Synergy between Quantum Circuits and Tensor Networks

Abhishek Abhishek, Daniel Kong, Harmeeta Dahiya, Mushahid Khan

Synergy Between Quantum Circuits and Tensor Networks: Short-cutting the Race to Practical Quantum Advantage

Manuel S. Rudolph,[1] Jacob Miller,[1] Jing Chen,[2] Atithi Acharya,[2,3] and Alejandro Perdomo-Ortiz[1,*]

Decomposition of Matrix Product States into Shallow Quantum Circuits

Manuel S. Rudolph,[1] Jing Chen,[2] Jacob Mille

Unsupervised Generative Modeling Using Matrix Product States

ng Fan,[2,4] Lei Wang,[2,4,*] and Pan Zhang[3,†]

g University, Beijing 100871, China

cademy of Sciences, Beijing 100190, China

l Physics, Institute of Theoretical Physics,

Sciences, Beijing 100190, China

in Topological Quantum Computation,

my of Sciences, Beijing 100190, China

ARTICLE OPEN

A generative modeling approach for benchmarking and training shallow quantum circuits

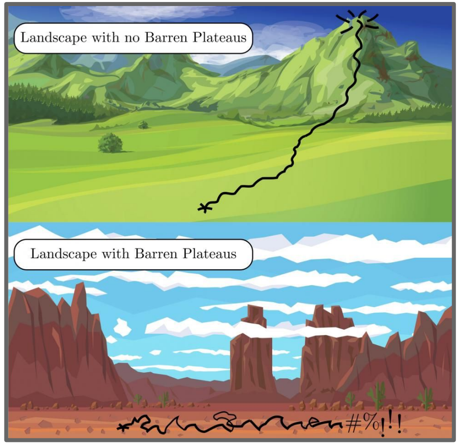
Marcello Benedetti[1,2], Delfina Garcia-Pintos[3], Oscar Perdomo[3,4,5], Vicente Leyton-Ortega[3,4], Yunseong Nam[6] and Alejandro Perdomo-Ortiz[1,3,4,7,8]

Main idea

❌ Generic initialization schemes for parameterized quantum circuits (PQCs)

😟 **barren plateaus**

😟 **local minima**



✅ Use classical resources to find good task-specific initializations and further optimize using quantum hardware

🤝 **classical + quantum** 🥳

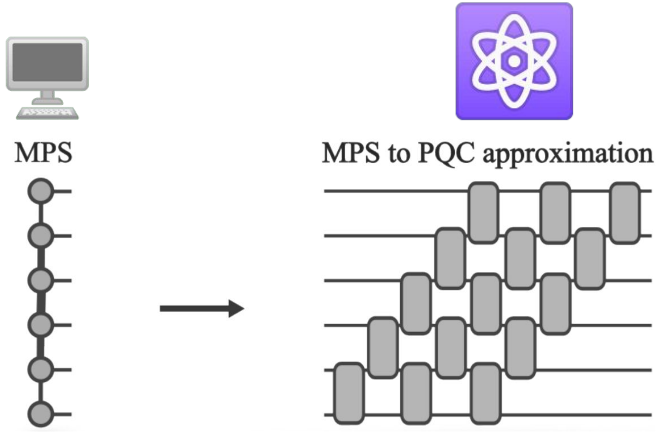
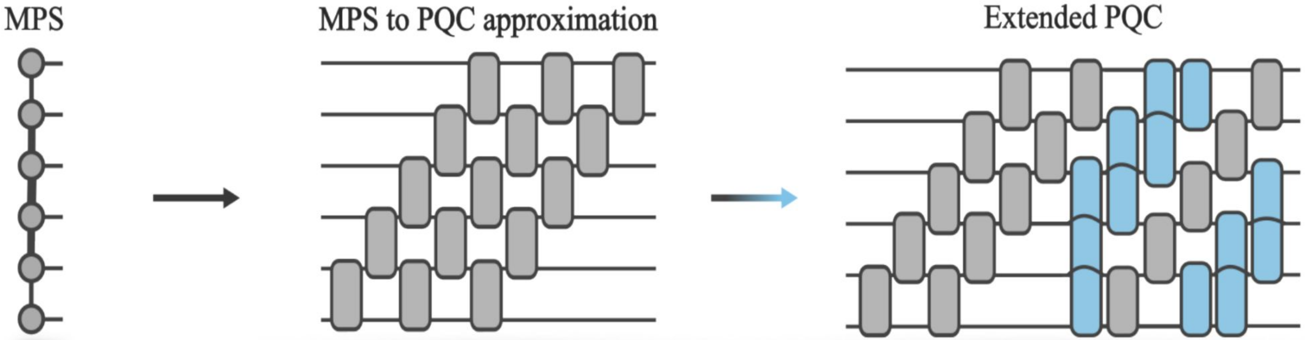


Image credit: https://discover.lanl.gov/news/0319-barren-plateaus/

Image credit: Fig 1. https://arxiv.org/abs/2208.13673

Overview of Approach

- Train a Tensor Network (TN) called Matrix Product State (MPS) as a generative model on the bars and stripes dataset
- Transfer the trained MPS to a MPS-PQC circuit
- Extend and train the MPS-PQC circuit with additional gates
- Generate bars and stripes images



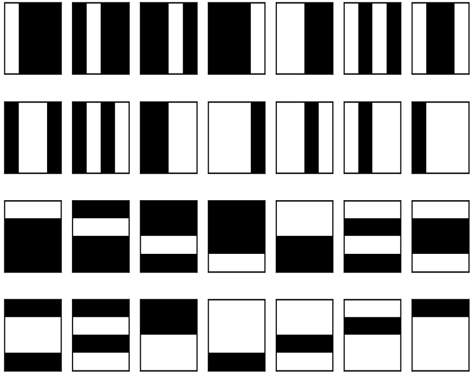
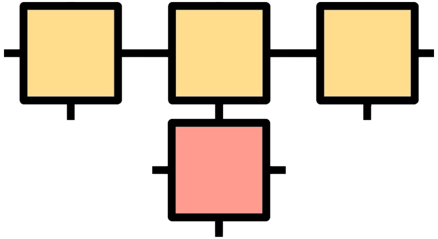
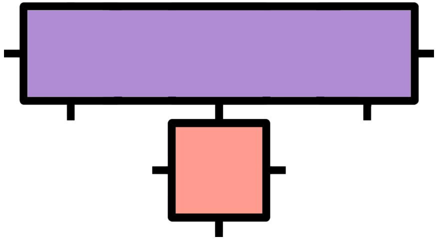


Image Credit: https://arxiv.org/abs/2208.13673

Train MPS

- TNs are linear algebraic models for classically simulating complex many-body quantum systems, that have been employed recently as machine learning models
- No. of nodes in a TN = No. of qubits in the quantum computer
- the topology of the N-node graph determines the forms of entanglement
- MPS are computationally tractable TN models whose cores are connected along a line graph





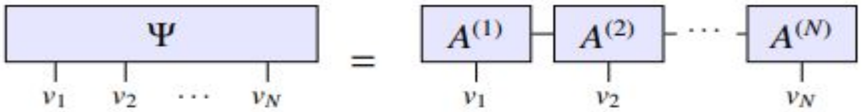


Image Credit: https://arxiv.org/pdf/1709.01662.pdf

MPS to PQC

How do we map a Matrix Product State (MPS) tensor network to a parameterized quantum circuit ?

1. The parameters of the model are stored in the tensor cores
2. These 3-d tensors are clearly not **unitaries** but can be converted to one

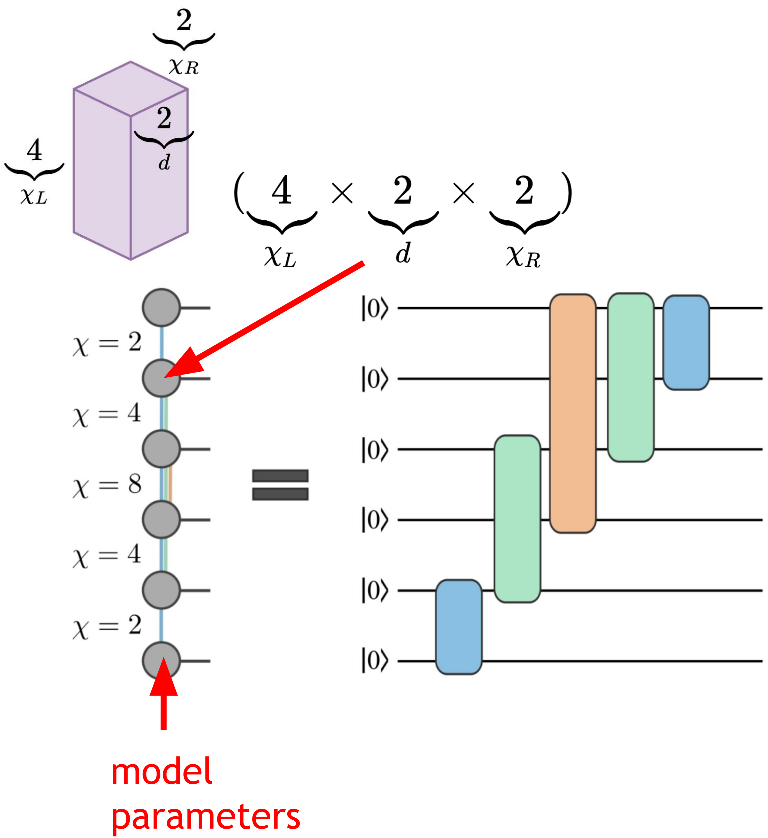
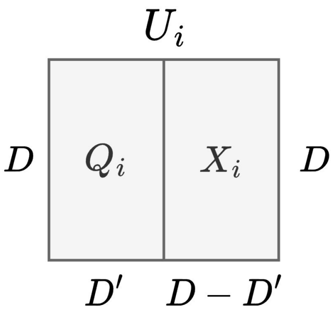


Image credits: https://arxiv.org/abs/2209.00595, https://muellergroup.lassp.cornell.edu/bt2020chap4.pdf

MPS to PQC

Two key steps:

1. Left-normalization of the MPS via iterated QR factorizations
2. Extend the left-isometries in the MPS to multi-qubit unitaries



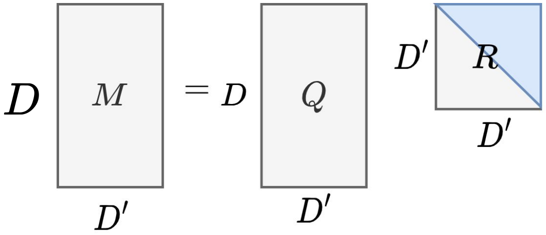
$$X_i^\dagger X_i = I_{D-D' \times D-D'}$$

$$Q_i^\dagger X_i = 0$$

$$U_i = [\, Q_i \;\; X_i \,]$$

$$U_i^\dagger U_i = U_i U_i^\dagger = I_{D\times D}$$

QR-factorization

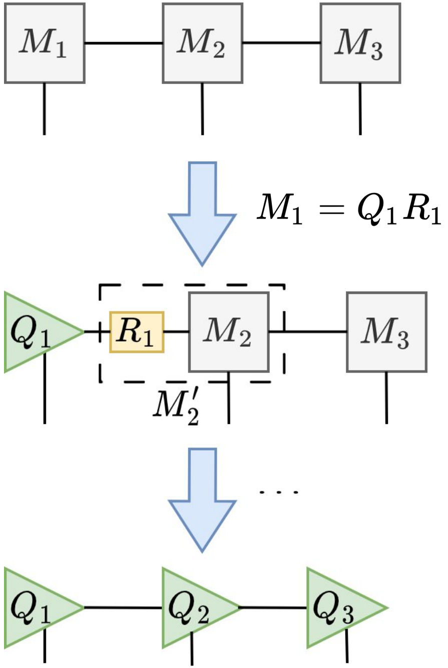


$$Q^\dagger Q = I_{D' \times D'}$$

$$QQ^\dagger \neq I_{D\times D}$$

Left-isometry

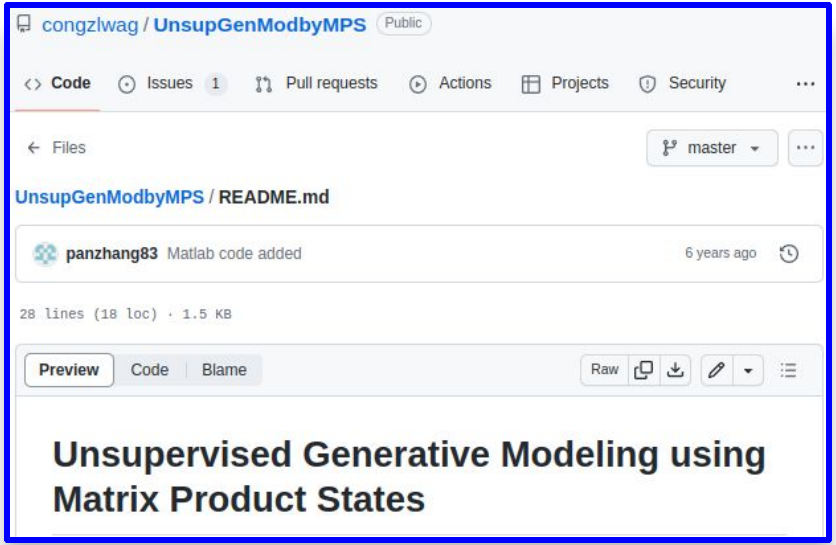
Left-normalization



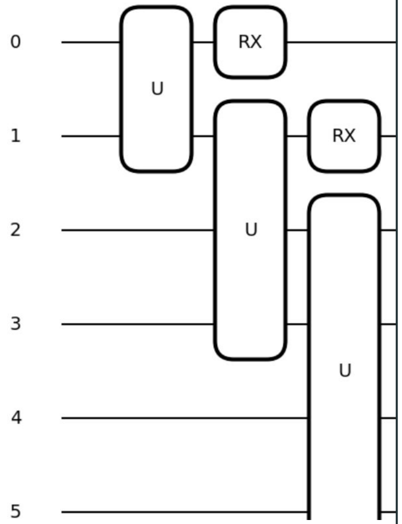
Reference: https://arxiv.org/abs/2209.00595

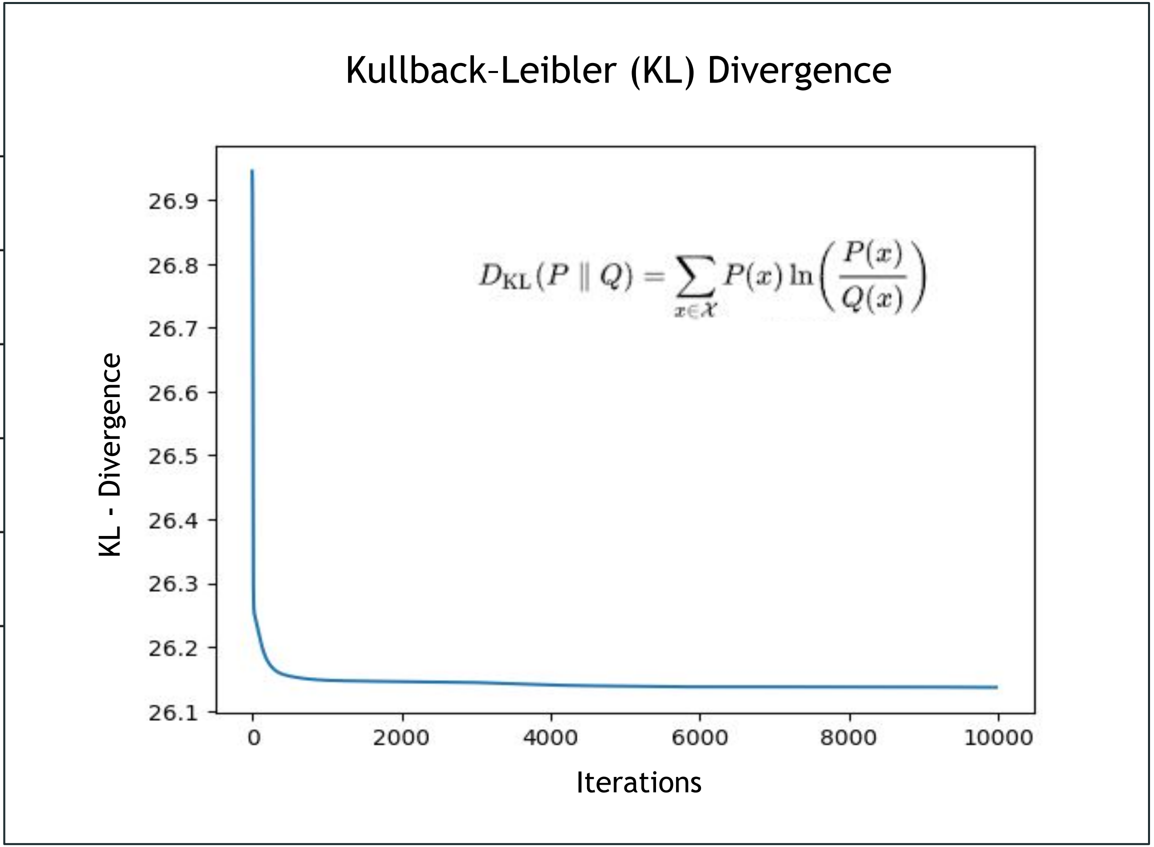
Code Overview

- ➔ Existing codebase for MPS training and normalization.
- ➔ Numpy.linalg and scipy.linalg for extracting the multi-qubit unitaries from the MPS.
- ➔ Pennylane to define and further train parameterized quantum circuits



Results


0
1
2
3
4
5
U
RX
U
RX
U


Kullback-Leibler (KL) Divergence
$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \ln\left(\frac{P(x)}{Q(x)}\right)$
KL - Divergence
26.9
26.8
26.7
26.6
26.5
26.4
26.3
26.2
26.1
0
2000
4000
6000
8000
10000
Iterations

Future Work

- To have a chance at improving the previously found MPS results, one needs to extend the linear layers with additional gates and train gates generated by MPS
- This will require increasing the circuit depth, more flexible entangling topologies, or both
- Train the unitary matrices generated by MPS alongside the additional gates added
- After training extended PQC, compare results with a randomly initialized circuit