Xanadu Research Presents

HERE COMES THE SU(N)!

Research paper by:

Roeland Wiersema
Dylan Lewis
David Wierichs
Juan Carrasquilla
Nathan Killoran

Abstract

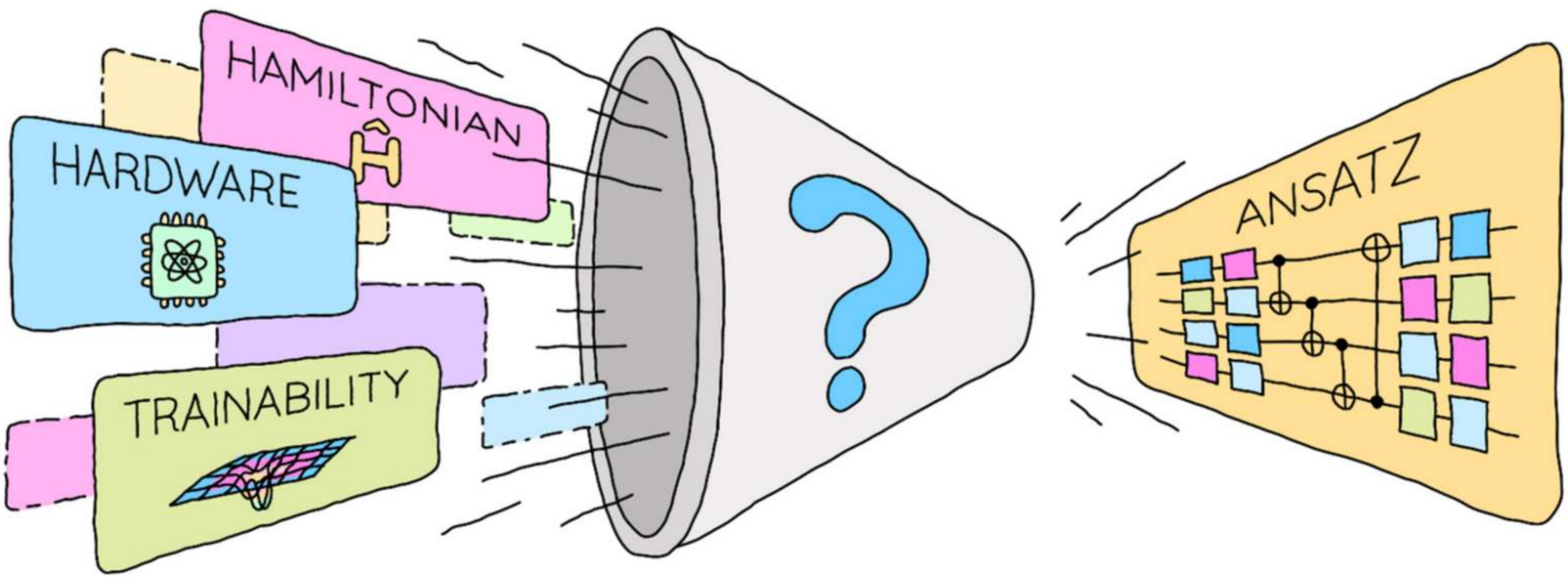
Variational quantum algorithms use non-convex optimization methods to find the optimal parameters for a parametrized quantum circuit in order to solve a computational problem. The choice of the circuit ansatz, which consists of parameterized gates, is crucial to the success of these algorithms. Here, we propose a gate which fully parameterizes the special unitary group SU(N). This gate is generated by a sum of non-commuting operators, and we provide a method for calculating its gradient on quantum hardware. In addition, we provide a theorem for the computational complexity of calculating these gradients by using results from Lie algebra theory. In doing so, we further generalize previous parameter-shift methods. We show that the proposed gate and its optimization satisfy the quantum speed limit, resulting in geodesics on the unitary group. Finally, we give numerical evidence to support the feasibility of our approach and show the advantage of our gate over a standard gate decomposition scheme. In doing so, we show that not only the expressibility of an ansatz matters, but also how it's explicitly parameterized.

This paper introduces a new style of multivariate gate and shows how to differentiate it on quantum hardware.

Join us in exploring Lie groups, geometry of circuit, and a new parameter-shift rule.

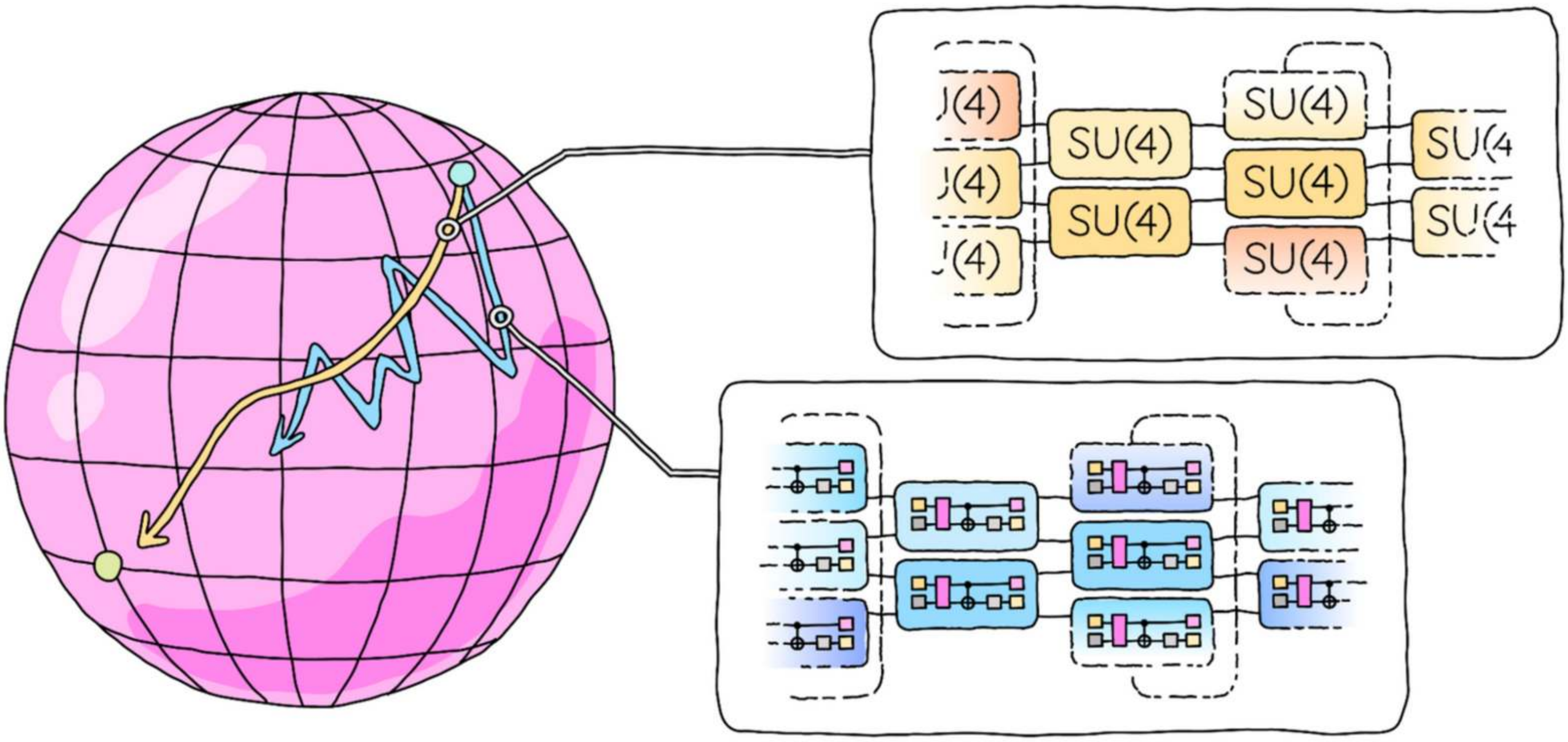
There are many circuit ansätze in variational quantum computing, but which one will get us a time-efficient circuit and good trainability?

Roeland Wiersema, Dylan Lewis, David Wierichs, Juan Carrasquilla, and Nathan Killoran tackle this question using multivariate quantum gates.



A lot is still unknown about the connection between circuit structures and the resulting properties.

This work explores gate speed limits, biases in gradient-based training as well as trainability in practice.



For this new SU(N) gate, we parametrize the group of all local operations via the (canonical) coordinates of the Lie algebra su(N).

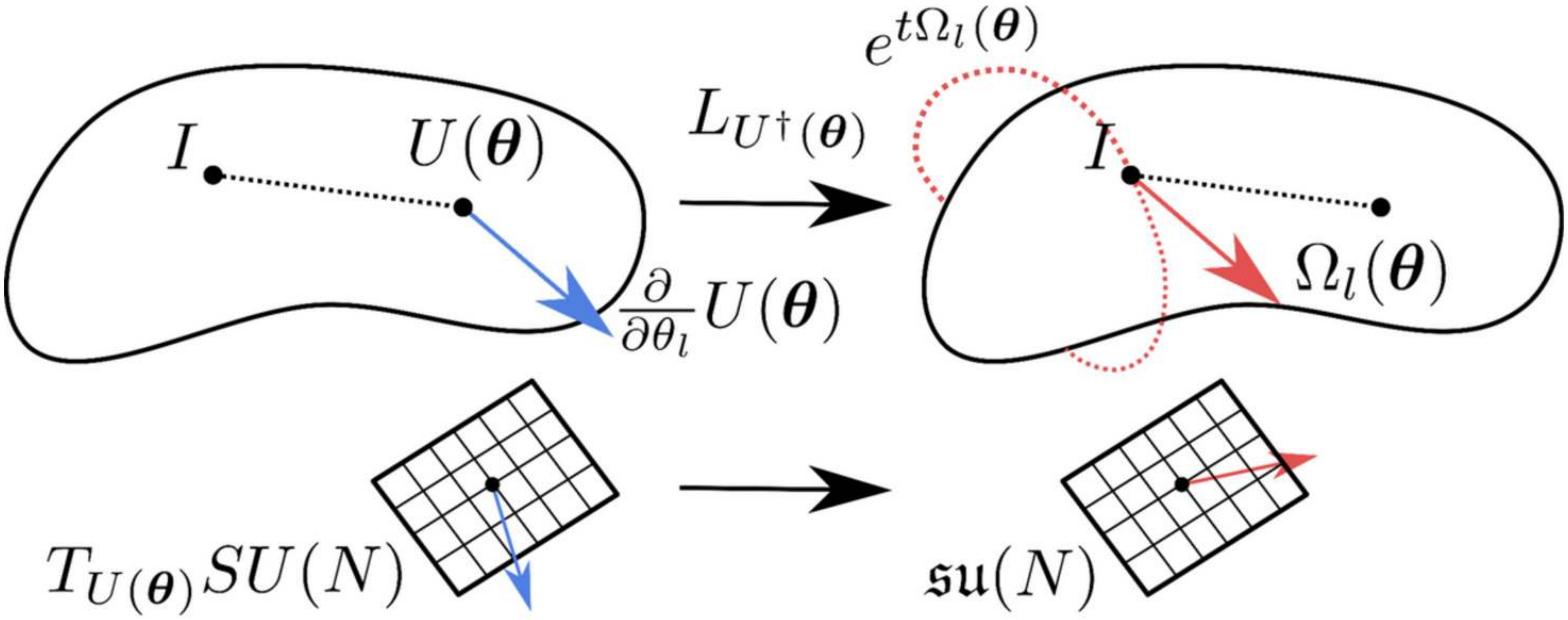
The exponential map allows us to map these coordinates of the algebra to the group:

$$U(\boldsymbol{\theta}) = \exp\left\{\sum_{m=1}^{d} \theta_m G_m\right\}$$

To use a multi-parameter gate in gradient-based training of variational quantum circuits, we need to compute, well, the gradient. Previously the stochastic parameter-shift rule was proposed or we could use good ol' finite differences.

Instead, one can consider the effective gate generator Ω for each parameter and compute the gradient by applying a generalized parameter-shift rule to it.

Mathematically speaking, the effective generator is the tangent vector of the SU(N) gate for the chosen parameter.



The effective generator can be found with automatic differentiation for gates that do not act on too many qubits.

$$\frac{\partial}{\partial x_l} U_{nm}(\boldsymbol{x}) = \partial_{x_l} \mathfrak{Re}[U_{nm}(\boldsymbol{x})] + i \partial_{x_l} \mathfrak{Im}[U_{nm}(\boldsymbol{x})]$$

$$\Omega_l(\boldsymbol{\theta}) = U^\dagger(\boldsymbol{\theta}) \left(\frac{\partial}{\partial x_l} U(\boldsymbol{x}) \bigg|_{\boldsymbol{\theta}} \right)$$

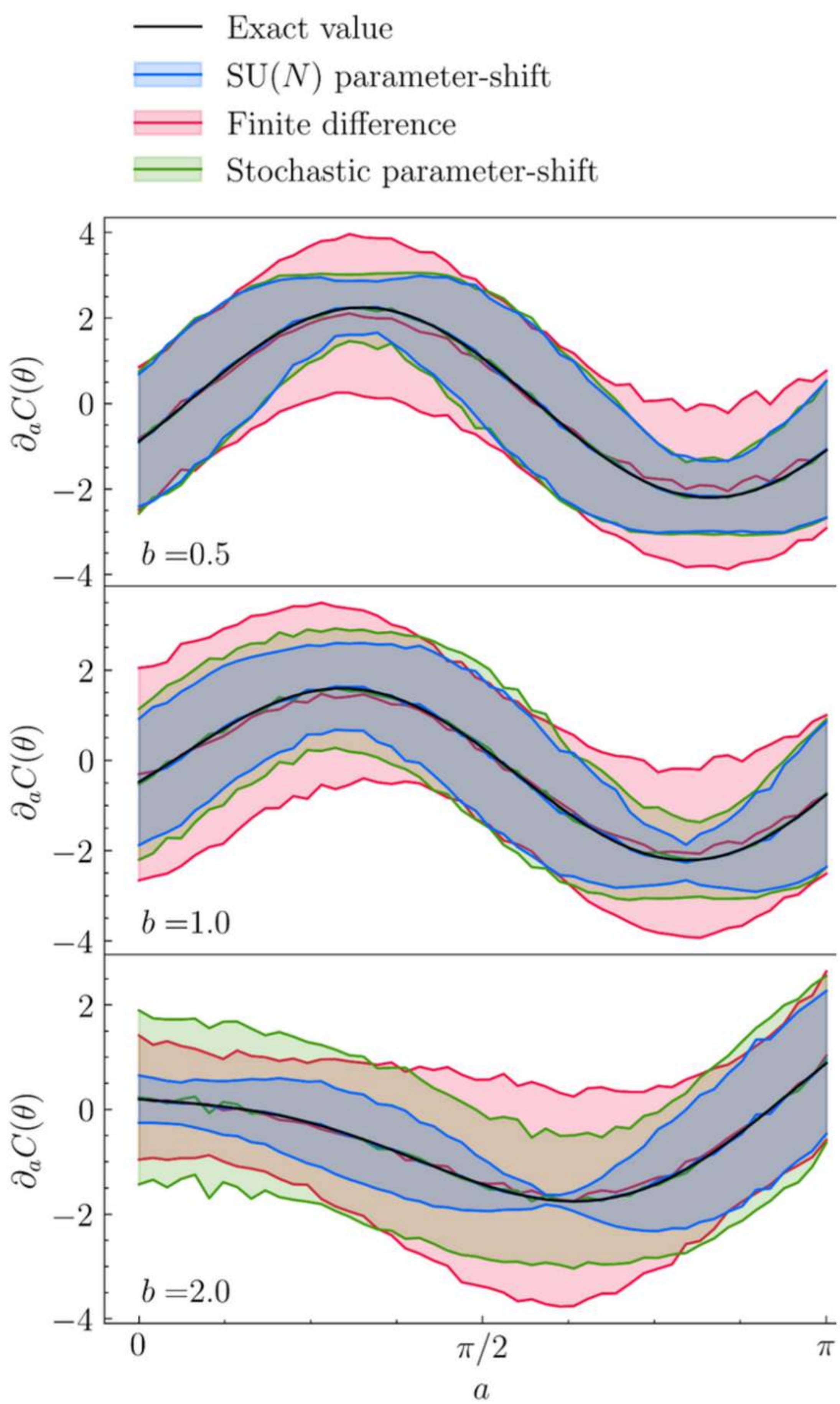
The applicable generalized shift rules can be found e.g. in https://arxiv.org/abs/2108.01218 or https://quantum-journal.org/papers/q-2022-03-30-677.

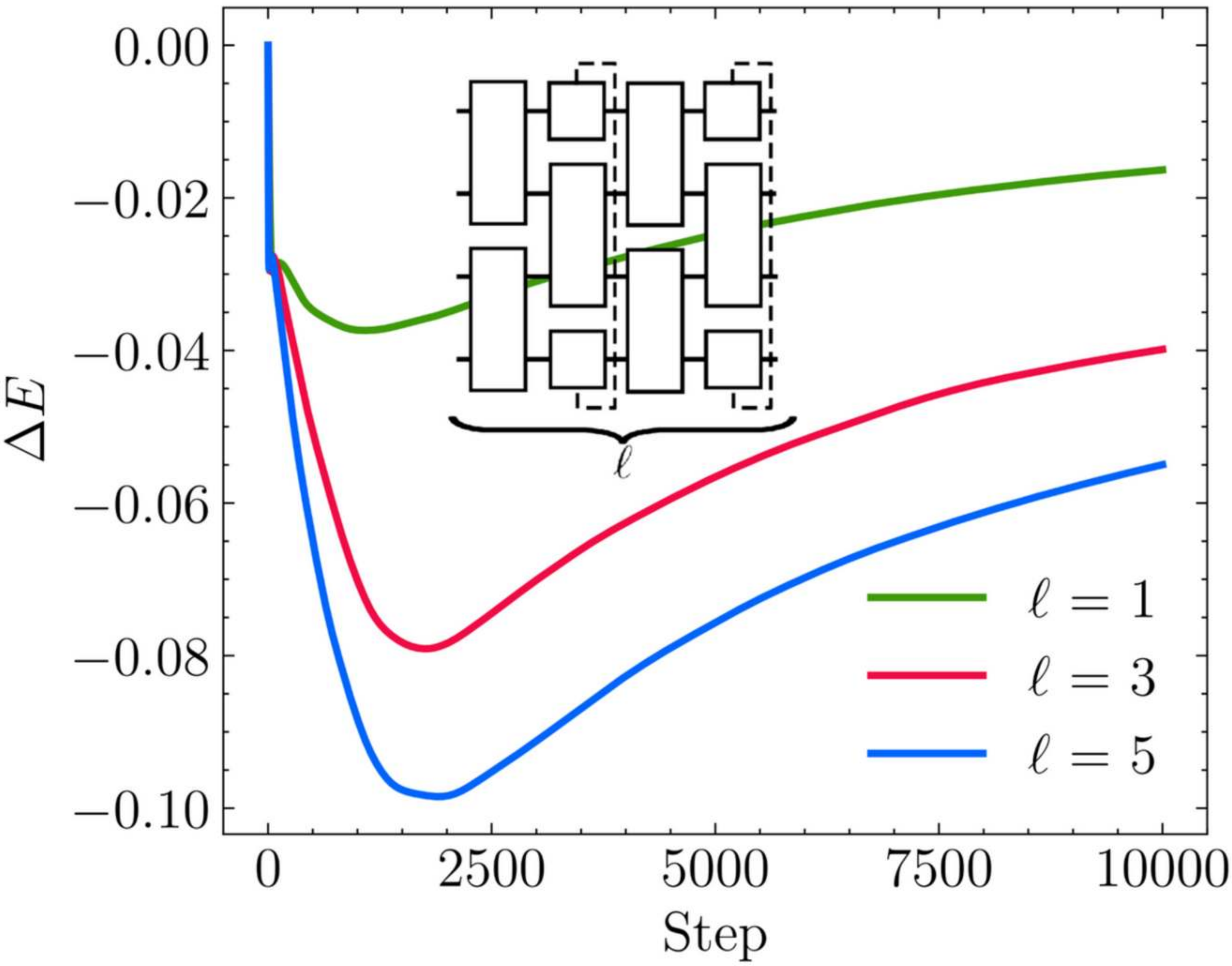
Alternatively, one can express all generators Ω in the Pauli basis, compute the two-term shift rule for the basis elements, and combine them into the gradient.

$$\Omega_\ell(\boldsymbol{\theta}) = \sum_{m=1}^{d} \omega_{\ell m}(\boldsymbol{\theta}) G_m$$

In the paper, finite differences, the stochastic shift rule and the new effective-generator shift rule are compared using their mean-squared error.

For a single-qubit unitary, Fig.4 shows the estimated single-shot error across a range of parameters, coded in PennyLane.



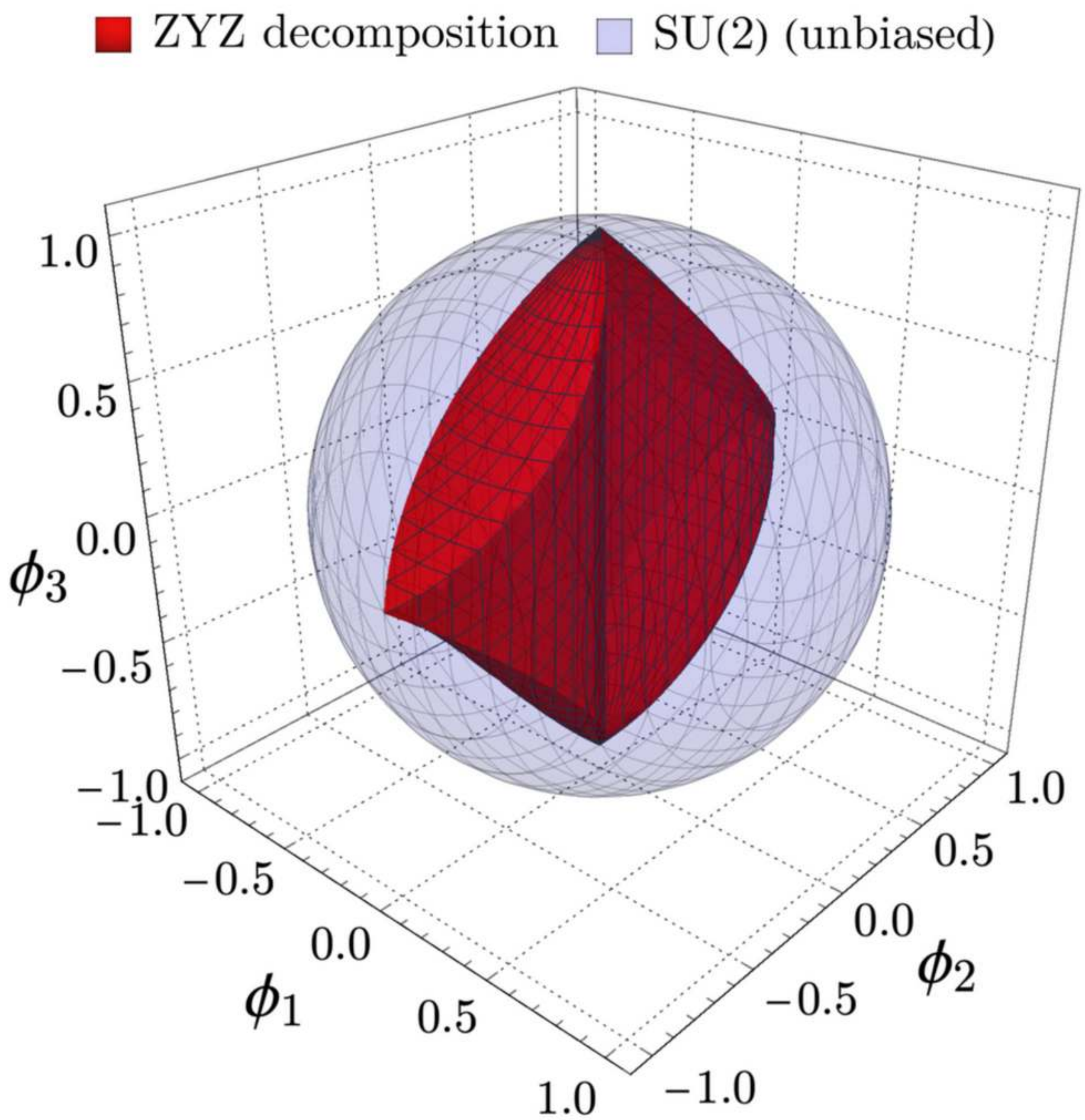


With a differentiation method chosen, the SU(N) gate can be used in an optimization workflow!

In the paper, the energy of a random Hamiltonian is optimized for up to 10 qubits, showing improved optimization behaviour compared to other fully-expressive circuit blocks.

This numerical result is supported by an analysis of the bias in the gradient direction for circuit building blocks made of elementary gates.

This bias is not present with a global SU(N) gate and the numerics show that avoiding this bias locally is useful as well.



But there is more!

Theorem 1 shows that the (global) SU(N) gate realizes the fastest possible time evolution for given initial and final states, and we compute the additional evolution time required by a simple decomposition into multiple gates.

$$\Delta t = t_d - \arccos\big(\cos(t_1)\cos(t_2) \\ - \boldsymbol{\phi}^{(1)} \cdot \boldsymbol{\phi}^{(2)} \sin(t_1)\sin(t_2)\big) \geq 0.$$

The SU(N) gate implements a time-efficient evolution and can be trained better than decompositions. But at what cost?

So far, we've considered the exponent of the SU(N) gate to be dense in the Pauli basis, causing exponential cost in the number of qubits of the gate.

To avoid this, we may parametrize a gate using just a few out of the many possible Pauli words.

This parametrizes a subgroup of the global special unitary group, and the gate may have a subalgebra as dynamical Lie algebra.

In this case, all processing steps and the shift rule itself will scale with the Lie algebra rather than Hilbert space:

Theorem 2 shows that the number of frequencies generated by Ω—and thus the differentiation cost—is bounded by the dynamical Lie algebra.

Theorem 2. *The number of unique spectral gaps R of $\Omega_l(\boldsymbol{\theta})$ is upper bounded by the number of roots $|\Phi|$ of any maximal semi-simple DLA,*

$$R \leq |\Phi|/2. \tag{19}$$

Bonus content in the paper:
a connection to Riemannian gradients
and more Lie theory.
Read details at
https://arxiv.org/abs/2303.11355,

dive into the PennyLane implementation
of the SU(N) gate at
https://docs.pennylane.ai/en/stable/code/api/pennylane.SpecialUnitary.html

or explore the paper repository at
https://github.com/dwierichs/Here-comes-the-SUN

XANADU

xanadu.ai

XanaduAI | XanaduAI