Hamiltonian Decoded Quantum Interferometry for General Pauli Hamiltonians

Kaifeng Bu

Math Picture Language Seminar @ Harvard

04/29/2026

Based on joint work with Weichen Gu, Dax Koh, Xiang Li
[arXiv:2508.10725, 2601.18773]

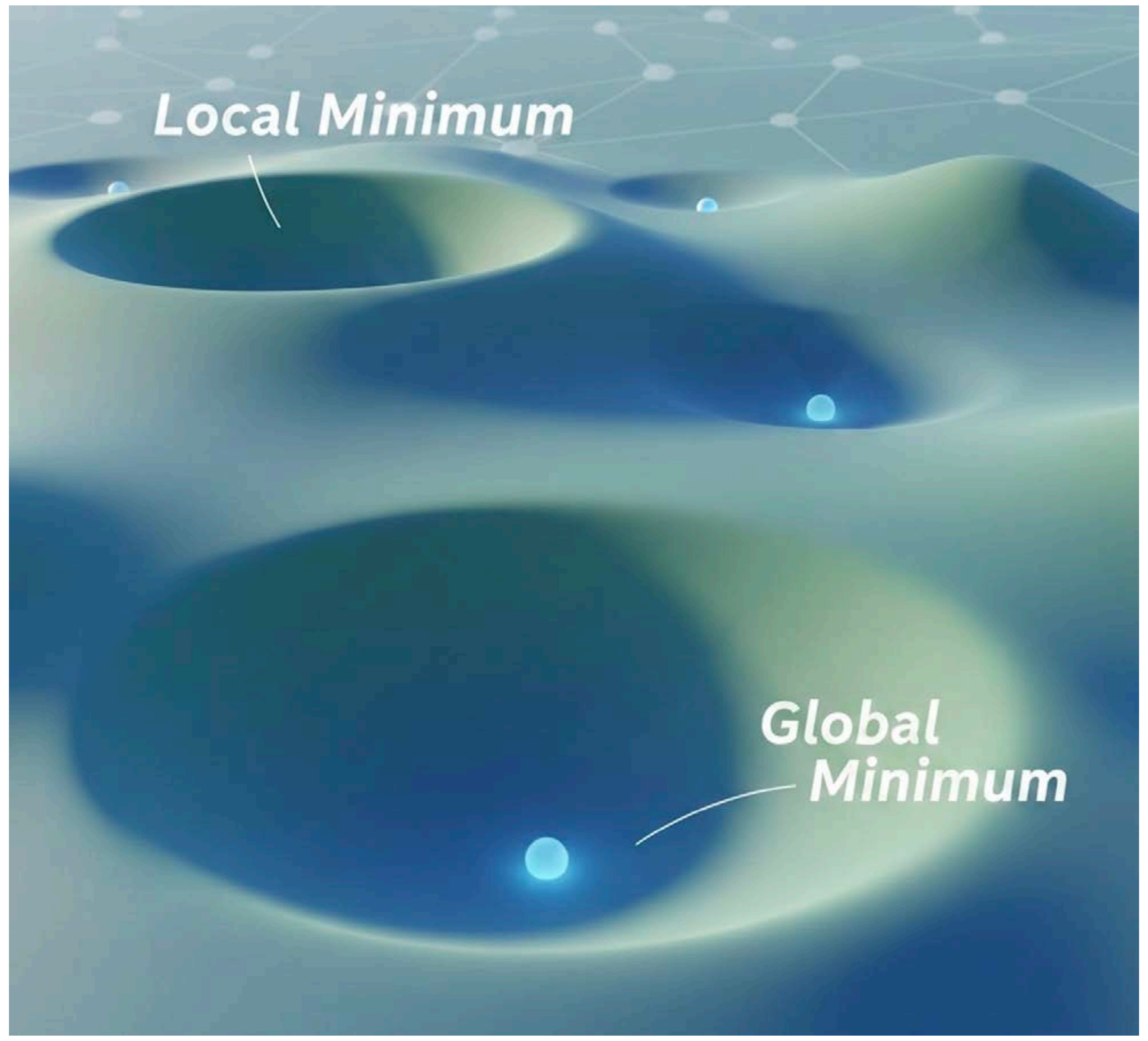
Background

Optimization problem:

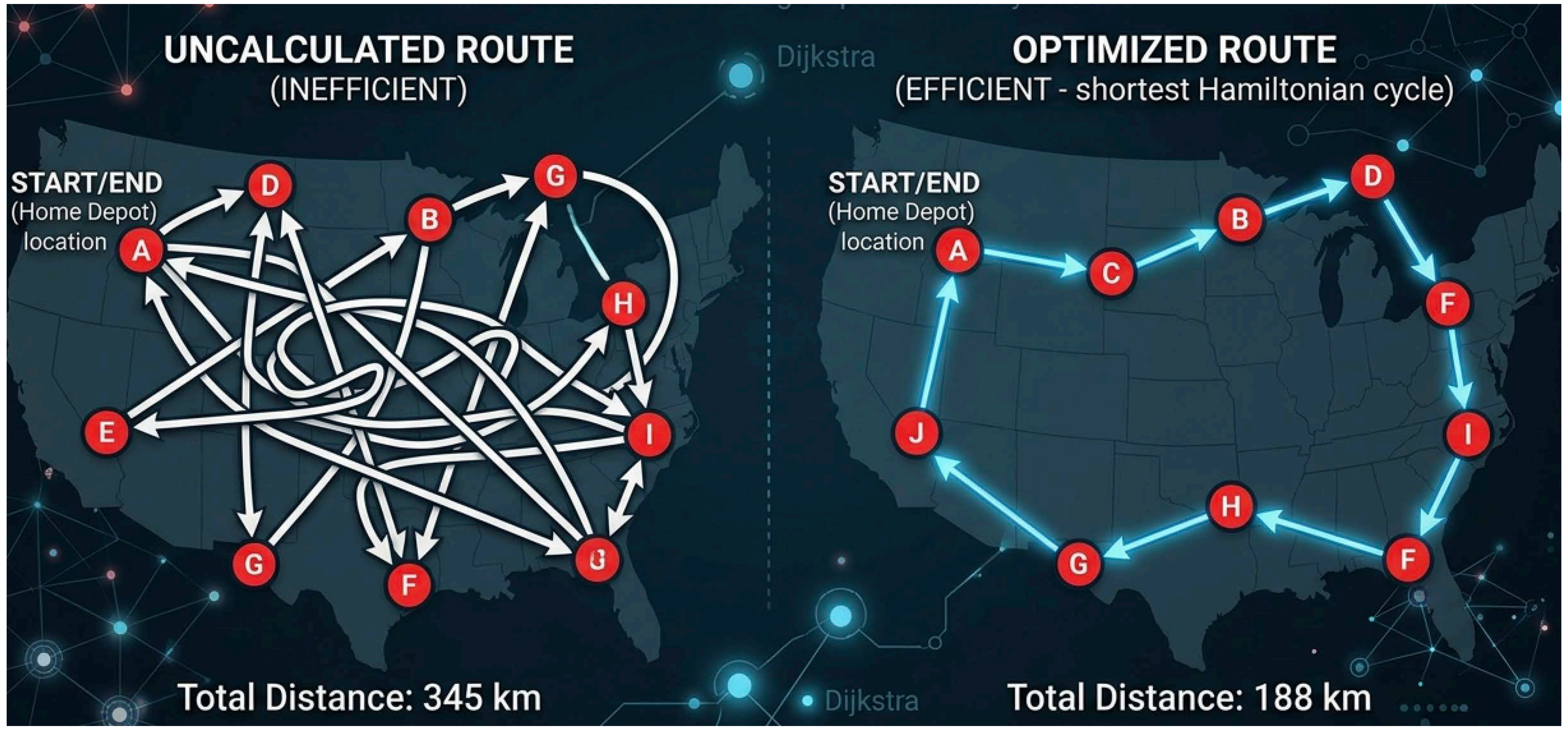
$$\max_x / \min_x f(x)$$

(Hard to solve in general)

Examples:



Finding The Ground Energy



The Traveling Salesman Problem

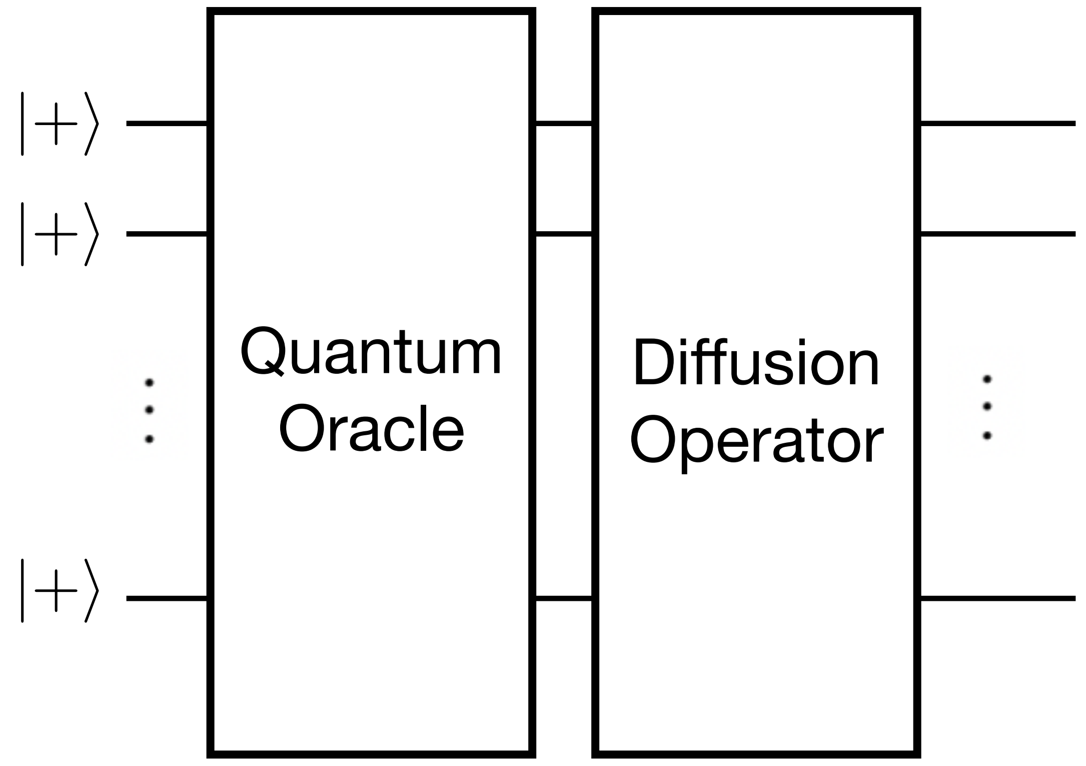
Background

Optimization problem:

$$\max_x \,/\, \min_x f(x)$$

(Hard to solve in general)

Known quantum algorithms for optimization: Grover's algorithm, quantum adiabatic algorithms, VQE, QAOA …



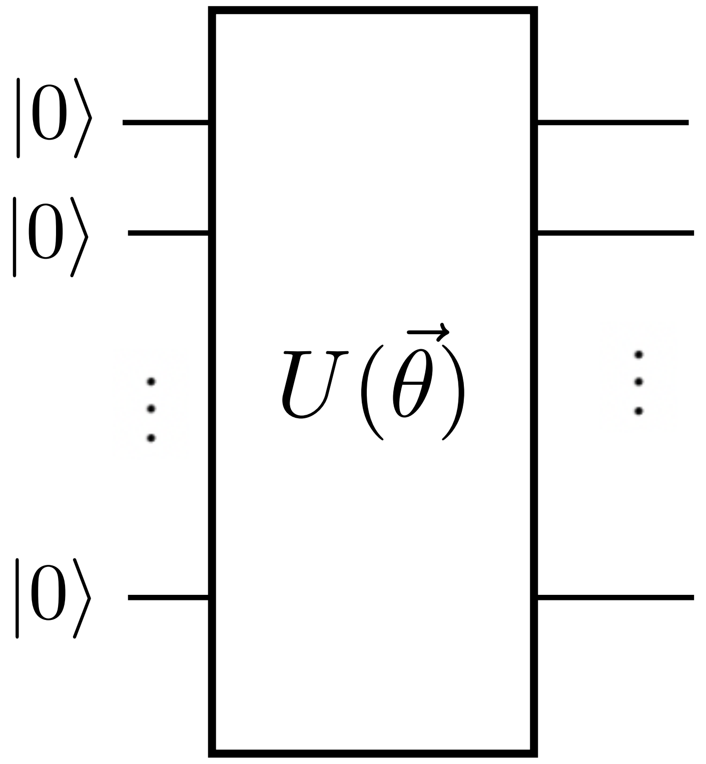
Grover's algorithm

$$H_t = (1-t)H_0 + tH_f$$

H_0 → H_f

Initial Hamiltonian — Target Hamiltonian

Quantum adiabatic algorithm



Measure $\langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle$

and

Update $\vec{\theta}$

VQE

[Grover, STOC 1996] $f(x) = 1, x = w$
$f(x) = 0, x \neq w$

[Farhi, Goldstone, Gutmann,Sipser, *arXix: quant-ph/0001106*]
[Aharonov, Dam, Kempe, Landau, Lloyd, Regev, FOCS 2004]

[Peruzzo, McClean, Shadbolt, Yung, Zhou, Love, Aspuru-Guzik, O'brien, Nat. Com. 2014]

Overview of DQI

$$\max_x / \min_x f(x)$$

Quantum Algorithm →

Classical Decoding

(Basic Idea)

Article | Open access | Published: 22 October 2025

Optimization by decoded quantum interferometry

Stephen P. Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, Tanuj Khattar & Ryan Babbush

Nature **646**, 831–836 (2025) | Cite this article

Overview of Hamiltonian DQI

Focus of today's talk

Hamiltonian

$$H = \sum_{i=1}^{m} c_i \mathrm{Pauli}_i, c_i \in \mathbb{R}$$

Polynomial P

→

Quantum process

+

Classical decoding

→

$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

(Basic Idea)

If the degree l is large enough, then output state $\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$ is close to the Gibbs state $\frac{e^{-\beta H}}{\mathrm{Tr}[e^{-\beta H}]}$

[Schmidhuber, Lu, Shutty, Jordan, Poremba, Quek, *arXiv:2510.07913*] $H = \sum_i (-1)^{v_i} \mathrm{Pauli}_i$

[Bu, Gu, Li, arXiv:2601.18773] $H = \sum_i c_i \mathrm{Pauli}_i, c_i \in \mathbb{R}$

Part 1: Basic Knowledge of DQI

Introduction of DQI

$$\max_x / \min_x f(x)$$

Quantum Algorithm →

Classical Decoding

(Basic Idea)

Article | Open access | Published: 22 October 2025

Optimization by decoded quantum interferometry

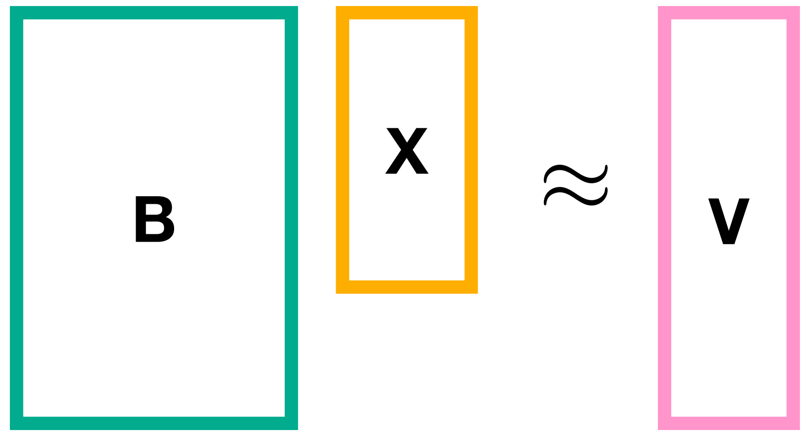
Stephen P. Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, Tanuj Khattar & Ryan Babbush

Nature **646**, 831–836 (2025) | Cite this article

Introduction of DQI

Example: MAX-LINSAT

Input: $B \in \mathbb{F}_2^{m \times n}$ $\quad v \in \mathbb{F}_2^{m}$



Goal: Find an assignment x such that it satisfies as many constraints as possible

$$\max_{x \in \mathbb{F}_2^n} \left| \{ i : b_i x = v_i \} \right|$$

b_i is the i-th row of B

$$\max_{x \in \mathbb{F}_2^n} f(x), f(x) = \sum_{i=1}^{m} (-1)^{b_i x + v_i}$$

Introduction of DQI

Example: MAX-LINSAT

Input: $B \in \mathbb{F}_2^{m \times n}$ $\quad v \in \mathbb{F}_2^m$

Goal: $\max_{x \in \mathbb{F}_2^n} f(x), f(x) = \sum_{i=1}^{m} (-1)^{b_i x + v_i}$ $\quad b_i$ is the i-th row of B

DQI can prepare a "DQI" state

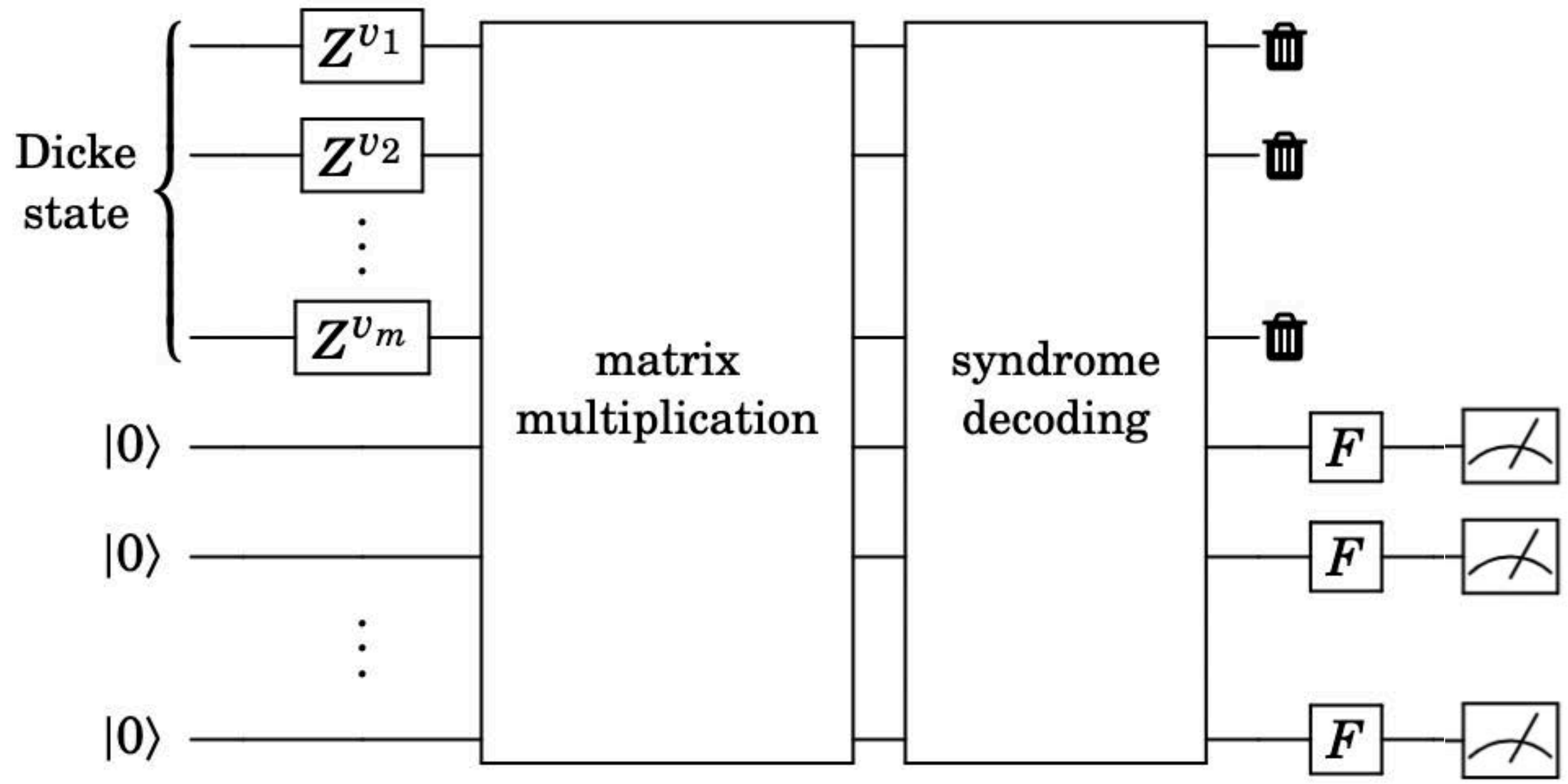
$$|P(f)\rangle \propto \sum_{x \in \mathbb{F}_2} P(f(x))|x\rangle$$

P is any polynomial of degree l

using the decoding on the code

$$C^{\perp} = \{y \in \mathbb{F}_2^m : B^{\top} y = 0\}$$

Introduction of DQI



$$|\text{Dicke}\rangle \otimes |0\rangle \qquad |\text{Dicke}\rangle \propto \sum_{y:|y|=l} |y\rangle$$

Apply Pauli Z →

$$Z^v|\text{Dicke}\rangle \otimes |0\rangle$$

Matrix Multiplication →

$$\sum_{y:|y|=l} (-1)^{v\cdot y}|y\rangle \otimes |B^T y\rangle$$

Decoding →

$$\sum_{y:|y|=l} (-1)^{v\cdot y}|0\rangle \otimes |B^T y\rangle$$

Apply Fourier transform →

$$\sum_x P(f(x))|x\rangle$$

Hence, if we recover y from $B^\top y$ for $|y| \leq l$

That is, decode l error for $C^\perp = \{y \in \mathbb{F}_2^m : B^\top y = 0\}$

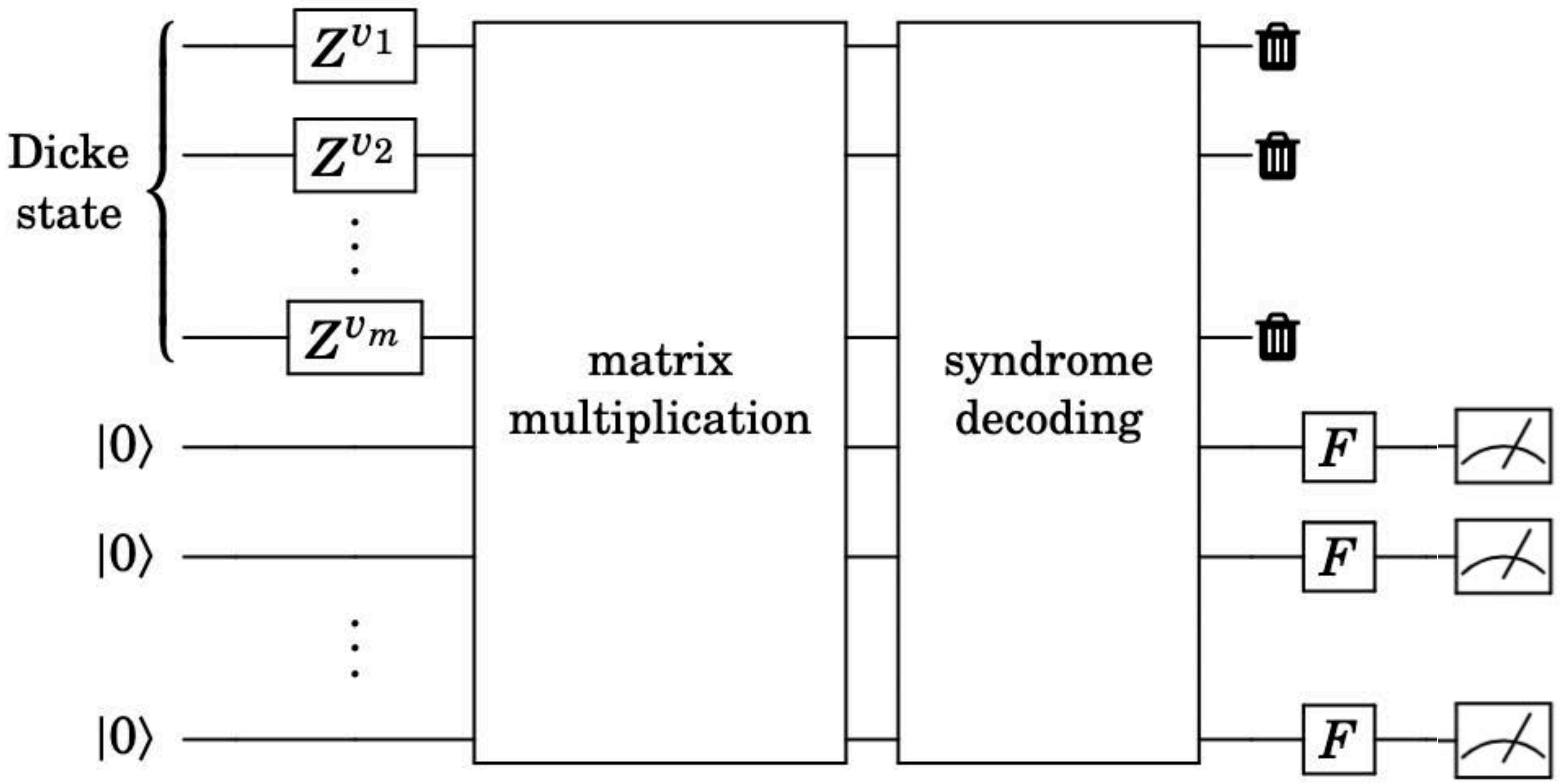
→ $|P(f)\rangle \propto \sum_{x\in\mathbb{F}_2} P(f(x))|x\rangle$

→ Sampling x w.r.t probability $P(f(x))^2$

$$\langle\text{Satisfied constraints}\rangle = \sum_x f(x)P(f(x))^2$$

[Jordan, Shutty, Wootters, Zalcman, Schmidhuber, King, Isakov, Khattar, Babbush, Nature 2025]

Introduction of DQI



Efficient decoding

↓

Efficiency of DQI

Example: Optimal Polynomial Interpolation (OPI) over $\mathbb{F}_p$ **p is a large prime number**

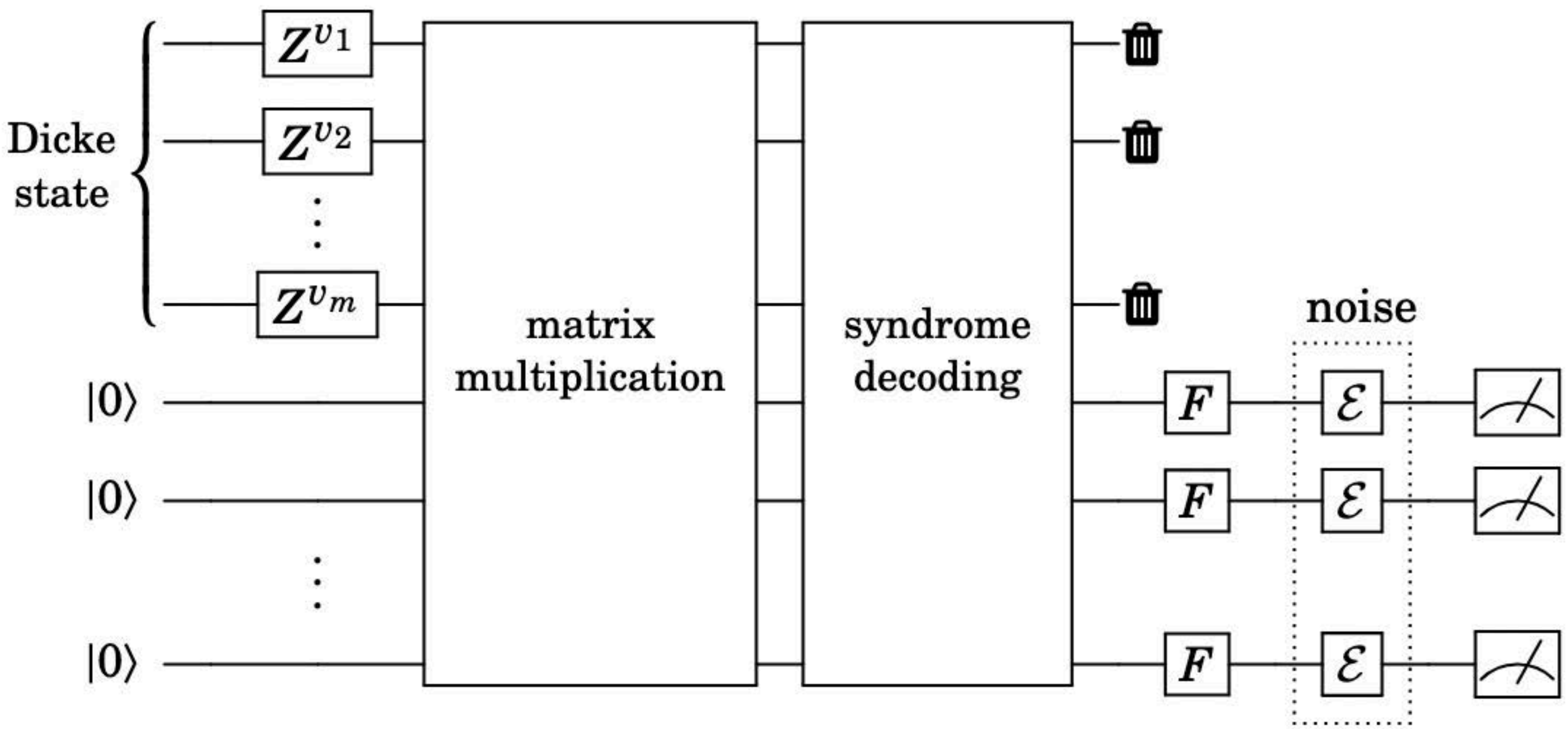
$$B_{ij} = \gamma^{i \cdot j}, i \in \{0, .., p-2\}, j \in \{0, ..., n-1\}, \gamma \in \mathbb{F}_p$$

Reed–Solomon code

There exists efficient decoding algorithm for $|y|$ **less than 1/2 of the code distance （Berlekamp-Massey algorithm）**

[Jordan, Shutty, Wootters, Zalcman, Schmidhuber, King, Isakov, Khattar, Babbush, Nature 2025]

Introduction of Noisy DQI



Noise:1-qubit depolarizing noise with noise rate p

$$\mathcal{E}(X) = (1-p)X + p\frac{\mathrm{Tr}[X]I}{2}$$

$$\langle \text{Satisfied constraints} \rangle_{\text{Noisy}} = [\mathbb{E}_i (1-p)^{|b_i|}] \langle \text{Satisfied constraints} \rangle_{\text{Noisless}}$$

(Exponential decay with respect to the weight of the matrix B)

[Bu, Gu, Koh, Li, Quantum Science and Technology, 2026]

From DQI to Hamiltonian DQI

DQI in Hamiltonian form

$$f(x) = \sum_{i=1}^{m} (-1)^{b_i x + v_i} \quad \rightarrow \quad \text{Hamiltonian} \quad H_f = \sum_i (-1)^{v_i} Z^{b_i} \qquad H_f|x\rangle = f(x)|x\rangle$$

DQI generate the sampling according to $P(f(x))^2$

$\downarrow$

Hamiltonian DQI (HDQI) [Schmidhuber, Lu, Shutty, Jordan, Poremba, Quek, *arXiv:2510.07913*]

Consider Hamiltonian $H = \sum_i (-1)^{v_i} \mathrm{Pauli}_i \qquad v_i \in \mathbb{F}_2$

HDQI aim to generate some density matrix

$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

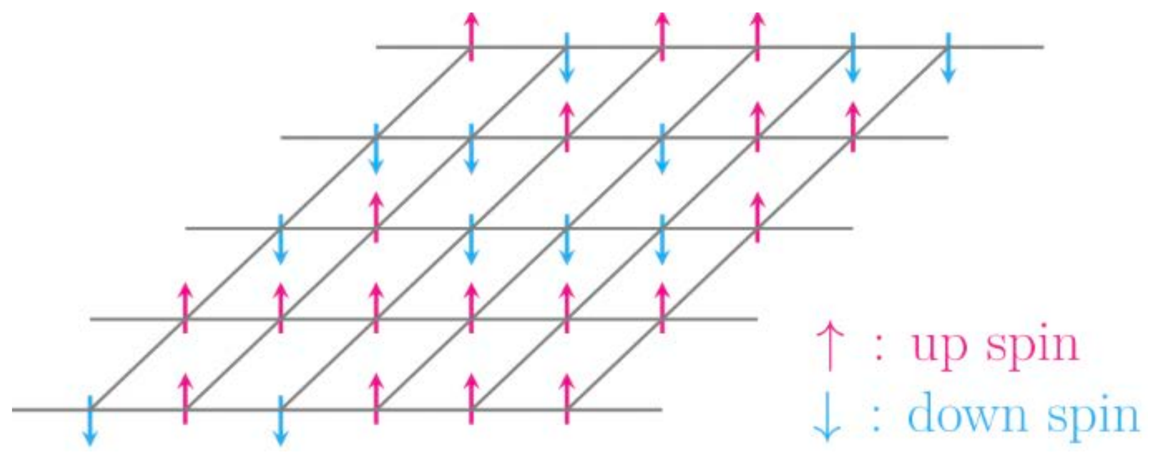
If the polynomial P is a good approximation of $e^{-\beta x/2}$ $\rightarrow$ $\rho_P(H)$ is a good approximation of the Gibbs state

A nature question arise:
How about the physics/chemistry Hamiltonians?

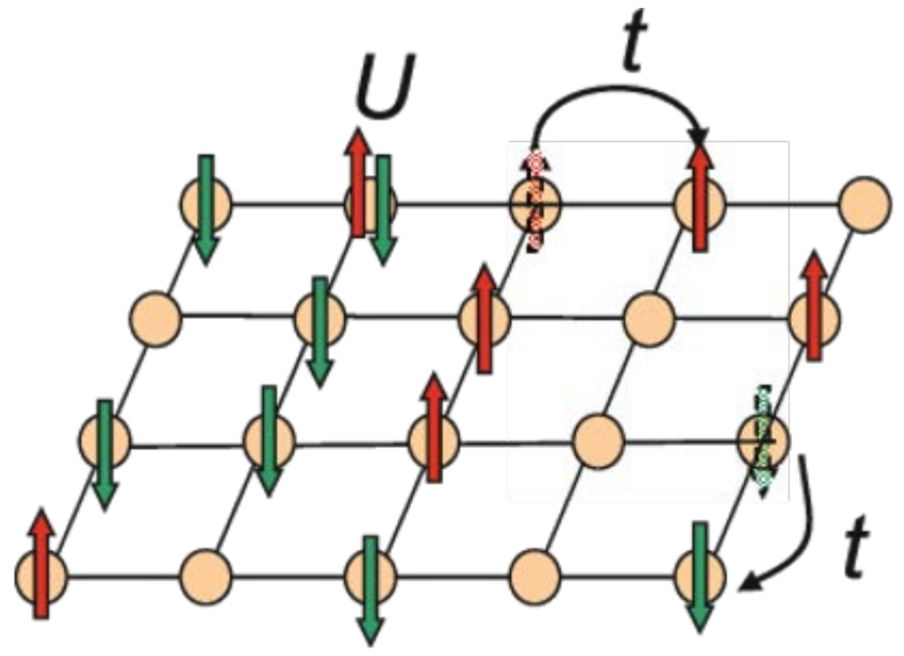
Part 2: HDQI for General Hamiltonians

Hamiltonian DQI for general Pauli Hamiltonians

Given an n-qubit Hamiltonian $H = \sum_{i=1}^{m} c_i P_i, c_i \in \mathbb{R}, |c_i| \leq 1$



Ising model



Hubbard model

Goal: prepare the state $\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$ where P is any degree-l polynomial

[Bu, Gu, Li, arXiv:2601.18773]

Basic Knowledge of Pauli operators

Pauli operators and its symplectic representation

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

1-qubit Pauli can be written as $W(\alpha, \beta) = i^{-\alpha\beta} Z^{\alpha} X^{\beta}$ $\qquad \alpha, \beta \in \mathbb{F}_2$

$$Y = W(1,1) = i^{-1} ZX = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

$$W(\alpha, \beta) W(\alpha', \beta') = (-1)^{\langle (\alpha,\beta),(\alpha',\beta') \rangle_s} W(\alpha', \beta') W(\alpha, \beta),$$

$$\langle (\alpha, \beta), (\alpha', \beta') \rangle_s = \alpha\beta' - \beta\alpha'$$

Basic Knowledge of Pauli operators

n-qubit Pauli can be written as $W(\vec{\alpha}, \vec{\beta}) = W(\alpha_1, \beta_1) \otimes W(\alpha_2, \beta_2) \otimes \ldots \otimes W(\alpha_n, \beta_n),$

$$W(\vec{\alpha}, \vec{\beta})W(\vec{\alpha}', \vec{\beta}') = (-1)^{\langle(\vec{\alpha}, \vec{\beta}), (\vec{\alpha}', \vec{\beta}')\rangle_s} W(\vec{\alpha}', \vec{\beta}')W(\vec{\alpha}, \vec{\beta}),$$

So the vector $(\vec{\alpha}, \vec{\beta}) \in \mathbb{F}_2^{2n}$ is called its symplectic representation for a Pauli operator P, denoted by $\mathrm{symp}(P)$

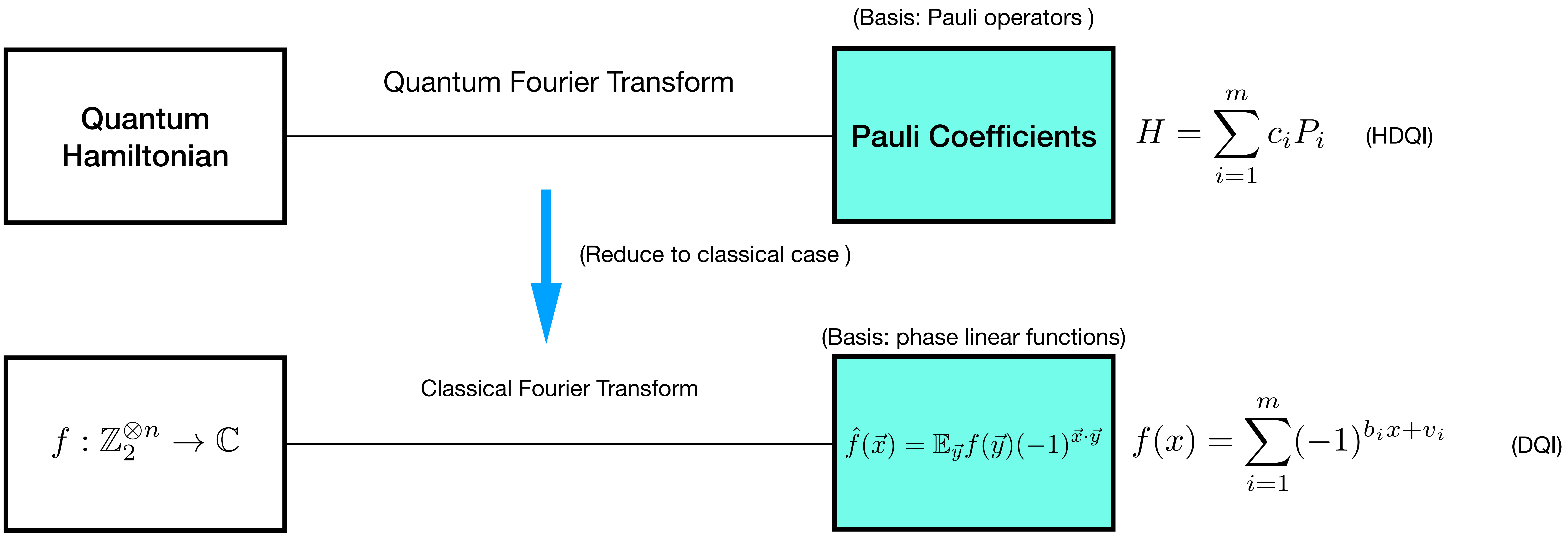
Moreover, $W(\vec{\alpha}, \vec{\beta})W(\vec{\alpha}', \vec{\beta}') = i^{\langle(\vec{\alpha}, \vec{\beta}), (\vec{\alpha}', \vec{\beta}')\rangle_s} W(\vec{\alpha} + \vec{\alpha}', \vec{\beta} + \vec{\beta}')$

$\Rightarrow$ Group homomorphism: $\mathrm{symp}(PQ) = \mathrm{symp}(P) \oplus \mathrm{symp}(Q)$

$|\mathrm{symp}(P)\rangle \xleftrightarrow{\text{CNOT, H}} P \otimes I|\mathrm{Bell}_n\rangle$

Pauli operators (via Fourier analysis)

(Pauli operators:ONB for $B(\mathcal{H}^{\otimes n})$ with respect to $\langle A, B \rangle = \frac{1}{2^n}\mathrm{Tr}[A^\dagger B]$)



Remark: Pauli operators are taken as the “quantum polynomials” of degree 1 in quantum higher-order Fourier analysis [Bu, Gu, Jaffe, PNAS 2025]

Hamiltonian DQI for general Pauli Hamiltonians

Using symplectic representation

$$H = \sum_{i=1}^{m} c_i P_i \quad \longrightarrow \quad B^\top = \begin{bmatrix} | & | & & | \\ \mathrm{symp}(P_1) & \mathrm{symp}(P_2) & \cdots & \mathrm{symp}(P_m) \\ | & | & & | \end{bmatrix} \in \mathbb{F}_2^{2n\times m}$$

(Classical Linear Code) $C^\perp = \{y \in \mathbb{F}_2^m : B^\top y = 0\}$

Assume there is an efficient and perfect decoder

$$D_H^{(l)} |y\rangle |B^\top y\rangle = |0\rangle |B^\top y\rangle, \forall |y| \leq l$$

For example, if the columns are linear independent, then using Gaussian elimination to decode

Commuting case

Given: $H = \sum_{i=1}^{m} c_i P_i$ Polynomial P

Goal: $\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$

Step 0: rewrite the polynomial P(H)

For any degree-l polynomial $P(x) = \sum_{j=0}^{l} a_j x^j$ consider $P(H) = \sum_{j=0}^{l} a_j H^j$

It can be rewritten as $P(H) = \sum_{\mathbf{y} \in \mathbb{F}_2^m} w_{\mathbf{y}} P^{\mathbf{y}}$ $P^{\mathbf{y}} = P_1^{y_1} \cdot P_2^{y_2} \cdot \ldots \cdot P_m^{y_m}$

$$w_{\mathbf{y}} = \sum_{j=0}^{l} a_j j! \sum_{\mu \in \mathbb{Z}_+^m : |\mu| = j, \mu \equiv y \mod 2} \frac{c^\mu}{\mu!}$$

$$c^\mu = c_1^{\mu_1} \cdot c_2^{\mu} \cdot \ldots \cdot c_m^{\mu_m}$$

$$\mu! = \mu_1! \cdot \mu_2! \cdot \ldots \cdot \mu_m!$$

Commuting case

Step 1: Prepare a reference state to encode P(H) $|R^l(H)\rangle = \frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle$ Recall that $P(H) = \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} P^{\mathbf{y}}$

Step 2: Apply control-Pauli operators between the reference state and Bell states on 2n qubits

$$\frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle \otimes (P^{\mathbf{y}} \otimes I)|\mathrm{Bell}_n\rangle \longleftrightarrow \frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle \otimes |\mathrm{Symp}(P^{\mathbf{y}})\rangle$$

$$\mathrm{Symp}(P^{\mathbf{y}}) = B^T \mathbf{y}$$

$$B^\top = \begin{bmatrix} | & | & & | \\ \mathrm{symp}(P_1) & \mathrm{symp}(P_2) & \cdots & \mathrm{symp}(P_m) \\ | & | & & | \end{bmatrix}$$

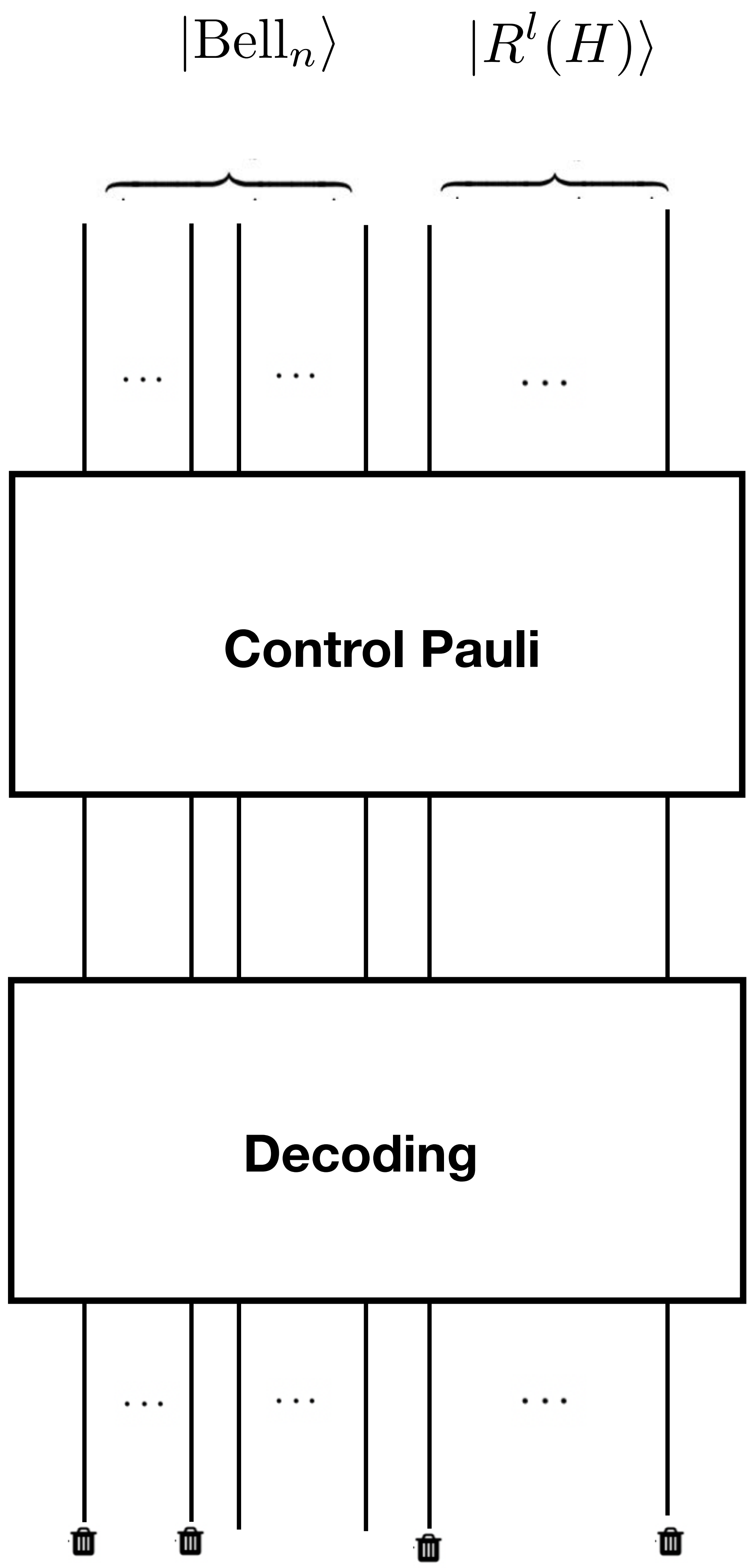
Step 3: Decoding

$$\frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} \cancel{|0\rangle} \otimes |\mathrm{Symp}(P^{\mathbf{y}})\rangle \longleftrightarrow \frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} (P^{\mathbf{y}} \otimes I)|\mathrm{Bell}_n\rangle = P(H) \otimes I|\mathrm{Bell}_n\rangle$$

Step 4: Take partial trace

$$P(H) \otimes I|\mathrm{Bell}_n\rangle \longrightarrow \rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

Commuting case



$$|R^l(H)\rangle \otimes |\mathrm{Bell}_n\rangle$$

$$\downarrow$$

$$\frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle \otimes (P^{\mathbf{y}} \otimes I)|\mathrm{Bell}_n\rangle$$

$$= \frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle \otimes |\mathrm{Symp}(P^{\mathbf{y}})\rangle$$

$$\downarrow$$

$$\frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |0\rangle \otimes |\mathrm{Symp}(P^{\mathbf{y}})\rangle \quad = P(H) \otimes I|\mathrm{Bell}_n\rangle$$

$$\downarrow$$

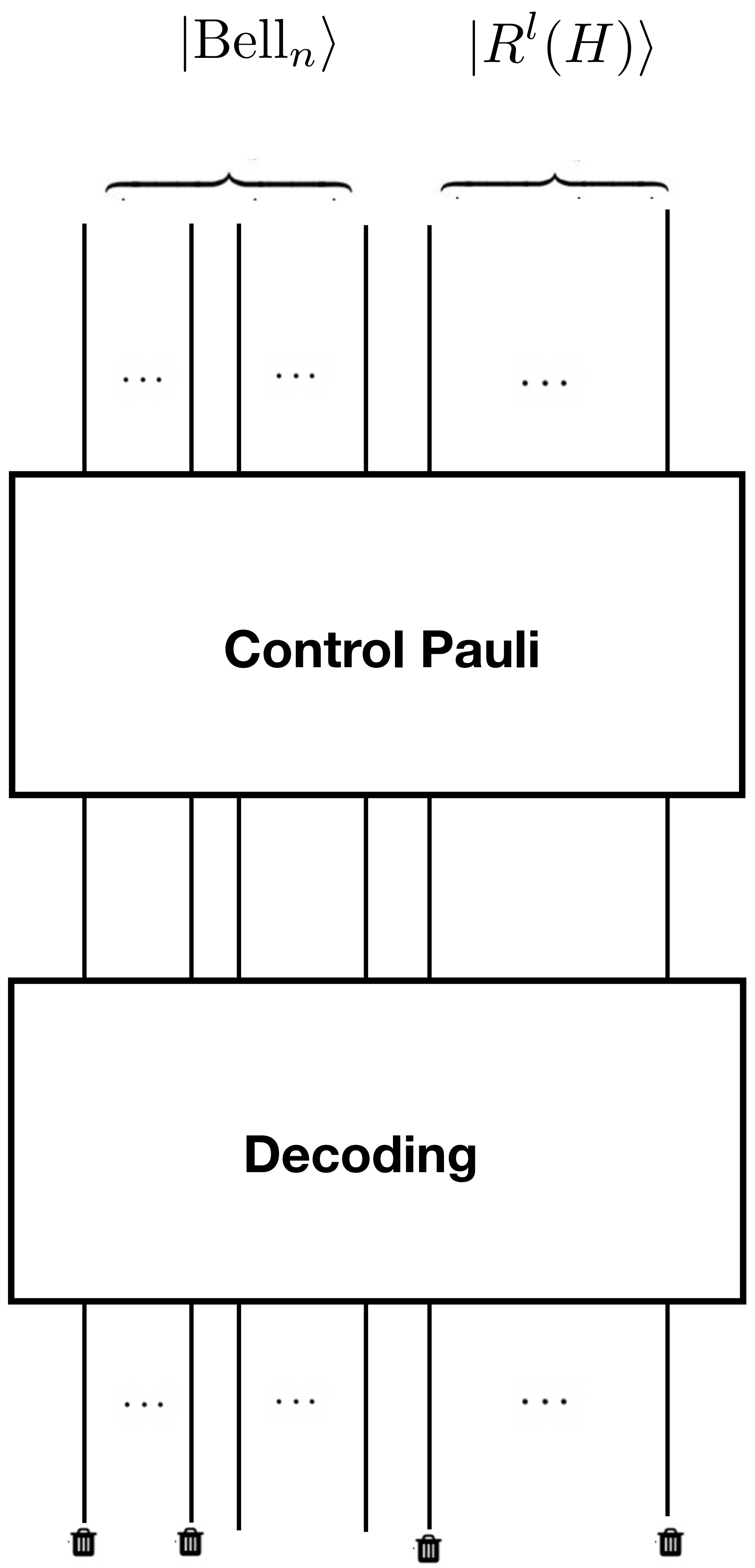
$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

$$P(H) = \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} P^{\mathbf{y}}$$

$$\mathrm{Symp}(P^{\mathbf{y}}) = B^T \mathbf{y}$$

$$B^\top = \begin{bmatrix} | & | & & | \\ \mathrm{symp}(P_1) & \mathrm{symp}(P_2) & \cdots & \mathrm{symp}(P_m) \\ | & | & & | \end{bmatrix}$$

Commuting case



$$|R^l(H)\rangle \otimes |\text{Bell}_n\rangle$$

$$\frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle \otimes (P^{\mathbf{y}} \otimes I)|\text{Bell}_n\rangle$$

$$= \frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |\mathbf{y}\rangle \otimes |\text{Symp}(P^{\mathbf{y}})\rangle$$

$$\frac{1}{\mathcal{N}} \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} |0\rangle \otimes |\text{Symp}(P^{\mathbf{y}})\rangle \quad = P(H) \otimes I|\text{Bell}_n\rangle$$

$$\rho_P(H) = \frac{P(H)^2}{\text{Tr}[P(H)^2]}$$

$$P(H) = \sum_{\mathbf{y}\in\mathbb{F}_2^m} w_{\mathbf{y}} P^{\mathbf{y}}$$

$$\text{Symp}(P^{\mathbf{y}}) = B^T \mathbf{y}$$

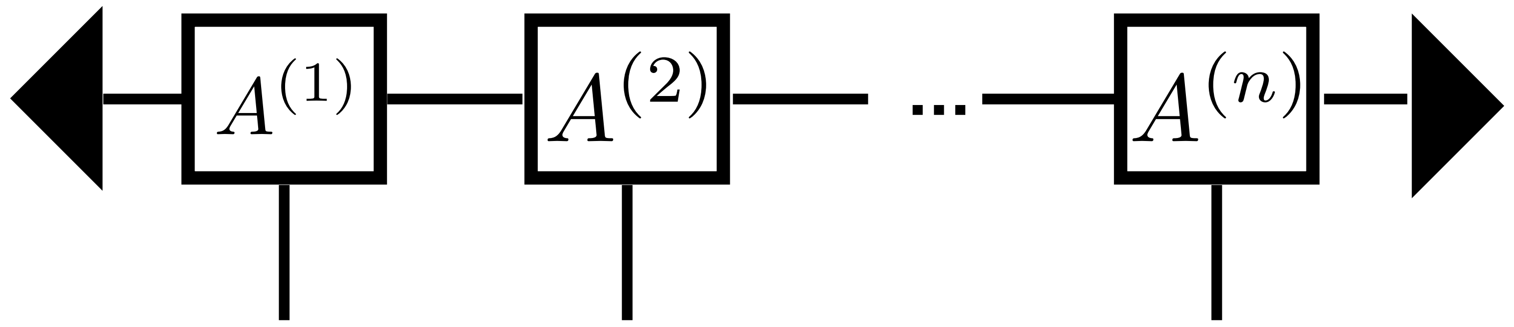
$$B^\top = \begin{bmatrix} | & | & & | \\ \text{symp}(P_1) & \text{symp}(P_2) & \cdots & \text{symp}(P_m) \\ | & | & & | \end{bmatrix}$$

A nature question arise:
can the reference state be efficiently prepared

Efficient preparation of reference state

Result: The reference state is an Matrix Product State (MPS) with bond dimension l+1

$$|R^l(H)\rangle = \sum_{y \in \mathbb{F}_2^m} v_L^\top A^{(1)}(y_1) \cdots A^{(m)}(y_m) v_R |y\rangle$$

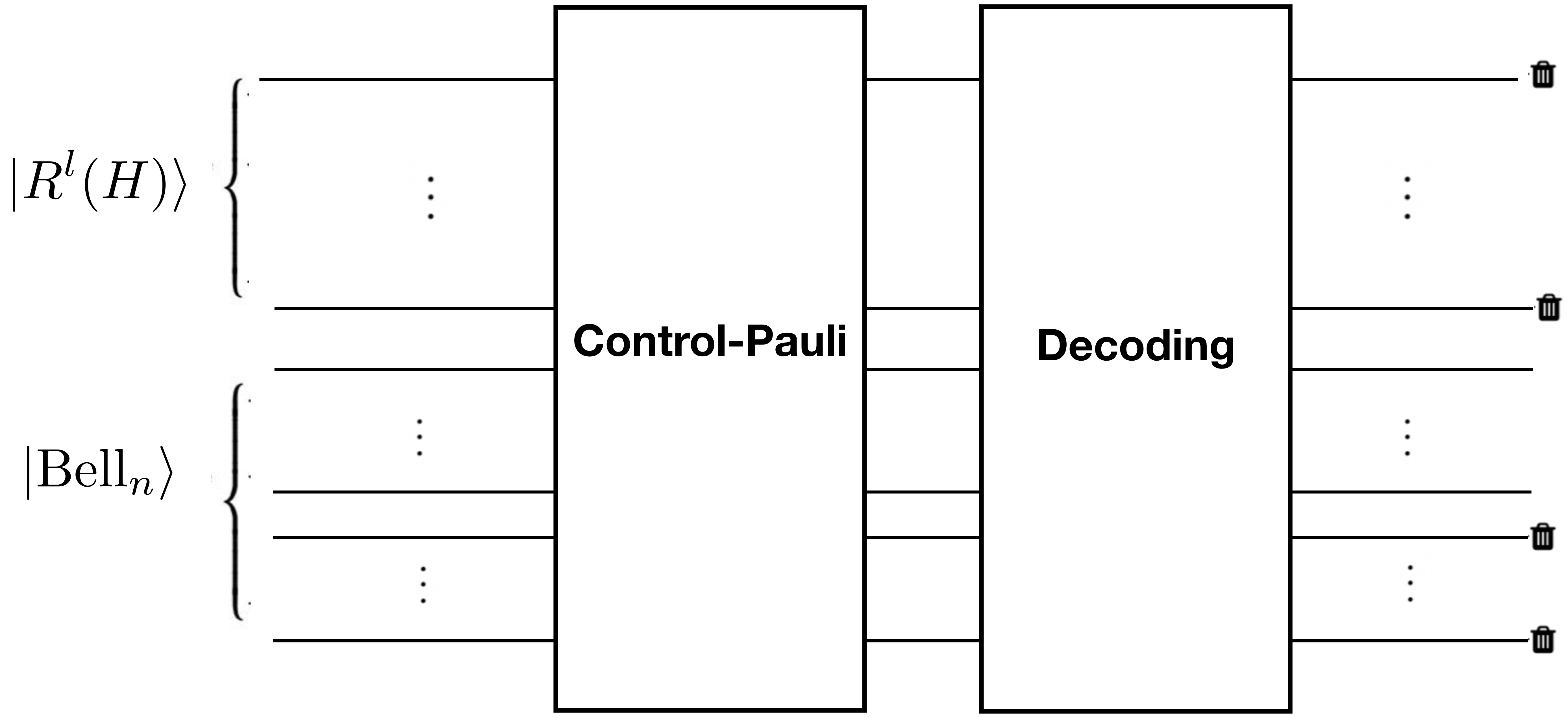


where $v_L = (\frac{1}{\mathcal{N}}, 0, \cdots, 0)^\top, \quad v_R = (a_0 \cdot 0!, a_1 \cdot 1!, \cdots, a_l \cdot l!)^\top$

$$A_0^{(k)} = \begin{pmatrix} 1 & 0 & \frac{c_k^2}{2!} & 0 & \frac{c_k^4}{4!} & \cdots & \cdots \\ 0 & 1 & 0 & \frac{c_k^2}{2!} & 0 & \ddots & \vdots \\ 0 & 0 & 1 & 0 & \frac{c_k^2}{2!} & \ddots & \frac{c_k^4}{4!} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & \frac{c_k^2}{2!} \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix}_{(l+1)\times(l+1)}, \quad A_1^{(k)} = \begin{pmatrix} 0 & \frac{c_k}{1!} & 0 & \frac{c_k^3}{3!} & 0 & \cdots & \cdots \\ 0 & 0 & \frac{c_k}{1!} & 0 & \frac{c_k^3}{3!} & \ddots & \vdots \\ 0 & 0 & 0 & \frac{c_k}{1!} & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \frac{c_k^3}{3!} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \frac{c_k}{1!} \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{pmatrix}_{(l+1)\times(l+1)}.$$

→ The reference state (MPS) can be prepared in time $O(m \cdot \mathrm{poly}(l))$

Commuting case

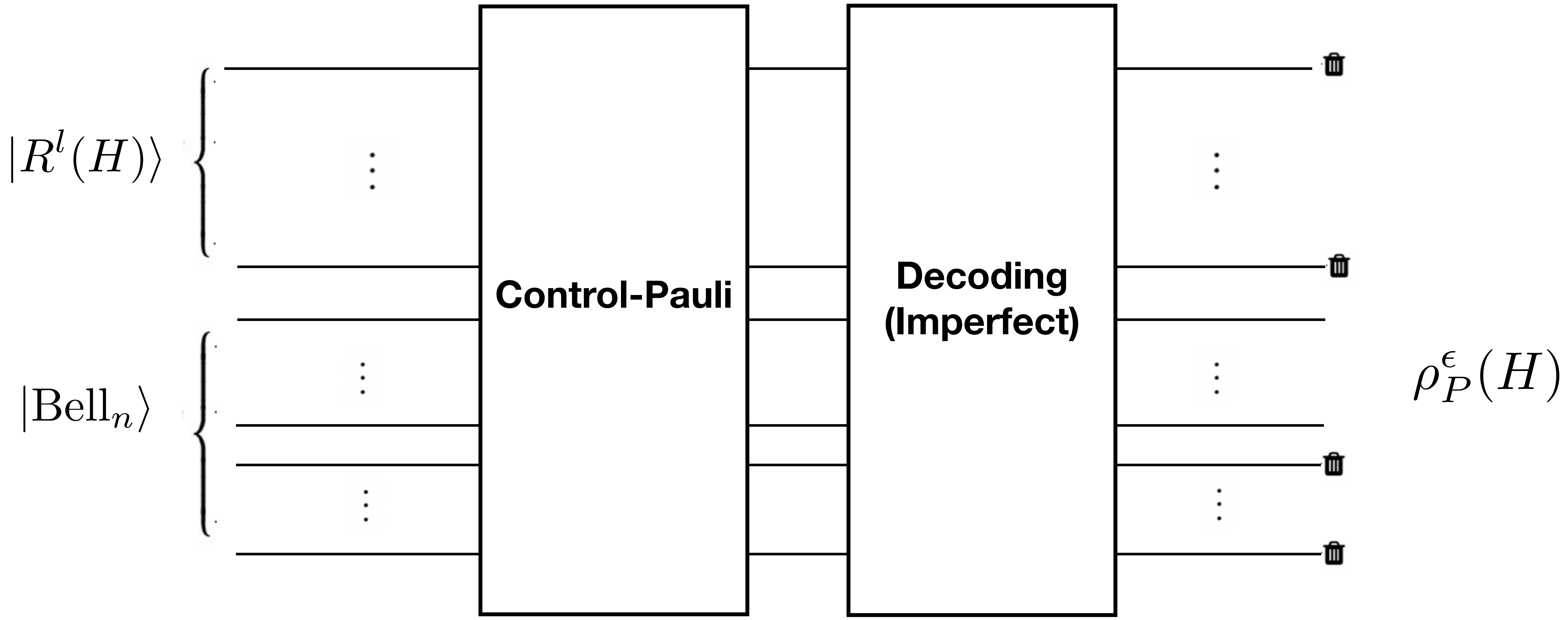


$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

$|R^l(H)\rangle$ can be prepared efficiently + Efficient decoding

Efficiently prepare $\rho_P(H)$

Commuting case



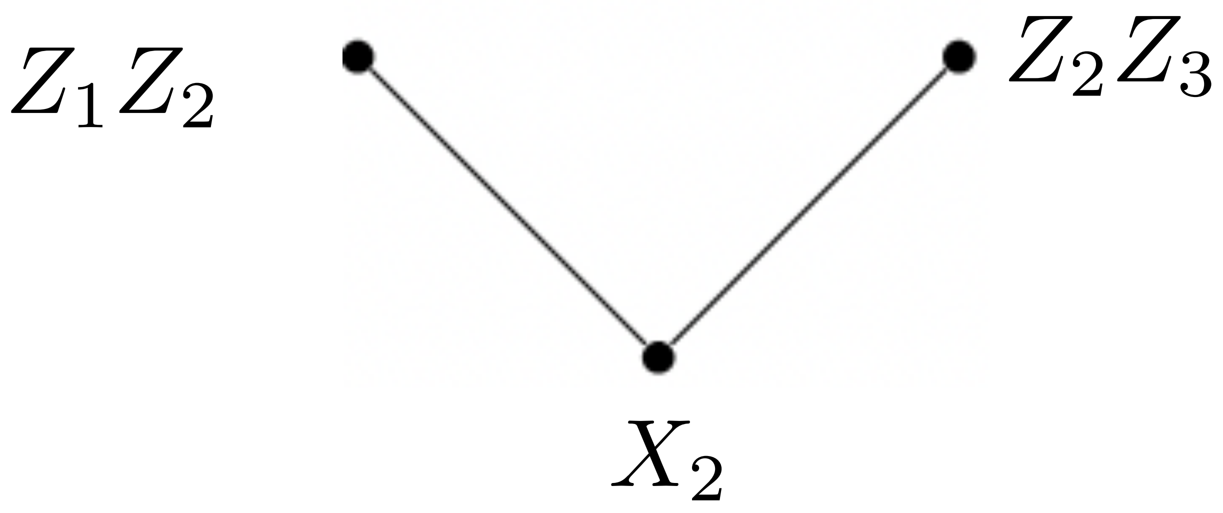
Imperfect Decoding: $D_H^{(l,\epsilon)}|y\rangle|B^\top y\rangle = \sum_{y'} \sqrt{p(y'|B^\top y)}|y \oplus y'\rangle|B^\top y\rangle, \forall |y| \leq l \quad p(y|B^\top y) \geq 1 - \epsilon$

$\|\rho_P^\epsilon(H) - \rho_P(H)\|_1 \leq 2\sqrt{\epsilon}$ (Robustness to imperfect decoding)

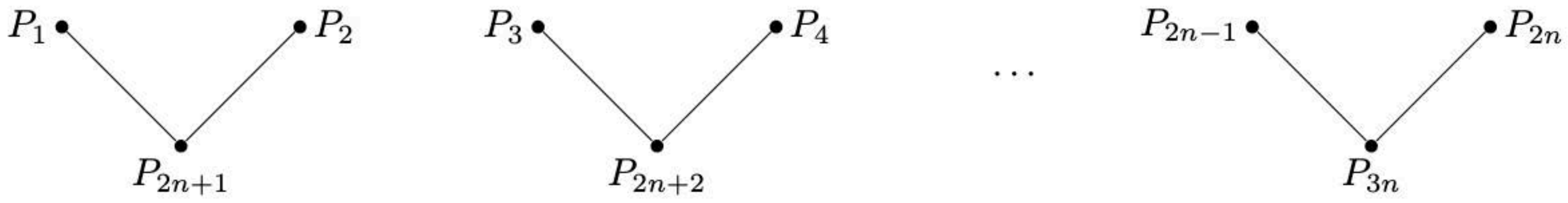
Noncommuting case

We need introduce the commutation graph based on the commute and anti-commute relations of the Pauli operators

E.g. $H = Z_1 Z_2 + Z_2 Z_3 + g X_2$

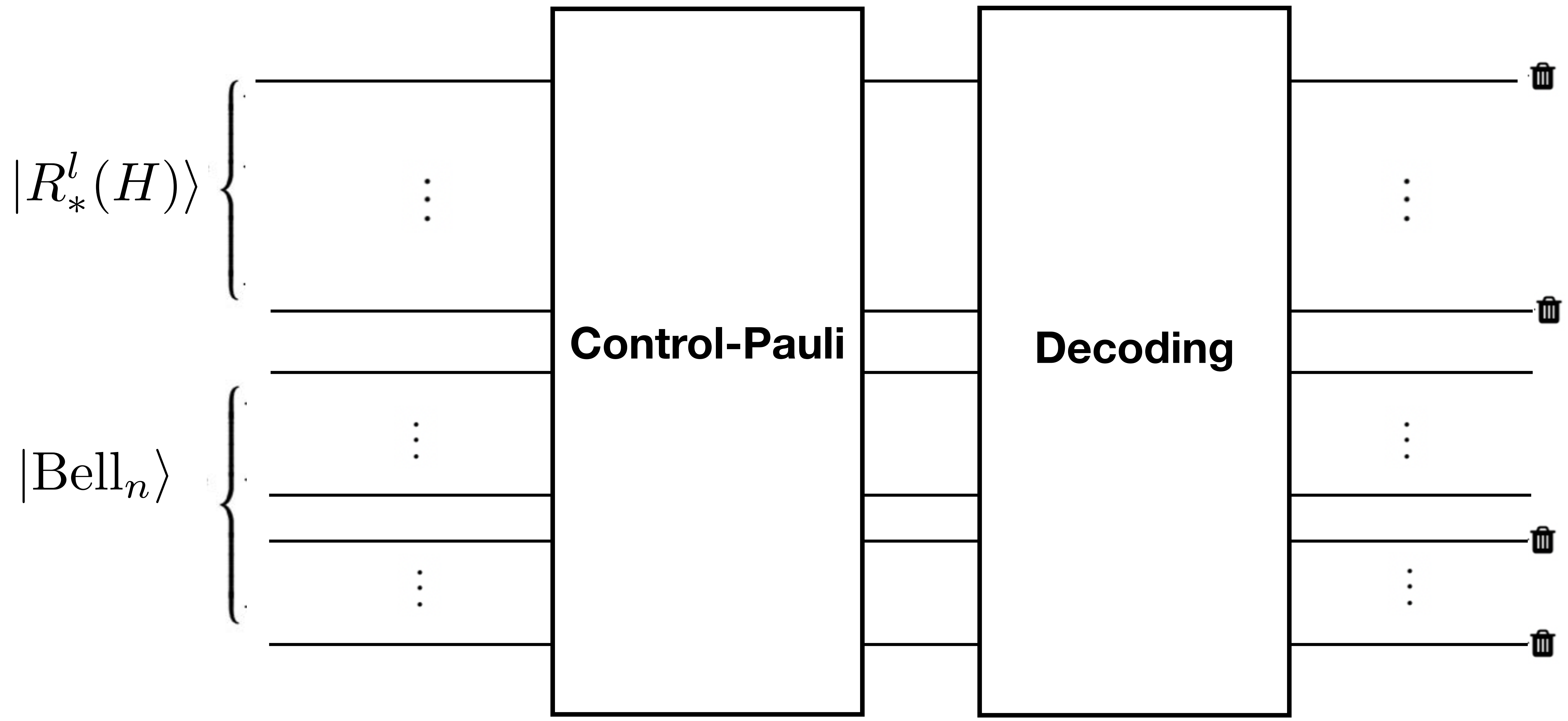


$$H = \sum_{i=1}^{2n} Z_i Z_{i+1} + g \sum_{i=1}^{n} X_{2i},$$



$$P_i = \begin{cases} Z_i Z_{i+1} & \text{for } 1 \leqslant i \leqslant 2n, \\ X_{2(i-2n)} & \text{for } 2n+1 \leqslant i \leqslant 3n, \end{cases}$$

Noncommuting case



$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

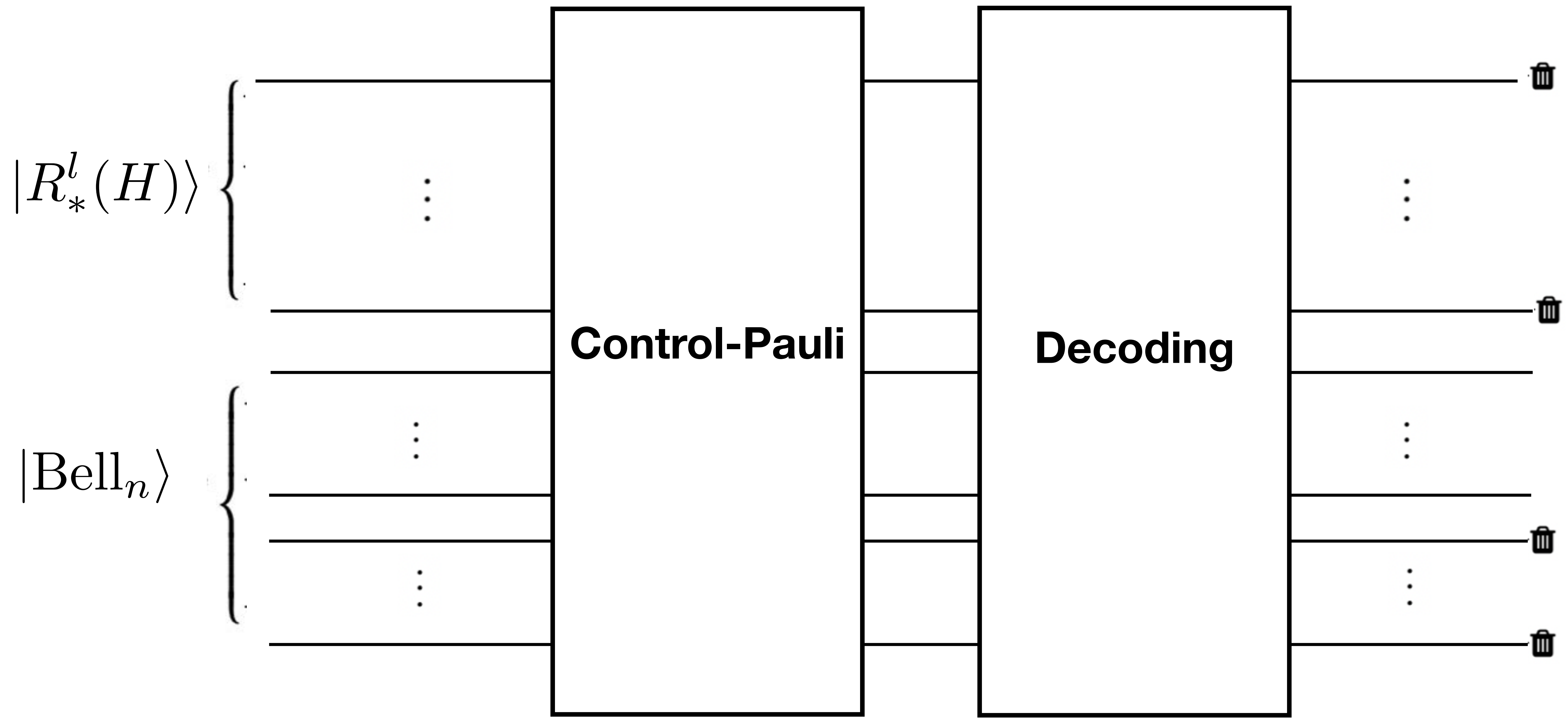
In the noncommuting case, the reference state is quite complicated

However, we can still express it into an MPS, and time complexity for preparation is

$$O(m \cdot \mathrm{poly}(l) \cdot \exp(\mathcal{M}))$$

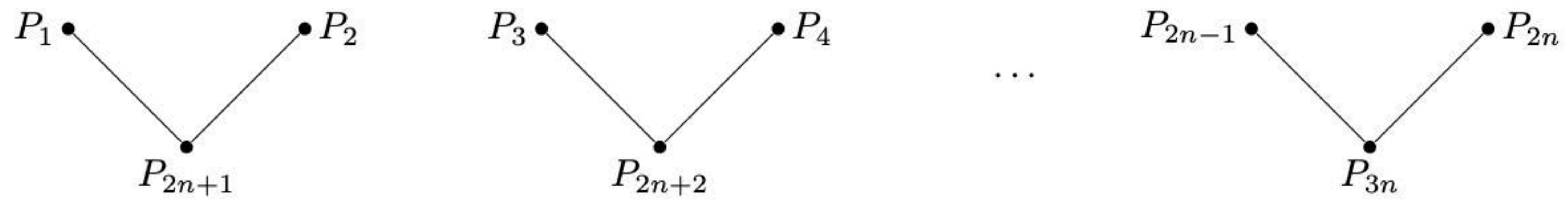
$\mathcal{M}$: maximal size of the connected components in the commutation graph of H

Noncommuting case



$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

For example, $$H = \sum_{i=1}^{2n} Z_i Z_{i+1} + g\sum_{i=1}^{n} X_{2i},$$

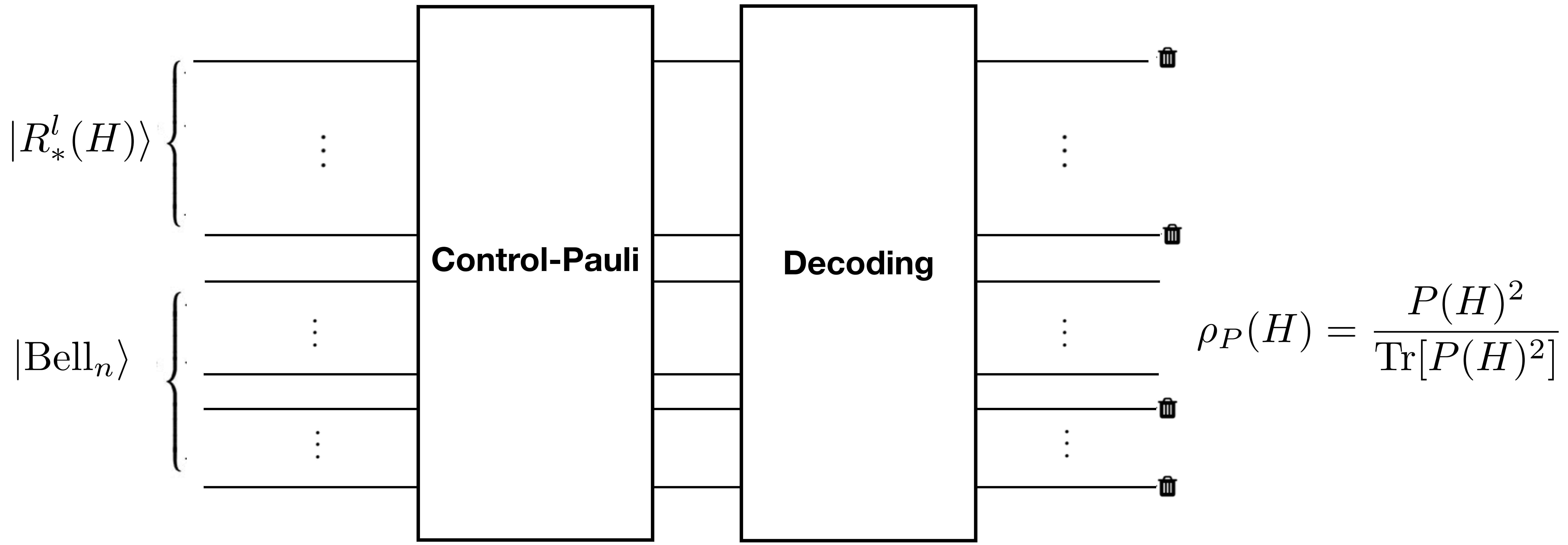


In this example, the maximal size of the connected components is 3, i.e., $\mathcal{M} = 3$

So the time complexity to prepare $|R_*^l(H)\rangle$ is $O(m \cdot \mathrm{poly}(l))$

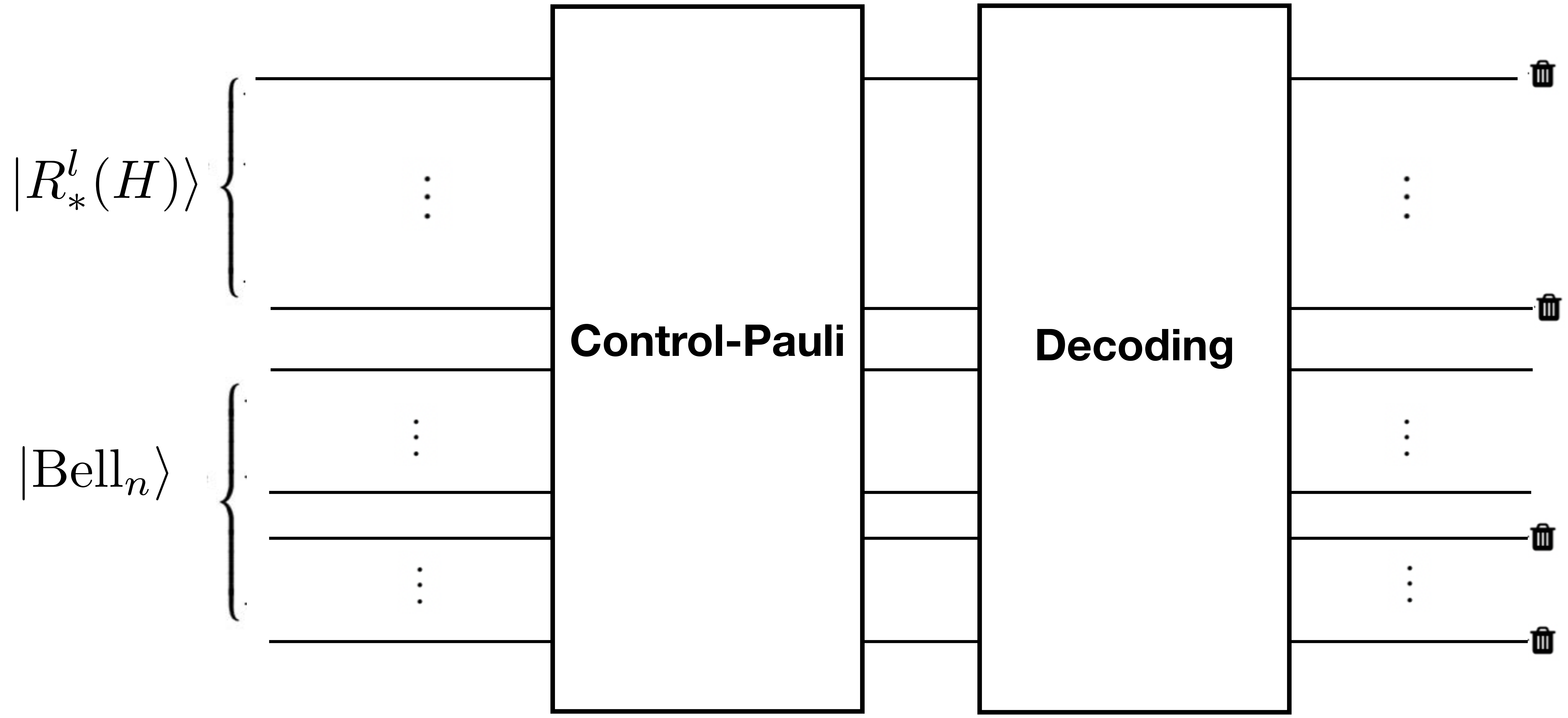
Details in [Bu, Gu, Li, arXiv:2601.18773]

Application: approximating Gibbs state



If the degree l is large enough, then output state $\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$ is close to the Gibbs state $\frac{e^{-\beta H}}{\mathrm{Tr}[e^{-\beta H}]}$

Application: approximating Gibbs state



$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

For example, $H = \sum_{i=1}^{2n} Z_i Z_{i+1} + g \sum_{i=1}^{n} X_{2i}$, we can choose some polynomial P with degree $l \leq 1.12\,(2+|g|)\beta n + 0.648 \ln \frac{2}{\delta}$

$$\|\rho_P(H) - e^{-\beta H}/\mathrm{Tr}[e^{-\beta H}]\|_1 \leq \delta$$

And the total running time is $\mathrm{poly}(l, n) = \mathrm{poly}((2+|g|)\beta, \ln \frac{1}{\delta}, n)$

Summary and outlook

In this work, we introduce a quantum algorithm to generate a function calculus on Hamiltonian $H = \sum_{i=1}^{m} c_i \mathrm{Pauli}_i, c_i \in \mathbb{R}, |c_i| \leq 1$

Hamiltonian

$$H = \sum_{i=1}^{m} c_i \mathrm{Pauli}_i, c_i \in \mathbb{R}$$

Polynomial P

→

Reference state preparation (MPS)

+

Quantum subroutine

+

Classical decoding

→

$$\rho_P(H) = \frac{P(H)^2}{\mathrm{Tr}[P(H)^2]}$$

Future directions: (1) More efficient algorithm for noncommuting case

(2) Other applications

.
.
.

Thank You!