

# Qudit Quantum Programming with Projective Cliffords

Jennifer Paykin

University of Vermont  
Intel

[jpaykin@uvm.edu](mailto:jpaykin@uvm.edu)

POPL  
January 13, 2026

Sam Winnick

Simon Fraser University  
University of Waterloo

[swinnick@sfu.ca](mailto:swinnick@sfu.ca)



# Unitary transformations

$$|\varphi\rangle \mapsto U |\varphi\rangle$$



$$P \mapsto U P U^\dagger$$

$P \in$  Pauli group  $\mathcal{P}_n$  :

unitary transformations generated by (tensor products of)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

# Clifford operators

$$|\varphi\rangle \mapsto U |\varphi\rangle$$



$$P \in \mathcal{P}_n \mapsto UPU^\dagger \in \mathcal{P}_n$$

$P \in$  Pauli group  $\mathcal{P}_n$  :  
unitary transformations generated by (tensor products of)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Clifford group

$$\{U \mid \forall P \in \mathcal{P}_n, UPU^\dagger \in \mathcal{P}_n\}$$

(also generated by Clifford gates H, S, CNOT)

Projective Cliffords

$$[U] = \{U' \mid UPU^\dagger = U'P(U')^\dagger\}$$

# Projective Cliffords

$$P \mapsto P'$$

$P \in$  Pauli group  $\mathcal{P}_n$  :  
unitary transformations generated by (tensor products of)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Clifford group

$$\{U \mid \forall P \in \mathcal{P}_n, UPU^\dagger \in \mathcal{P}_n\}$$

$\lambda^{PC}$

A language for functions over the Pauli group

where

well-typed functions  $\cong$  Projective Cliffords

pure  $\lambda$ -calculus with  
operational  
semantics

can be compiled to  
circuits

new ways of  
thinking about  
quantum algorithms

# Compilation to circuits via Pauli tableaux

- Every Pauli  $P$  can be written  $i^r \Delta_v = i^r X_1^{x_1} \star Z_1^{z_1} \star \dots \star X_n^{x_n} \star Z_n^{z_n}$ 
  - where  $Y = X \star Z = i X Z$
- Projective Cliffords distribute over  $\star$
- A projective Clifford on  $n$  qubits can be expressed as

$i$	$U X_i U^\dagger$	$U Z_i U^\dagger$
1	$P_1$	$Q_1$
...	...	...
$n$	$P_n$	$Q_n$

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Aaronson, Scott, and Daniel Gottesman. "Improved simulation of stabilizer circuits." *Physical Review A—Atomic, Molecular, and Optical Physics* 70.5 (2004): 052328.

# Compilation to circuits via Pauli tableaux

- Efficient simulation ( $O(n^2)$ ) of stabilizer (Clifford) circuits
- Efficient synthesis into Clifford circuits
- Used for circuit optimization
- A data structure, not a programming feature

Note:  
Cliffords are not  
universal!

$i$	$U X_i U^\dagger$	$U Z_i U^\dagger$
1	$P_1$	$Q_1$
...	...	...
$n$	$P_n$	$Q_n$

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Aaronson, Scott, and Daniel Gottesman. "Improved simulation of stabilizer circuits." *Physical Review A—Atomic, Molecular, and Optical Physics* 70.5 (2004): 052328.

Schmitz, Albert T., et al. "Graph optimization perspective for low-depth Trotter-Suzuki decomposition." *Physical Review A* 109.4 (2024): 042418.

# $\lambda^{PC}$ Examples

Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$HXH^\dagger = Z$$

$$HZH^\dagger = X$$

```
hadamard :: |Pauli -o Pauli|  
hadamard |X| = Z  
hadamard |Z| = X
```

```
$ hadamard Y  
->* hadamard (X*Z)  
->* hadamard X * hadamard Z  
->* Z * X  
->* -Y
```

$$Y = X * Z = iXZ$$

# $\lambda^{PC}$ Examples

controlled-not

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

i	CNOT $X_i$ CNOT <sup>†</sup>	CNOT $Z_i$ CNOT <sup>†</sup>
1	$X_1 X_2$	$Z_1$
2	$X_2$	$Z_1 Z_2$

```
cnot :: |Pauli ⊗ Pauli -o Pauli ⊗ Pauli|
cnot |in1 X| = in1 X * in2 X
cnot |in1 Z| = in1 Z
cnot |in2 X| = in2 X
cnot |in2 Z| = in1 Z * in2 Z
```

# $\lambda^{PC}$ Examples

Any Pauli  $P$  is also Clifford

$$PQP^\dagger = (-1)^{\omega(P,Q)} Q$$

Any two Paulis either commute or anticommute:

$$PQ = (-1)^{\omega(P,Q)} QP$$

```
omega :: tau -> tau -> Z2
omega (in i X) (in j X) = 0
omega (in i Z) (in j Z) = 0
omega (in i X) (in j Z) = if i=j then 1 else 0
omega (in i Z) (in j X) = if i=j then 1 else 0
```

```
pauli_to_clifford :: |tau| -> |tau -> tau|
pauli_to_Clifford p |q| = <omega(p,q)> q
```

$$\langle r \rangle q = (-1)^r q$$

# A type system for projective Cliffords

- A function  $\text{Pauli} \rightarrow \text{Pauli}$  corresponds to a projective Clifford iff :
  - it is a center-fixing automorphism over the Pauli group
- Want:
  - Properties can be checked efficiently by a type system
  - Compositionality
- Encodings (tableaux):  $\Delta_v \mapsto (-1)^{\mu(v)} \Delta_{\phi(v)}$

# From qubits to qudits

- Qubits:  $\alpha_0|0\rangle + \alpha_1|1\rangle$
- Qudits:  $\alpha_0|0\rangle + \dots + \alpha_{d-1}|d-1\rangle$
- Encodings of projective Cliffords (tableaux) for odd dimensions are straightforward:
  - $\Delta_v \mapsto \zeta^{\mu(v)} \Delta_{\phi(v)}$  where  $\zeta^d = 1$
  - $v \in V = (\mathbb{Z}/d\mathbb{Z})^{2n}$
  - $\mu : V \rightarrow \mathbb{Z}/d\mathbb{Z}$ 
    - must be a linear transformation
  - $\phi : V \rightarrow V$ 
    - must be a linear transformation
    - must be a symplectomorphism:  $\omega(\phi(v_1), \phi(v_2)) = \omega(v_1, v_2)$

Claim:

These properties can be enforced by a type system and are appropriately compositional

# From qubits to qudits

- Typical encodings of projective Cliffords (tableaux) for even dimensions are **NOT** straightforward:
  - $\Delta_v \mapsto \zeta^{\mu(v)} \Delta_{\phi(v)}$  where  $\zeta^d = 1$
  - $v \in V' = (\mathbb{Z}/2d\mathbb{Z})^{2n}$
  - $\mu : V' \rightarrow \mathbb{Z}/d\mathbb{Z}$ 
    - may or may not be linear...
  - $\phi : V' \rightarrow V'$ 
    - must be a linear transformation
    - is a symplectomorphism iff  $\mu$  is linear...

# Condensed encodings

## Condensed encodings of projective Clifford operations in arbitrary dimension

Cite as: J. Math. Phys. 66, 072202 (2025); doi: 10.1063/5.0239974  
Submitted: 23 September 2024 • Accepted: 8 July 2025 •  
Published Online: 30 July 2025



Sam Winnick<sup>1,a)</sup> and Jennifer Paykin<sup>2</sup>

- Qudits are treated uniformly whether  $d$  is even or odd

- $\Delta_b \mapsto \zeta^{\mu(b)} \Delta_{\phi(b)}$

where  $\zeta^d = 1$

- $\Delta_b \in \{X_i, Z_i\}_i$

- $\mu : V \rightarrow \mathbb{Z}/d\mathbb{Z}$

- must be a linear transformation

- $\phi : V \rightarrow V$

- must be a linear transformation

- must be a symplectomorphism:  $\omega(\phi(v_1), \phi(v_2)) = \omega(v_1, v_2)$

Caveat:

evaluation on arbitrary Paulis  $\Delta_v$   
requires phase correction:

$$\Delta_v \mapsto \pm \zeta^{\mu(v)} \Delta_{\phi(v)}$$

# A type system for qudit projective Cliffords

$$\Theta \vdash t : \tau$$

- Linear types to enforce linearity of vectors & phases
- Symplectomorphism check:

$$\frac{x_1 : \tau, x_2 : \tau \vdash \omega(t^\psi \{x_1/x\}, t^\psi \{x_2/x\}) \equiv \omega(x_1, x_2)}{x : \tau \vdash^s t : \tau}$$

- Beta reduction performs phase adjustment computations

# Also in the paper...

- Programming examples with qudits
- Case study: stabilizer error-correcting codes
- Categorical semantics of condensed encodings
- Full description of type system & operational semantics
- Additional language details:
  - Pauli data structures and non-quantum/non-linear data
  - Compiling to circuits

# Extensions/future work

- Implementation: [github.com/jpaykin/LambdaPC-ocaml](https://github.com/jpaykin/LambdaPC-ocaml)



- Extensions of the language
  - Stabilizer semantics including measurement
  - Combine dimensions of qubits
  - Clifford hierarchy
  - and more...

# Qudit Quantum Programming with Projective Cliffords

Jennifer Paykin

University of Vermont  
Intel

[jpaykin@uvm.edu](mailto:jpaykin@uvm.edu)

POPL  
January 13, 2026

Hiring  
PhD  
students!

Sam Winnick

Simon Fraser University  
University of Waterloo

[swinnick@sfu.ca](mailto:swinnick@sfu.ca)

Questions?



# Bonus Slides

# What are the building blocks of fault-tolerant quantum computation?

- “the definition of a cnot gate that we care about is this map on operators:  $x$  on control gets copied to both,  $z$  on control goes straight through...
- that is the definition of cnot for the purposes of fault-tolerance in a 2d nearest neighbor square grid”
- Austin Fowler (Google), Munich Quantum Software Forum 2024