Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits

Quantum Technology Initiative Journal Club

Saverio Monaco May 27, 2026

Outline

arXiv: `https://arxiv.org/abs/2512.16674`

`https://github.com/desyqml/Pauli-Propagator`

Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits

Saverio Monaco [1,2] Jamal Slim [2] Florian Rehm [3] Dirk Krücker [2] and Kerstin Borras [1,2]

[1] *RWTH Aachen University, Aachen 52062, Germany*
[2] *Deutsches Elektronen-Synchrotron DESY, 22603 Hamburg, Germany*
[3] *European Organization for Nuclear Research (CERN), Geneva 1211, Switzerland*

(Dated: December 18, 2025)

Quantum Machine Learning models typically require expensive on-chip training procedures and often lack efficient gradient estimation methods. By employing Pauli propagation, it is possible to derive a symbolic representation of observables as analytic functions of a circuit's parameters. Although the number of terms in such functional representations grows rapidly with circuit depth, suitable choices of ansatz and controlled truncations on Pauli weights and frequency components yield accurate yet tractable estimators of the target observables. With the right ansatz design, this approach can be extended to system sizes beyond the reach of classical simulation, enabling scalable training for larger quantum systems. This also enables a form of classical pre-training through gradient-based optimization prior to deployment on quantum hardware. The proposed approach is demonstrated on the Variational Quantum Eigensolver for obtaining the ground state of a spin model, showing that accurate results can be achieved with a scalable and computationally efficient procedure.

I. INTRODUCTION

Parametrized quantum circuits (PQCs) have attracted considerable attention in recent years due to their potential applications in Quantum Machine Learning (QML) and the growing availability of hardware capable of executing quantum algorithms for tasks of practical interest.

Despite the progress in hardware, a major challenge for QML remains the training time of these models. Evaluating and optimizing PQCs on quantum hardware is resource-intensive, with the added consideration that quantum devices are currently limited and experimental.

Obtaining gradients for PQC further exacerbates

In this work, we propose a method to represent an observable as an explicit function of the circuit parameters in symbolic form, independent of any particular parameter assignment.

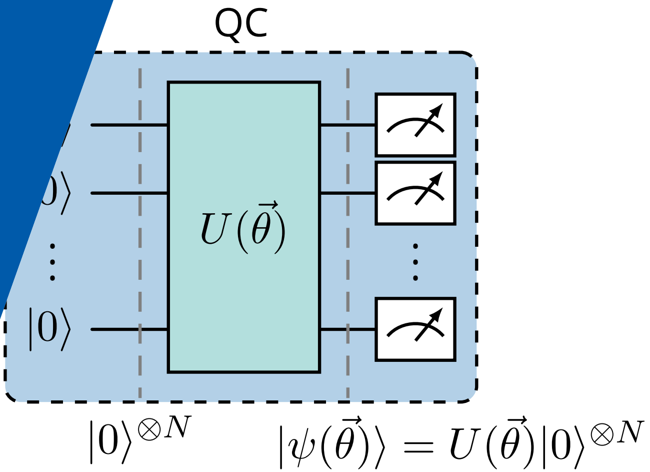
This approximate representation can be used to pretrain PQCs for the tasks described above, and subsequently fine-tuned via on-chip training or alternative optimization techniques.

In general, computing the exact functional representation of an observable is computationally challenging, as the number of terms grows (in the worst case scenario) exponentially with the number of parameters. To address

QUANTUM TECHNOLOGY INITIATIVE

Introduction

Setting and Objectives



Introduction

Pauli propagation is a well-known classical technique, based on the Heisenberg picture, for computing the expectation value of an observable associated with a given quantum circuit [1].

Introduction

Pauli propagation is a well-known classical technique, based on the Heisenberg picture, for computing the expectation value of an observable associated with a given quantum circuit [1].

The exact method is known to fail for circuits of even modest size. In *"Classically Estimating Observables of Noiseless Quantum Circuits"* by Angrisani *et al.* [2], the approach is extended to larger circuits by restricting to a specific subclass of circuits and introducing a cutoff in the propagation.

Introduction

	Conditions	Output
Angrisani et al.	Locally Scrambling + Weight Cutoff	$\langle O \rangle \to 0.4$
Our work	Locally Scrambling + Weight Cutoff + Frequency Cutoff	$\langle O(\vec{\theta}) \rangle \to -\sin(\theta_2)\sin(\theta_3)\cos(\theta_0) + \ldots$


Classically estimating observables of noiseless quantum circuits


Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits

Introduction

Angrisani et al.


Classically estimating observables of noiseless quantum circuits

Conditions

Output

Locally Scrambling
+ Weight Cutoff

$$\langle O \rangle \to 0.4$$

Our work


Symbolic Pauli Propagation for Gradient-Enabled Pre-Training of Quantum Circuits

Locally Scrambling
+ Weight Cutoff
+ Frequency Cutoff

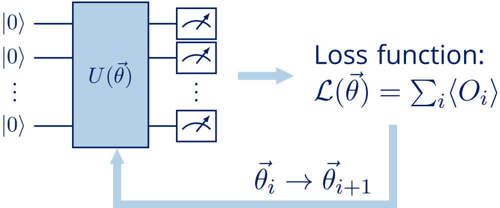
$$\langle O(\vec{\theta}) \rangle \to -\sin(\theta_2)\sin(\theta_3)\cos(\theta_0) + \dots$$

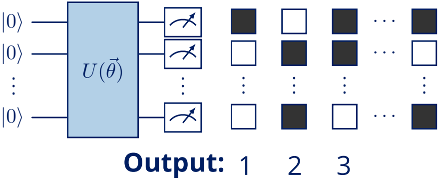
Access to the explicit function of the observable allows for a single propagation needed, and gradient descent

Introduction

Potential applications: algorithms which output is a single wavefunction collapse (measurement), but the loss function can be formalized through expectation values:

Training on classical resources (Pauli Propagation) $\rightarrow$ Deployment on quantum resources

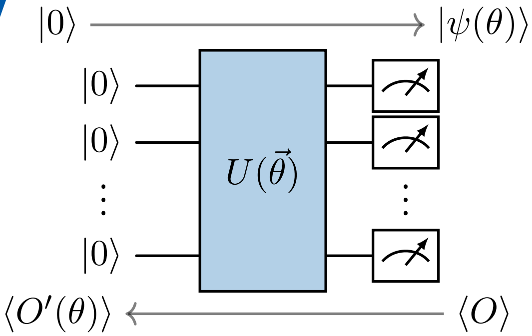




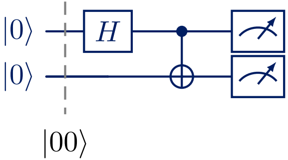
Applications: Variational Quantum Eigensolver [3], Compression (Autoencoder) [4], Generation [5, 6]

Evolution Pictures

Schrŏedinger & Heisenberg Pictures



Propagation

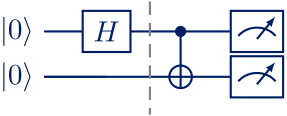


Schrödinger Picture:
Propagate the Wavefunction through the circuit according to the rule:

$$|\psi\rangle' = U|\psi\rangle$$

$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \boxed{H} \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

Propagation



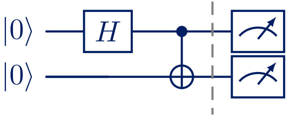
$$\frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right)$$

Schrödinger Picture:
Propagate the Wavefunction through the circuit according to the rule:

$$|\psi\rangle' = U|\psi\rangle$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \boxed{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

Propagation



$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

Schrödinger Picture:

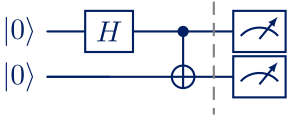
Propagate the Wavefunction through the circuit according to the rule:

$$|\psi\rangle' = U|\psi\rangle$$

$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\equiv \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

Propagation



$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

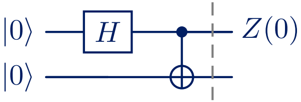
Schrödinger Picture:
Propagate the Wavefunction through the circuit according to the rule:

$$|\psi\rangle' = U|\psi\rangle$$

$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \boxed{H} \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

Example: $\langle Z(0)\rangle = \langle\psi|Z\,I\,|\psi\rangle = 0$

Propagation



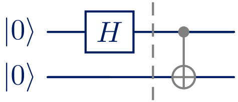
Heisenberg Picture:
Evolve the operator *"backwards"* through the circuit according to the rule:

$$V' = U^{\dagger} V U$$

(Similarity transformation)

$Z(0)I(1)$

Propagation



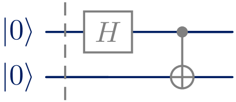
Heisenberg Picture:
Evolve the operator *"backwards"* through the circuit according to the rule:

$$V' = U^{\dagger} V U$$

(Similarity transformation)

$$Z(0)I(1) = CNOT^{\dagger} Z(0)I(1)CNOT$$

Propagation



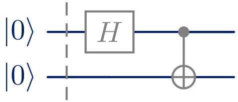
$$X(0)I(1) = H^\dagger Z(0)I(1)H$$

Heisenberg Picture:
Evolve the operator *"backwards"* through the circuit according to the rule:

$$V' = U^\dagger V U$$

(Similarity transformation)

Propagation



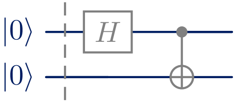
$$\langle 00|X(0)I(1)|00\rangle = 0$$

Heisenberg Picture:
Evolve the operator *"backwards"* through the circuit according to the rule:

$$V' = U^\dagger V U$$

(Similarity transformation)

Propagation



Heisenberg Picture:
Evolve the operator *"backwards"* through the circuit according to the rule:

$$V' = U^\dagger V U$$

(Similarity transformation)

$$\langle 00|X(0)I(1)|00\rangle = 0 \rightarrow \langle 00|U^\dagger Z(0)U|00\rangle = 0$$

Definitions

A Pauli word is a **product of Pauli operators** applied to different qubits.

Pauli Operators:

- Pauli-X (X): Bit-flip operator.
- Pauli-Y (Y): Bit-flip and phase-flip operator.
- Pauli-Z (Z): Phase-flip operator.
- Identity (I): (Can be omitted)

Examples:

- $X(0)I(1)I(2) \rightarrow$ Apply Pauli-X to the first qubit and identity (I) to the second and third qubits.
- $Z(0)X(1)Y(2) \rightarrow$ Apply Pauli-Z to the first qubit, Pauli-X to the second qubit, and Pauli-Y to the third qubit.

Definitions

The **weight** refers to the number of qubits on which the Pauli word has a non-identity component.
It measures how many qubits are actively involved in the Pauli word.

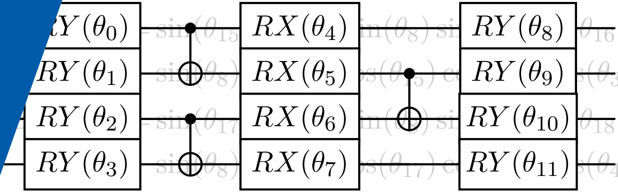
Examples:

- $X(0)I(1)I(2)$ (Weight 1)
- $Z(0)X(1)Y(2)$ (Weight 3)

QUANTUM TECHNOLOGY INITIATIVE

Propagation of Observables

Evolving Observables through a Parametrized Quantum Circuit



Propagation

Paper example

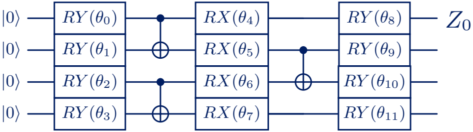


Figure: Four-qubit local entangler ansatz

$$\begin{aligned}
Y \otimes I &\longrightarrow Y \otimes X \\
Z \otimes I &\longrightarrow Z \otimes I \\
I \otimes X &\longrightarrow I \otimes X \\
I \otimes Y &\longrightarrow Z \otimes Y \\
I \otimes Z &\longrightarrow Z \otimes Z \\
X \otimes X &\longrightarrow X \otimes I \\
Z \otimes Z &\longrightarrow I \otimes Z
\end{aligned}$$

$\boxed{RX(\theta)}$

$$\begin{aligned}
I &\longrightarrow I \\
X &\longrightarrow X \\
Y &\longrightarrow +\cos\theta Y \\
&\qquad -\sin\theta Z \\
Z &\longrightarrow +\cos\theta Z \\
&\qquad +\sin\theta Y
\end{aligned}$$

$\boxed{RY(\theta)}$

$$\begin{aligned}
I &\longrightarrow I \\
X &\longrightarrow +\cos\theta X \\
&\qquad +\sin\theta Z \\
Y &\longrightarrow Y \\
Z &\longrightarrow +\cos\theta Z \\
&\qquad -\sin\theta X
\end{aligned}$$

Propagation

Paper example

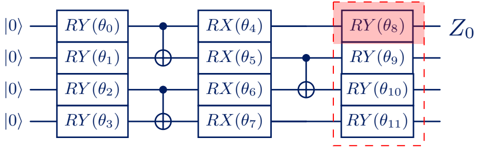


Figure: Four-qubit local entangler ansatz

- **Layer 5:** At this stage, the observable Z_0 is only affected by the gate $RY(\theta_8)j$. This evolves as:

$$Z_0 \to \cos\theta_8 Z_0 - \sin\theta_8 X_0$$

$$\begin{aligned} Y\otimes I &\longrightarrow Y\otimes X\\ Z\otimes I &\longrightarrow Z\otimes I\\ I\otimes X &\longrightarrow I\otimes X\\ I\otimes Y &\longrightarrow Z\otimes Y\\ I\otimes Z &\longrightarrow Z\otimes Z\\ X\otimes X &\longrightarrow X\otimes I\\ Z\otimes Z &\longrightarrow I\otimes Z \end{aligned}$$

$$\boxed{RX(\theta)}$$

$$\begin{aligned} I &\longrightarrow I\\ X &\longrightarrow X\\ Y &\longrightarrow +\cos\theta Y\\ & \qquad -\sin\theta Z\\ Z &\longrightarrow +\cos\theta Z\\ & \qquad +\sin\theta Y \end{aligned}$$

$$\boxed{RY(\theta)}$$

$$\begin{aligned} I &\longrightarrow I\\ X &\longrightarrow +\cos\theta X\\ & \qquad +\sin\theta Z\\ Y &\longrightarrow Y\\ Z &\longrightarrow +\cos\theta Z\\ & \qquad -\sin\theta X \end{aligned}$$

Propagation

Paper example

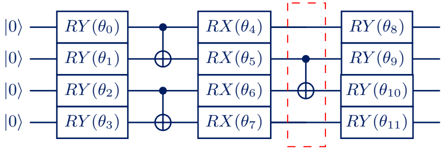


Figure: Four-qubit local entangler ansatz

- **Layer 4:** Both terms of the observable commute with CNOT$(1, 2)$, hence no transformation occurs:

$$\cos\theta_8 Z_0 - \sin\theta_8 X_0 \to \cos\theta_8 Z_0 - \sin\theta_8 X_0$$

$$\begin{aligned}
Y \otimes I &\longrightarrow Y \otimes X \\
Z \otimes I &\longrightarrow Z \otimes I \\
I \otimes X &\longrightarrow I \otimes X \\
I \otimes Y &\longrightarrow Z \otimes Y \\
I \otimes Z &\longrightarrow Z \otimes Z \\
X \otimes X &\longrightarrow X \otimes I \\
Z \otimes Z &\longrightarrow I \otimes Z
\end{aligned}$$

$\boxed{RX(\theta)}$

$$\begin{aligned}
I &\longrightarrow I \\
X &\longrightarrow X \\
Y &\longrightarrow +\cos\theta Y \\
&\qquad -\sin\theta Z \\
Z &\longrightarrow +\cos\theta Z \\
&\qquad +\sin\theta Y
\end{aligned}$$

$\boxed{RY(\theta)}$

$$\begin{aligned}
I &\longrightarrow I \\
X &\longrightarrow +\cos\theta X \\
&\qquad +\sin\theta Z \\
Y &\longrightarrow Y \\
Z &\longrightarrow +\cos\theta Z \\
&\qquad -\sin\theta X
\end{aligned}$$

Propagation

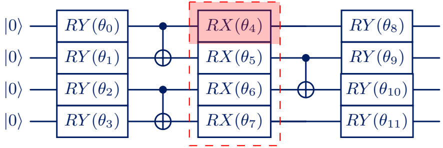


Figure: Four-qubit local entangler ansatz

- **Layer 3:** Here only the gate $RX(\theta_4)_0$ affects the propagated observable:

$$\begin{aligned}\cos\theta_8 Z_0 - \sin\theta_8 X_0 \rightarrow & + \cos\theta_8 \cos\theta_4 Z_0 \\ & + \cos\theta_8 \sin\theta_4 Y_0 \\ & - \sin\theta_8 X_0\end{aligned}$$

$$\begin{aligned} Y \otimes I &\longrightarrow Y \otimes X \\ Z \otimes I &\longrightarrow Z \otimes I \\ I \otimes X &\longrightarrow I \otimes X \\ I \otimes Y &\longrightarrow Z \otimes Y \\ I \otimes Z &\longrightarrow Z \otimes Z \\ X \otimes X &\longrightarrow X \otimes I \\ Z \otimes Z &\longrightarrow I \otimes Z \end{aligned}$$

$\boxed{RX(\theta)}$

$$\begin{aligned} I &\longrightarrow I \\ X &\longrightarrow X \\ Y &\longrightarrow + \cos\theta Y \\ & \quad\;\; - \sin\theta Z \\ Z &\longrightarrow + \cos\theta Z \\ & \quad\;\; + \sin\theta Y \end{aligned}$$

$\boxed{RY(\theta)}$

$$\begin{aligned} I &\longrightarrow I \\ X &\longrightarrow + \cos\theta X \\ & \quad\;\; + \sin\theta Z \\ Y &\longrightarrow Y \\ Z &\longrightarrow + \cos\theta Z \\ & \quad\;\; - \sin\theta X \end{aligned}$$

Propagation

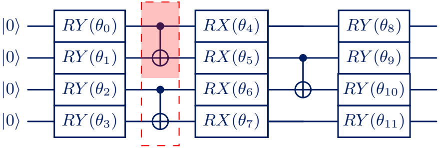


Figure: Four-qubit local entangler ansatz

- **Layer 2:** The $\mathrm{CNOT}(0,1)$ gate acts on all observables involving the first two qubits:

$$+\cos\theta_8\cos\theta_4 Z_0 + \cos\theta_8 \sin\theta_4 Y_0 - \sin\theta_8 X_0$$
$$\downarrow$$
$$+\cos\theta_8\cos\theta_4 Z_0 + \cos\theta_8 \sin\theta_4 Y_0 X_1$$
$$-\sin\theta_8 X_0 X_1$$

$$\begin{aligned} Y\otimes I &\longrightarrow Y\otimes X\\ Z\otimes I &\longrightarrow Z\otimes I\\ I\otimes X &\longrightarrow I\otimes X\\ I\otimes Y &\longrightarrow Z\otimes Y\\ I\otimes Z &\longrightarrow Z\otimes Z\\ X\otimes X &\longrightarrow X\otimes I\\ Z\otimes Z &\longrightarrow I\otimes Z \end{aligned}$$

$\boxed{RX(\theta)}$

$$\begin{aligned} I &\longrightarrow I\\ X &\longrightarrow X\\ Y &\longrightarrow +\cos\theta Y\\ & \qquad -\sin\theta Z\\ Z &\longrightarrow +\cos\theta Z\\ & \qquad +\sin\theta Y \end{aligned}$$

$\boxed{RY(\theta)}$

$$\begin{aligned} I &\longrightarrow I\\ X &\longrightarrow +\cos\theta X\\ & \qquad +\sin\theta Z\\ Y &\longrightarrow Y\\ Z &\longrightarrow +\cos\theta Z\\ & \qquad -\sin\theta X \end{aligned}$$

Propagation

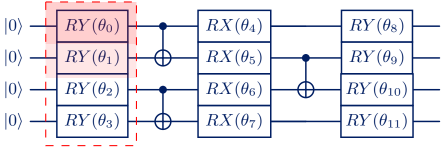


Figure: Four-qubit local entangler ansatz

- **Layer 1:** In this final layer, both $RY(\theta_0)_0$ and $RY(\theta_1)_1$ contribute to the evolution.
  - **Action of $RY(\theta_1)_1$:**

$$
\begin{aligned}
&+\cos\theta_8\cos\theta_4 Z_0 + \cos\theta_8\sin\theta_4 Y_0 X_1\\
&-\sin\theta_8 X_0 X_1\\
&\quad\downarrow\\
&\quad+\cos\theta_8\cos\theta_4 Z_0\\
&\quad+\cos\theta_8\sin\theta_4\cos\theta_1 Y_0X_1 - \cos\theta_8\sin\theta_4\sin\theta_1 Y_0Z_1\\
&\quad-\sin\theta_8\cos\theta_1 X_0X_1 + \sin\theta_8\sin\theta_1 X_0Z_1
\end{aligned}
$$

$$
\begin{aligned}
Y\otimes I &\longrightarrow Y\otimes X\\
Z\otimes I &\longrightarrow Z\otimes I\\
I\otimes X &\longrightarrow I\otimes X\\
I\otimes Y &\longrightarrow Z\otimes Y
\end{aligned}
$$

$RX(\theta)$

$$
\begin{aligned}
I &\longrightarrow I\\
X &\longrightarrow X\\
Y &\longrightarrow +\cos\theta Y
\end{aligned}
$$

$RY(\theta)$

$$
\begin{aligned}
I &\longrightarrow I\\
X &\longrightarrow +\cos\theta X\\
&\quad\;\; +\sin\theta Z
\end{aligned}
$$

Propagation

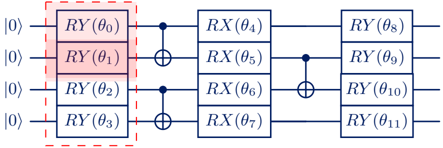


Figure: Four-qubit local entangler ansatz

- **Layer 1:** In this final layer, both $RY(\theta_0)_0$ and $RY(\theta_1)_1$ contribute to the evolution.
  - **Action of $RY(\theta_0)_0$:**

$$
\begin{aligned}
&+\cos\theta_8\cos\theta_4 Z_0\\
&+\cos\theta_8\sin\theta_4\cos\theta_1 Y_0X_1 - \cos\theta_8\sin\theta_4\sin\theta_1 Y_0Z_1\\
&-\sin\theta_8\cos\theta_1 X_0X_1 + \sin\theta_8\sin\theta_1 X_0Z_1\\
&\quad\downarrow\\
&\quad+\cos\theta_8\cos\theta_4\cos\theta_0 Z_0 - \cos\theta_8\cos\theta_4\sin\theta_0 X_0\\
&\quad+\cos\theta_8\sin\theta_4\cos\theta_1 Y_0X_1 - \cos\theta_8\sin\theta_4\sin\theta_1 Y_0Z_1\\
&\quad-\sin\theta_8\cos\theta_1\cos\theta_0 X_0X_1 - \sin\theta_8\cos\theta_1\sin\theta_0 Z_0X_1\\
&\quad+\sin\theta_8\sin\theta_1\cos\theta_0 X_0Z_1 - \sin\theta_8\sin\theta_1\sin\theta_0 Z_0Z_1
\end{aligned}
$$

$Y\otimes I \longrightarrow Y\otimes X$	$I \longrightarrow I$	$I \longrightarrow I$

Propagation

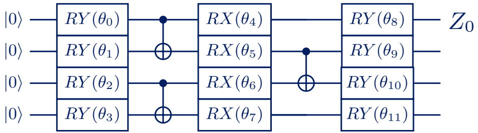


Figure: Four-qubit local entangler ansatz

Propagated observable:

$$
\begin{aligned}
Z_0 \rightarrow &+ \cos\theta_8 \cos\theta_4 \cos\theta_0 Z_0 - \cos\theta_8 \cos\theta_4 \sin\theta_0 X_0 \\
&+ \cos\theta_8 \sin\theta_4 \cos\theta_1 Y_0 X_1 - \cos\theta_8 \sin\theta_4 \sin\theta_1 Y_0 Z_1 \\
&- \sin\theta_8 \cos\theta_1 \cos\theta_0 X_0 X_1 - \sin\theta_8 \cos\theta_1 \sin\theta_0 Z_0 X_1 \\
&+ \sin\theta_8 \sin\theta_1 \cos\theta_0 X_0 Z_1 - \sin\theta_8 \sin\theta_1 \sin\theta_0 Z_0 Z_1
\end{aligned}
$$

saverio.monaco@desy.de

Propagation

Paper example

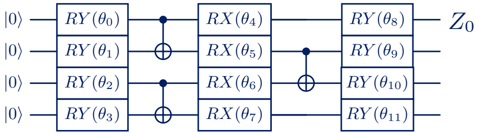


Figure: Four-qubit local entangler ansatz

Propagated observable:

$$
\begin{aligned}
Z_0 \to & + \cos\theta_8 \cos\theta_4 \cos\theta_0 Z_0 - \cos\theta_8 \cos\theta_4 \sin\theta_0 X_0 \\
& + \cos\theta_8 \sin\theta_4 \cos\theta_1 Y_0 X_1 - \cos\theta_8 \sin\theta_4 \sin\theta_1 Y_0 Z_1 \\
& - \sin\theta_8 \cos\theta_1 \cos\theta_0 X_0 X_1 - \sin\theta_8 \cos\theta_1 \sin\theta_0 Z_0 X_1 \\
& + \sin\theta_8 \sin\theta_1 \cos\theta_0 X_0 Z_1 - \sin\theta_8 \sin\theta_1 \sin\theta_0 Z_0 Z_1
\end{aligned}
$$

When computing the expectation value, each propagated observable is sandwiched between ${}^{\otimes N}\langle 0|$ and $|0\rangle^{\otimes N}$. Only Pauli strings composed solely of Z or I operators survive this projection, leading to:

$$\langle 0|U^\dagger Z_0 U|0\rangle(\vec{\theta}) = -\sin\theta_8 \sin\theta_1 \sin\theta_0 + \cos\theta_8 \cos\theta_4 \cos\theta_0$$

Propagation

Non-Clifford Gates:

$$\boxed{RX(\theta)} \equiv \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$\boxed{RY(\theta)} \equiv \begin{pmatrix} +\cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ -\sin(\frac{\theta}{2}) & +\cos(\frac{\theta}{2}) \end{pmatrix}$$

$$\boxed{RZ(\theta)} \equiv \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{+i\theta/2} \end{pmatrix}$$

Propagation causes **branchings** and **frequency** increase.

Propagation

Non-Clifford Gates:

$$\boxed{RX(\theta)} \equiv \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

$$\boxed{RY(\theta)} \equiv \begin{pmatrix} +\cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ -\sin(\frac{\theta}{2}) & +\cos(\frac{\theta}{2}) \end{pmatrix}$$

$$\boxed{RZ(\theta)} \equiv \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{+i\theta/2} \end{pmatrix}$$

Propagation causes **branchings** and **frequency** increase.

Effects:

- **Branching**:
  - $RY(\theta)^\dagger \mathbf{X}(\mathbf{1}) RY(\theta)$ $= \cos(\theta)X(1) + \sin(\theta)Z(1).$

  Branching causes non-commuting words to double.

- **Frequency increase:**:
  - $RY(\theta_2)^\dagger (\boldsymbol{\cos(\theta_1)}\mathbf{X}(\mathbf{1}))RY(\theta_2)$ $= \cos(\theta_2)\cos(\theta_1)X(1)$ $+ \sin(\theta_2)\cos(\theta_1)Z(1).$

  Frequency defined as the number of trigonometric functions of the parameters multiplied together

Propagation

Clifford Gates:

$$\boxed{H} \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Clifford Gates **do not** cause *branchings* or *frequency increase*.

Propagation

Clifford Gates:

$$-\boxed{H}- \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$$

$$\equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Clifford Gates **do not** cause *branchings* or *frequency increase*.

Effects:

- **Weight increase**:
  - $CX^\dagger I(1)Z(2)CX = Z(1)Z(2)$.

  Multi-qubit gates can increase the weight of the Pauli word.

- **Ideal propagation:**:
  - $H^\dagger X(1)H = Z(1)$.

  Ideal propagation does not increase the weight of the Pauli word, it does not cause branching or frequency increase.

Exact propagation

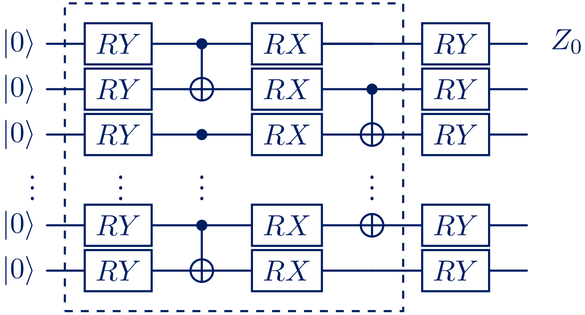
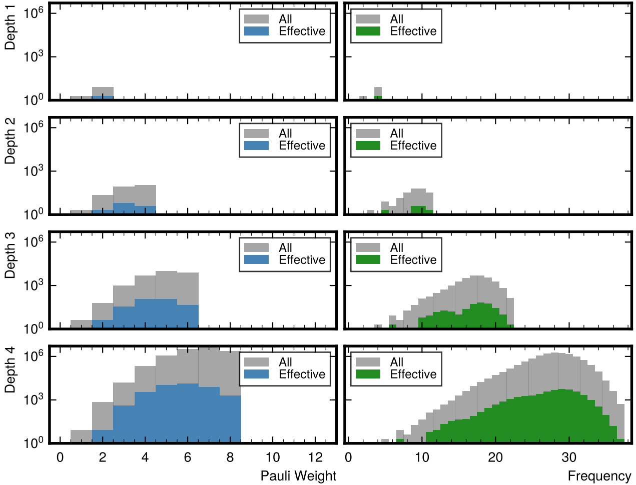


Figure: Local entangler Ansatz

Small circuit to perform exact propagation of observable Z_0.
number of qubits: 12 (rows).

Exact propagation

Small circuit example



Example: $\underbrace{\sin\theta_8 \sin\theta_1 \cos\theta_0}_{\text{Frequency } (3)} \; \underbrace{X_0 Z_1}_{\text{Weight } (2)}$

- Deeper circuits lead to more branching, doubling the number of Pauli words.
- Most Pauli words do not contribute to the expectation value of the observable and are discarded during the trimming process (gray).
- Branching causes a large number of Pauli words to be tracked, making cutoffs necessary.

Cutoffs: Weight cutoff

From **Angrisani et al.**:
Supposing U is the unitary of a *locally scrambling* Ansatz, for $k \geq 0$, we have

$$\mathbb{E}_U \left(\Delta f_U^{(k)}(O) \right) \leq \left(\frac{2}{3} \right)^{k+1} ||O||^2,$$

where $||O||_{\text{Pauli},2} := (2^{-n}\text{Tr}[O^\dagger O])^{1/2}$ is the normalized Hilbert-Schmidt norm.

$$\mathbb{E}_U \left(\Delta f_U^{(k)}(O) \right) := \mathbb{E}_U \left[\left(f_U(O) - f_U^{(k)}(O) \right)^2 \right]$$

where:

- $f_U(O)$ is the excact expectation value of the observable O under the unitary U.
- $f_U^{(k)}(O)$ is the expectation value of the observable O truncated with max weight k.

Cutoffs: Frequency cutoff

Idea: For random parameters, high frequency words contribute less because the product of $\sin$ and $\cos$ will be close to 0.

- $\sin\theta_1 \cos\theta_0$ (high average contribution)
- $\sin\theta_5 \cos\theta_4 \sin\theta_3 \cos\theta_2 \sin\theta_1 \cos\theta_0$ (low average contribution)

For $\nu \geq 0$, we have

$$\mathbb{E}_{p\sim U}\left(\Delta f_{\vec{p}}^{(\nu)}(O)\right) \sim \frac{1}{2^\nu}; \qquad \mathbb{E}_{p\sim U}\left(\Delta f_{\vec{p}}^{(\nu)}(O)\right) := \mathbb{E}_{p\sim U}\left[\left(f_{\vec{p}}(O) - f_{\vec{p}}^{(\nu)}(O)\right)^2\right]$$

where:

- $f_{\vec{p}}(O)$ is the excact expectation value of the observable O with parameters $\vec{p}$ drawn uniformly.
- $f_{\vec{p}}^{(\nu)}(O)$ is the expectation value of the observable O truncated with max frequency ν.

Cutoffs

Cutoffs produce an approximate expectation value of the observable, and training on this approximation may still yield valuable results, though not fully optimal

$\mathcal{L}(\langle O'_{\mathrm{cut}_{\mathrm{low}}}(\theta)\rangle)$ $\mathcal{L}(\langle O'_{\mathrm{cut}_{\mathrm{med.}}}(\theta)\rangle)$ $\mathcal{L}(\langle O'_{\mathrm{cut}_{\mathrm{high}}}(\theta)\rangle)$

QUANTUM
TECHNOLOGY
INITIATIVE

Code Showcase

Examples from the Github code

https://github.com/desyqml/Pauli-Propagator

…pagation of Parametrized Circuits

…o.17989023

… expectation values of a circuit's observables into explicit functions of the …rameters.

…e

…irst import all the necessary libraries:

```
import pennylane as qml # To define the circuit
from pprop.propagator import Propagator # For Pauli Propagation
```

- Define the circuit as a function of the parameters

```
# Function of parameters
def ansatz(params : list[float]):
    qml.RX(params[0], wires=0)
    qml.RX(params[1], wires=1)

    qml.RY(params[2], wires=0)
    qml.RY(params[3], wires=1)
```

Tutorial

1: Import necessary libraries

```python
import pennylane as qml # To define the circuit
from pprop.propagator import Propagator # For Pauli Propagation
```


saverio.monaco@desy.de

Tutorial

2: Define the circuit as a Pennylane function of the parameters, put the observables to propagate as return values

```python
# Function of parameters
def ansatz(params : list[float]):
    qml.RX(params[0], wires=0); qml.RX(params[1], wires=1)
    qml.RY(params[2], wires=0); qml.RY(params[3], wires=1)
    qml.Hadamard(wires = 2)

    qml.CNOT(wires = [0, 1]); qml.CNOT(wires = [1, 2])

    qml.RY(params[4], wires=0)
    qml.RY(params[5], wires=1)
    qml.RY(params[6], wires=2)

    return [qml.expval(qml.PauliZ(qubit)) for qubit in range(1)] + [qml.exp
```

3: Initialize `Propagator` class

```
prop = Propagator(
    ansatz,
    k1 = None, # Cutoff on the Pauli Weight
    k2 = None, # Cutoff on the frequencies
)
```

Tutorial


```
>>> prop
Propagator
  Number of qubits : 3
  Trainable parameters : 7
  Cutoff 1: None | Cutoff 2: None
  Observables [Z(0), X(0) @ X(1) @ X(2), Y(2), -1.0 * (X(0) @ X(1) @ X(2))
0: ──RX──RY──||─╭●─────||──RY─┤  <Z> ╭<X@X@X>      ╭<𝓗>
1: ──RX──RY──||─╰X─╭●──||──RY─┤      ├<X@X@X>      ├<𝓗>
2: ──H───────||────╰X──||──RY─┤      ╰<X@X@X>  <Y> ╰<𝓗>
```

4: Propagate

```
prop.propagate()
Propagating -1.0 * (X(0) @ X(1) @ X(2)) + 13.0 * Z(2): 100%|██████████| 4/4
```

Tutorial

Once propagated, we have access to the functional forms of the observables

```
prop.expression()
```

$$Z0 = -sin(\theta_2) * sin(\theta_3) * sin(\theta_4) * cos(\theta_0) * cos(\theta_1) + cos(\theta_0) * cos(\theta_2) * cos(\theta_4)$$

...

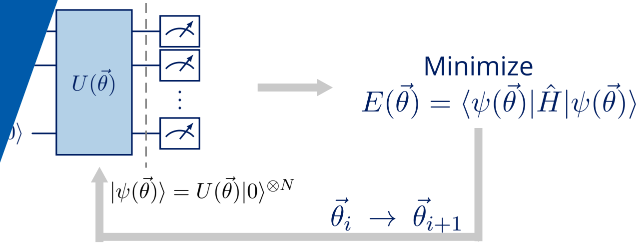
5: Which can be evaluated through

```
random_params = qml.numpy.arange(prop.num_params)
prop_output = prop.eval(random_params)
[ 0.32448207 -0.5280619   0.          4.16046337]
```

QUANTUM TECHNOLOGY INITIATIVE

Applications

Example application of Symbolic Propagation: Variational Quantum Eigensolver



Application: VQE

Finding the ground state of a spin model

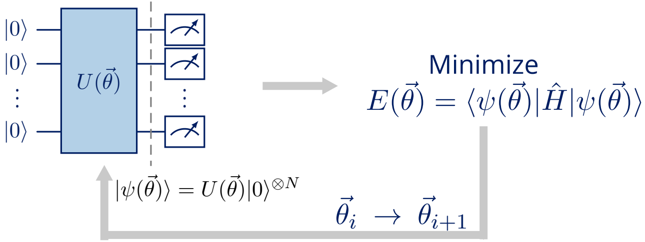
Goal: Get the ground-state of an Hamiltonian H.

Key idea:

- Prepare a parameterized quantum state $|\psi(\boldsymbol{\theta})\rangle$ with a variational circuit.
- Measure the expectation value

$$E(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$$

- Use a classical optimizer to adjust $\boldsymbol{\theta}$ and minimize $E(\boldsymbol{\theta})$.



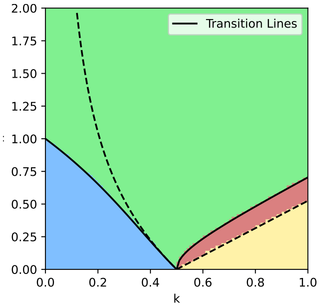
Application: VQE

Finding the ground state of a spin model

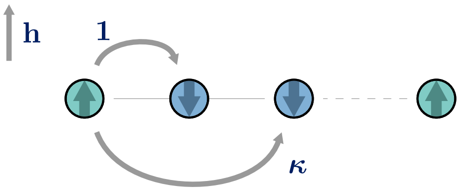
Hamiltonian:

$$H(\kappa, h) = -J \sum_{i=1}^{N} \Big(X(i)X(i+1) - \kappa\, X(i)X(i+2) + h\, Z(i) \Big),$$

Phase diagram:



Spin system:



Number of spins: $N = 18$

Application: VQE

Finding the ground state of a spin model

1. Split the Hamiltonian:

$$H(\kappa, h) = -J \sum_{i=1}^{N} \Big(X(i)X(i+1) - \kappa\, X(i)X(i+2) + h\, Z(i) \Big),$$

Application: VQE

Finding the ground state of a spin model

1. Split the Hamiltonian:

$$H(\kappa, h) = -J \sum_{i=1}^{N} \Big(X(i)X(i+1) - \kappa\, X(i)X(i+2) + h\, Z(i) \Big),$$

$$H(\kappa, h) = \underbrace{-J \sum_{i=1}^{N} \Big(X(i)X(i+1) \Big)}_{O_1} + \underbrace{J\kappa \sum_{i=1}^{N} \Big(X(i)X(i+2) \Big)}_{O_2} \underbrace{-Jh \sum_{i=1}^{N} \Big(Z(i) \Big)}_{O_3}.,$$

Application: VQE

Finding the ground state of a spin model

$$H(\kappa, h) = \underbrace{-J\sum_{i=1}^{N}\Big(X(i)X(i+1)\Big)}_{O_1} \underbrace{+J\kappa\sum_{i=1}^{N}\Big(X(i)X(i+2)\Big)}_{O_2} \underbrace{-Jh\sum_{i=1}^{N}\Big(Z(i)\Big)}_{O_3}.,$$

2. Propagate the Observables $O_1, O_2, O_3 \rightarrow O_1', O_2', O_3'$:

Application: VQE

Finding the ground state of a spin model

$$H(\kappa, h) = \underbrace{-J\sum_{i=1}^{N}\Big(X(i)X(i+1)\Big)}_{O_1} + \underbrace{J\kappa\sum_{i=1}^{N}\Big(X(i)X(i+2)\Big)}_{O_2} - \underbrace{Jh\sum_{i=1}^{N}\Big(Z(i)\Big)}_{O_3}.,$$

2. Propagate the Observables $O_1, O_2, O_3 \to O_1', O_2', O_3'$:
3. Expectation value of the Hamiltonian is given by:

$$\langle 0|U^\dagger(\vec{\theta})|H(\kappa, h)|U(\vec{\theta})|0\rangle = -J\langle O_1'(\vec{\theta})\rangle + J\kappa\langle O_2'(\vec{\theta})\rangle - Jh\langle O_3'(\vec{\theta})\rangle.$$

Application: VQE

Finding the ground state of a spin model

$$H(\kappa,h) = \underbrace{-J\sum_{i=1}^{N}\Big(X(i)X(i+1)\Big)}_{O_1} \underbrace{+J\kappa\sum_{i=1}^{N}\Big(X(i)X(i+2)\Big)}_{O_2} \underbrace{-Jh\sum_{i=1}^{N}\Big(Z(i)\Big)}_{O_3}.,$$

2. Propagate the Observables $O_1, O_2, O_3 \to O'_1, O'_2, O'_3$:
3. Expectation value of the Hamiltonian is given by:

$$\langle 0|U^\dagger(\vec{\theta})|H(\kappa,h)|U(\vec{\theta})|0\rangle = -J\langle O'_1(\vec{\theta})\rangle + J\kappa\langle O'_2(\vec{\theta})\rangle - Jh\langle O'_3(\vec{\theta})\rangle.$$

4. Minimize through gradient descent for the desired (κ, h)

Application: VQE

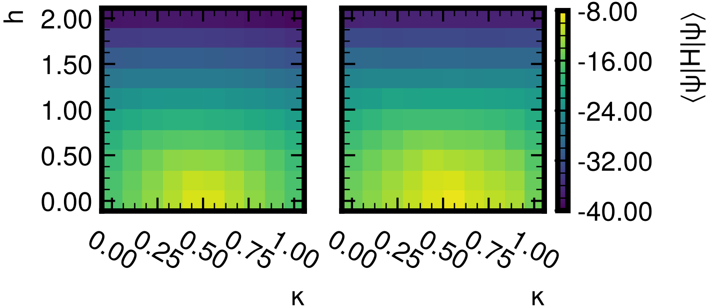


Figure: Energies of the 18-spin ANNNI model at various values of κ and h, obtained using exact diagonalization (left) and by minimizing the function of the propagated Hamiltonian via gradient descent (right).

Application: VQE

Question: How do the two cutoffs affect the final training performance?

- Higher cutoffs $\rightarrow$ improved training performance, although they require more computational resources for Pauli propagation.
- The performance, however, saturates already at relatively low cutoff values; higher-order terms do not significantly improve the training.

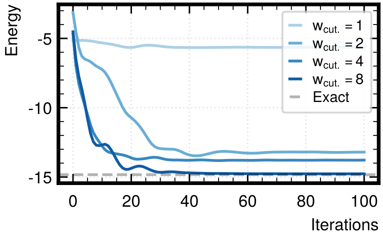


Figure: VQE training with fixed $\nu_{\text{cut}} = 20$ and varying $w_{\text{cut}} \in \{1, 2, 4, 8\}$.

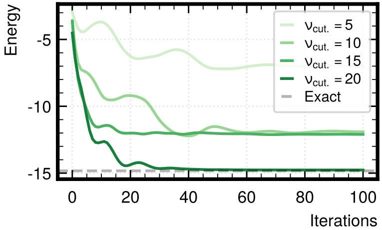


Figure: VQE training with fixed $w_{\text{cut}} = 8$ and varying $\nu_{\text{cut}} \in \{5, 10, 15, 20\}$.

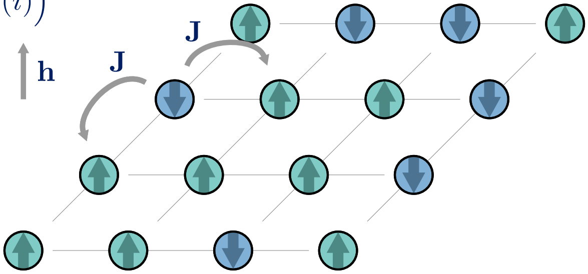
Application: Larger VQE

2D Square lattice (8x8) with nearest neighbour interactions

Number of qubits: 64

Hamiltonian:

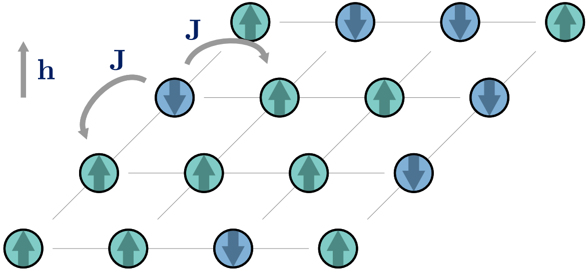
$$H(J,h) = \frac{1}{N}\Big(-J\sum_{\langle i,j\rangle} Z(i)Z(j) - h\sum_{i} X(i)\Big)$$



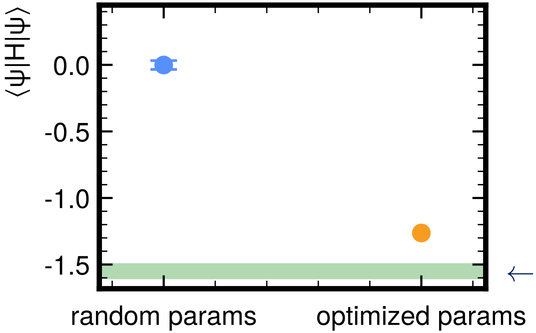
Application: Larger VQE

Warm start **procedure:**

1. Define circuit
2. Propagate Hamiltonian through the circuit
3. Assign spin system's parameters $(J = h = 1)$
4. Minimize $\langle H'(J = 1, h = 1, \vec{\theta})\rangle$ and obtain optimal parameters
5. Deploy on quantum



Application: Larger VQE

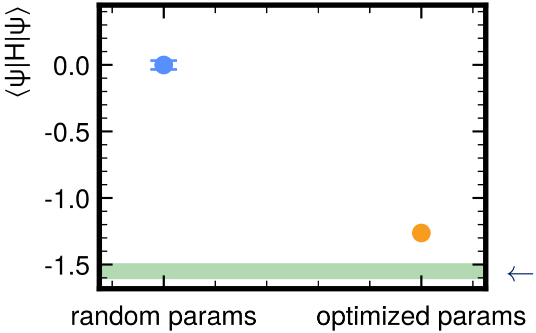


$\leftarrow$ Actual ground state energy

Info:

- Deployed on *ibm_fez*
- Ansatz: see backup
- Number of shots: 2048
- Random parameters:
  - 30 runs
  - $p \sim \mathcal{N}(0, 1)$

Application: Larger VQE



$\leftarrow$ Actual ground state energy, i guess

Info:

- Deployed on *ibm_fez*
- Ansatz: see backup
- Number of shots: 2048
- Random parameters:
  - 30 runs
  - $p \sim \mathcal{N}(0, 1)$

saverio.monaco@desy.de

Thank you!

ENGAGE has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement No. 101034267.

Backup

Backup slides

Simulating noisy circuits

From "Simulating quantum circuits with arbitrary local noise using Pauli Propagation" [7]

Model of the noise:

Circuits represented as a sequence of layers

$$C = C_L C_{L-1} \cdots C_1$$

where each layer C_j consists of a unitary U_j followed by a local (single-qubit) noise channel $N^{\otimes n}$.
(This includes depolarizing-like noise, dephasing-like noise, amplitude damping.)

Computational and accuracy guarantees:

Polynomial complexity with error bounds that are inversely polynomially small.

Locally-scrambling circuits

For the truncation on the Pauli weight, we require an L-layered circuit to be of the form [2]

$$U = U_L U_{L-1} \cdots U_1$$

where each layer U_j is

- sampled independently from a *locally scrambling* distribution[1]

> That is, independent layers where the distribution of gates ensures that any local bias or structure is erased by random rotation.

- reasonably shallow.

[1] The condition on local scrambling might be relaxed without significantly compromising accuracy.

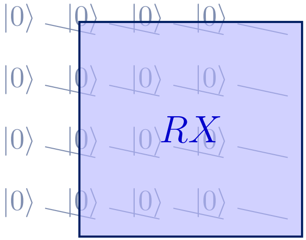
Application: Larger VQE

Ansatz: 2D brickwork with alternating entanglers

$$\begin{array}{cccc} |0\rangle & |0\rangle & |0\rangle & |0\rangle \\ |0\rangle & |0\rangle & |0\rangle & |0\rangle \\ |0\rangle & |0\rangle & |0\rangle & |0\rangle \\ |0\rangle & |0\rangle & |0\rangle & |0\rangle \end{array}$$

Application: Larger VQE

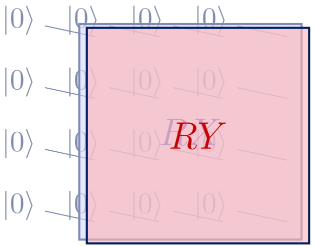
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)

Application: Larger VQE

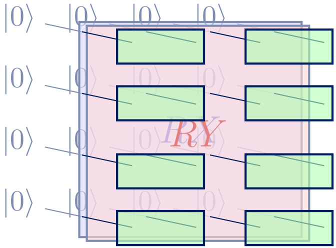
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)

Application: Larger VQE

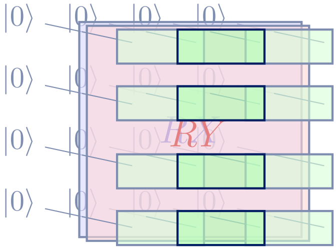
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)

Application: Larger VQE

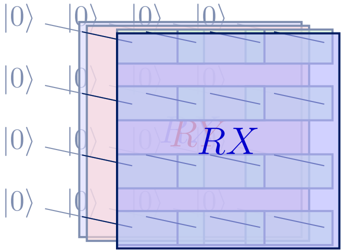
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)

Application: Larger VQE

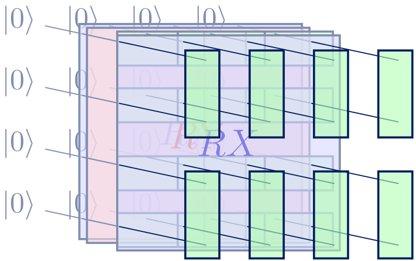
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)
- RX on all qubits (different params.)

Application: Larger VQE

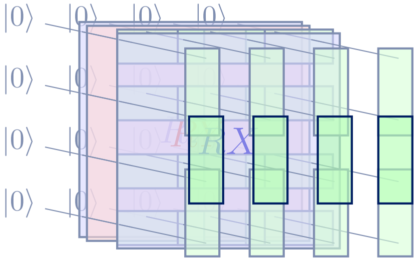
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)
- RX on all qubits (different params.)
- Vertical CNOT (offset 0)

Application: Larger VQE

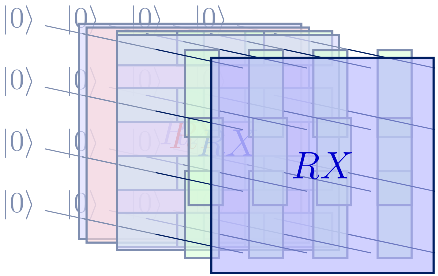
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)
- RX on all qubits (different params.)
- Vertical CNOT (offset 0)
- Vertical CNOT (offset 1)

Application: Larger VQE

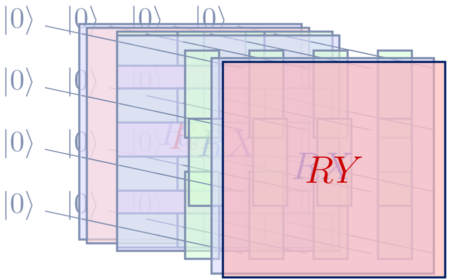
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)
- RX on all qubits (different params.)
- Vertical CNOT (offset 0)
- Vertical CNOT (offset 1)
- RX on all qubits (different params.)

Application: Larger VQE

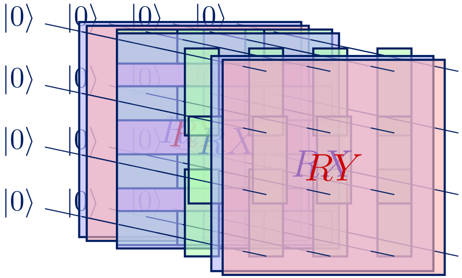
Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)
- RX on all qubits (different params.)
- Vertical CNOT (offset 0)
- Vertical CNOT (offset 1)
- RX on all qubits (different params.)
- RY on all qubits (different params.)

Application: Larger VQE

Ansatz: 2D brickwork with alternating entanglers



- RX on all qubits (different params.)
- RY on all qubits (different params.)
- Horizontal CNOT (offset 0)
- Horizontal CNOT (offset 1)
- RX on all qubits (different params.)
- Vertical CNOT (offset 0)
- Vertical CNOT (offset 1)
- RX on all qubits (different params.)
- RY on all qubits (different params.)

QUANTUM TECHNOLOGY INITIATIVE

References I

[1] Daniel Gottesman.
The heisenberg representation of quantum computers, 1998.

[2] Armando Angrisani, Alexander Schmidhuber, Manuel S. Rudolph, M. Cerezo, Zoë Holmes, and Hsin-Yuan Huang.
Classically estimating observables of noiseless quantum circuits, 2025.

[3] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien.
A variational eigenvalue solver on a quantum processor, 2013.

[4] Korbinian Kottmann, Friederike Metz, Joana Fraxanet, and Niccolo Baldelli.
Variational quantum anomaly detection: Unsupervised mapping of phase diagrams on a physical quantum computer, 2022.

QUANTUM TECHNOLOGY INITIATIVE

References II

[5] Erik Recio-Armengol and Joseph Bowles.
Iqpopt: Fast optimization of instantaneous quantum polynomial circuits in jax, 2025.

[6] Manuel S. Rudolph, Sacha Lerch, Supanut Thanasilp, Oriel Kiss, Sofia Vallecorsa, Michele Grossi, and Zoë Holmes.
Trainability barriers and opportunities in quantum generative modeling, 2023.

[7] Armando Angrisani, Antonio A. Mele, Manuel S. Rudolph, M. Cerezo, and Zoë Holmes.
Simulating quantum circuits with arbitrary local noise using pauli propagation, 2025.