Variational Fast Forwarding (VFF) for Quantum Simulations[1]

- Hamiltonian Simulation is one of the most anticipated application of quantum computing. However, known algorithm with optimal asymptotic scaling will only be relevant on fault-tolerant quantum devices, as they are limited by the decoherence time of the hardware.

- Variational Fast Forwarding uses short depth circuits even for long simulation time, suitable in NISQ - era. It aims at approximating a diagonal representation of the Hamiltonian by optimizing an ansatz represented as a quantum circuit to approximate the eigenstates and the eigenvalues of the Hamiltonian.

- This enables a fast forward time evolution, as we can efficiently exponentiate the Hamiltonian in the following way: $e^{-iHt} = (e^{-iH\Delta t})^N \approx W(\theta)D(\gamma,\Delta t)^N W(\theta)^\dagger$. This permits to forward beyond the decohenrence time.

Can we find better ansatz for W?
Can we use perturbative layerweise training to tackle complexe hamiltonians?
Is this method competitive against Trotter evolution or exact diagonalisation?

[1] Cristina Cîrstoiu et al., *Variational fast forwarding for quantum simulation beyond the coherence time,* NpJ Quantum Information 6:82 (2020).

Technical details:

Goal: build a qiskit package of the VFF algorithm, which would permits the user to compute the time evolution for any Ising-type Hamiltonian beyond the decoherence time. We also hope to find better ansatz and to test a layerweise training strategy while perturabting the Hamiltonian[2].

- Ansatz: trainable feature map, build on the inherent symmetry of the Hamiltonian
- Layerweise training: start with a simple Hamiltonian and train W. Then add a term to the Hamiltonian and layers to W initialiased at the identity and train W again.

We will benchmark our results with classical strategy and Trotter evolution, both on simulator and harware.

Impact: Study Hamiltonian evolution more efficiently in the NISQ-era.

- Permits time evolution beyond the decoherence time.
- Uses less computational ressources than exact method, while being accurate.

[2] Andrea Skolik et al., *Layerwise learning for quantum neural networks,* Quantum Machine Intelligence 3 (2021).

Implementation plan:

Benchmark

Trotter

Exact diagonalisation

Build Hamiltonian

Build Optimization Strategy

Build Ansatz

Trainable feature map, symmetry

Experiment

Layerweise training

Time evolution of Observable

Team:

- Georgios Ts (Software Dv, Computer science, QML)
- Oriel Kiss (Physics, QML, Qinfo)
- Atithi Acharya (Computer science, QML, Physics)

Material:

- Qiskit machine learning
- Qiskit Aqua Operator
- Papers listed before