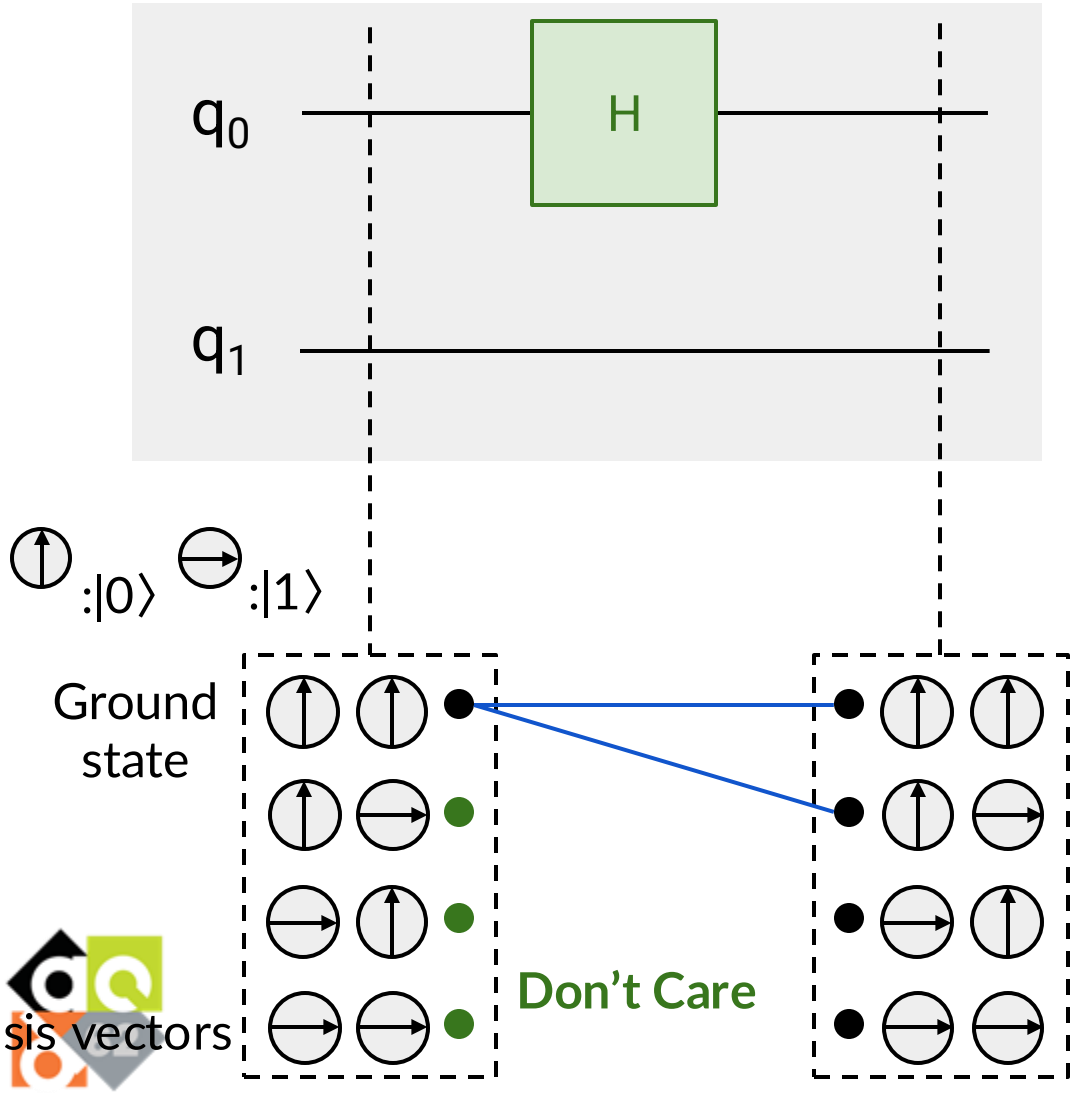
Don’t Care conditions:
Changing local function does not affect global functionality.

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

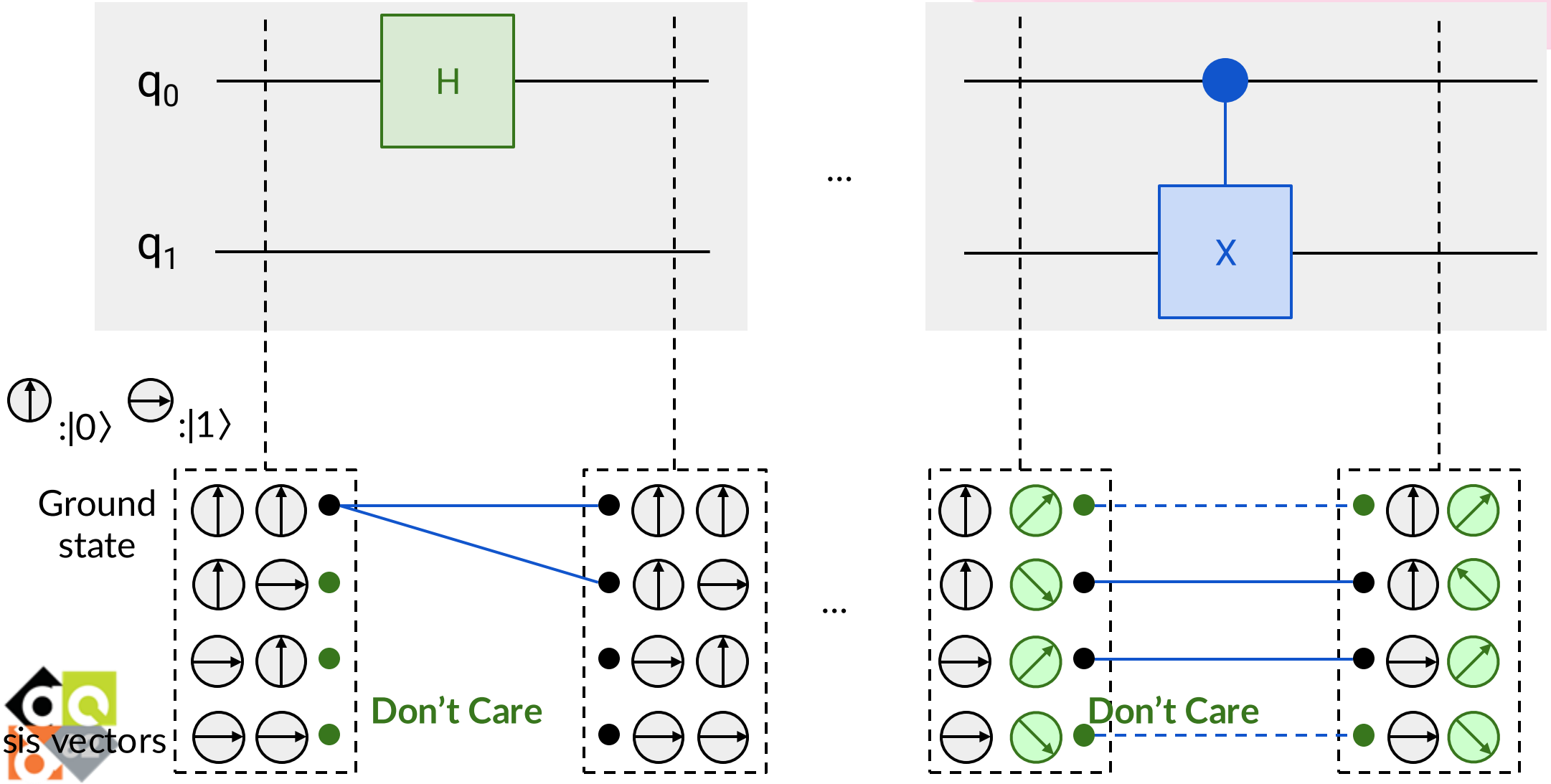


Don’t Care conditions:
Changing local function does not affect global functionality.

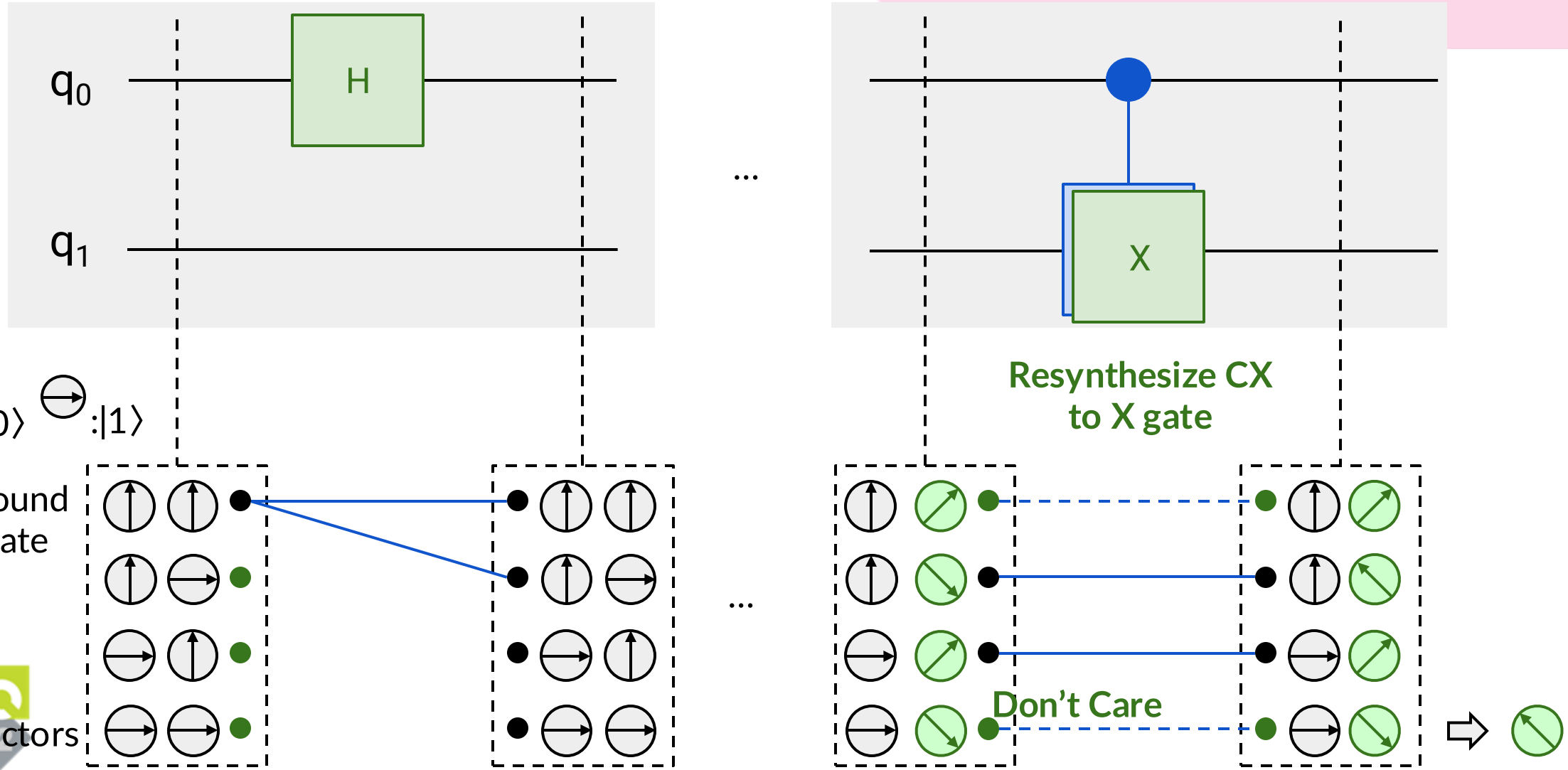
QSP Exploiting Don’t Cares [Wang+, ICCAD’24]



QSP Exploiting Don’t Cares [Wang+, ICCAD’24]



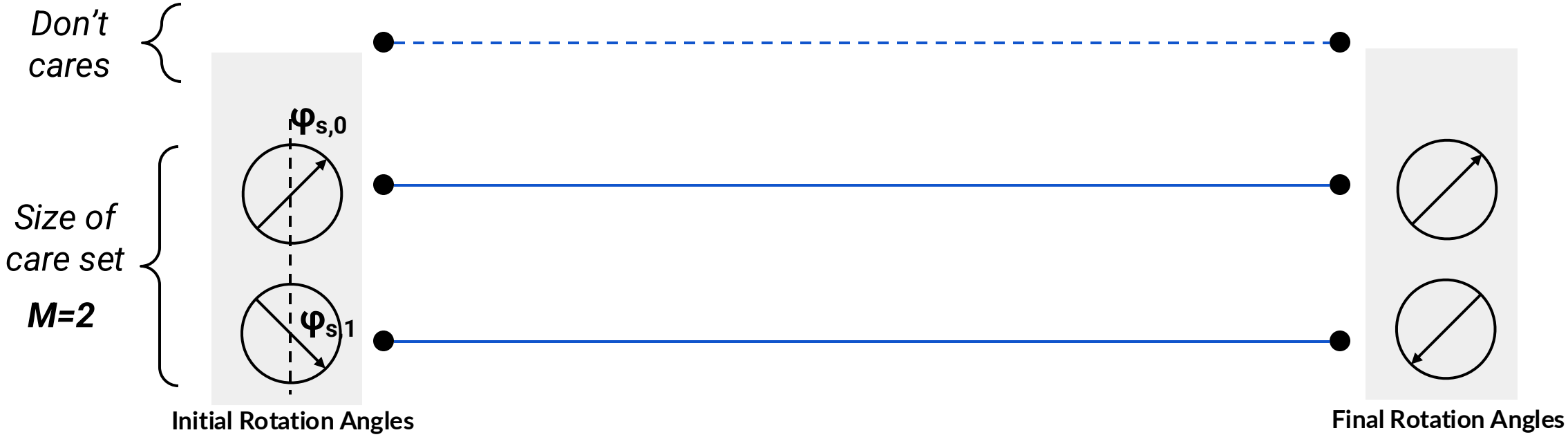
QSP Exploiting Don’t Cares [Wang+, ICCAD’24]



QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Problem formulation:

- Given initial rotation angles $\varphi_{s,0}$ $\varphi_{s,1}$, …$\varphi_{s,M-1}$ and final rotation angles $\varphi_{t,0}$ $\varphi_{t,1}$, …$\varphi_{t,M-1}$
- Find a sequence of gates that accomplish the rotations.



$\varphi_{s,0}$

$\varphi_{s,1}$

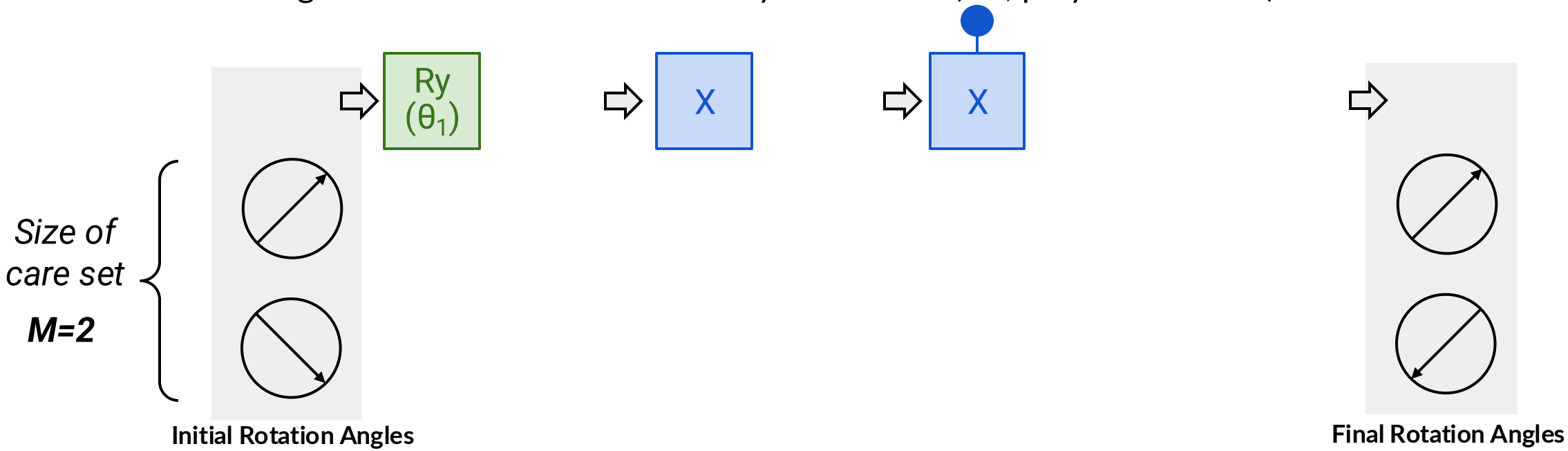
$\varphi_{t,0}$

$\varphi_{t,1}$

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Key observations:

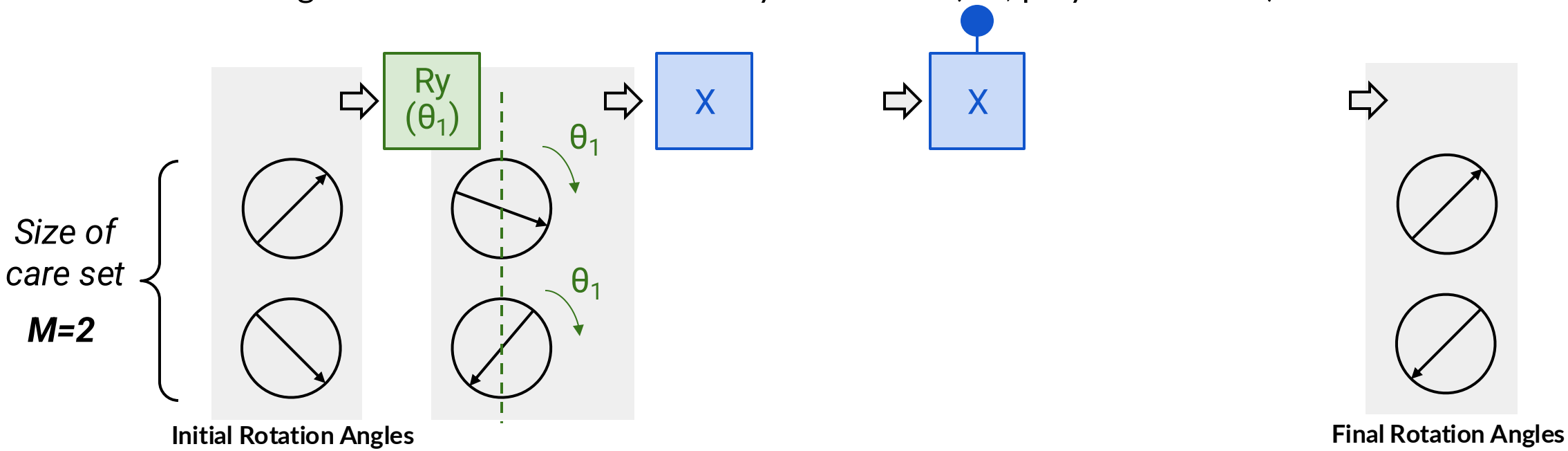
- Finding the circuit directly is **hard**,
- Checking a candidate circuit’s feasibility is **efficient** (i.e., polynomial-time).



QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Key observations:

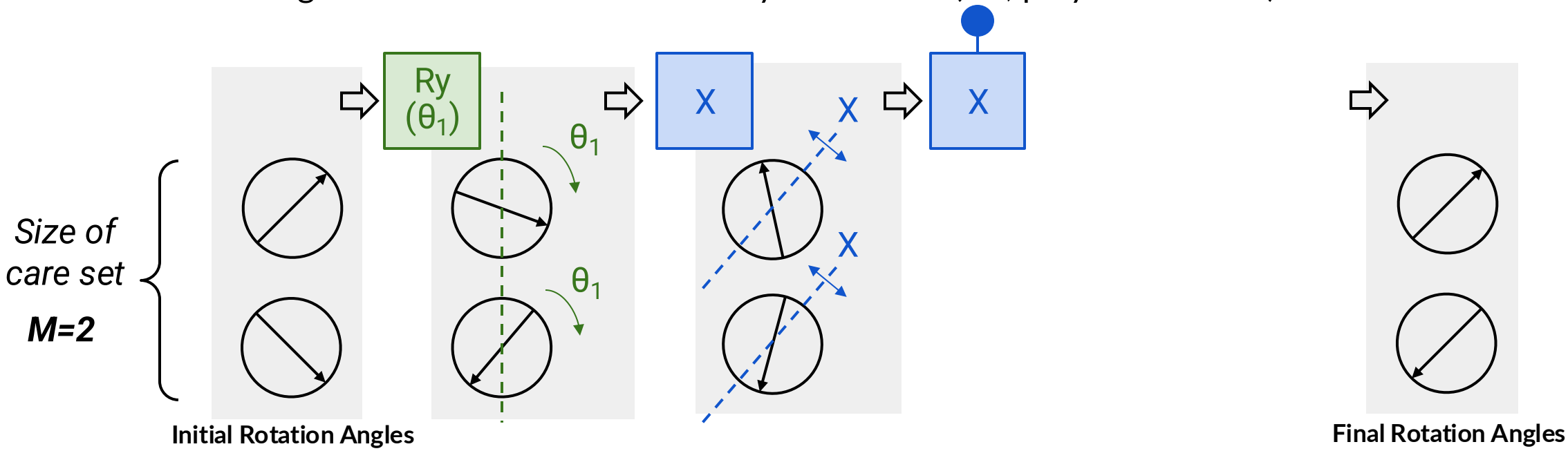
- Finding the circuit directly is **hard**,
- Checking a candidate circuit’s feasibility is **efficient** (i.e., polynomial-time).



QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Key observations:

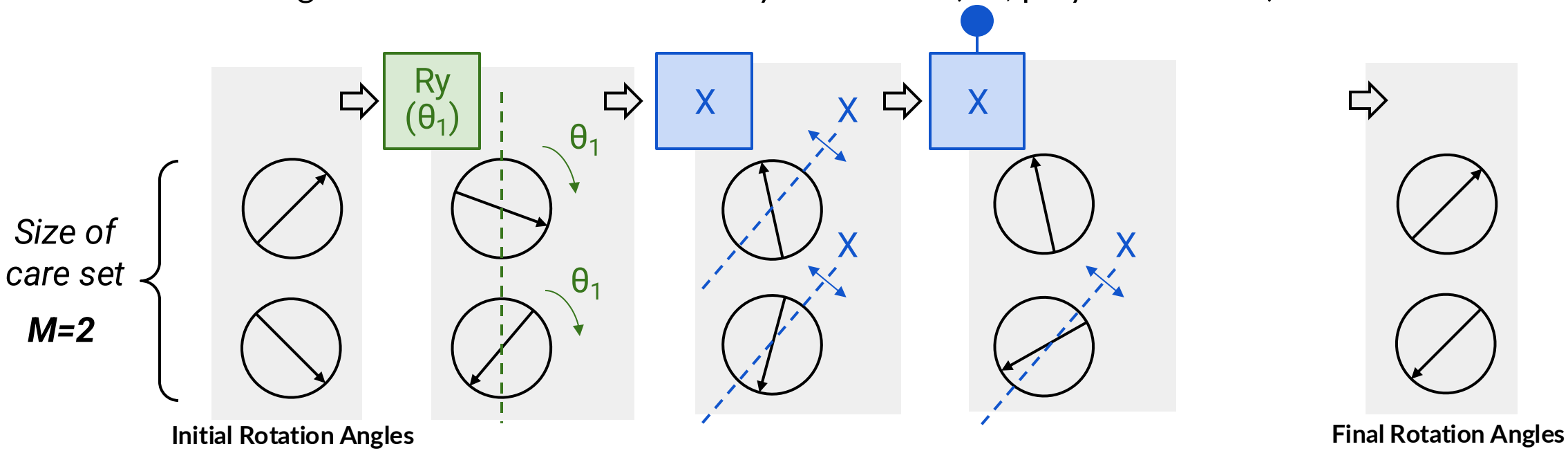
- Finding the circuit directly is **hard**,
- Checking a candidate circuit’s feasibility is **efficient** (i.e., polynomial-time).



QSP Exploiting Don't Cares [Wang+, ICCAD'24]

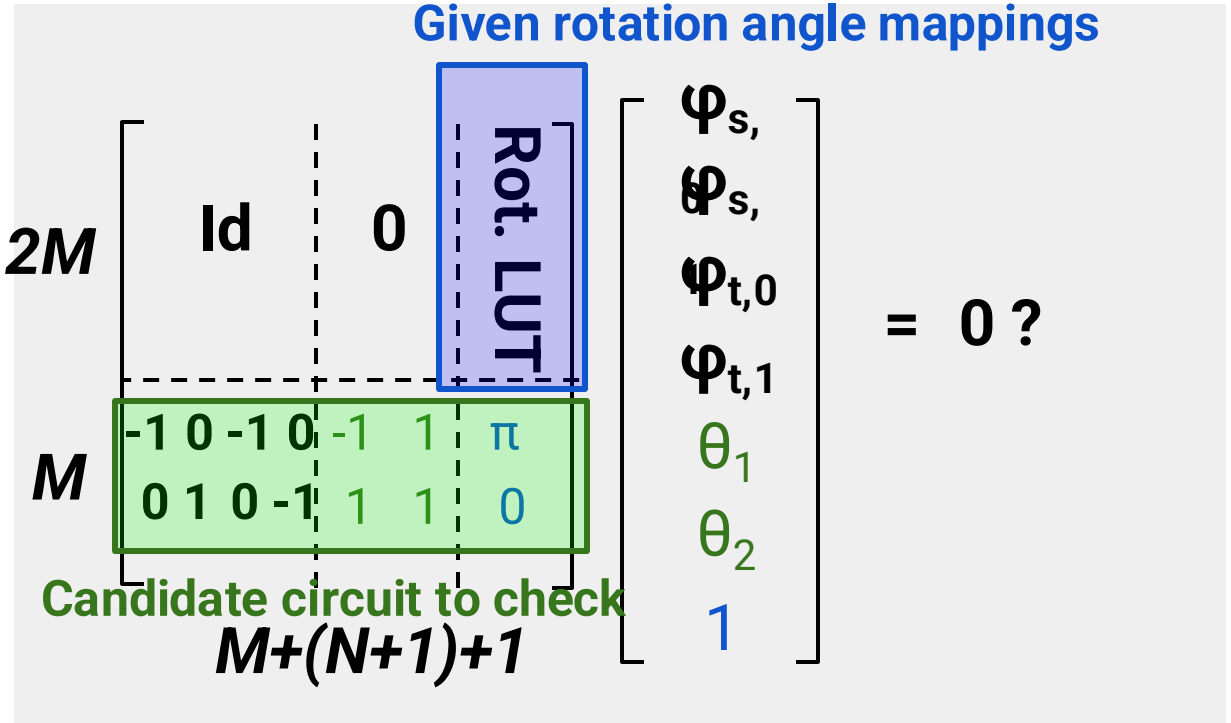
Key observations:

- Finding the circuit directly is **hard**,
- Checking a candidate circuit's feasibility is **efficient** (i.e., polynomial-time).



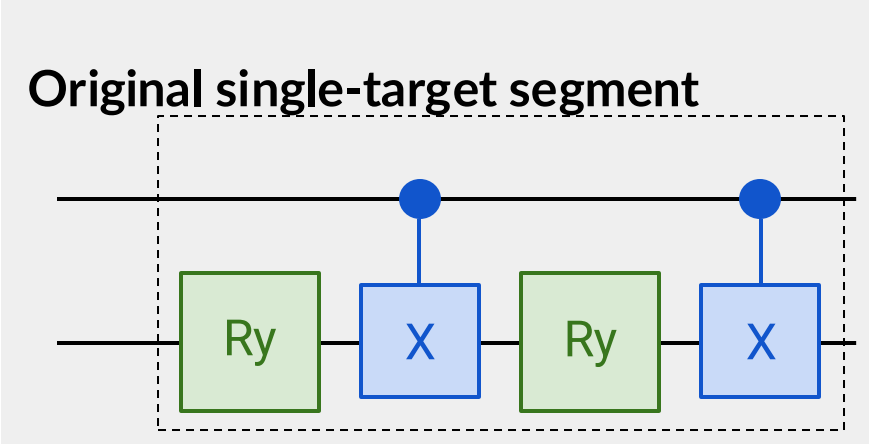
QSP Exploiting Don't Cares [Wang+, ICCAD'24]

Check feasibility of candidate circuits: $O(M^2(M+N))$

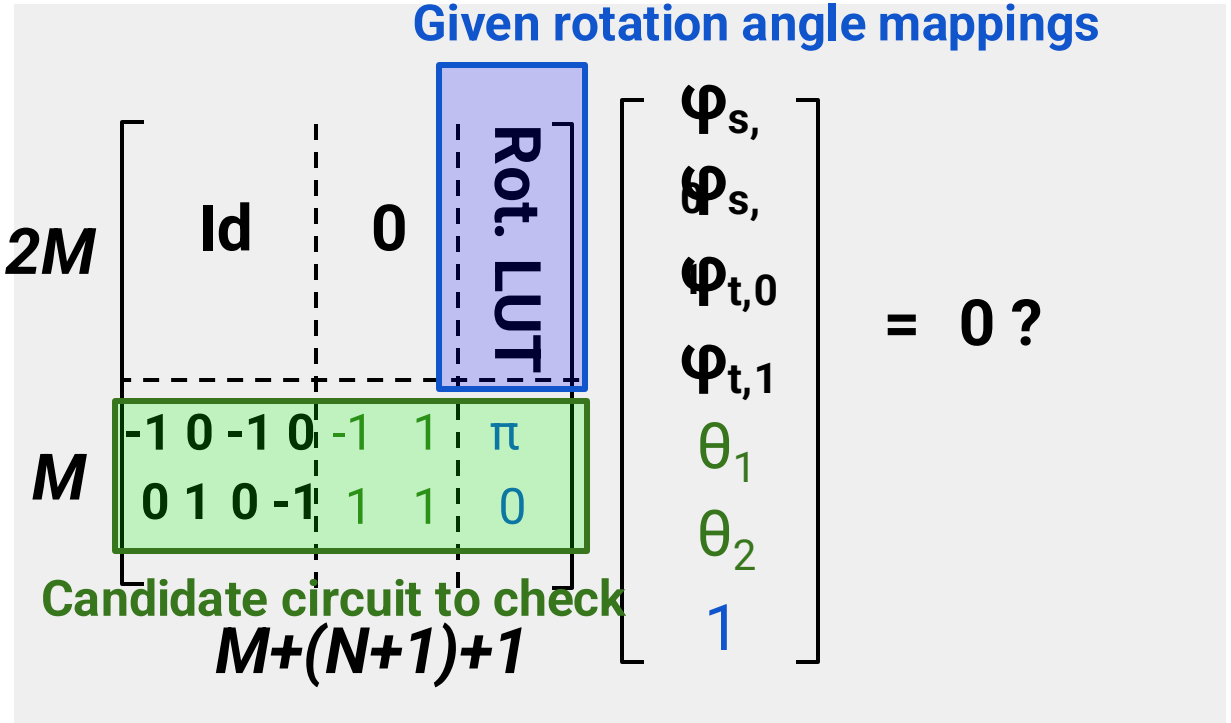


***M**: size of care set, **N**: CNOT count*

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]



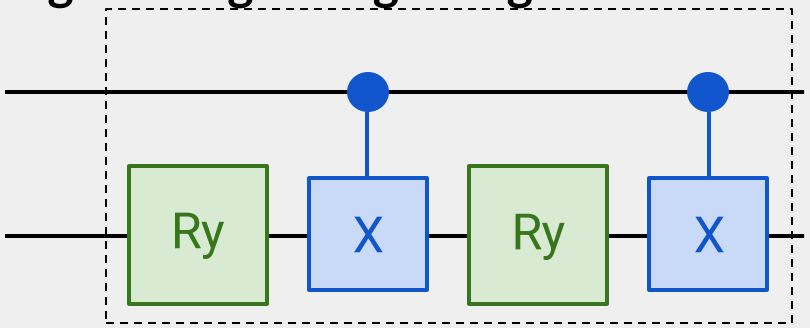
Check feasibility of candidate circuits: $O(M^2(M+N))$



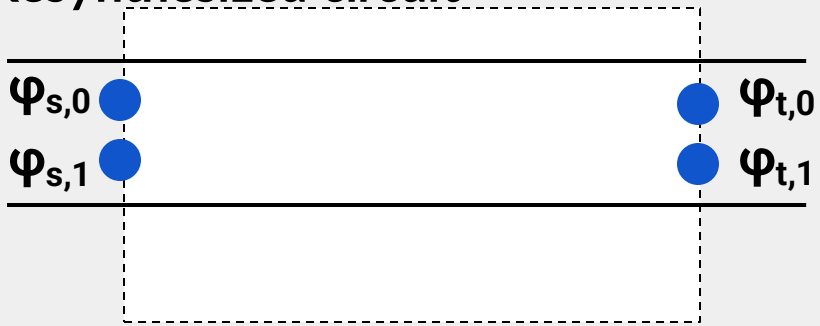
M: *size of care set,* ***N***: *CNOT count*

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Original single-target segment

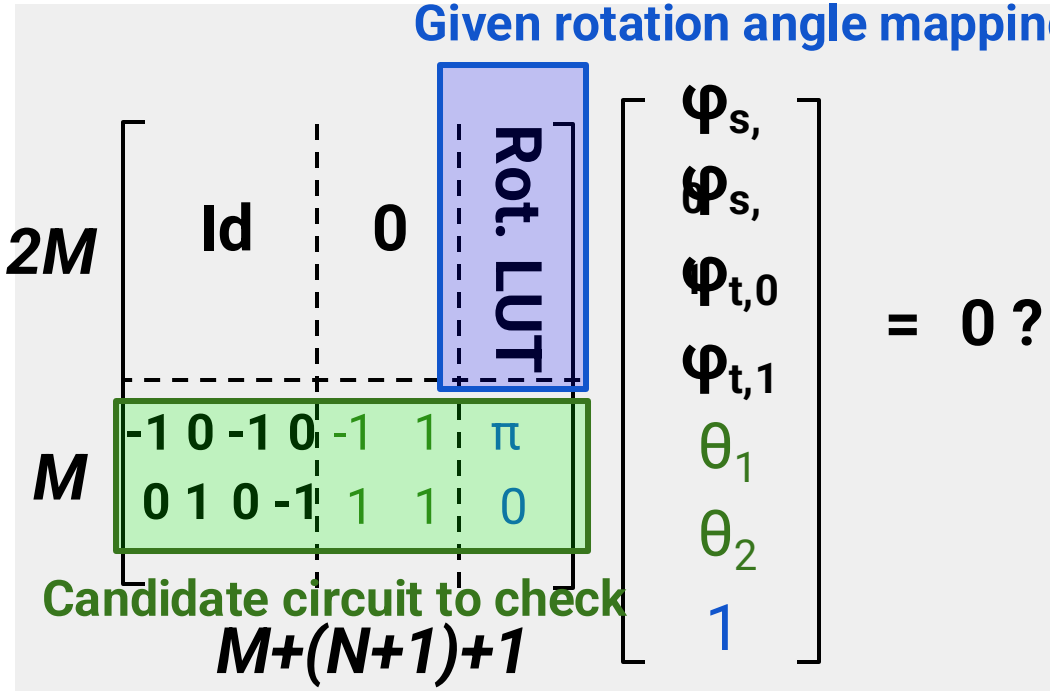


Resynthesized circuit



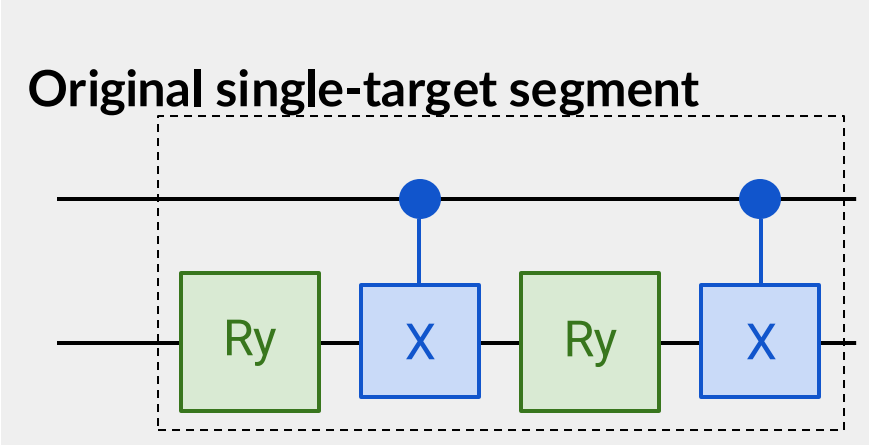
Check feasibility of candidate circuits: $O(M^2(M+N))$

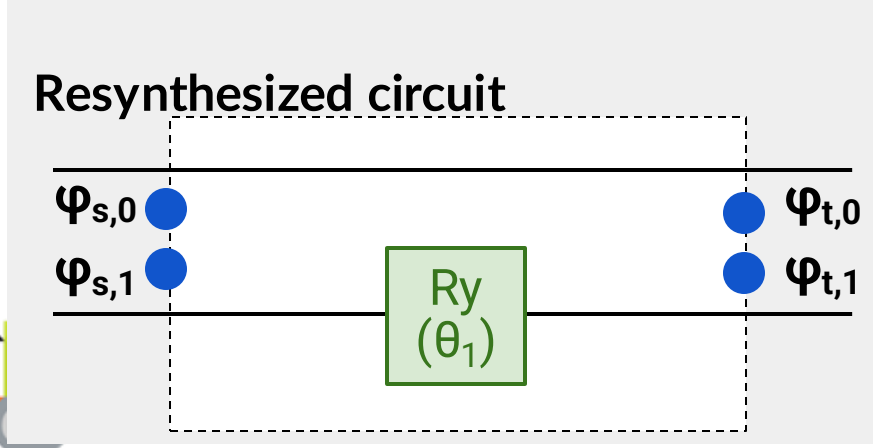
Given rotation angle mappings



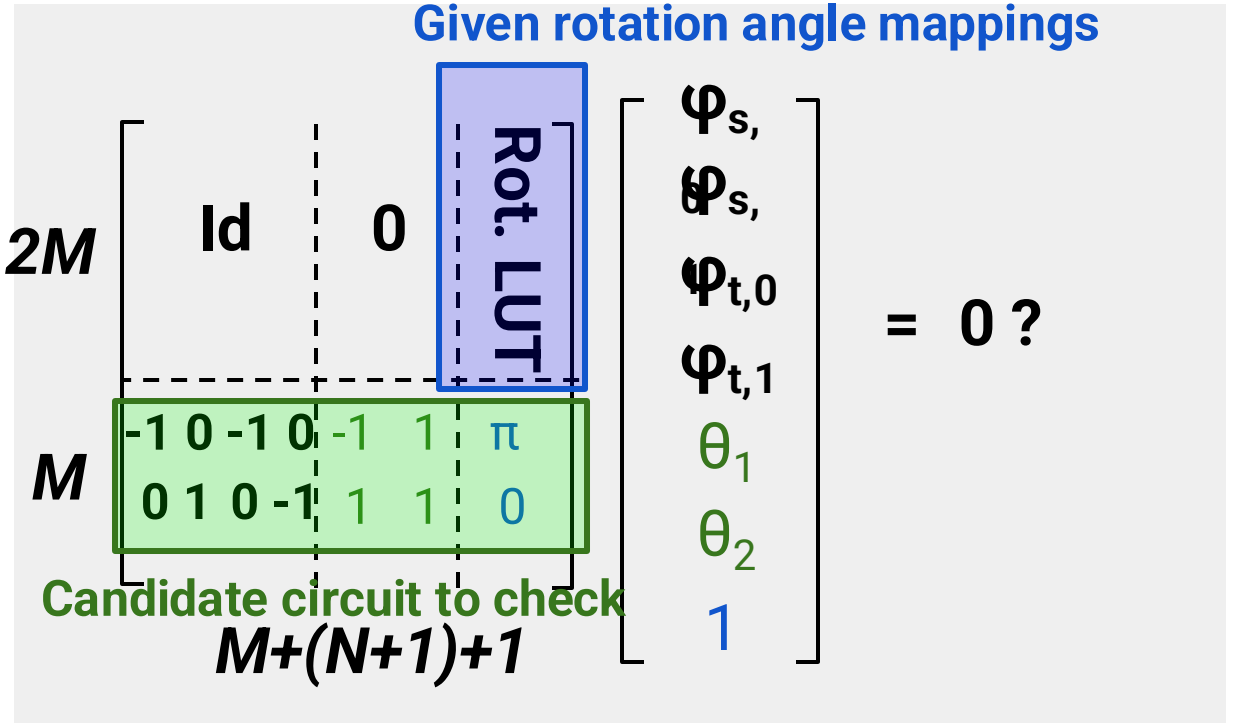
M: *size of care set,* ***N***: *CNOT count*

QSP Exploiting Don't Cares [Wang+, ICCAD'24]





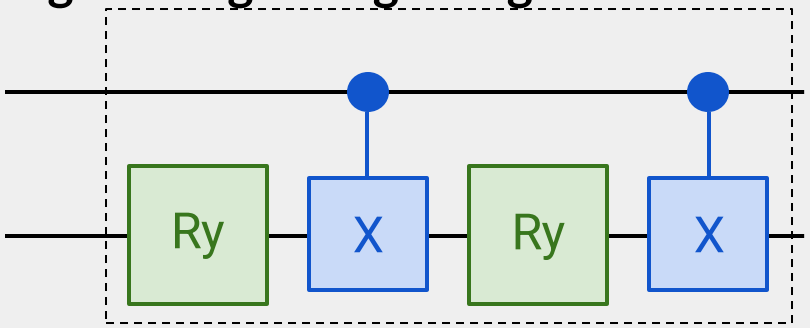
Check feasibility of candidate circuits: $O(M^2(M+N))$



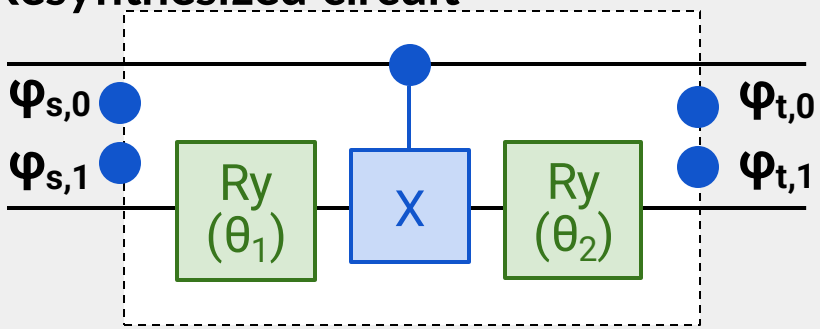
M: *size of care set,* ***N***: *CNOT count*

QSP Exploiting Don't Cares [Wang+, ICCAD'24]

Original single-target segment

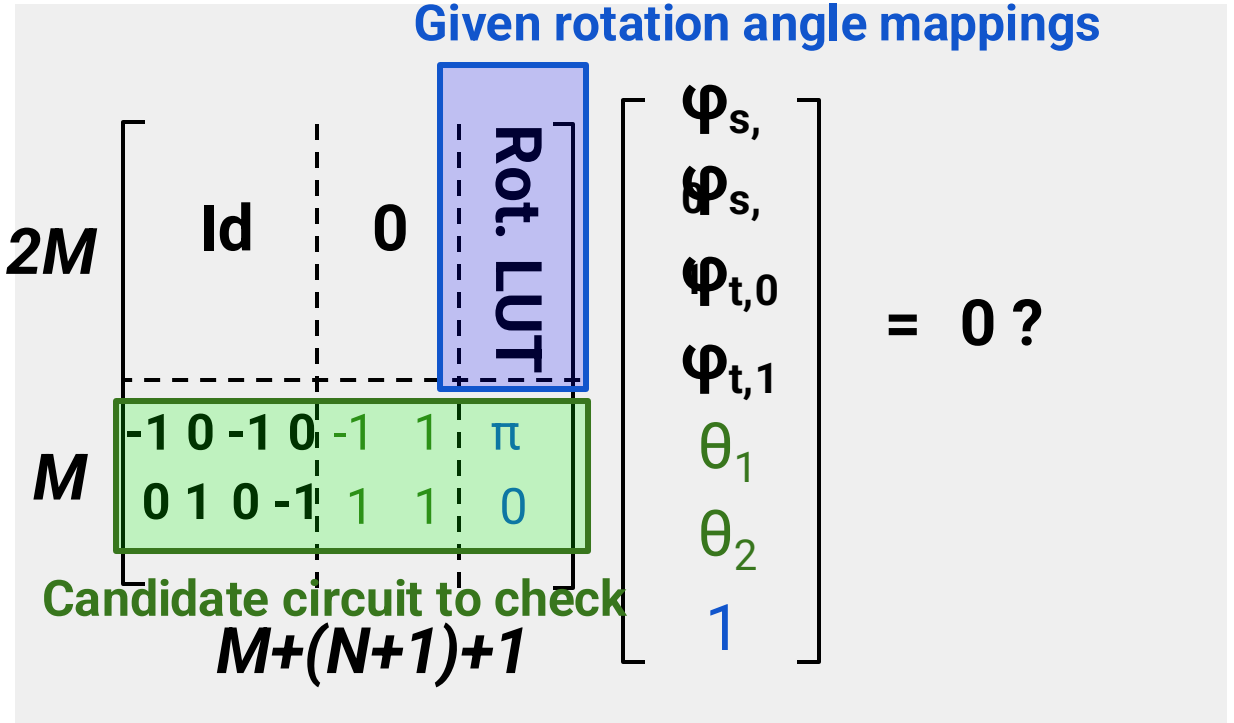


Resynthesized circuit



Check feasibility of candidate circuits: $O(M^2(M+N))$

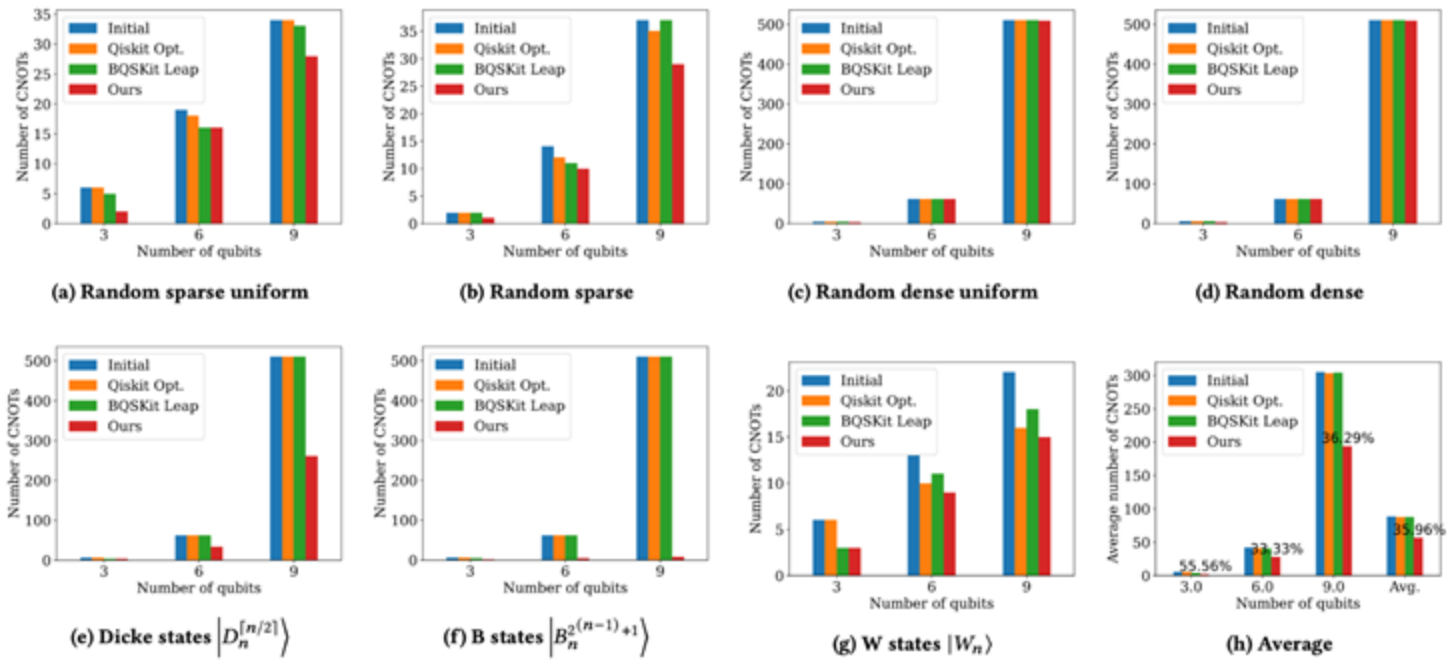
Given rotation angle mappings



M: size of care set, *N*: CNOT count

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Effectiveness:
36% CNOT reduction over Qiskit and BQSKit flows

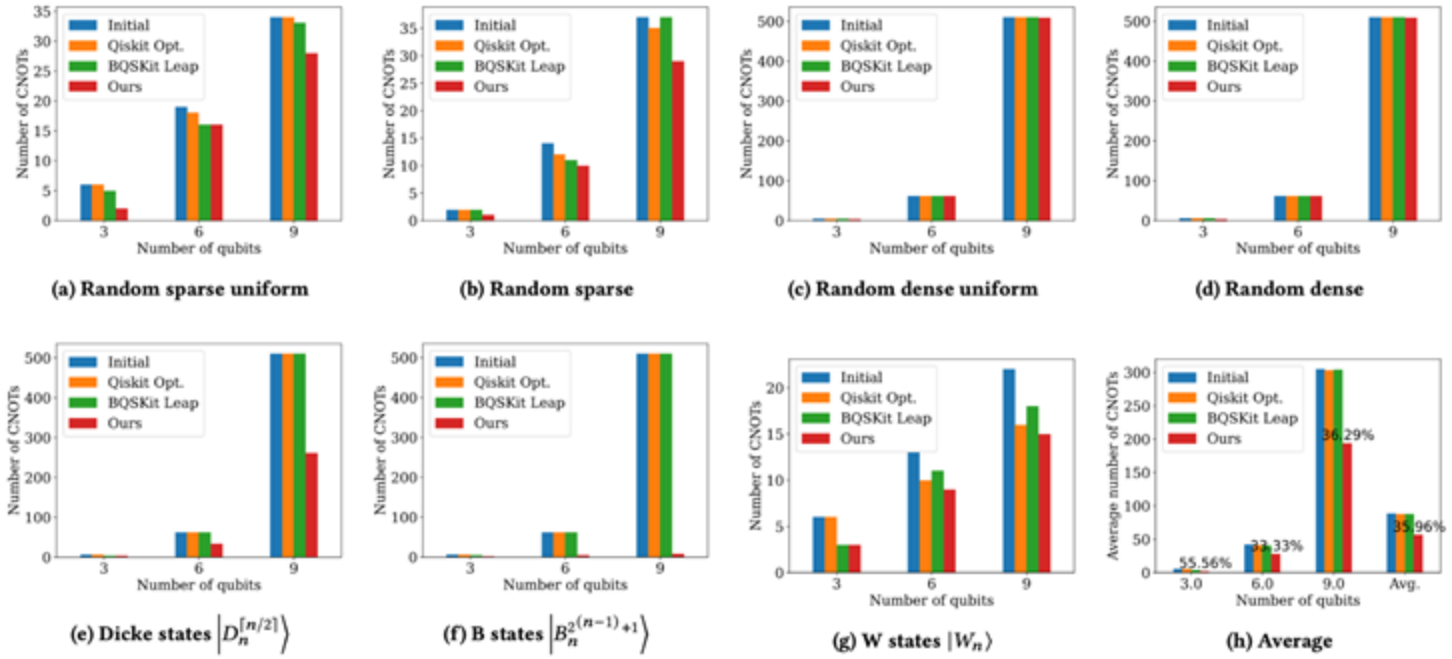


(a) Random sparse uniform (b) Random sparse (c) Random dense uniform (d) Random dense

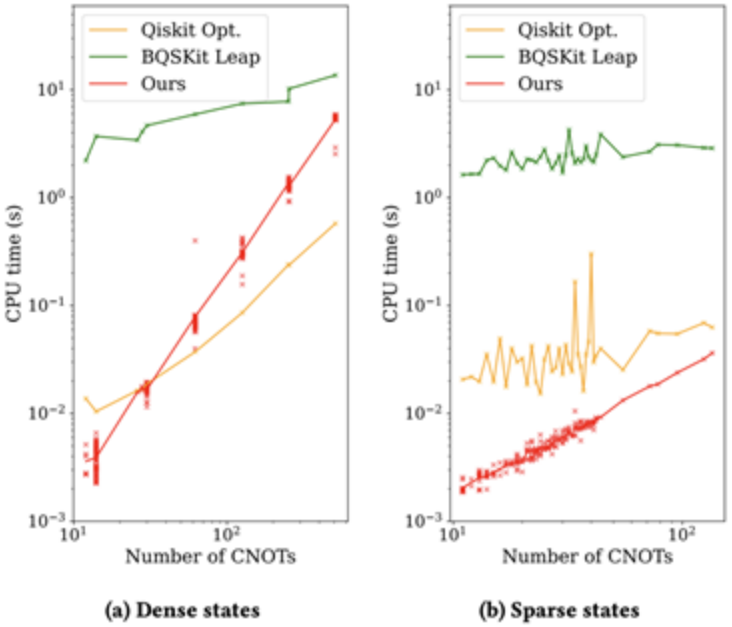
(e) Dicke states $\left|D_n^{\lceil n/2 \rceil}\right\rangle$ (f) B states $\left|B_n^{2^{(n-1)}+1}\right\rangle$ (g) W states $|W_n\rangle$ (h) Average

QSP Exploiting Don’t Cares [Wang+, ICCAD’24]

Effectiveness:
36% CNOT reduction over Qiskit and BQSKit flows



Efficiency:
Poly(#CNOTs) Complexity



Conclusions and Future Works:

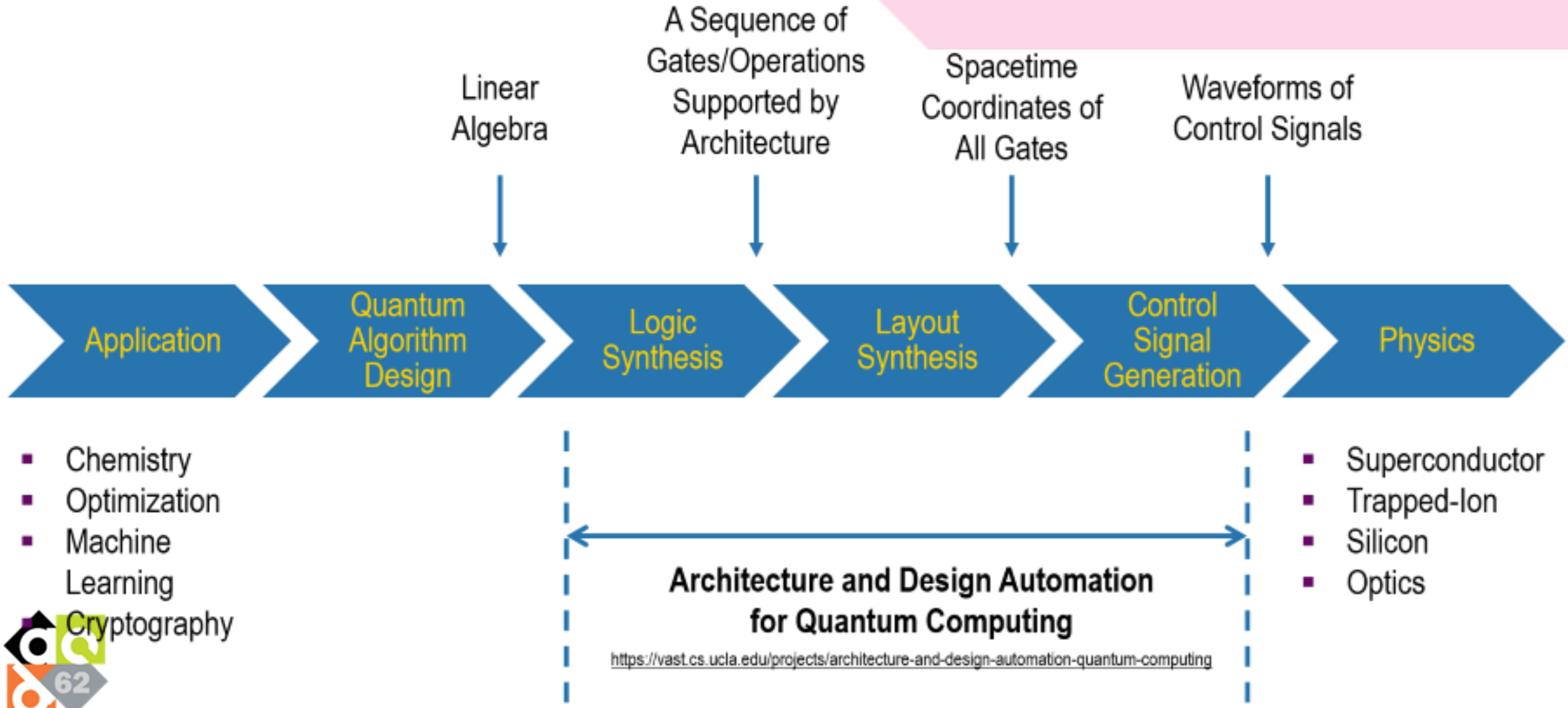
What we've done: Quantum state preparation

- Using an exact CNOT synthesis formulation
  - Boolean method
  - High-quality solutions for small circuits
- Exploiting Don't Cares
  - Algebraic method
  - Scalable optimization for larger circuits

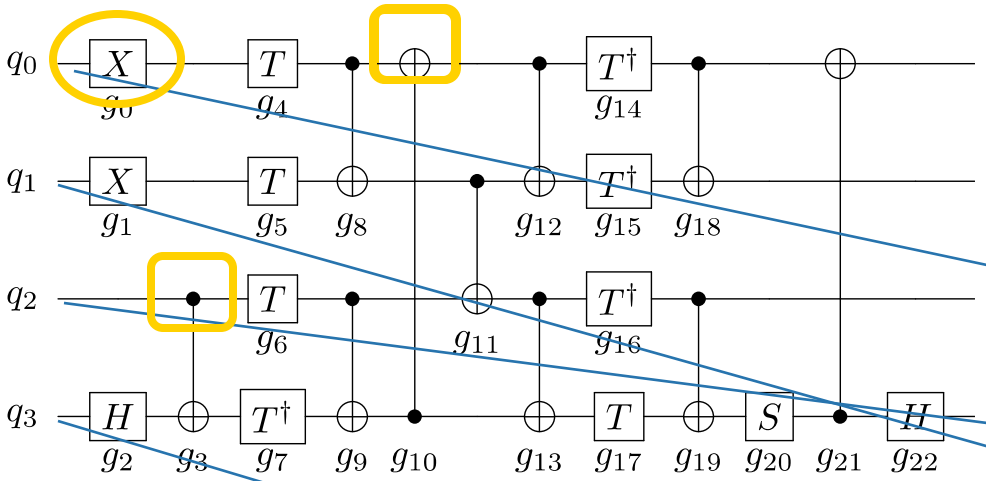
What we haven't:

- Phase oracle synthesis using X/Z rotations
- Optimizing costs beyond #CNOTs

Compilation Flow for Quantum Computing



Quantum Layout Synthesis



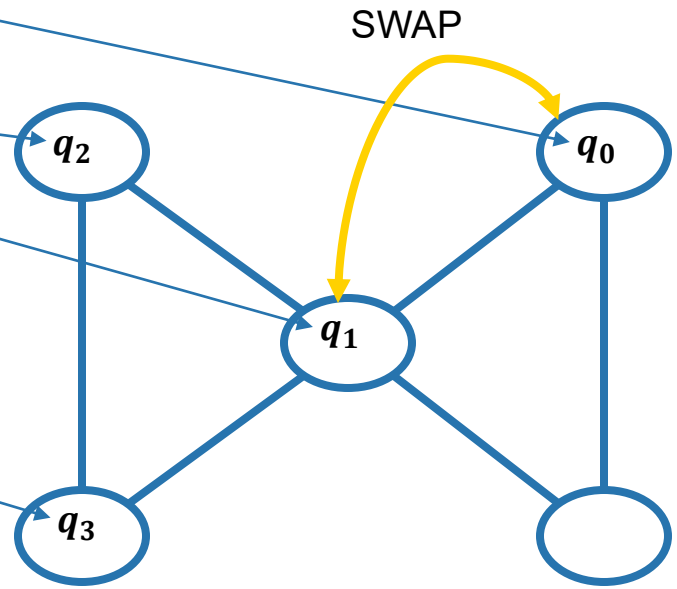
Input Circuit of Adder

```
# Input quantum program
x q[0];
x q[1];
h q[3];
cx q[2], q[3];
t q[0];
```

CX on a pair of adjacent qubits, OK.

CX on a pair of non-adjacent qubits!

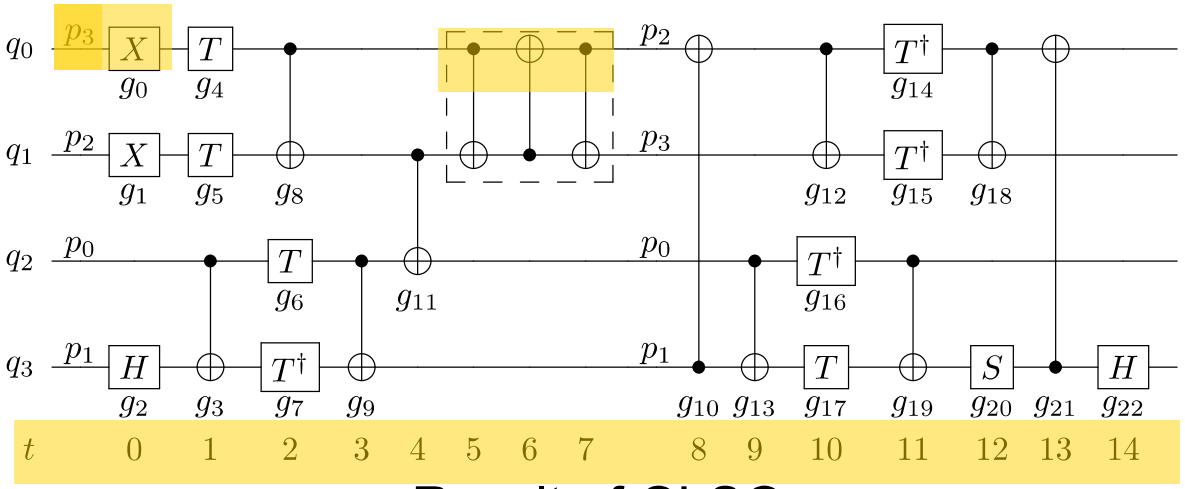
Insert SWAP gate to change the mapping



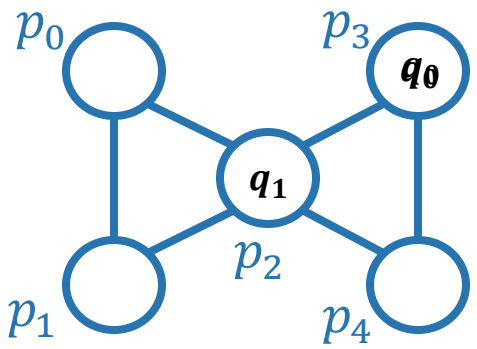
Coupling Graph of IBM QX 2

OLSQ (Optimal Layout Synthesis for Quantum Computing)

- Variables in OLSQ
  - Spacetime Coordinates (x_l, t_l) for every gate g_l
    - If g_l is a single-qubit gate, x_l is a physical qubit; if g_l is a two-qubit gate, x_l is an edge
  - Mapping π_q^t: at time t, logical qubit q is mapped to the physical qubit π_q^t
  - Use of SWAP σ_e^t: $\sigma_e^t = 1$ iff. there is a SWAP on edge e and its last time step is t

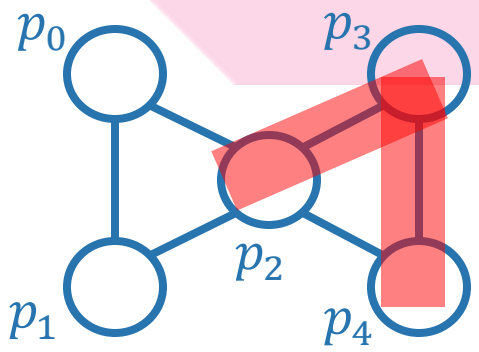


Result of OLSQ



OLSQ: Constraints

- Mapping transformed by SWAPs
  - $\left[\left(\pi_0^0 = p_3\right) \wedge \left(\sum_{e:\, p_3 \in e} \sigma_e^0 = 0\right)\right] \Rightarrow \left(\pi_0^1 = p_3\right)$
  - $\left[\left(\pi_0^7 = p_3\right) \wedge \left(\sigma_{(p_2,p_3)}^7 = 1\right)\right] \Rightarrow \left(\pi_0^8 = p_2\right)$

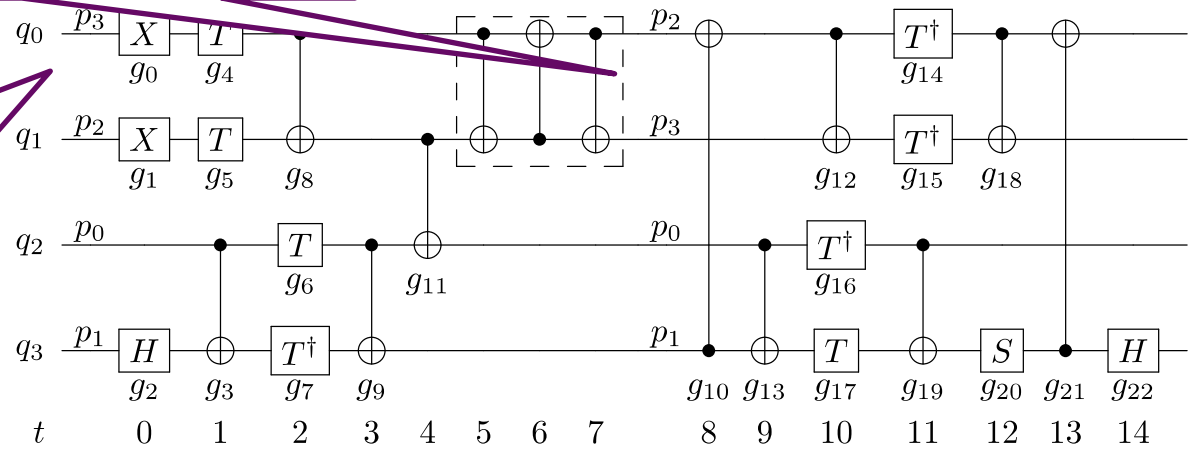


There is a SWAP on (p_2, p_3) finishing at time 7.

Mapping of q_0 changes at time 8.

There are no SWAPs ending at time 0 on any edge connecting p_3.

Mapping of q_0 at time 1 is the same with that at time 0.



Constraints:

Validity

Injective Mapping

Dependency

Mapping constrains Spacetime Coordinates

No Overlap with Other SWAPs

No Overlap with Original Gates

Mapping transformed by SWAPs

Neutral Atom Arrays: Current Status

- >1000 qubits, continuous reload.
  Atom Computing, Munich
- 2Q gate: 99.5%, 1Q gate: 99.9%
  Evered *et al*, *Nature* **622**, 268–272 (2023).
- Reconfigurability: atom moving
  Bluvstein *et al*, *Nature* **604**, 451–456 (2022).
- Mid-circuit measurement, logical qubits and logical program Bluvstein *et al*, *Nature* **626**, 58–65 (2024).

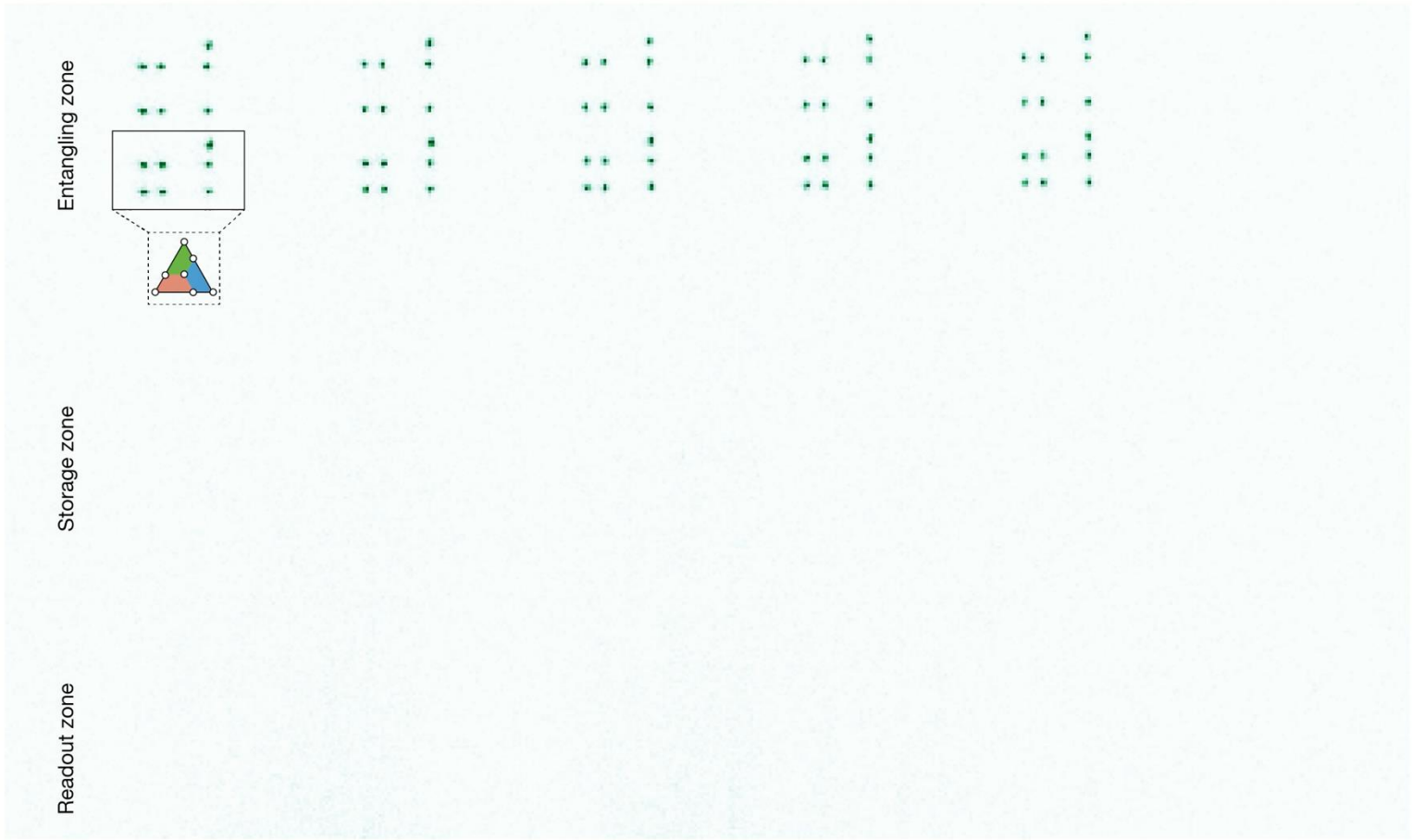
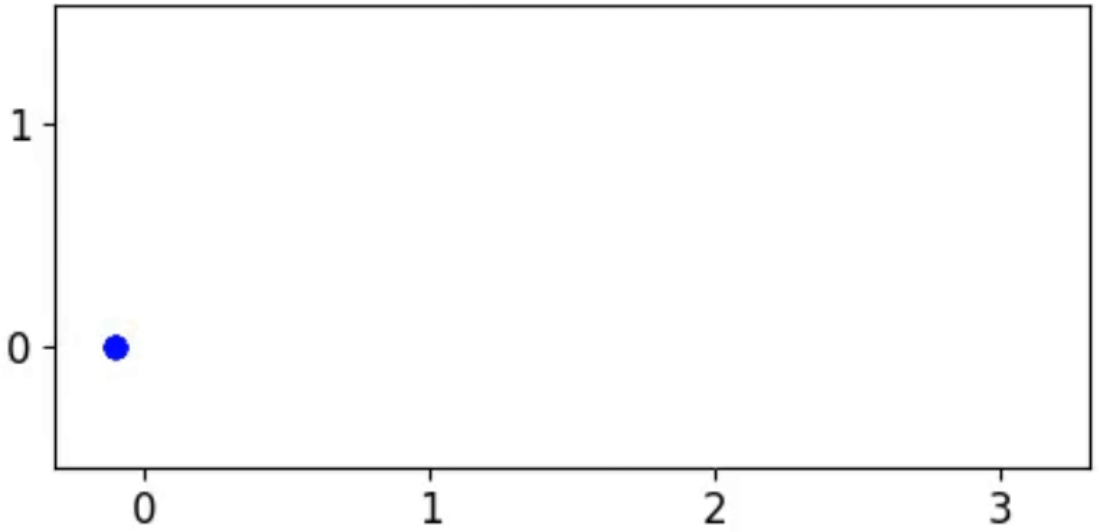


Figure: Bluvstein *et al*, *Nature* **626**, 58–65 (2024).

Compiling for Atom Movement

- SLM: isolated static traps.
- Pull pairs together for 2Q gate, avoid any pair not supposed for 2Q gate.
- AOD to pick up qubits and order preserving movements.



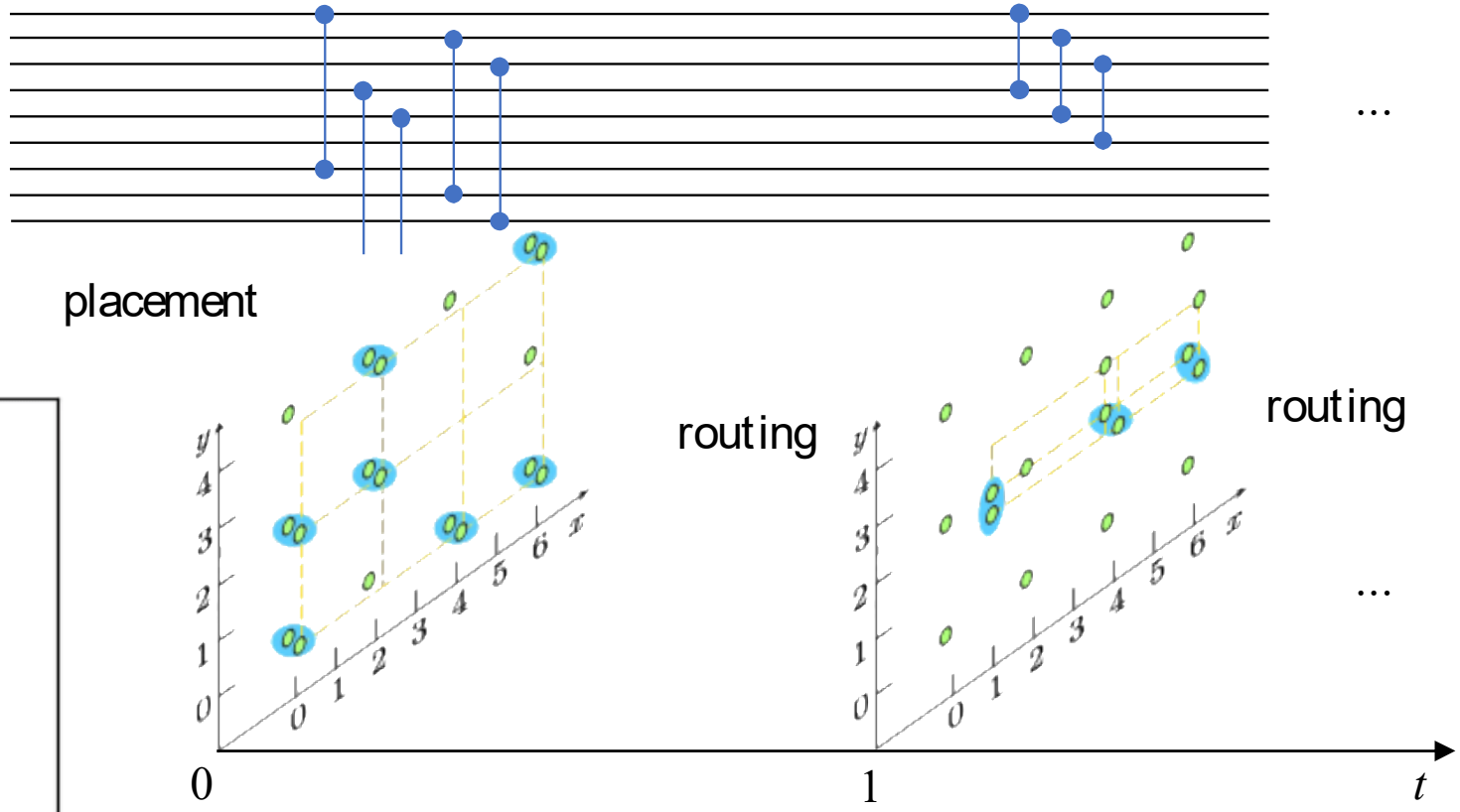


Figure: Tan *et al*, arXiv 2306.03487.

SMT Formulation for Atom Movement

- At each stage of 2Q gate, each qubit has location variables and AOD row&col variables.
- ICCAD'22: put all stages in the SMT model.
- Quantum'24: iterative peeling, as many gates at each stage, switch to multi-stage in the end.
- OLSQ-DPQA open-source repo

name	N_Q	OLSQ-DPQA 2Q	SABRE 2Q	2Q reduction	runtime
QAOA	90	135		**5.10X**	8E+4
ising	98	194	347	**1.79X**	3.25E+4
swap_test	83	328	628	**1.91X**	3.71E+4
dnn	51	392	632	**1.61X**	2.91E+3
adder	64	455	845	**1.86X**	8.32E+3
qugan	71	552	936	**1.70X**	1.36E+4

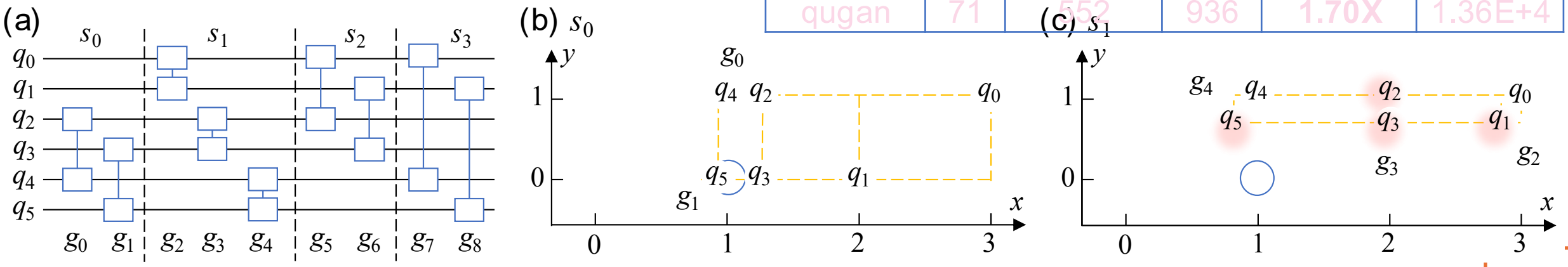
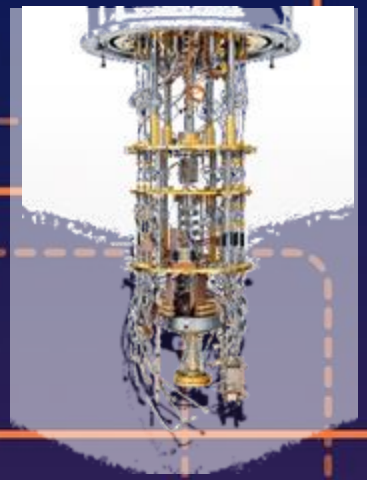
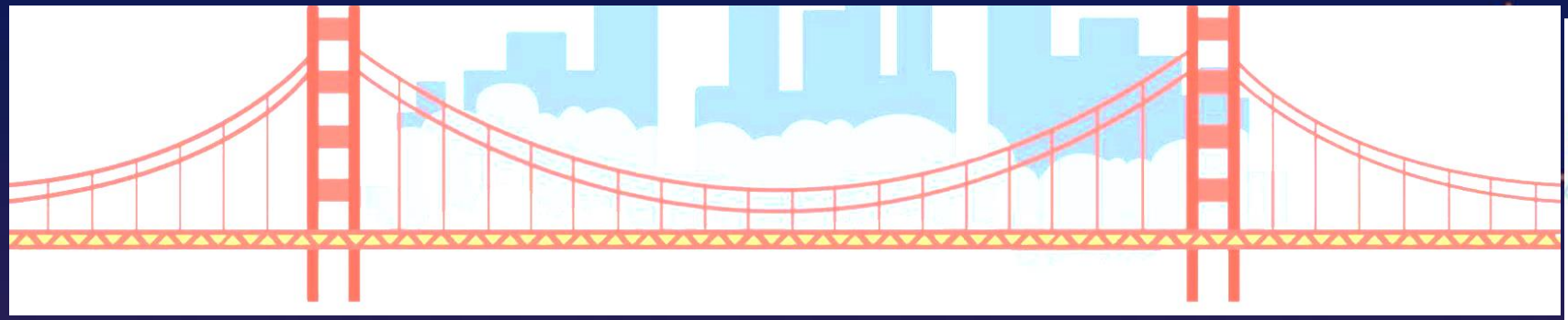


Figure: Tan *et al*, arXiv 2306.03487.

Design and Benchmark the Variational Quantum Circuit and the Quantum Circuit Simulation Cost



QC

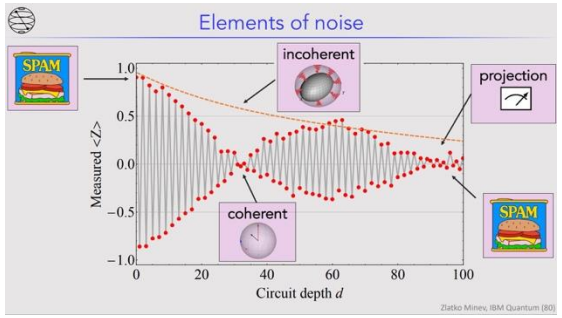


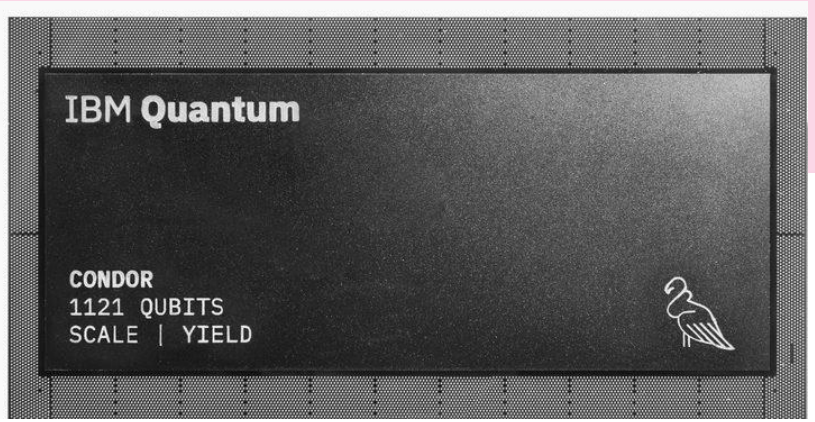
User

Weiwen Jiang, Ph.D.
Assistant Professor
Electrical and Computer Engineering
Junction of Quantum-Classical Computer-Aided Design Lab (JQub)
George Mason University
wjiang8@gmu.edu
https://jqub.ece.gmu.edu

THE CHIPS TO SYSTEMS CONFERENCE
SPONSORED BY CEDA sigda

We Are Entering a New Quanutm Era: Era of Utility

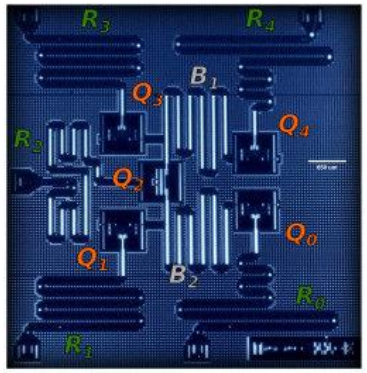




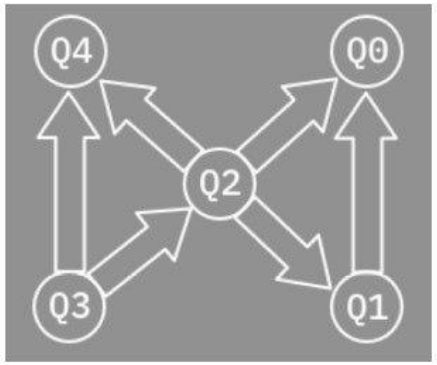
NISQ Era @ 2017

Era of Utility @ 2023

a



b



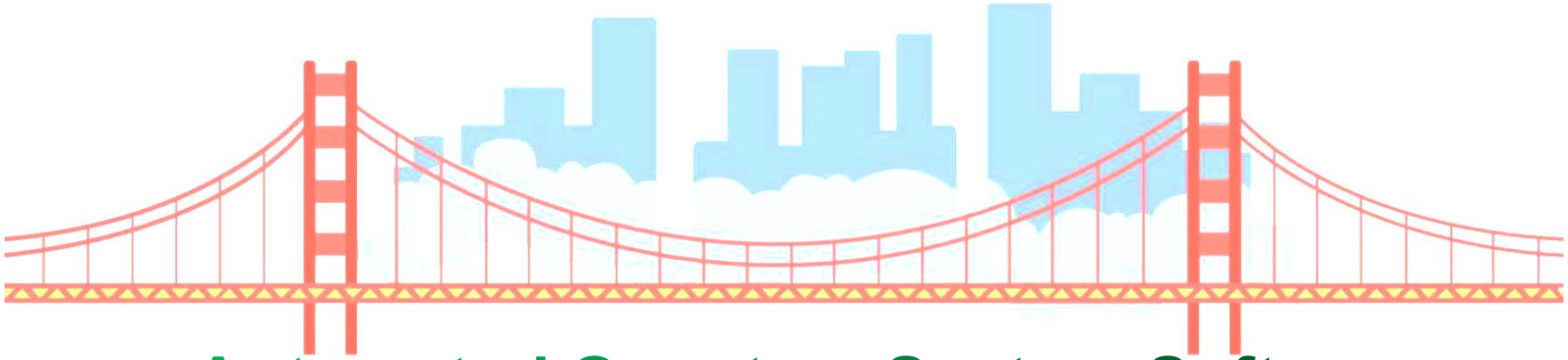
What is the Key in Quantum Utility (Answer from IBM)

The Era of Utility means a focus on **performance**, **stability** and **reliability**

The Era of Utility means **new users**, and **new tools**

Our Thoughts:

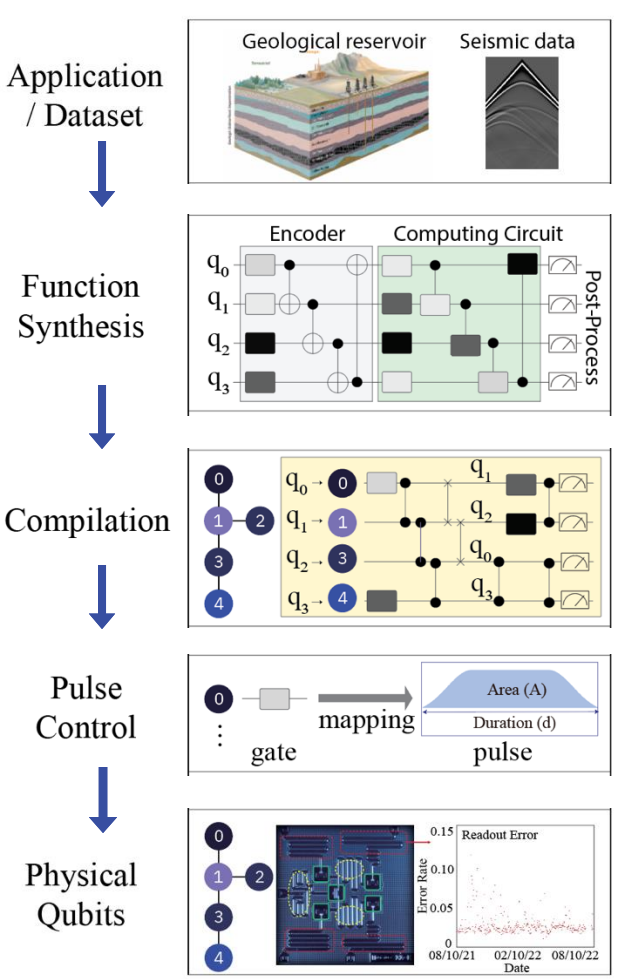
QC



Automated Quantum System Software will be the Bridge

QC Scientist

Outline: Quantum System Software Stack



- **Sythesizability of Domain App.**
  - **ST-VQC**: Data encoding
  - **QuGeo**: Problem scaling
- **Stability in Compilation and Control**
  - **QuCAD**: Circuit-level optimization
  - **QuPAD**: Pulse-level optimization
- **Conclusion**

Sythesizability

ST-VQC

A Novel Spatial-Temporal Variational Quantum Circuit to Enable Deep Learning on NISQ Devices

Published at QuantumWeek 2023

Jinyang Li[1], Zhepeng Wang[1], Zhirui Hu[1], Prasanna Date[2], Ang Li[3], Weiwen Jiang[1]

[1]George Mason University, [2]Oak Ridge National Lab, [3]Pacific Northwest National Lab

QuGeo

QuGeo: An End-to-end Quantum Learning Framework for Geoscience -- A Case Study on Full-Waveform Inversion

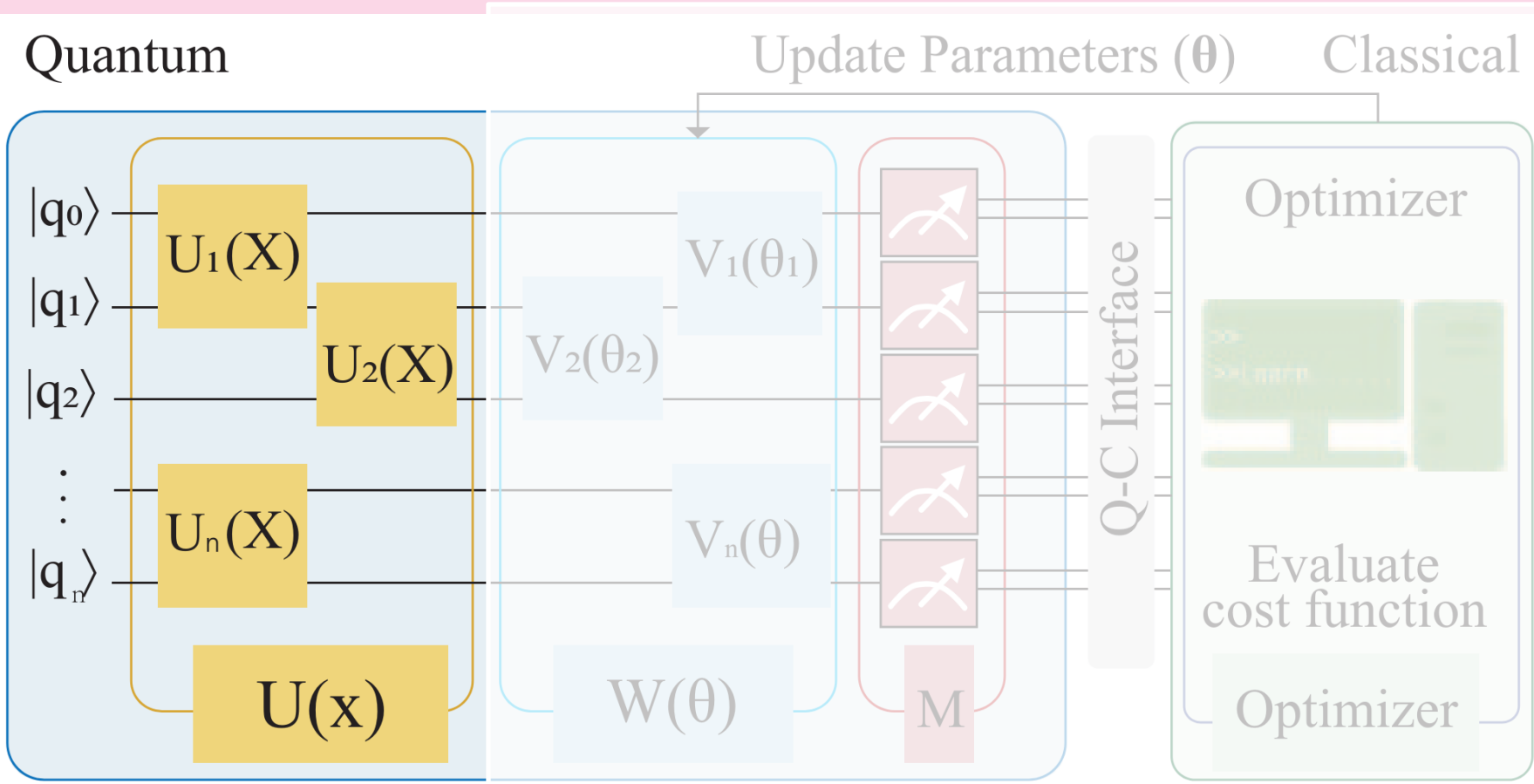
Published at DAC 2024

Weiwen Jiang[1], Youzuo Lin[2]

[1]George Mason University, [2]Los Alamos National Lab

SPONSORED BY CEDA sigda

Variational Quantum Circuits for Learning Tasks



ST-Encoder: Qubit-Depth Tradeoff and Duplication for Non-Linearity

Amplitude Encoding
2^n data to n *qubit*

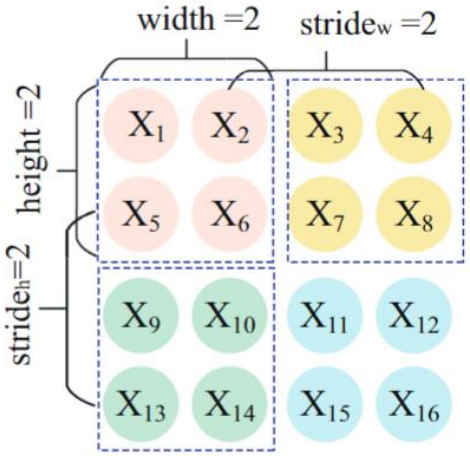
# Qubits	ibm_geneva (27 qubits)		ibmq_jakarta (7 qubits)	
	Circ. Len.	Average Err.	Circ. Len.	Average Err.
2	11	2.3450	11	0.7143
3	36	1.9496	35	2.6842
4	102	6.6860	91	8.1201
5	220	10.8722	211	13.3838
6	478	11.9897	495	20.4578

Long depth leads low fidelity

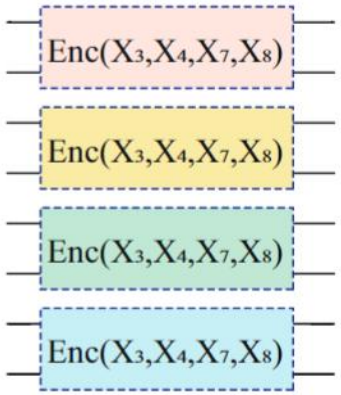
Angle Encoding
n data to O(n) *qubit*

	Amplitude encoding	Angle encoding
2-class	97.5%	90.0%
3-class	92.5%	73.0%

More qubits or info loss



Classical Data



Proposed Group Encoding

Better qubit-depth tradeoff

Results

EVALUATION OF DIFFERENT QUANTUM DESIGNS FOR MNIST-2 ON *ibm_cairo* QUANTUM PROCESSOR

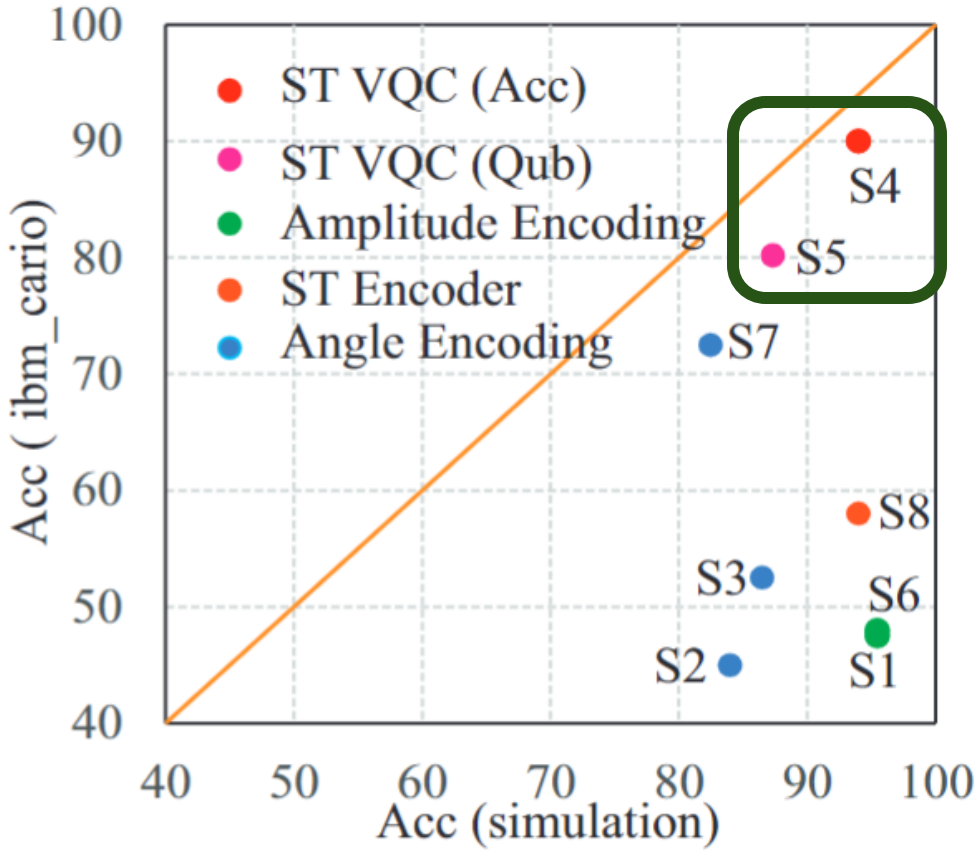
ID	Method	#Qubits	Cir. Depth	#Para.	Accuracy
S1	Amplitude VQC	4	422	40	47.5%
S2	Angle-4 VQC	4	226	40	45.0%
S3	Angle-8 VQC	8	233	44	52.5%
S4	ST-VQC (Acc)	8	309	54	90.0%
S5	ST-VQC (Qub)	6	191	34	80.5%

- Circuits by ST-VQC are **much shorter** than Amplitude VQC, even with more qubits

EVALUATION OF DIFFERENT ENCODING METHODS FOR MNIST-2 ON *ibm_cairo* QUANTUM PROCESSOR

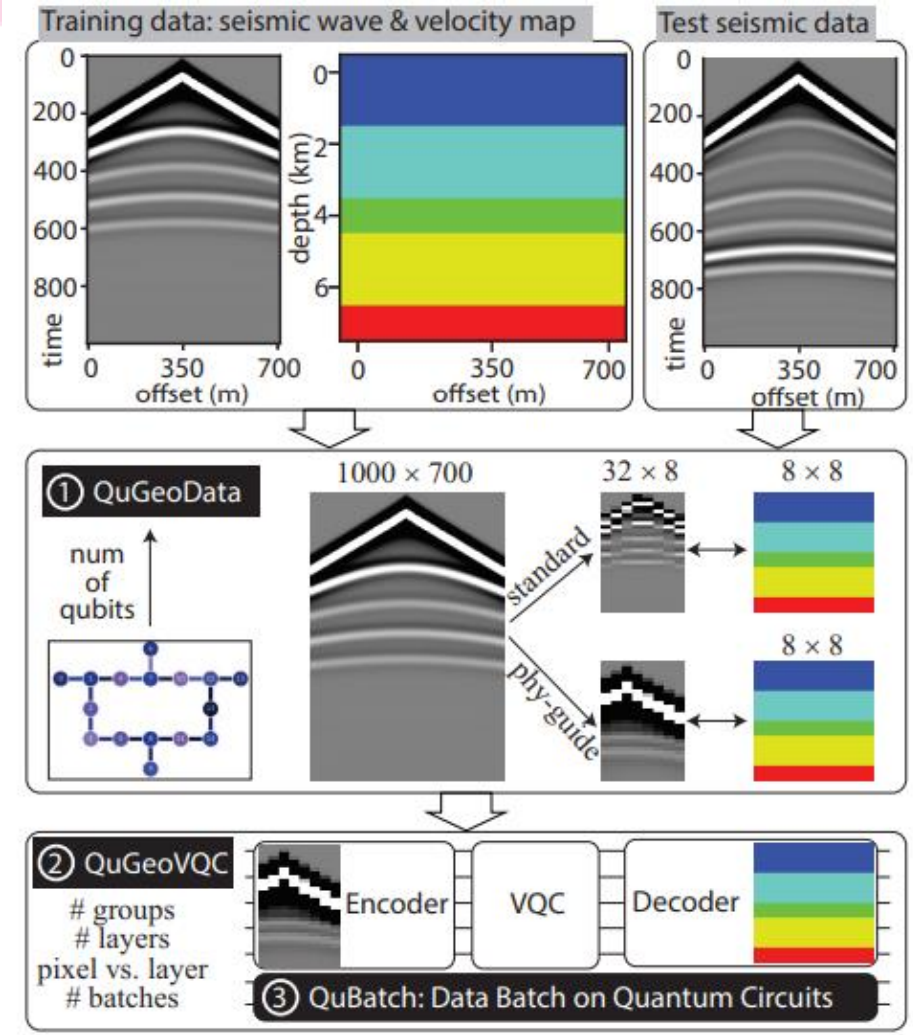
ID	Encoding	Computation	Enc. Depth	Deviation	Accuracy
S6	Amplitude	ST-Processor	99	0.0938	48.0%
S7	Angle-4	ST-Processor	5	0.0032	72.5%
S4	ST-Encoder	ST-Processor	11	0.0058	90.0%

- Circuits by ST-VQC has less info loss for **higher accuracy** over angle encoding



- ST-VQC **significantly outperforms** existing approaches in terms of **fidelity**

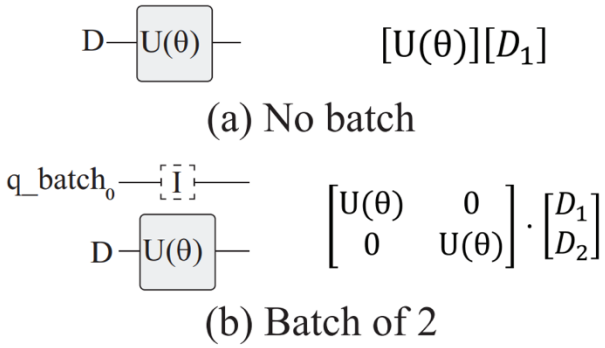
QuGeo Framework on Top of ST-VQC



1. **QuGeoData: Data Preprocessing**
   - Downsampling to fit limited qubits.
   - Encoding using ST-VQC

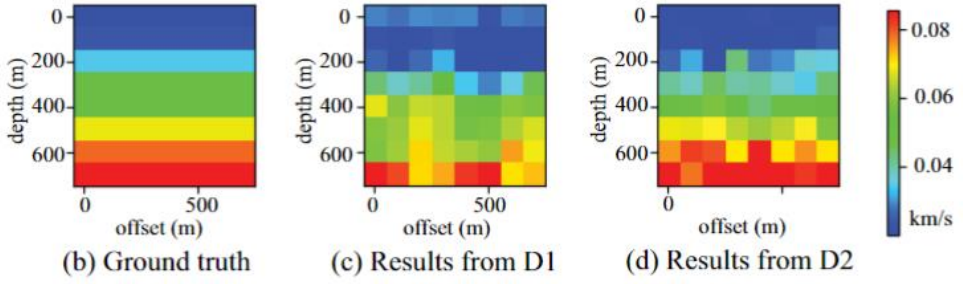
2. **QuGeoVQC: Quantum Learning Structure**
   - Coupled VQC in ST-VQC

3. **QuBatch: Batch Can be Performed with Little overhead**
   - Batch size exponentially increase with the # of qubits

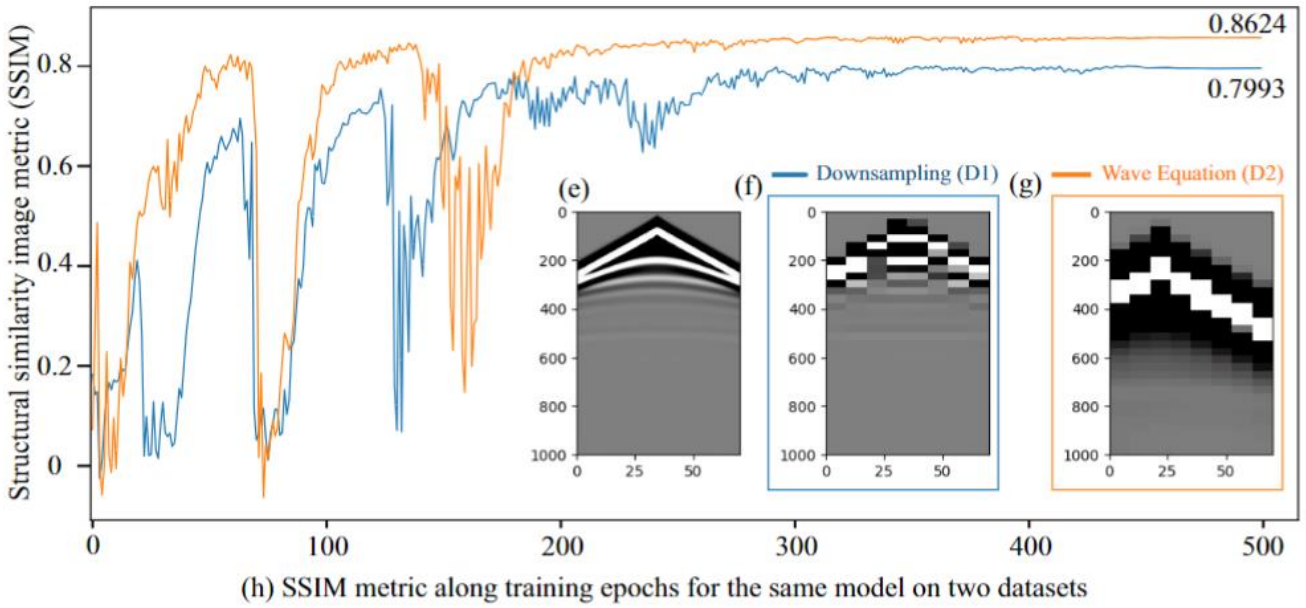


Problem Scaling Should Respect to Physics
VQC Optimization Can Leverage Property of Geophysics Problem

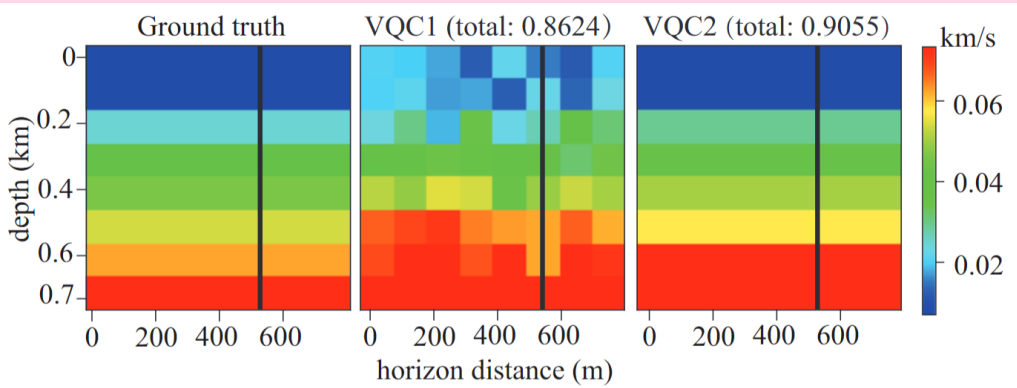
(a) Flat rock layers



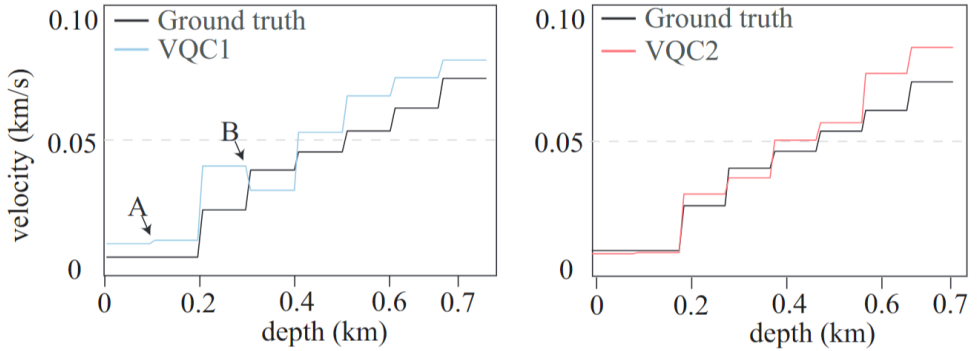
(b) Ground truth (c) Results from D1 (d) Results from D2



(h) SSIM metric along training epochs for the same model on two datasets



(a) Visualization of outputs from two VQCs



(b) Vertical velocity profiles of inversion results at x = 500 m

Quantum Learning Has the Potential to Better Extract Complicated Features, Compared with Classical ML at the Same Scale

Physics-guided data scaling

ML-Based data scaling

Model	Par.	Q-D-FW				**Q-D-CNN**			
		SSIM	vs.	MSE	vs.	SSIM	vs.	MSE	vs.
CNN-PX	634	0.870	BL	4.34E-04	BL	0.87	BL	4.38E-04	BL
CNN-LY	616	0.871	0.04%	4.36E-04	-0.43%	0.87	0.00%	4.36E-04	0.38%
Q-M-PX	576	0.859	-1.28%	4.61E-04	-6.10%	0.86	-0.98%	4.62E-04	-5.45%
Q-M-LY	**576**	0.893	**2.50%**	3.48E-04	**19.84%**	0.91	**3.87%**	3.28E-04	**25.17%**

Hands-on Tutorial of ST-VQC

Developed by Mason's Undergraduates via NSF REU 2024:
Noah M Khan nkhan34@gmu.edu, Hanad Elmi helmi2@gmu.edu,

Guided by PhD student:
Jinyang Li jli56@gmu.edu

Led by PI:
Weiwen Jiang wjiang8@gmu.edu

Sustainability

Carbon Emissions of Quantum Circuit Simulation: More than You Would Think

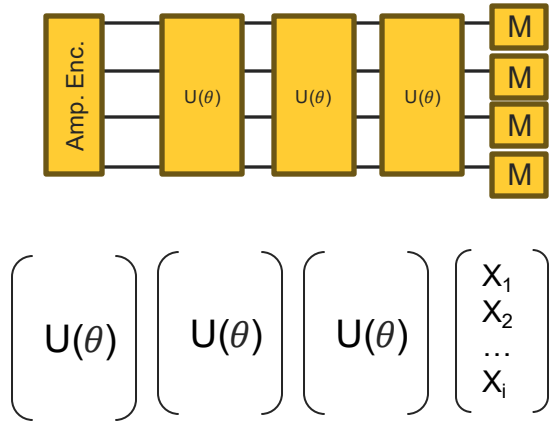
Published at IGSC 2023

Jinyang Li[1], Qiang Guan[2], Dingwen Tao[3], Weiwen Jiang[1]

[1]George Mason University, [2]Kent State University, [3]Indiana University Bloomington

Background on Quantum Circuit Simulation

- Quantum computing's transformative potential
- Hardware limitations & high costs
- Quantum circuits components: quantum states & quantum gates, state vector simulation
- Cloud-based simulators such as IBM quantum, Cirq, and etc.



Amplitude Encoding Quantum Circuit

Importance of Quantum Circuit Simulations and A Deeper Cost

Why it matters?

- Addressing Quantum Hardware Availability
- Noise influence
- Use for Algorithm Testing and Development.
- Quantum Error Mitigation.

A Deeper Cost: Carbon Emissions

- Normal evaluation metrics for quantum circuit simulations
- Energy cost of quantum circuit simulations leading to CO_2e emissions.
- The urgency of considering carbon footprint due to global greenhouse gas concerns.

Simulations vs. Everyday Activities

- One simulation can produce CO_2e up to 1.81 times of a Transatlantic flight
- And approximately 48 times of training a standard transformer base model

Table 1: Comparison of CO_2e emissions from human activities, and classical machine learning training.

Activity	CO_2e (kg)
Household electricity use for a day [4]	12.44
Driving a car for 100 km [2]	24.80
Transatlantic flight, 1 passenger [7]	313.90
Transformer_base training [16]	11.79
Transformer_large training [16]	87.09
ELMo training [16]	118.84
Quantum Circuit Simulation (43 qubits)	568.77

Simulation-Emission Model

- Sources of emissions: Embodied Emissions, Idle Power Consumption, and Dynamic Power Consumption.
- Factors affecting Dynamic Power Consumption in simulation, such as number of qubits, computational resources, etc.
- Quantifying the Emissions: $CO_2e = n \times Pp \times T \times CI \times PUE.$

Small Quantum Circuit Simulations

- Using the MNIST dataset showing results of CO2e emissions for quantum vs. classical neural networks.

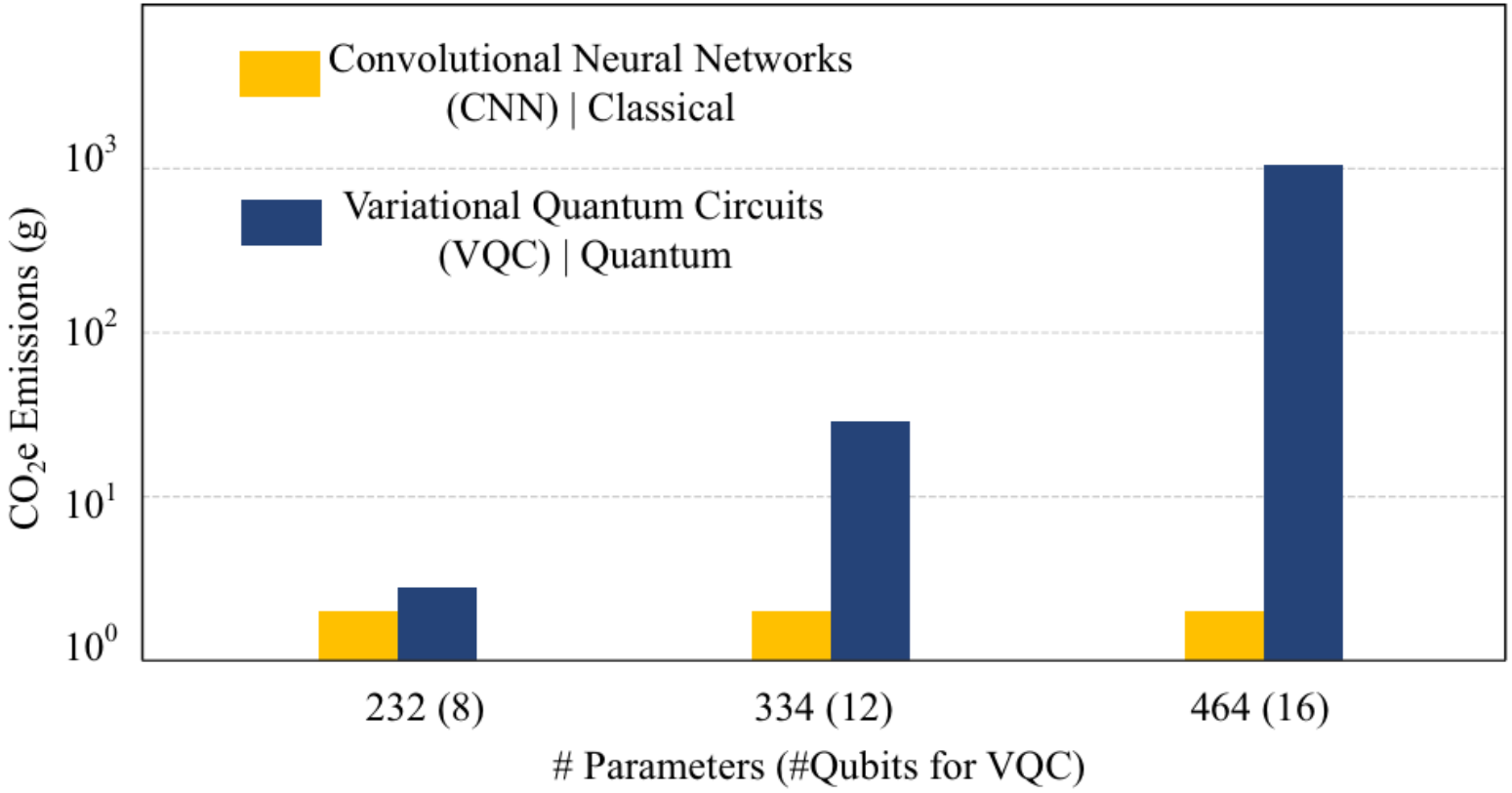


Figure 1: CO_2e Emission Result for CNN and VQC.

Large Quantum Circuit Simulations

- Emissions rise with qubit number
- A growing concern in simulation development

Table 2: emission result from aws large-scale quantum circuit simulation.

#Qubit	#Execution time (hours)	#Processor	CO_2e emissions (kg)
40	1.25	256	54.23
41	1.41	512	122.91
42	1.58	1024	274.74
43	1.64	2048	568.77

Hands-on Tutorial

Developed by PhD student:
Jinyang Li jli56@gmu.edu

Led by PI:
Weiwen Jiang wjiang8@gmu.edu

Thank you for attention

Weiwen Jiang
wjiang8@gmu.edu

62 THE CHIPS TO SYSTEMS CONFERENCE

SPONSORED BY CEDA sigda

Pulse-Level Noise Mitigation for Quantum Computing

Siyuan Niu

University of Central Florida

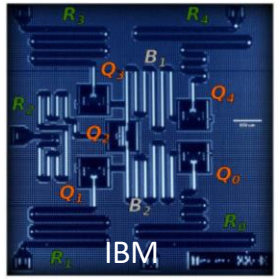
DAC 2025 - Quantum Computing Design Automation Tutorial

SPONSORED BY CEDA sigda

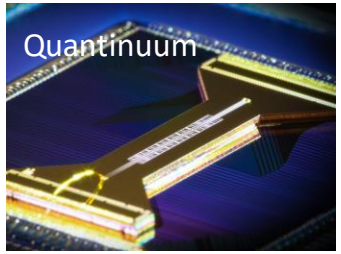
Main challenge in quantum hardware

- Quantum hardware has made great progress in recent years.

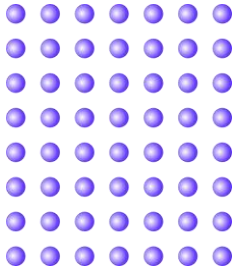
Superconducting qubits



Trapped ions

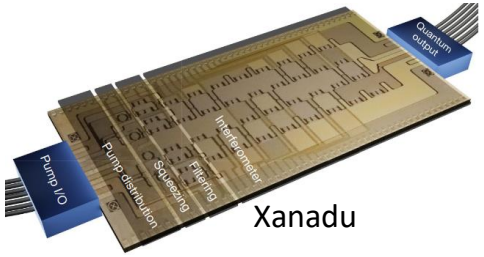


Neutral atoms



QuEra

Photonic chips

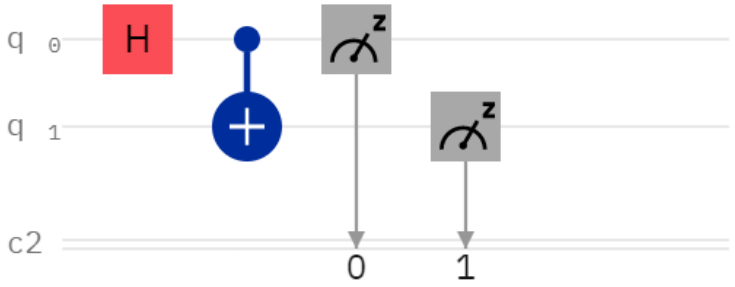


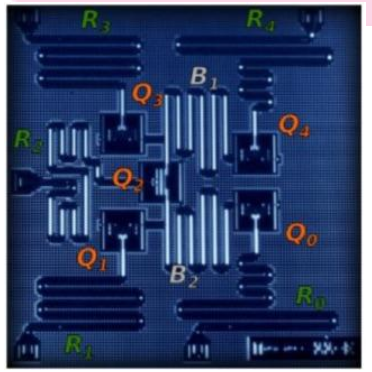
Xanadu

- No practical quantum advantage yet.
- Main challenge: noise in the quantum hardware.
- Roadmap: achieve fault-tolerant quantum computer by 2030.

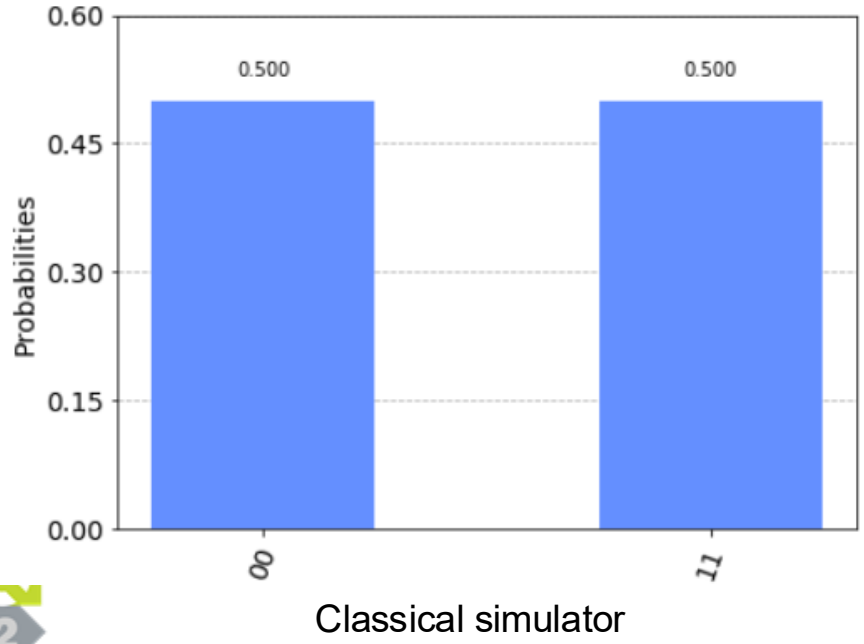
Errors of quantum circuit execution on hardware

- Quantum circuit execution on classical simulator and quantum hardware



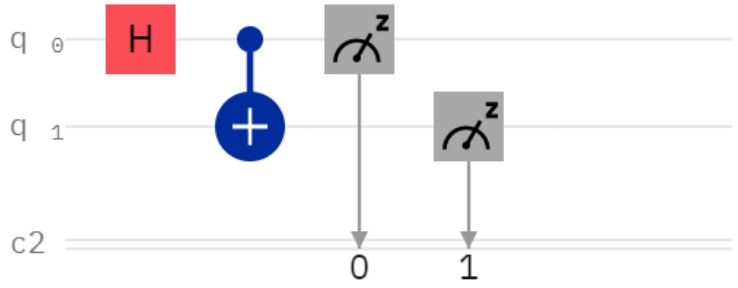


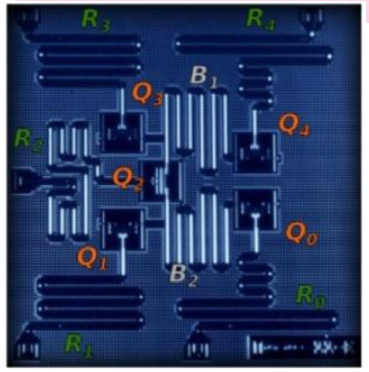
source: IBM



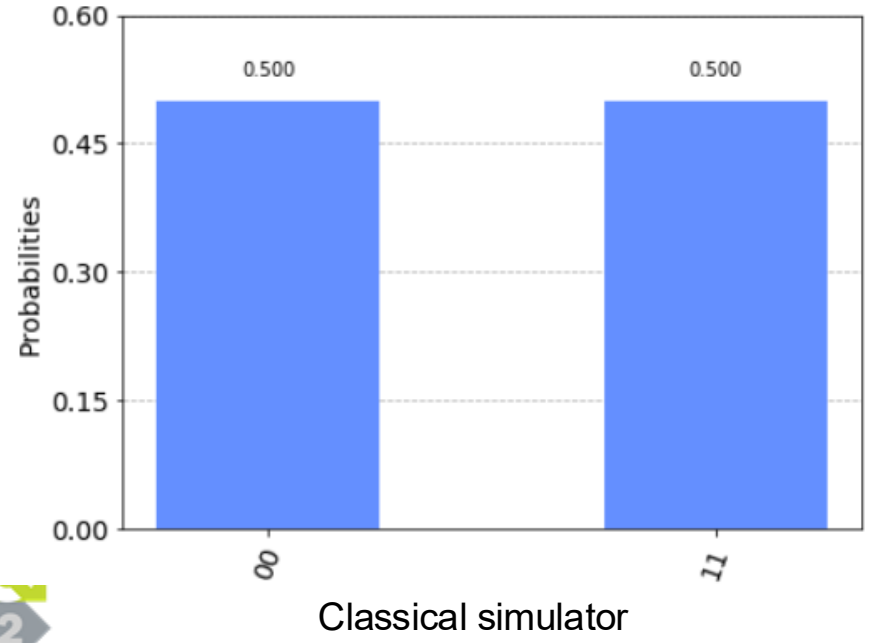
Errors of quantum circuit execution on hardware

- Quantum circuit execution on classical simulator and quantum hardware

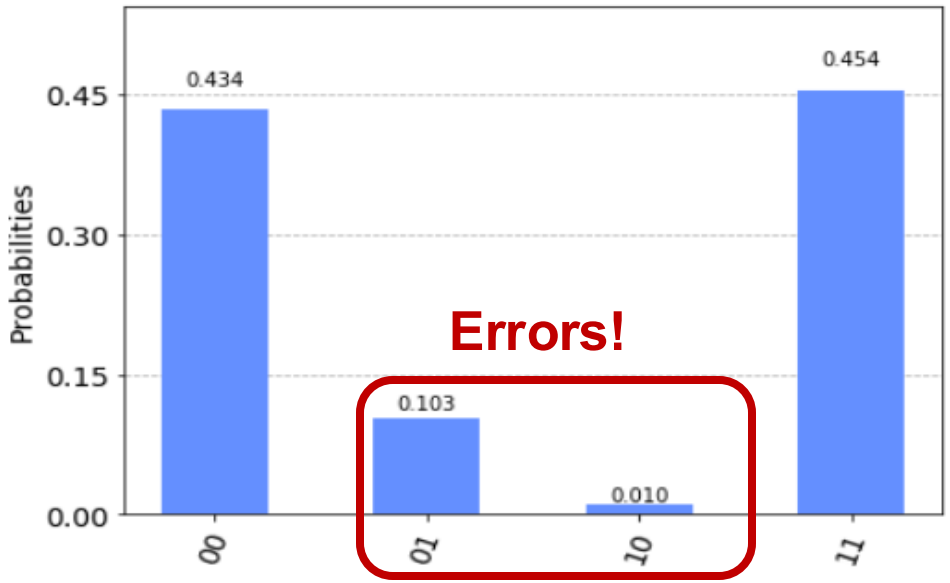




source: IBM



Classical simulator



Quantum hardware

Two categories of quantum algorithms

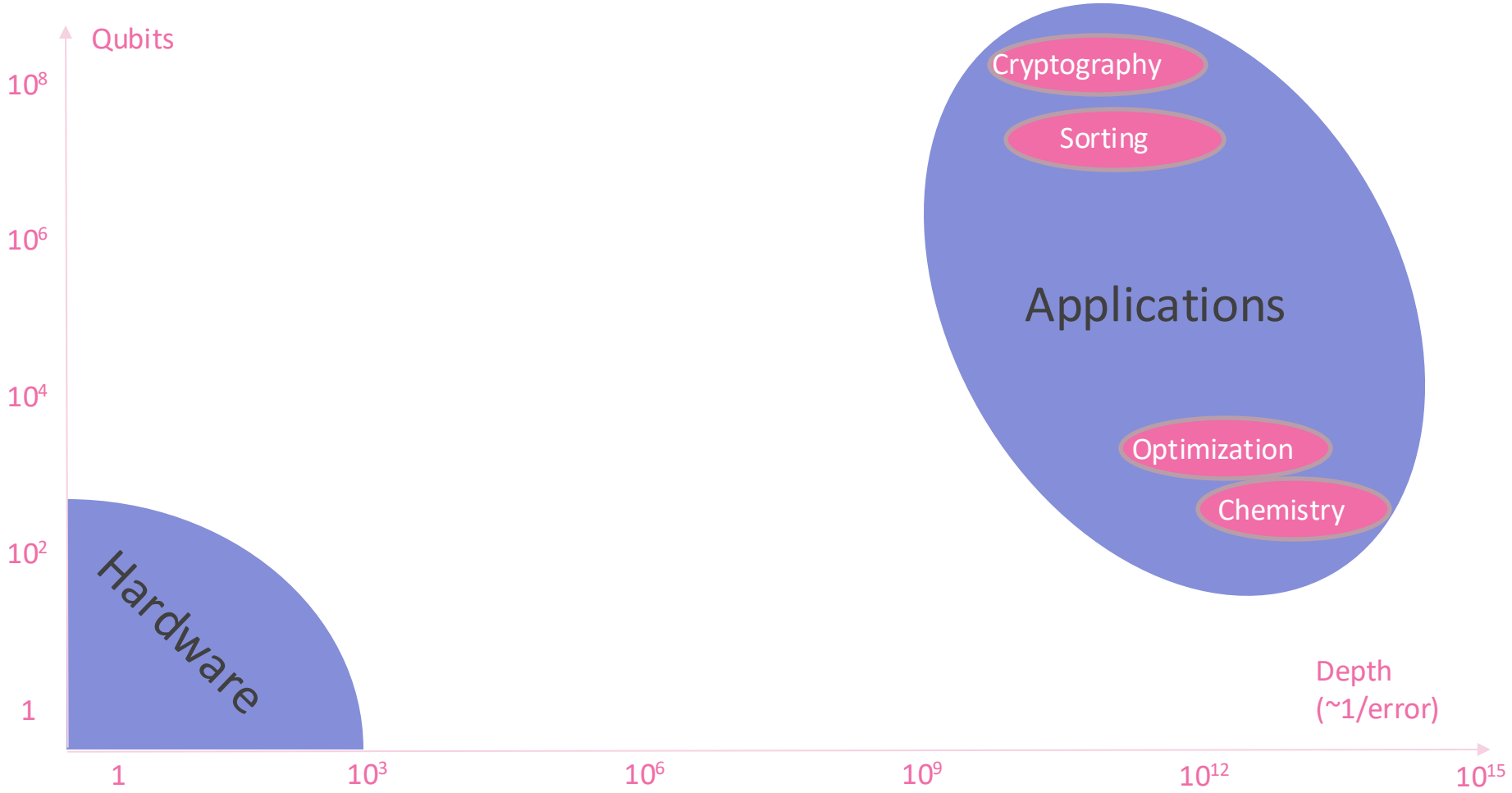
- **Fault-tolerant** quantum algorithms
  - Suitable for fault-tolerant quantum computer
  - Shor's algorithm - exponential speedup
    - Cryptography
  - Grover's algorithm - quadratic speedup
    - Searching problem

Two categories of quantum algorithms

- **Fault-tolerant** quantum algorithms
  - Suitable for fault-tolerant quantum computer
  - Shor's algorithm - exponential speedup
    - Cryptography
  - Grover's algorithm - quadratic speedup
    - Searching problem
- **Near-term** quantum algorithms
  - Suitable for near-term quantum computer
  - Quantum heuristic and speedup unclear
  - Variational quantum eigensolver (VQE)
    - Quantum chemistry
  - Quantum approximate optimization algorithm (QAOA)
    - Combinatorial optimization

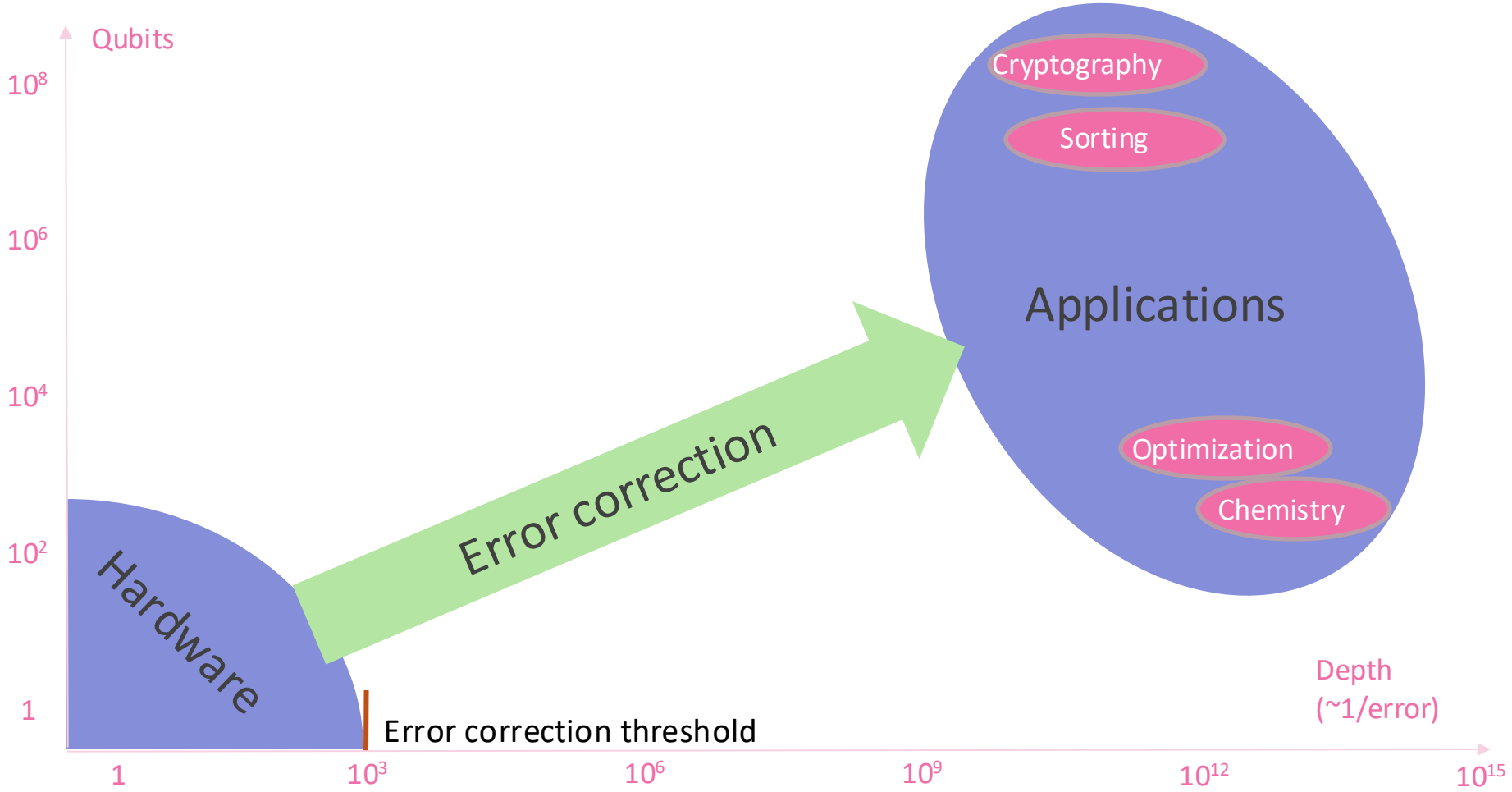
Gap between quantum hardware and applications

- Mismatch between current quantum hardware capability and application resource requirements.



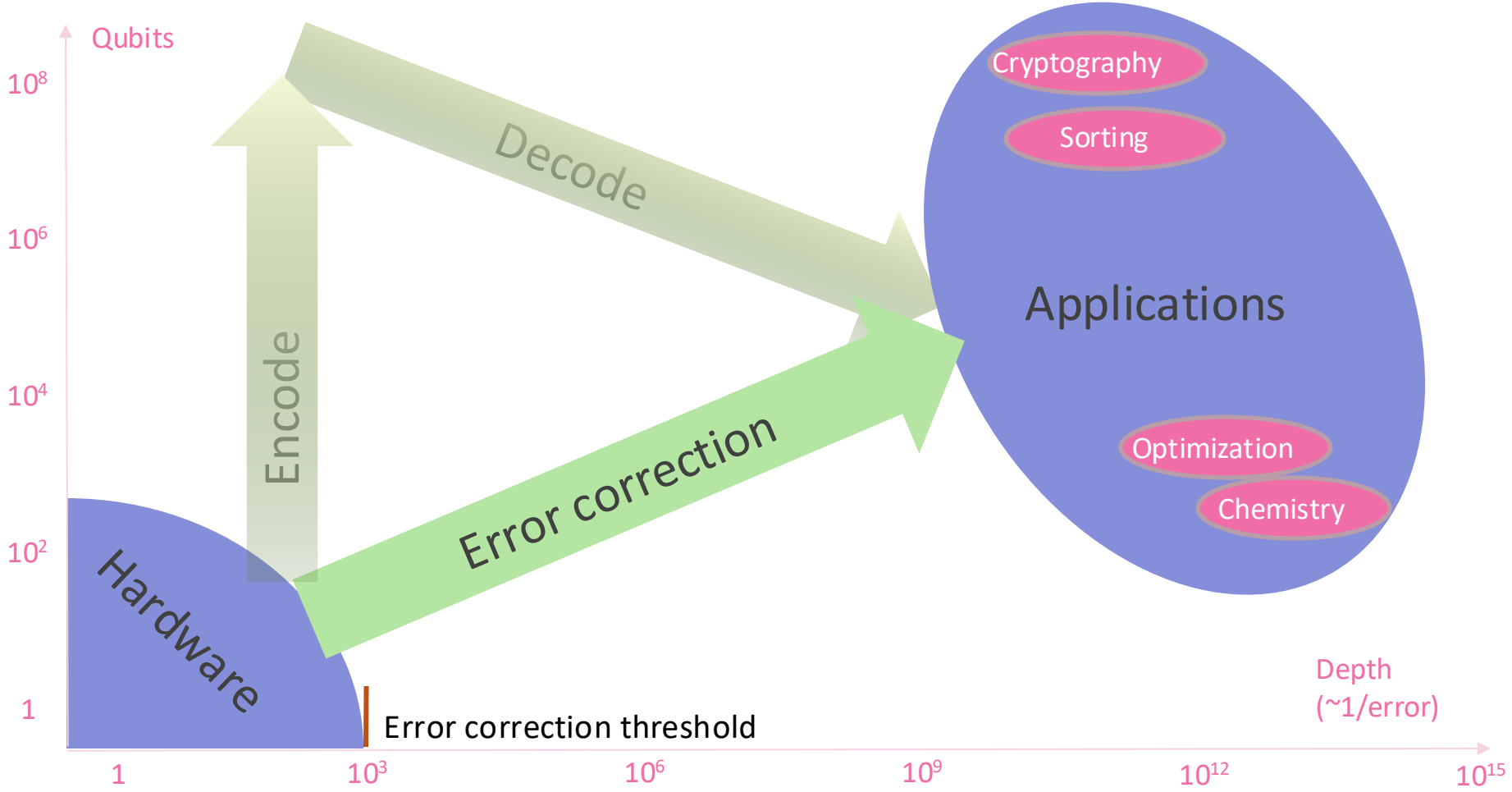
Gap between quantum hardware and applications

- Build a bridge using error correction if the error rate is below the threshold.



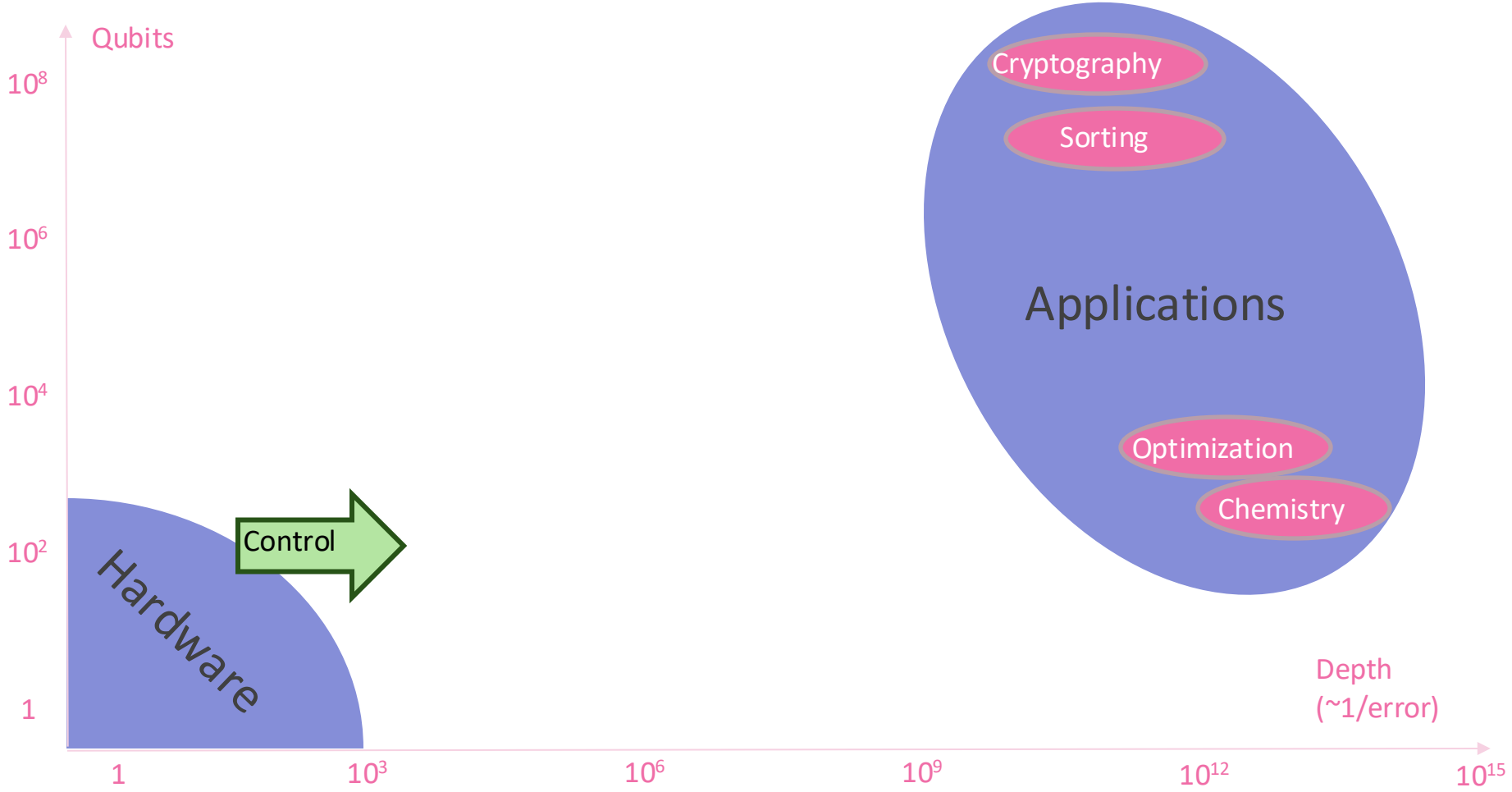
Gap between quantum hardware and applications

- Encoding and decoding require large number of qubits.
- Not practical for near-term quantum hardware.



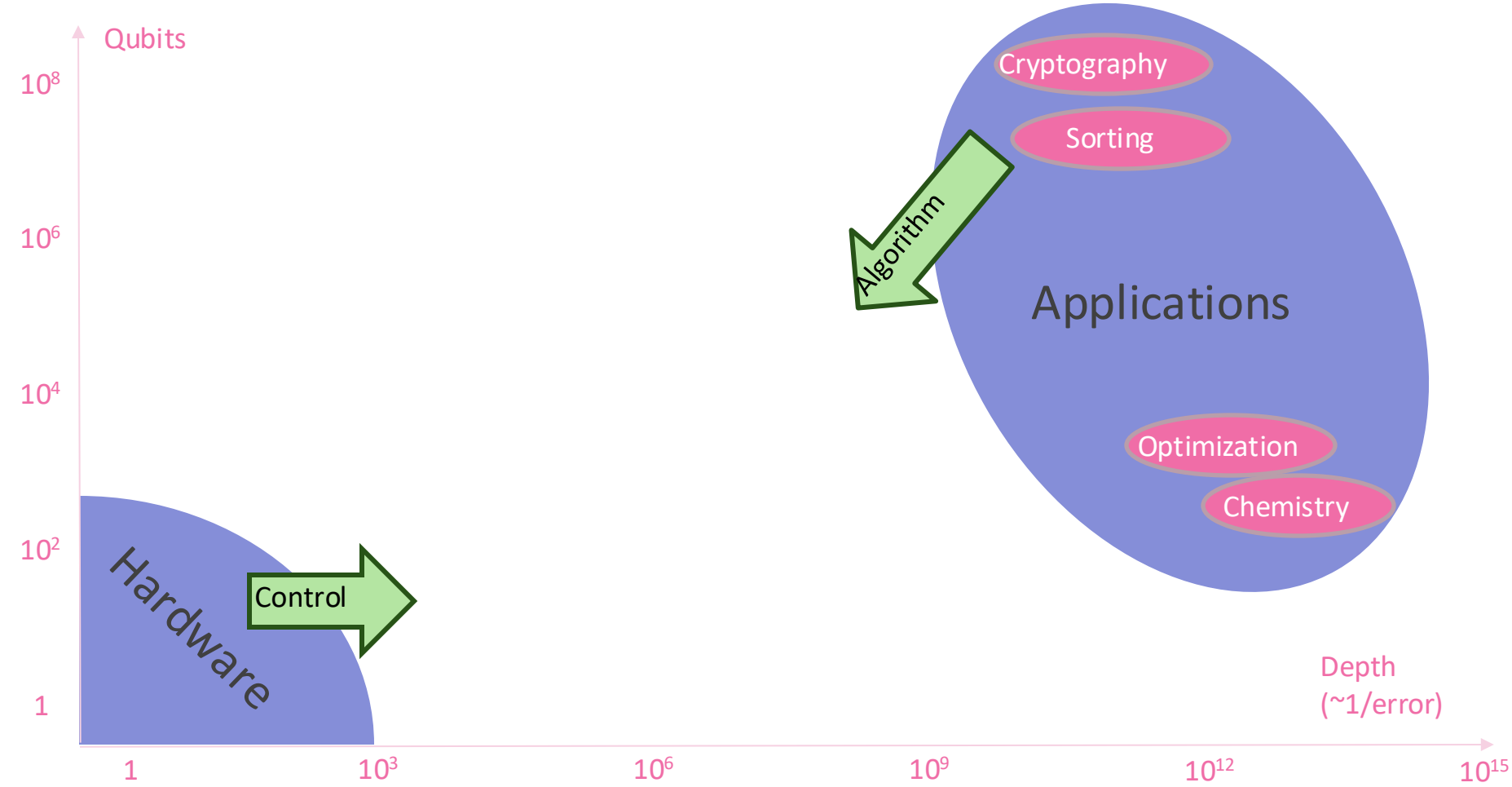
Gap between quantum hardware and applications

- Improve the hardware control for better qubits.



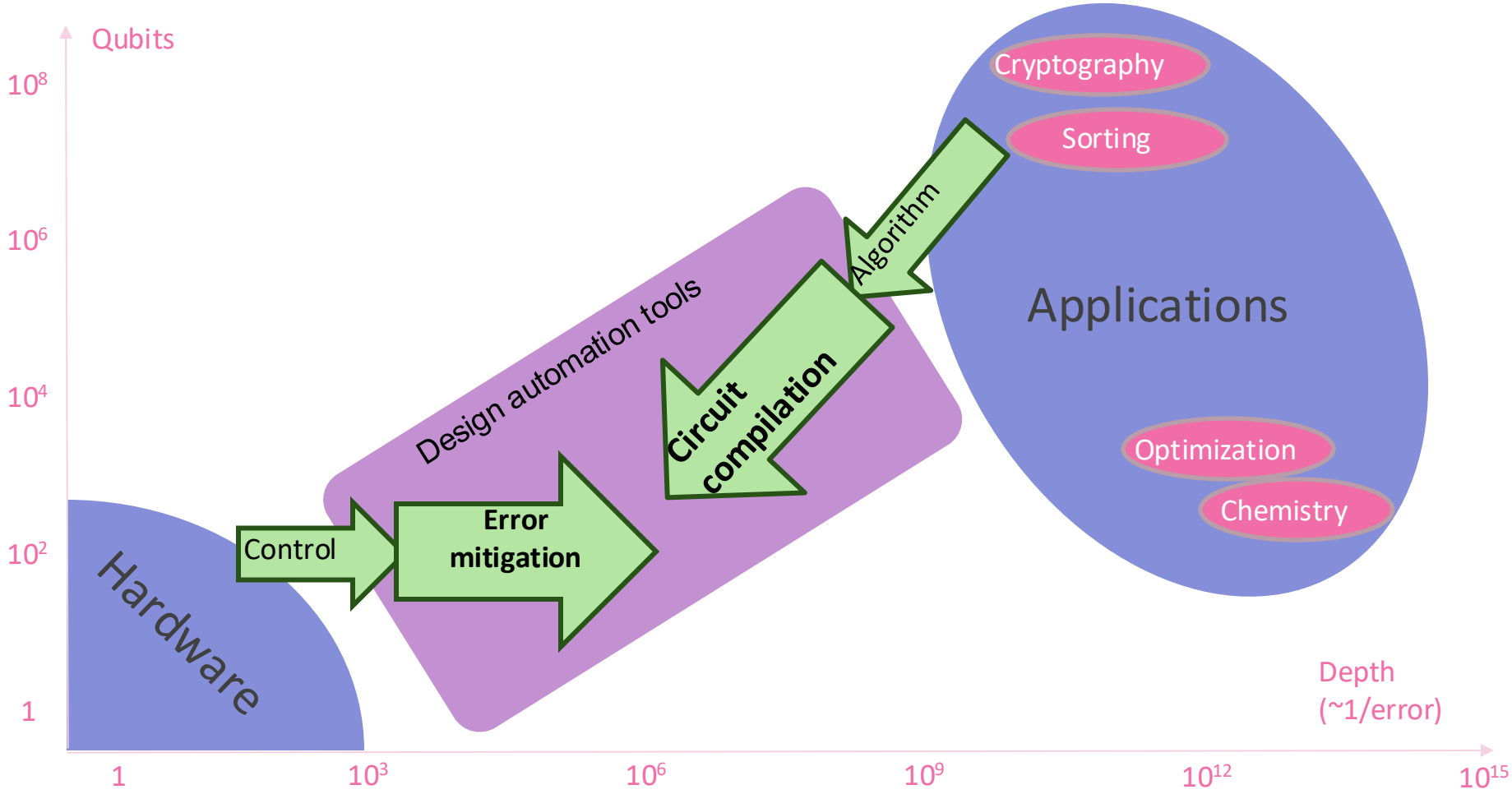
Gap between quantum hardware and applications

- Improve hardware control for better qubits.
- Improve algorithm design to reduce resource requirements.



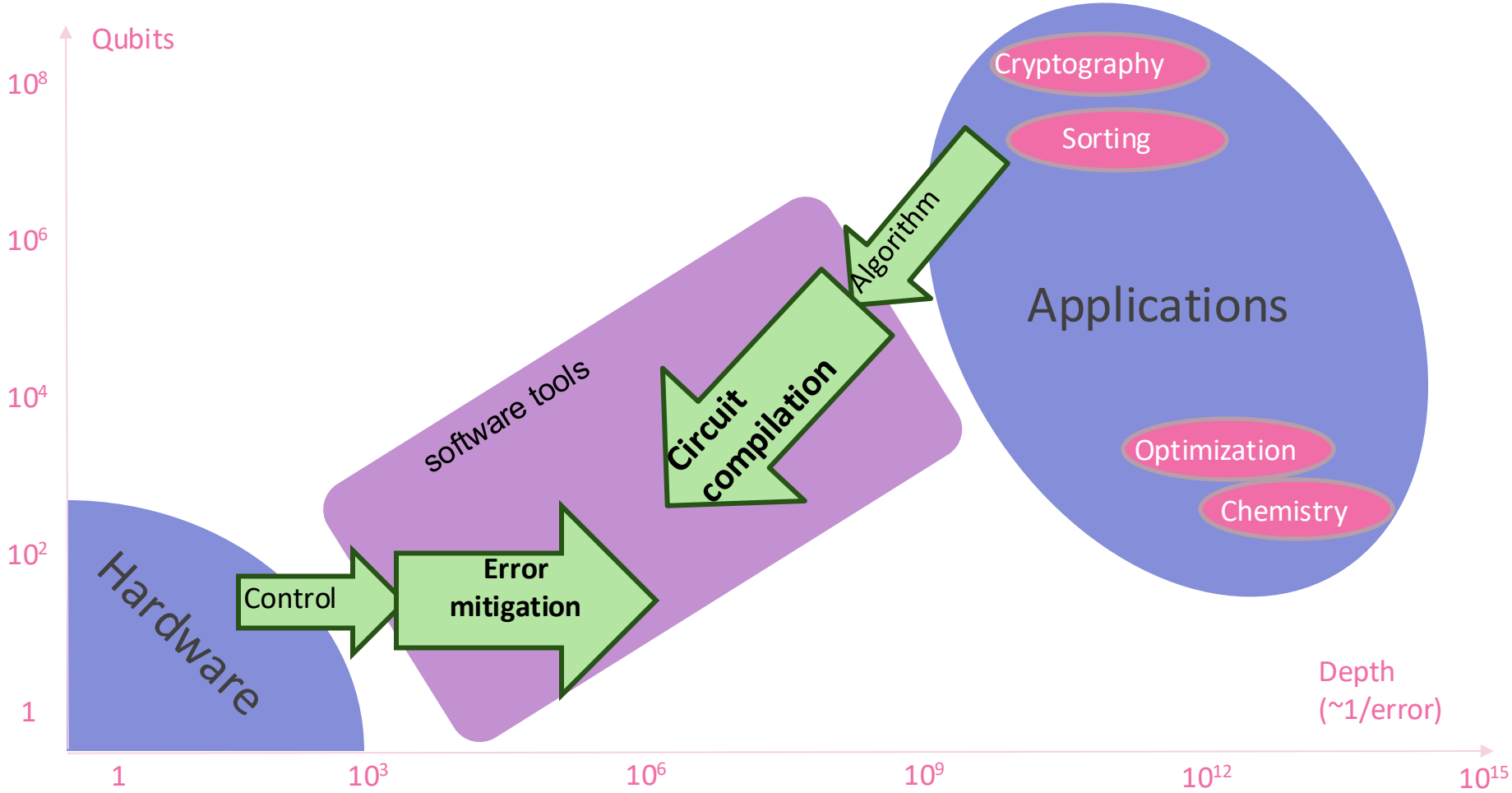
Gap between quantum hardware and applications

- Improve hardware control for better qubits.
- Improve algorithm design to reduce resource requirements.

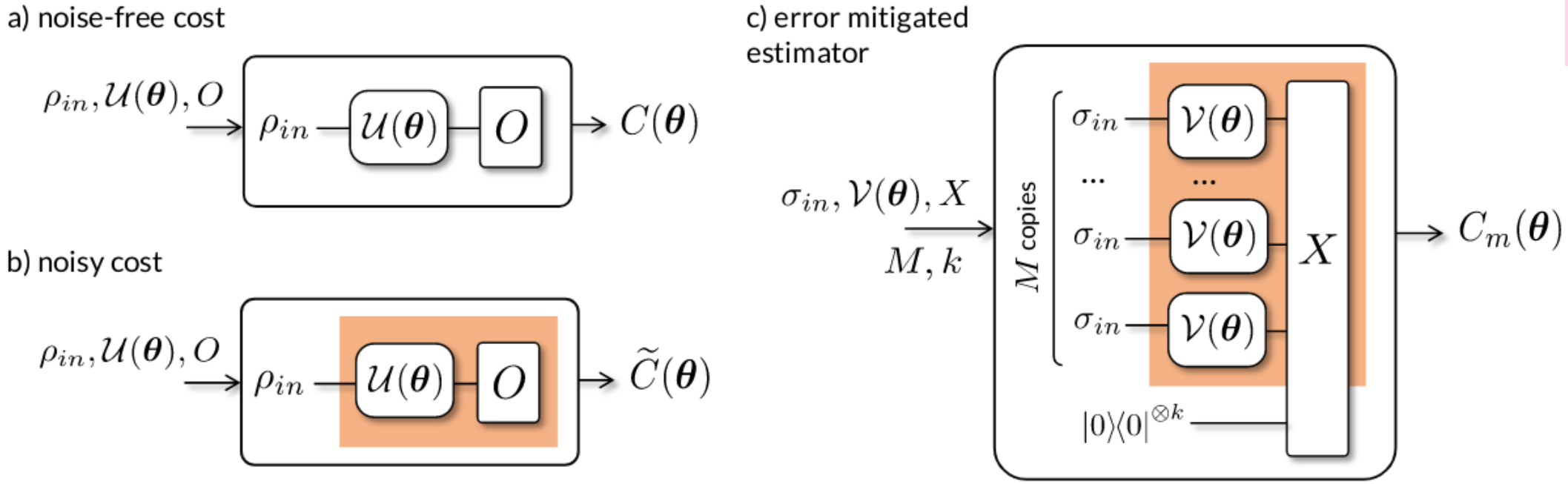


Gap between quantum hardware and applications

- Improve hardware control for better qubits.
- Improve algorithm design to reduce resource requirements.



Error mitigation



- Introduce errors and analyze the circuits in a model to derive the error mitigated expectation value
- Examples
  - Zero noise extrapolation
  - Probabilistic error cancellation

Wang et al. Quantum 8 (2024): 1287.

Error suppression

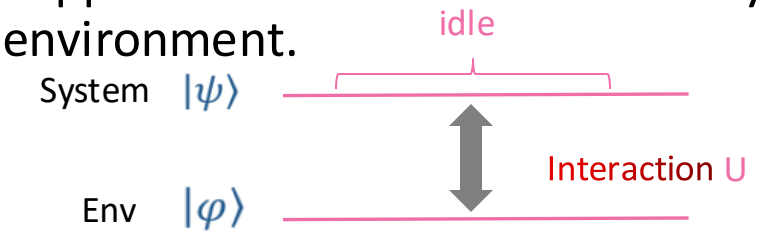
- Handling errors at the closest level to the hardware.
- Anticipate the hardware errors and correct them in the hardware level.
- Examples
  - Pulse scaling
  - Dynamical decoupling
  - Quantum feedback control
- No quantum resource overhead!

Error suppression

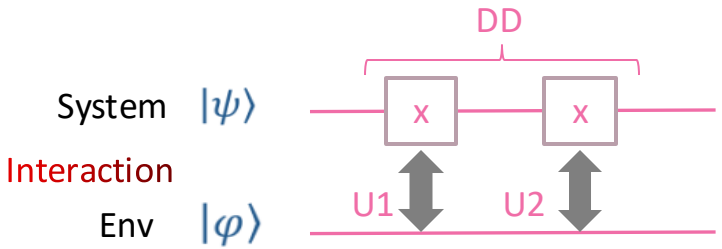
- Handling errors at the closest level to the hardware.
- Anticipate the hardware errors and correct them in the hardware level.
- Examples
  - Pulse scaling
  - Dynamical decoupling
  - Quantum feedback control
- No quantum resource overhead!

Introduction to dynamical decoupling (DD)

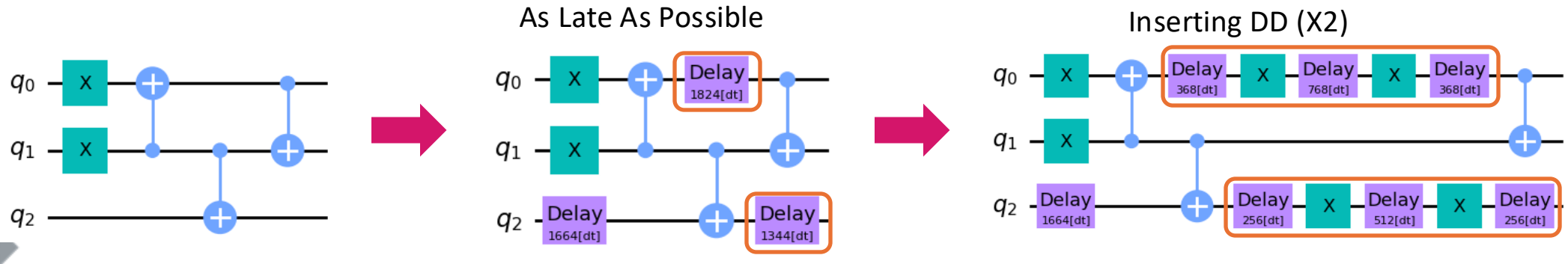
- Suppress the decoherence error by reducing the interaction between the system and the environment.



T1: energy loss
T2: phase loss



- Example of applying DD to a quantum circuit



DD suppresses decoherence for single-qubit system

- Hamiltonian of an open quantum system

$$H = H_S + H_B + H_{SB}$$

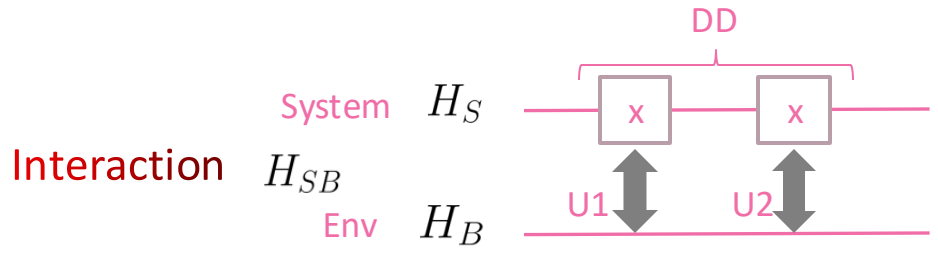
System Bath (env) System-bath interaction (unwanted)

- Suppose we only have dephasing (Z) for single-qubit system-bath coupling

$$H_{SB} = \sigma^z \otimes B^z$$

$$U_{SB} = I$$

Suppressed!



Different DD sequences

DD sequences	Pulse implementation
Hahn echo [39]	X or Y
CP/CPMG [6], [25]	$(X)^n$ or $(Y)^n$
XY4 [50]	$(X-Y-X-Y)^n$
XY8 [16]	$(XY4-\overline{XY4})^n$
XY16 [16]	$(XY8-\overline{XY8})^n$
UDD [48]	$(X)^n$ or $(Y)^n$ with space τ where $\tau = \sin^2(\frac{\pi j}{2n+2}), j \in \{1, 2, ..., n\}$
KDD [41]	$(\pi)_{\frac{\pi}{6}+\phi} - (\pi)_{\phi} - (\pi)_{\frac{\pi}{2}+\phi} - (\pi)_{\phi} - (\pi)_{\frac{\pi}{6}+\phi}$ where $(\theta)_{\phi} = e^{\frac{i\theta}{2}[\cos(\theta)\sigma_x + \sin(\theta)\sigma_y]}$, $\phi = 0, \frac{\pi}{2}, 0, \frac{\pi}{2}$

Note: n is the repetition time of the basic DD cycle and $(\theta)_{\phi}$ is a rotation of θ around the axis defined by ϕ.

Different DD sequences

DD sequences	Pulse implementation
Hahn echo [39]	X or Y
	$\phi = 0, \frac{\pi}{2}, 0, \frac{\pi}{2}$

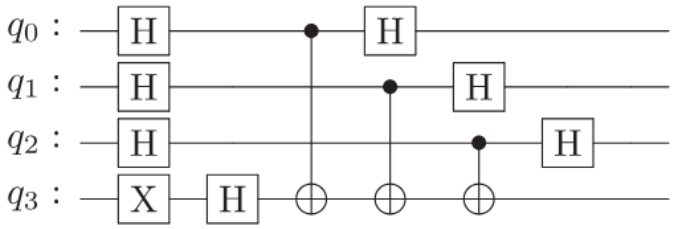
Note: n is the repetition time of the basic DD cycle and $(\theta)_\phi$ is a rotation of θ around the axis defined by ϕ.

What are the impacts of different DD sequences on quantum algorithms and quantum hardware?

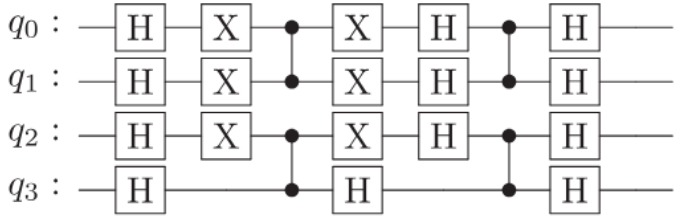
Selected IBM quantum hardware

IBM Q	Jakarta	Guadalupe	Toronto	Montreal
n	7	16	27	27
QV	16	16	32	128

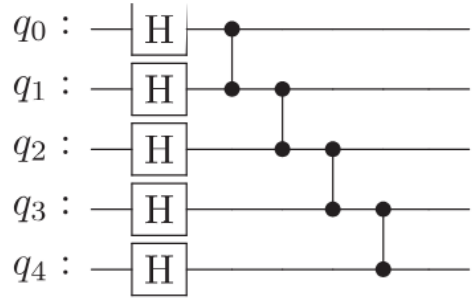
Selected quantum algorithms



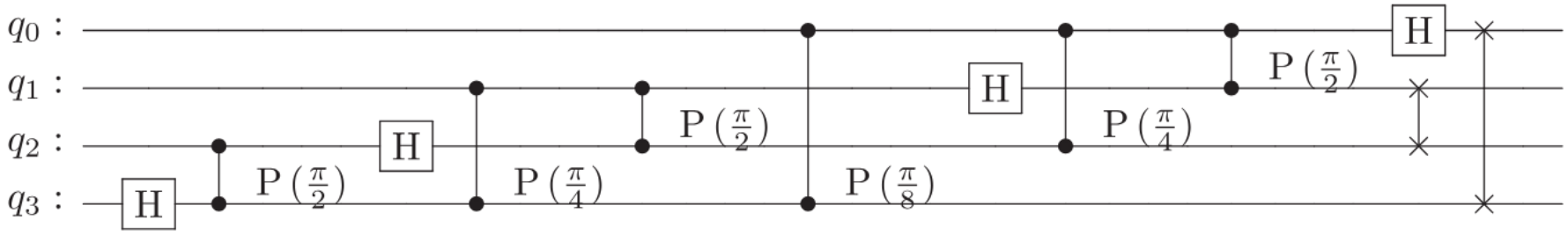
(a) Bernstein Vazirani



(b) Hidden Shift



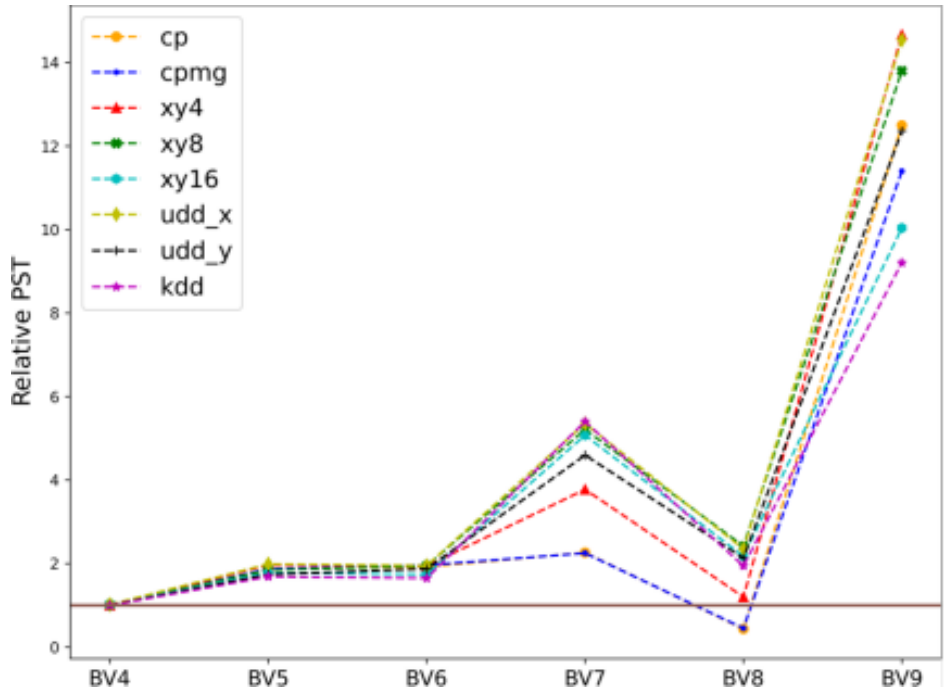
(c) Graph State



(d) QFT

Experimental results - BV algorithm

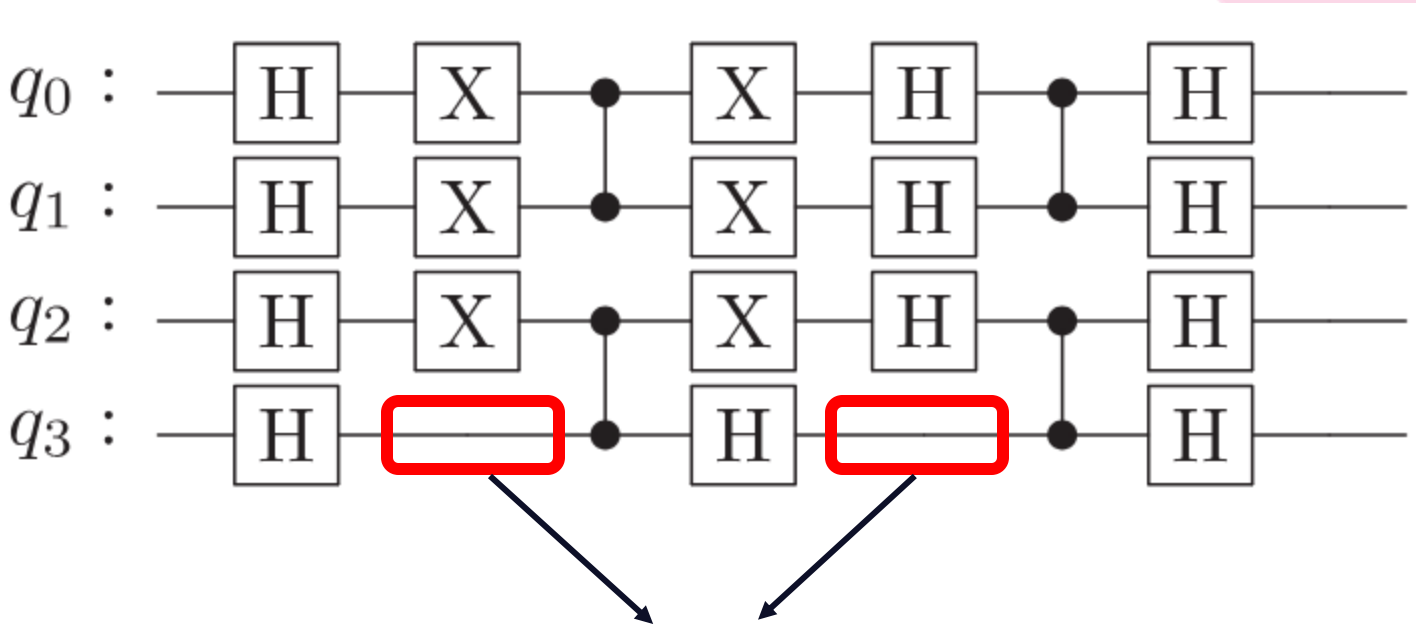
Relative PST results for BV circuits on IBM Q Guadalupe.



IBM Q Hardware	Relative PST		
	Max	Min	Avg
Jakarta	1.1 (UDD_X)	1.05 (KDD)	1.09
Guadalupe	4.53 (UDD_X)	3.16 (CPMG)	3.82
Toronto	3.18 (UDD_X)	2.44 (XY16)	2.79
Montreal	1.77 (UDD_X)	1.55 (KDD)	1.68

All the DDs can improve the BV circuit fidelity.
UDD_X usually performs the best for BV.

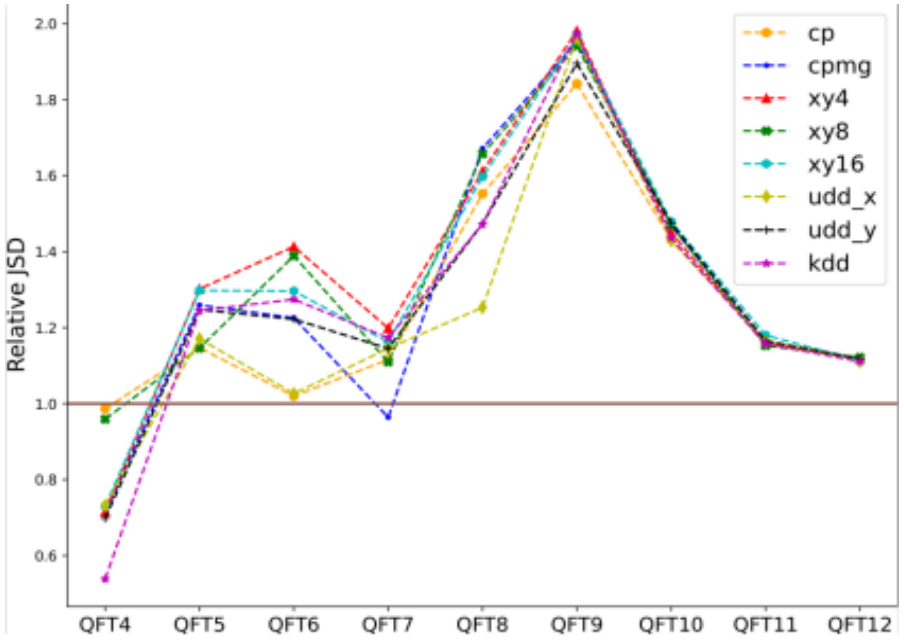
Experimental results - HS algorithm



Too short to fit any DD sequences!

Experimental results - QFT

Relative JSD results for QFT circuits on IBM Q Montreal.

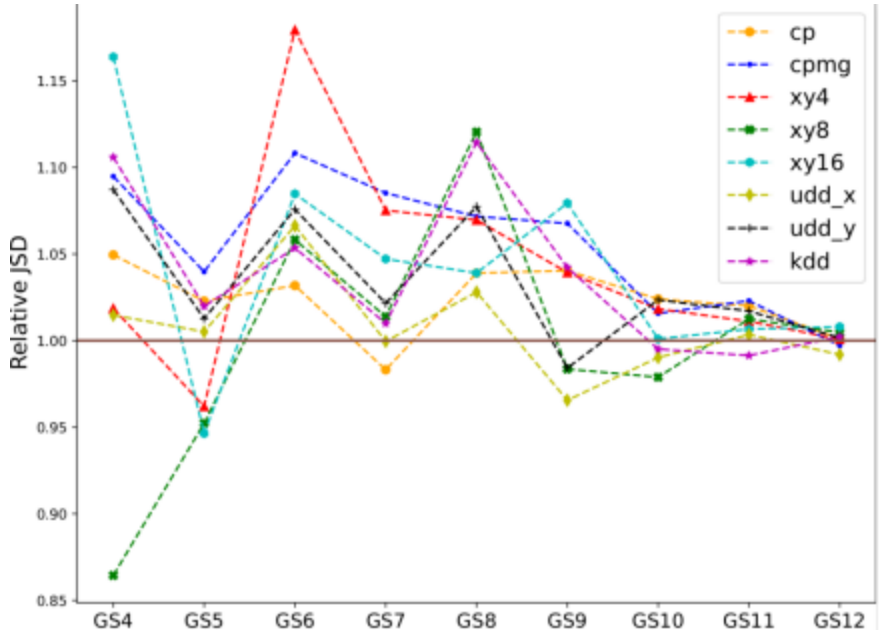


IBM Q Hardware	Relative JSD		
	Max	Min	Avg
Jakarta	1.1 (XY16)	0.89 (UDD_Y)	1.01
Guadalupe	1.17 (CPMG)	1.06 (UDD_X)	1.14
Toronto	1.23 (KDD)	1.14 (UDD_X)	1.18
Montreal	1.33 (XY4)	1.22 (UDD_X)	1.28

DD can be beneficial for QFT circuits but are not as large as for BV circuits.
DD sequence is more favorable when n > 5.

Experimental results - Graph State

Relative JSD results for Graph State circuits on IBM Q Toronto.



IBM Q Hardware	Relative JSD		
	Max	Min	Avg
Jakarta	1 (CP)	0.9 (UDD_Y)	0.95
Guadalupe	1.21 (XY4)	1.06 (UDD_Y)	1.17
Toronto	1.06 (CPMG)	1 (XY8)	1.03
Montreal	1.02 (CP)	0.96 (XY16)	0.99

DD performance is different across IBM devices.

Only beneficial for IBM Q Guadalupe.

Pulse-efficient transpilation (PE)

- Limitation of IBM Q hardware
  - Only one 2-q operation in basis gates -> CNOT
- CNOT is implemented by Rzx(π/2) and 1-q gates Pulse-efficient transpilation technique

$$R_{ZX}(\theta) = XR_{ZX}(-\tfrac{\theta}{2})XR_{ZX}(\tfrac{\theta}{2})$$

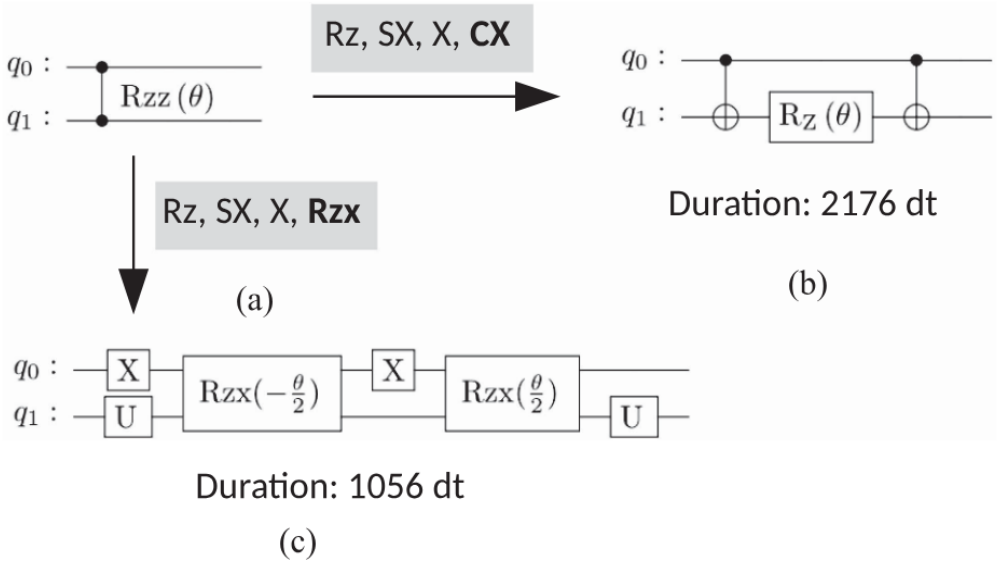


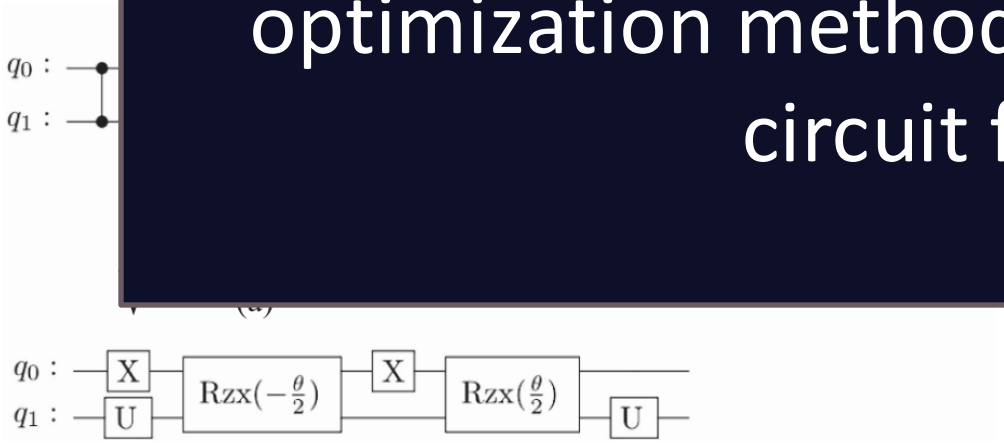
Fig. Comparison of CNOT-basis and Rzx-basis transpilation for Rzz gate.

Earnest et al. PRX. 2021.

Pulse-efficient transpilation (PE)

- Limitation of IBM Q hardware
  - Only one 2-q operation in basis gates -> CNOT
- CNOT is implemented by Rzx(π/2) and 1-q gates Pulse-efficient transpilation techn

Will the combination of DD and pulse-efficient optimization method further improve the circuit fidelity?



Duration: 1056 dt

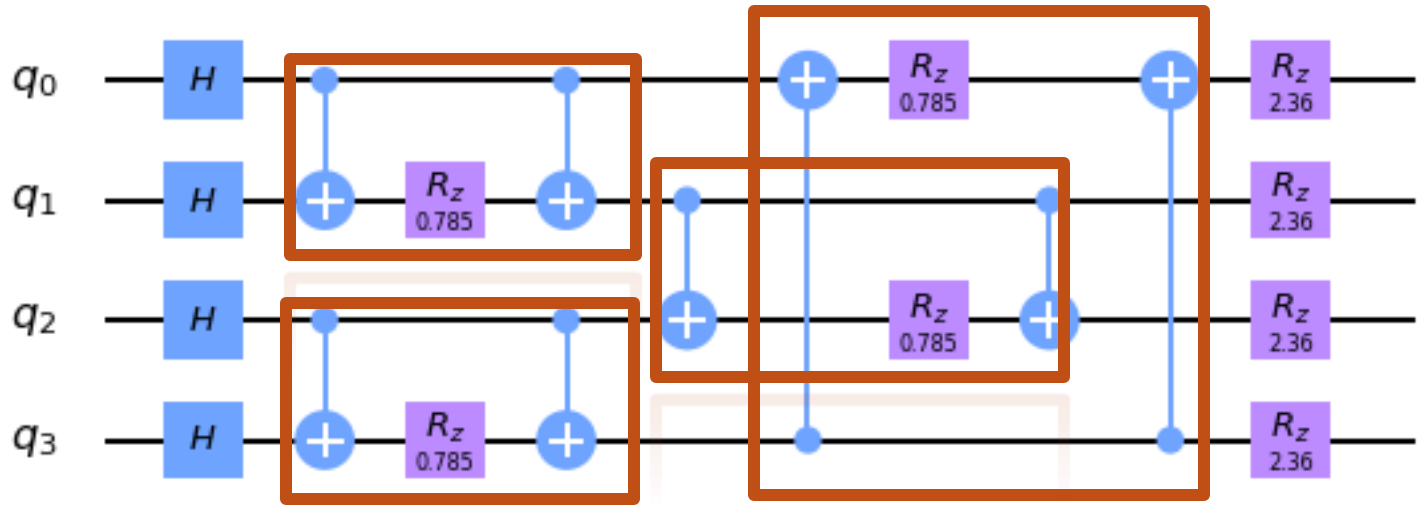
(c)

Earnest et al. PRX. 2021.

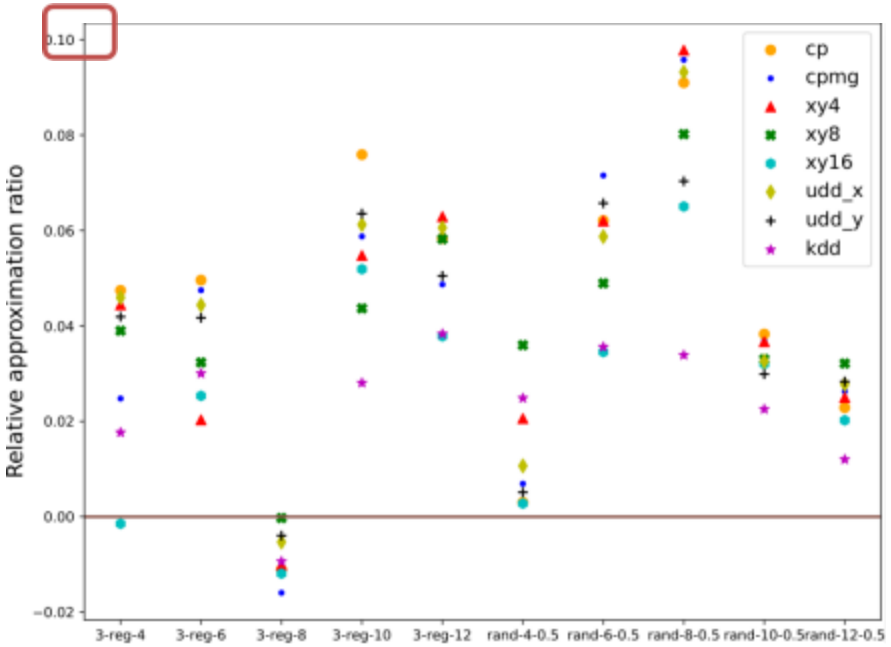
Evaluation algorithm - QAOA

- Quantum Approximate Optimization Algorithm
  - Solving combinatorial optimization problem

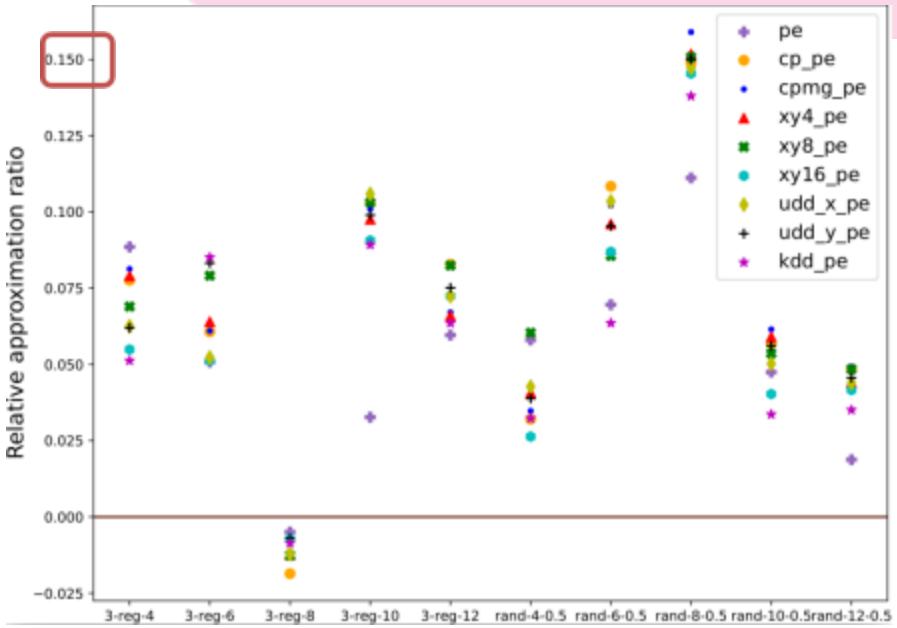
- QAOA circuits



Experimental results



(a) Only DD is applied.



(b) DD and PE technique are applied.

Fig. Relative AR for QAOA on IBM Q Toronto.

Experimental results

Methods	IBM Q Hardware													
	Jakarta				Guadalupe			Toronto			Montreal			
	Max		Min	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg	
Only DD	8.6 (UDD_X)		2.4 (KDD)	6.8	7.4 (CP)	2.5 (KDD)	5.9	4.4 (CP)	2.3 (KDD)	3.7	4.7 (CP)	1.6 (KDD)	3.7	
Only PE	8.9				5.5			5.3			4.8			
DD + PE	10.4 (UDD_X)		6 (KDD)	8.6	9.7 (CP)	6.8 (KDD)	8.8	7.2 (XY8)	5.8 (KDD)	6.7	7.4 (CP)	4.5 (KDD)	6.5	

- Only DD or only PE can already increase the circuit fidelity.
- The combination of DD and PE can further benefit the circuit fidelity.
- CP performs the best and KDD the worst in general.

Guidelines

- The performance of DD is highly **application-dependent.** It is recommended to use DD on BV and QAOA.
- It is recommended to check the **structure** of the circuit to verify if DD is suitable to be inserted.
- The **robust DD** sequence **KDD** does not work well on IBM quantum devices for most of the quantum algorithms tested.
- The pulse-efficient technique is favorable for QAOA circuits and combine it with DD can advance the circuit fidelity even better.

Paper
arxiv:2204.01471

Thank you for your attention!

siyuan.niu@ucf.edu