RICE UNIVERSITY | Northeastern

Park-n-Ride

Hardware-Aware Compilation and Execution of Bivariate Bicycle Codes on Neutral-Atom Systems



Jason Ludmir
Rice University



Aditya Ranjan
Northeastern University



Nicholas S. DiBrita
Rice University



Jason Han
Rice University



Tirthak Patel
Rice University

Roadmap

01	**Motivation**	Problem and contributions
02	**Background**	Neutral-atom hardware and bivariate bicycle codes
03	**Design**	Placement, shifts and joint-measurement scheduling
04	**Evaluation**	Runtime, communication, parallelism and cost
05	**Conclusion**	Summary

MOTIVATION · PROBLEM

BB Codes on Neutral-Atom Systems

Bivariate Bicycle (BB) Codes

Bravyi et al., Nature 2024

- Constant check weight and degree
- Constant-depth syndrome extraction
- Gross code [[144, 12, 12]]
- Long-range, toroidal checks

Neutral-Atom Arrays

Bluvstein et al., Nature 2024

- Large arrays, high-fidelity gates
- Reconfigurable via atom transport
- Zoned, with mid-circuit readout
- AOD and blockade limit motion

Existing neutral-atom compilers do not support BB-native operations; no hardware-aware execution model exists for BB codes on neutral atoms.

Park-n-Ride: BB-Native Execution on Atoms

Spectral Placement

Co-locates frequently interacting modules

Parallel Shift Automorphisms

Cyclic rolls without AOD crossings

Bridge-Based Interaction

Conflict-free bridge scheduling

45%
fewer bridge rounds

34%
less bridge travel

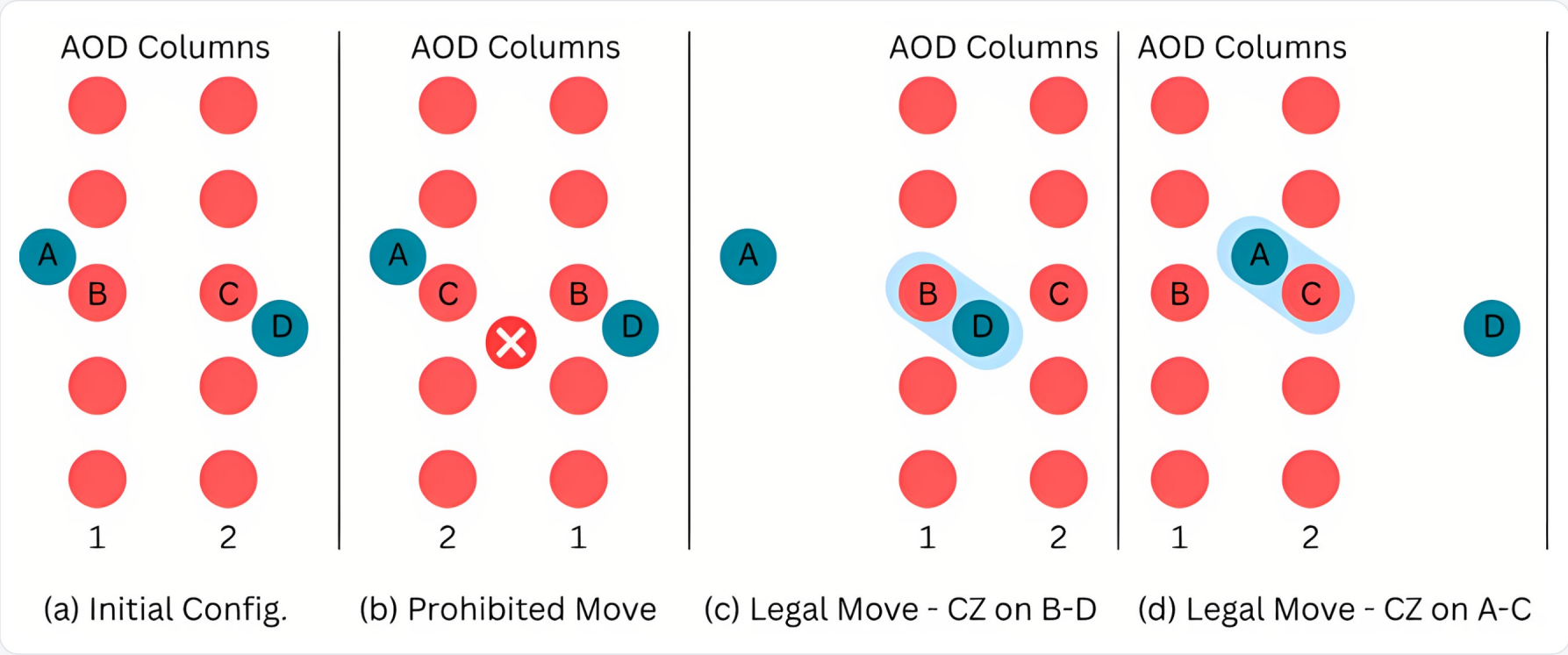
17.9%
lower runtime

40.7%
fewer atom-loss events

Roadmap

01	**Motivation**	Problem and contributions
02	**Background**	Neutral-atom hardware and bivariate bicycle codes
03	**Design**	Placement, shifts and joint-measurement scheduling
04	**Evaluation**	Runtime, communication, parallelism and cost
05	**Conclusion**	Summary

Neutral-Atom Hardware Constraints



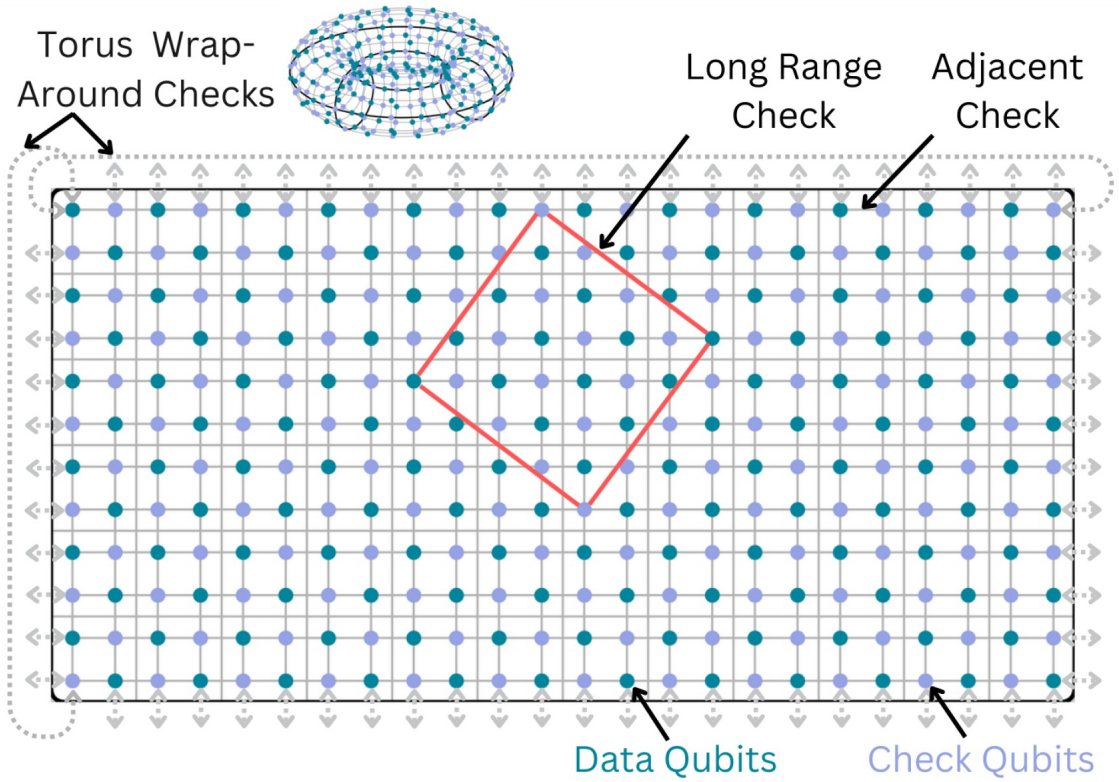
SLM traps: static 2D storage

Rydberg blockade: fixed gate radius

AOD columns: move together, never cross

Reordering: translations or trap changes

Bivariate Bicycle Codes



- Toroidal grid of identical unit cells
- Each cell: data qubits L and R, one X check, one Z check
- Weight-6 checks; each data qubit is in 3 X and 3 Z checks
- Local edges plus two long-range wrap-around edges

Code	[[n, k, d]]	Grid
Gross	[[144, 12, 12]]	12 × 6
Two-gross	[[288, 12, 18]]	12 × 12

BB Execution Interface

Instruction	Description	Implementation
Idle	Alternating X/Z syndrome rounds	Stabilizer cycles, idling subzone
Shift automorphism	Permutation of the toroidal layout	Two AOD rolls and a resync
In-module measurement	Local logical operation	Shift/measure subzone
Inter-module measurement	Joint operation on two modules	Bridge qubits, interaction subzone
T-injection	Magic-state injection	Per-column T-state factory

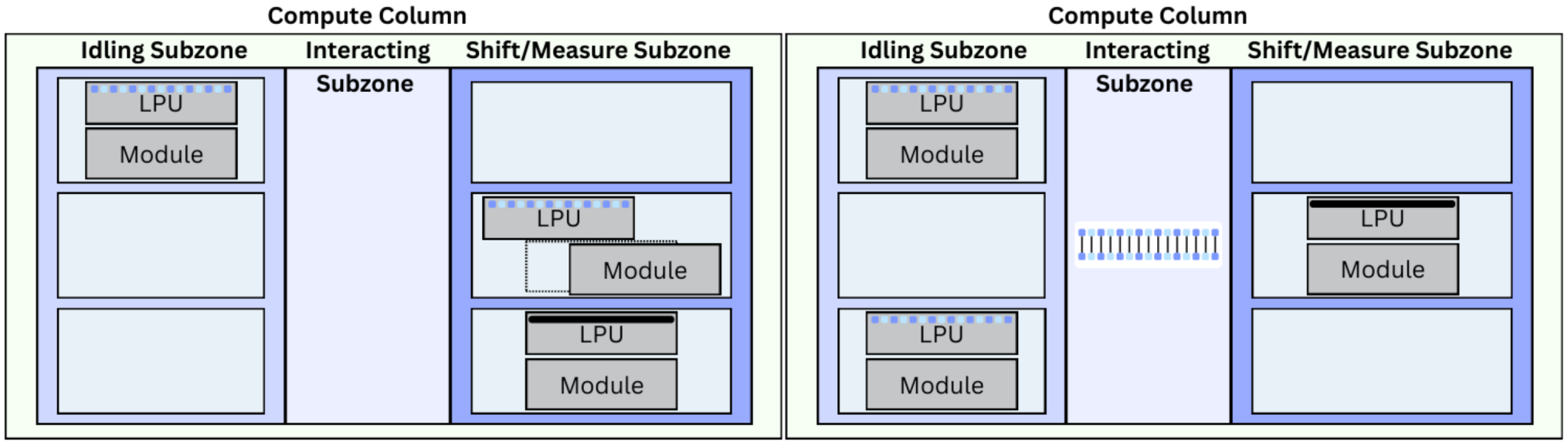
Roadmap

01	**Motivation**	Problem and contributions
02	**Background**	Neutral-atom hardware and bivariate bicycle codes
03	**Design**	Placement, shifts and joint-measurement scheduling
04	**Evaluation**	Runtime, communication, parallelism and cost
05	**Conclusion**	Summary

Design Challenges

Constraint	Approach
Wrap-around interactions must be realized by atom transport	**Spectral placement** co-locates interacting modules
AOD transport must be non-crossing and order-preserving	**Directional rolls** with an order-preserving resync
Joint measurements need collision-free bridge movement	**Interval-based bridge scheduling** with unit-distance moves

Compute Zone Organization



Compute Columns

Fixed stack of modules; motion stays in the column

Sublayer Moves

Only participating modules leave the idling subzone

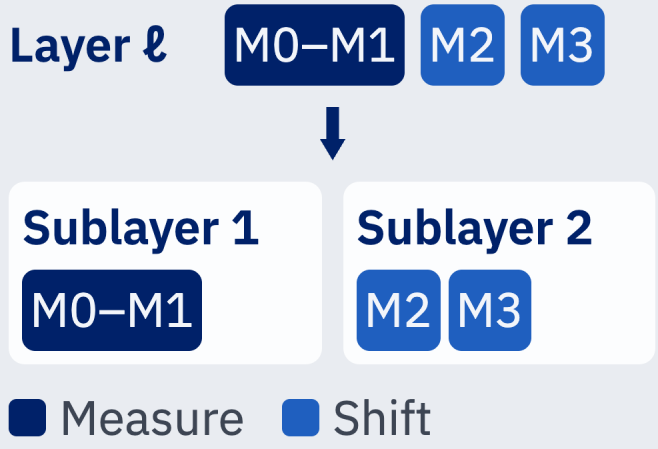
LPU per Module

Bridge qubits on top, stabilizer ancillas below

Compilation Pipeline

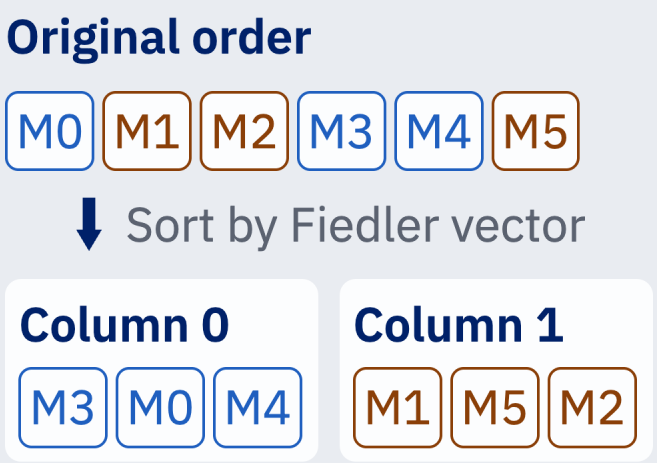
1. Layerization

DAG into layers and sublayers



2. Placement

Fiedler order packed into columns



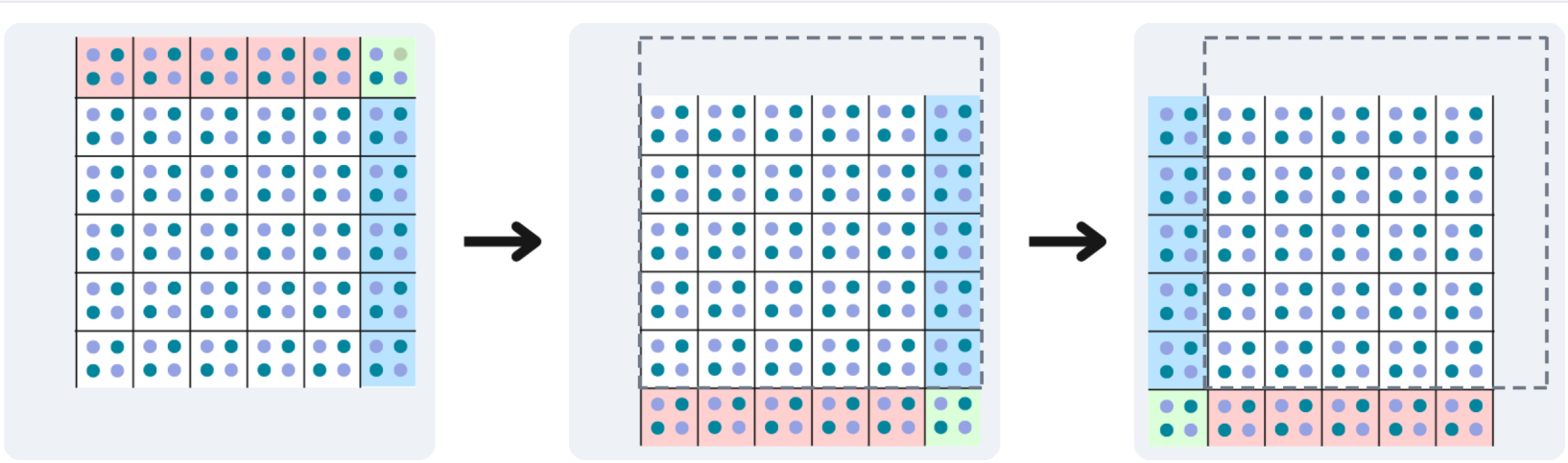
3. Scheduling

Per-sublayer subzone assignments



Shift Automorphism Decomposition

A shift automorphism translates every qubit of a module by (Δi, Δj) on its torus. The code maps onto itself, so the shift is a logical operation that needs only atom transport.



(a) Start **(b)** $\text{Roll}_i(+1)$ **(c)** $\text{Roll}_j(+1)$

$$T(\Delta i, \Delta j) = \text{Roll}_i(\Delta i) \circ \text{Roll}_j(\Delta j)$$

Each roll moves one wrap-around strip and can run either way: $\text{Roll}(\Delta) = \text{Roll}(\Delta - N)$.

Parallel Shifts and Resynchronization

Any mix of shifts in a compute column runs in parallel, with no AOD crossings.

1. Shared Directions

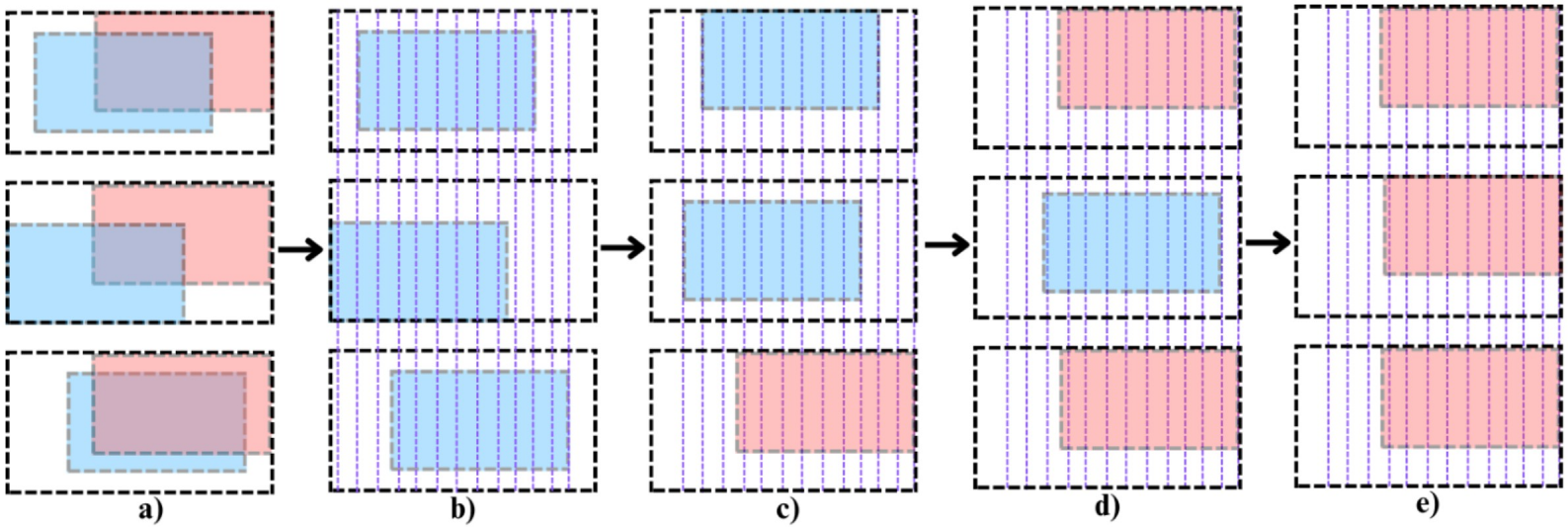
All modules in a column roll the same way

2. Residual Offsets

Unequal shift sizes leave modules off their sites (a)

3. Monotone Resync

Sweep by sorted residuals; drop when aligned (b–e)



Shifted module

Original site

Shared AOD traps

$\Delta_1 = r_{(1)}$

$\Delta_k = r_{(k)} - r_{(k-1)}$

One move per distinct offset

Spectral Module Placement

1. Interaction graph

W_{ij} = joint measurements of i and j

2. Graph Laplacian

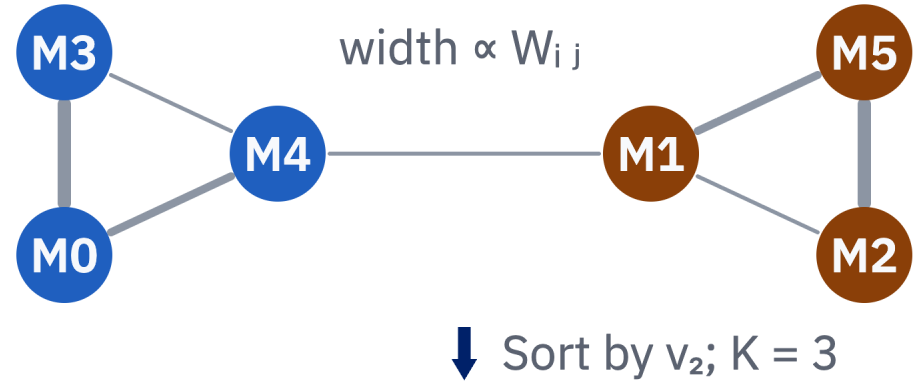
$L = D - W$, $D_{ii} = \Sigma_j W_{ij}$

3. Fiedler vector (unit $v \perp 1$)

$v_2 = \text{argmin}\ \Sigma_{ij} W_{ij} (v_i - v_j)^2$

4. Column packing

Sort by v_2; cut into blocks of capacity K



Baselines. Arbitrary: first available column. Greedy: high-degree hubs in the central columns, then each module at minimum weighted 1D distance.

Bridge-Based Joint-Measurement Scheduling

A joint measurement entangles bridge qubits from two modules in the interaction lane. Bridges ride AOD columns, so two bridges can never pass each other.

1. Corridors

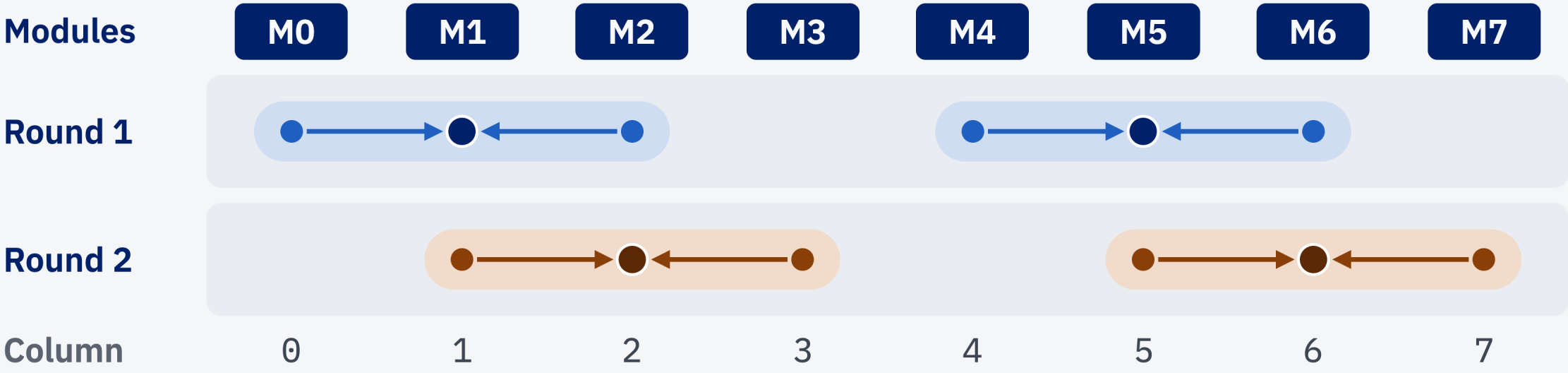
A pair moves within the span between its home columns

2. Rounds

Overlapping corridors go in different rounds

3. Lockstep Moves

Bridges step ±1 column to the midpoint, measure, return

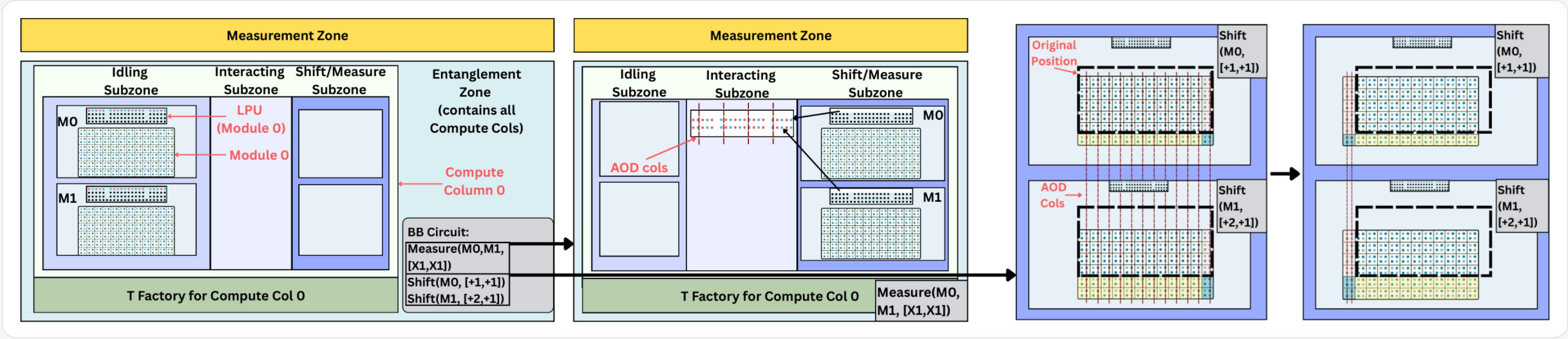


DESIGN · EXAMPLE

End-to-End Example

BB circuit:

```
Measure(M0, M1, [X1, X1])
Shift(M0, [+1, +1])
Shift(M1, [+2, +1])
```



(a) M0 and M1 are placed in the idling subzone

(b) Bridges meet to form Bell pairs

(c.1) Parallel shifts leave an offset

(c.2) Resync restores alignment

Roadmap

01	**Motivation**	Problem and contributions
02	**Background**	Neutral-atom hardware and bivariate bicycle codes
03	**Design**	Placement, shifts and joint-measurement scheduling
04	**Evaluation**	Runtime, communication, parallelism and cost
05	**Conclusion**	Summary

Experimental Setup

Workloads

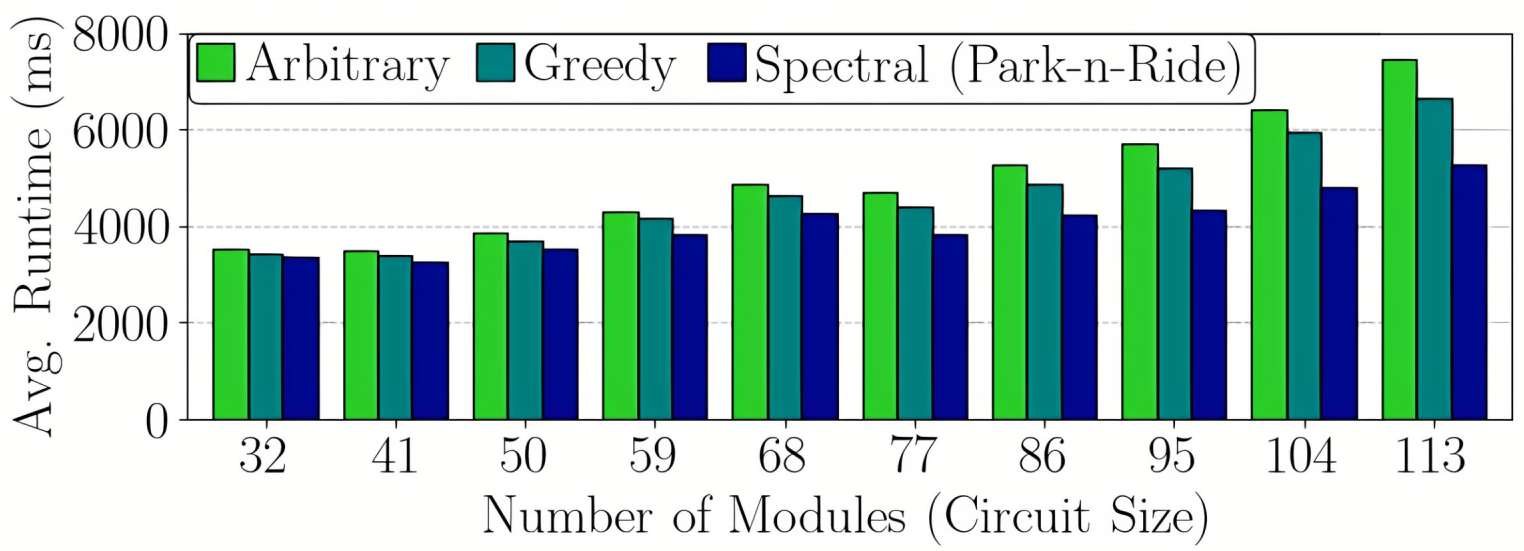
- 30 BB circuits, 32–113 modules
- 9,216–32,544 physical qubits
- Gross code; two-gross for scaling
- 10 ops/module, 25% joint measurements
- Column capacity of 2–10 modules
- Baselines:
  - **Arbitrary:** First available column
  - **Greedy:** Hubs first, then neighbors
- **Runtime:**
  - Shuttling
  - Gates
  - Trap Switches
  - Subzone Transitions

Hardware Model

Parameter	Value
AOD speed	55 µm/µs
Rydberg distance	2 µm
Lattice spacing	10 µm
2Q / 1Q gate	0.8 / 2.0 µs
Trap switch	100 µs
Module size	120 × 60 µm

EVALUATION · RUNTIME

Runtime vs. Circuit Size



17.9%
lower than arbitrary

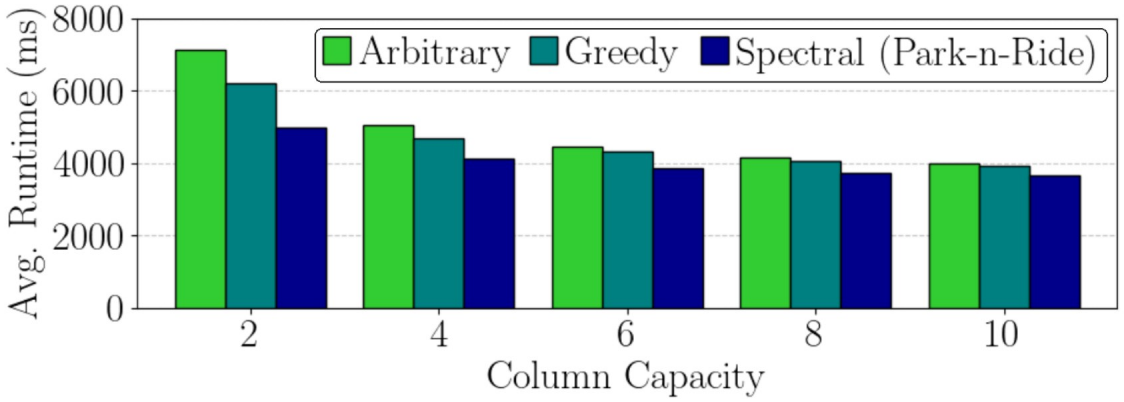
12.2%
lower than greedy

>2 s
faster at 113 modules

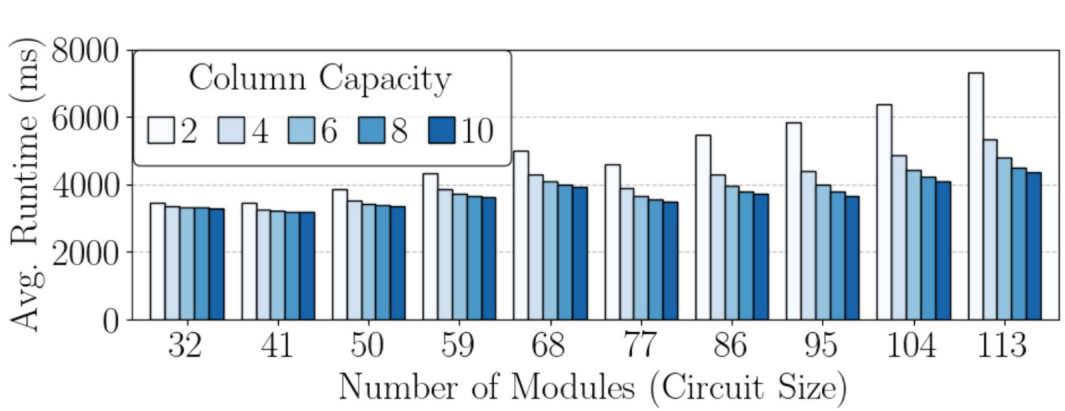
- Spectral is fastest at every size, and the gap grows with circuit size.
- At 113 modules, greedy recovers only about half of spectral's gain over arbitrary placement.

EVALUATION · COLUMN CAPACITY

Impact of Compute-Column Capacity



All placers. Capacity 2 → 4 cuts runtime by about 30%; gains level off beyond 6.



Spectral, by size. At 113 modules, capacity 2 → 10 cuts runtime by almost 40% (7.3 s → 4.3 s).

Placement can stand in for hardware: spectral at capacity 6 outperforms arbitrary at capacity 10.

Communication Cost

Size	Placer	Travel	Rounds	Steps
32	Greedy	239.93	2.08	146.0
32	Spectral	197.32	1.57	100.0
68	Greedy	355.82	5.78	528.0
68	Spectral	277.69	3.80	361.3
113	Greedy	550.59	14.62	1491.3
113	Spectral	361.91	8.02	810.0

Spectral reduction vs. greedy

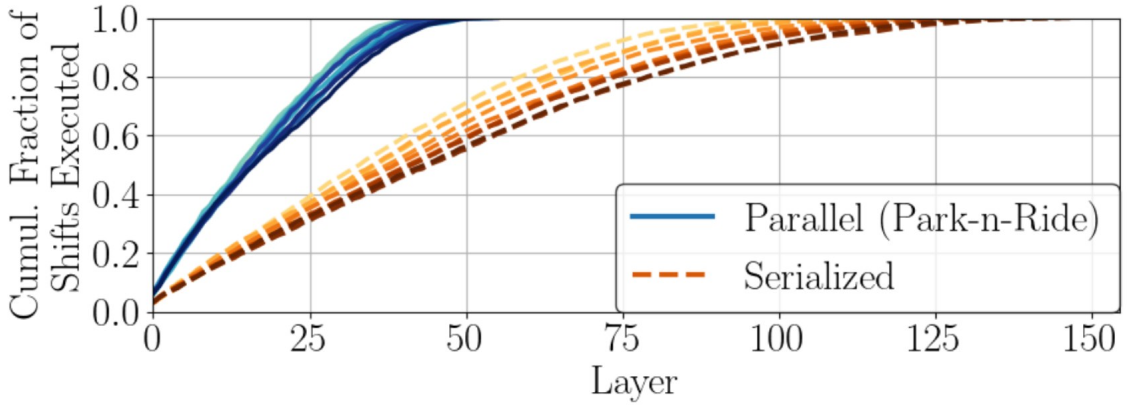
Size	Travel	Rounds	Steps
32	18%	25%	32%
68	22%	34%	32%
113	**34%**	**45%**	**46%**

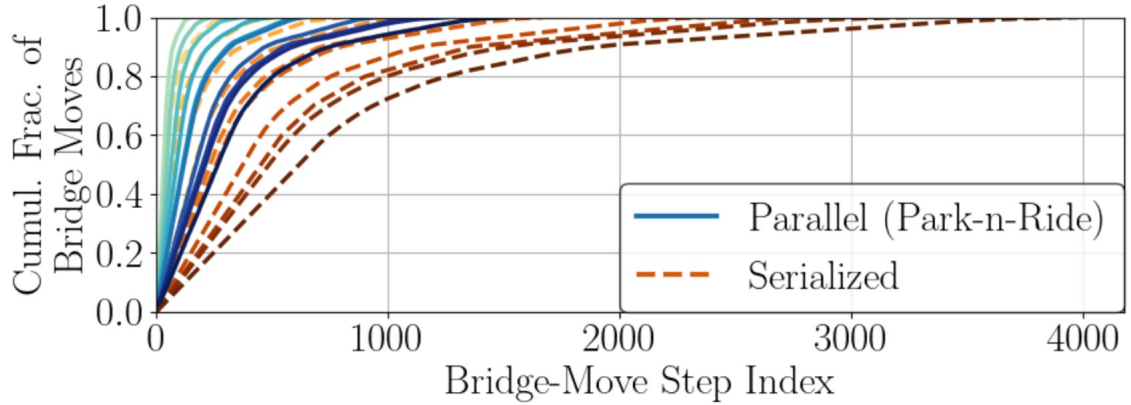
Savings grow with circuit size.

EVALUATION · PARALLELISM

Shift and Bridge Parallelism

Darker curves mean larger circuit sizes.





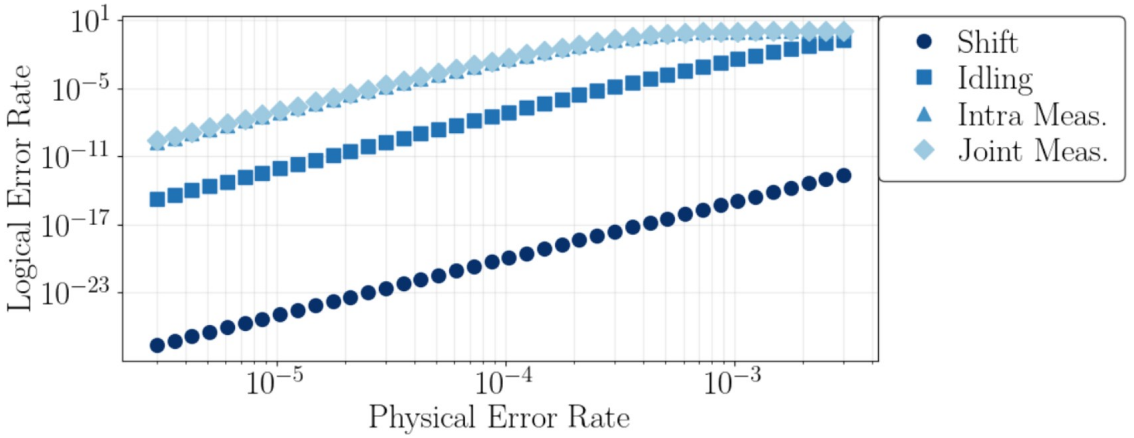
Shift Automorphisms

All shifts finish within ~50 layers, vs. up to ~150 when serialized.

Bridge Movement

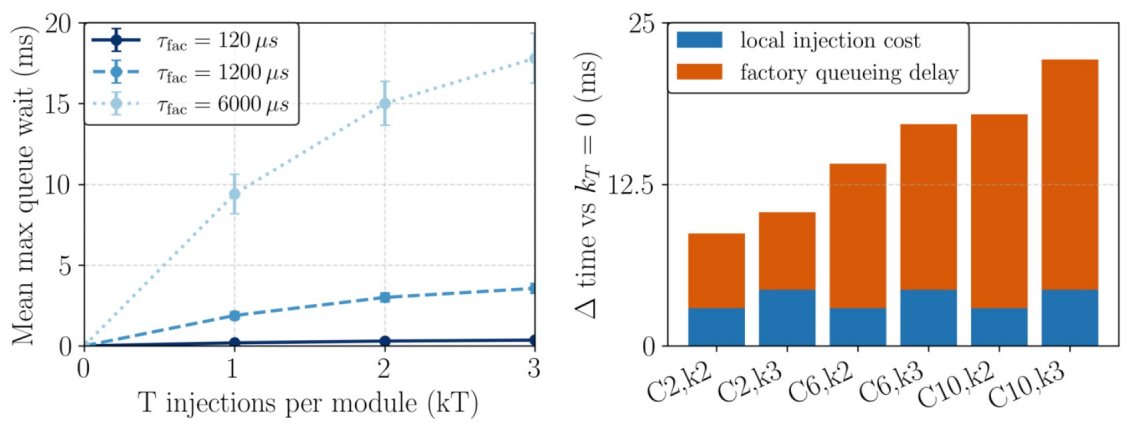
Bridges conflict under AOD constraints in large circuits.

Logical Error and T-Factory Pressure



Logical Error per Operation

Shifts are orders of magnitude below idling and measurement.



T-Factory Pressure

Negligible at τ_fac = 120 µs; a few ms at 1200 µs; O(10) ms at 6000 µs.

Roadmap

01	**Motivation**	Problem and contributions
02	**Background**	Neutral-atom hardware and bivariate bicycle codes
03	**Design**	Placement, shifts and joint-measurement scheduling
04	**Evaluation**	Runtime, communication, parallelism and cost
05	**Conclusion**	Summary

CONCLUSION · SUMMARY

Conclusion

Execution Model

- Zoned compute columns host BB modules
- Idle, bridge and shift/measure subzones
- Crossing-free AOD motion throughout
- Per-column T factories, local injection

Shift Automorphisms

- Two cyclic rolls, in either direction
- Any mix of shifts runs in parallel
- Monotone resync restores alignment
- Error far below idling and measurement

Placement & Scheduling

- Spectral seriation co-locates partners
- Interval coloring gives conflict-free bridges
- Placement can stand in for capacity
- Gains grow with circuit size

45% fewer bridge rounds

34% less bridge travel

17.9% lower runtime

40.7% fewer atom-loss events

RICE UNIVERSITY

Thank You!
Questions?

Jason Ludmir, jzl2@rice.edu

Code and data

This work was supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Science Center, and by the Rice Quantum Initiative (Smalley-Curl Institute and Ken Kennedy Institute).

BACKUP · SENSITIVITY

Sensitivity Analysis

Readout Time

Runtime 17.9% lower than arbitrary and 12.2% lower than greedy; 15.6% and 10.6% at 1500 µs readout.

Atom Loss

Expected loss events: 21.9 arbitrary, 18.0 greedy, 13.0 spectral; same order in all 16 settings.

Two-Gross Code

Runtime rises by less than 0.1%; spectral remains 14.4% and 10.1% lower than arbitrary and greedy.

Compile Time

113 modules, capacity 10: 28.8 s spectral, 36.5 s arbitrary, 37.6 s greedy.