Workshop Quantum Computing @ INFN, Bologna, 15/11/2022

Entanglement entropy production in Quantum Neural Networks

M. Ballarin, S. Mangini, S. Montangero, C. Macchiavello and R. Mengoni

arxiv:2206.02474

Department of Physics and Astronomy
Università degli Studi di Padova
INFN sezione di Padova

Marco Ballarin
marco.ballarin.6@phd.unipd.it

Digital quantum computing

Qubits are lines with time flowing from left to right

CNOT, apply X on crossed qubit (target) if dotted qubit (control) is in $|1\rangle$

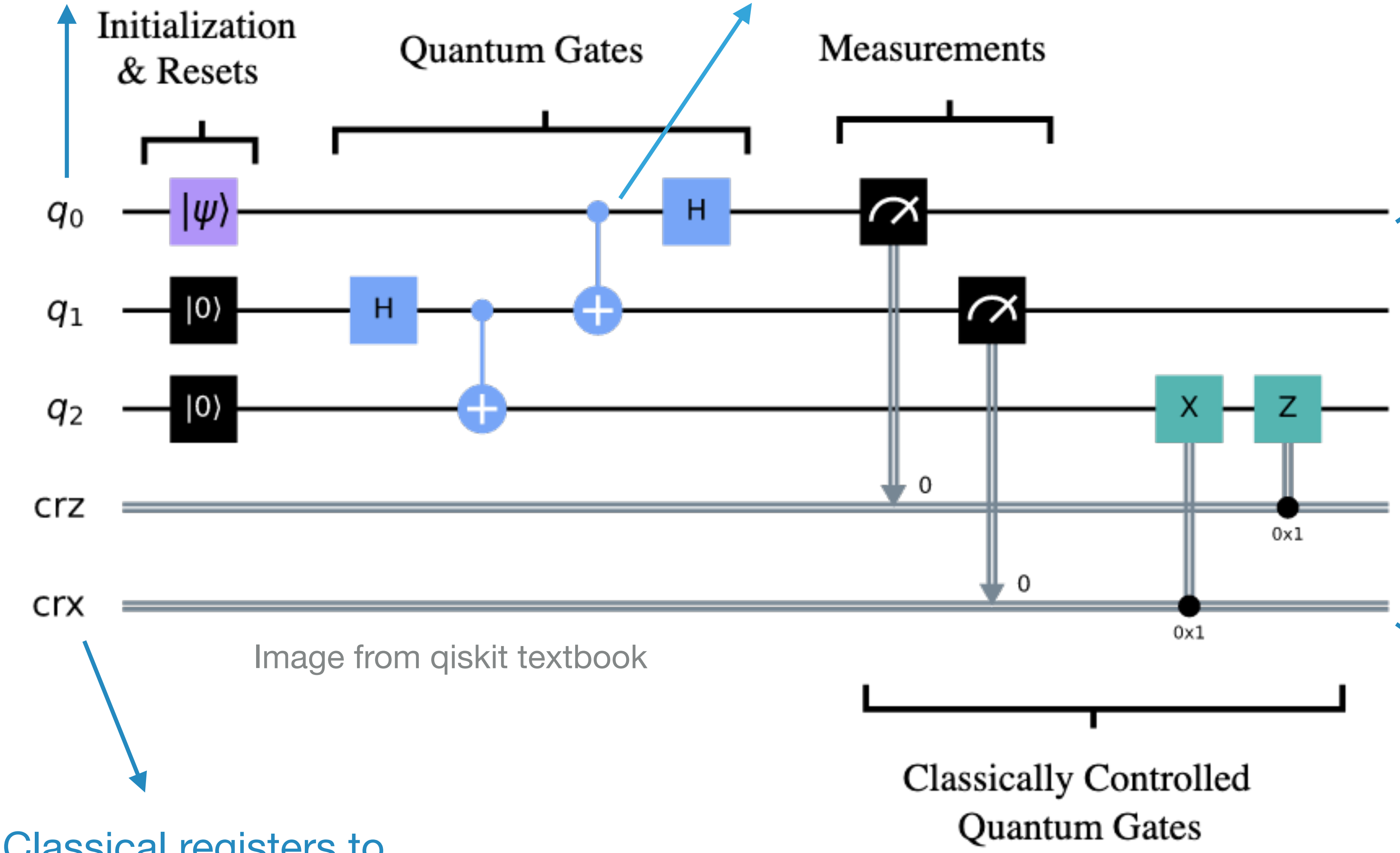


Image from qiskit textbook

Classical registers to store the results of measurements

DIGITAL QUANTUM SIMULATIONS

QUANTUM ALGORITHMS

QUANTUM CHEMISTRY

QUANTUM MACHINE LEARNING

Quantum machine learning

IRIS dataset

Iris Versicolor

Iris Setosa

Iris Virginica

Dataset

$$(x_i, y_i)_{i=1,\ldots,n}$$

Quantum machine learning

IRIS dataset

Iris Versicolor

Iris Setosa

Iris Virginica

Dataset

$$(\boxed{x_i}, y_i)_{i=1,\ldots,n}$$

Input features

Quantum machine learning

IRIS dataset

Iris Versicolor

Iris Setosa

Iris Virginica

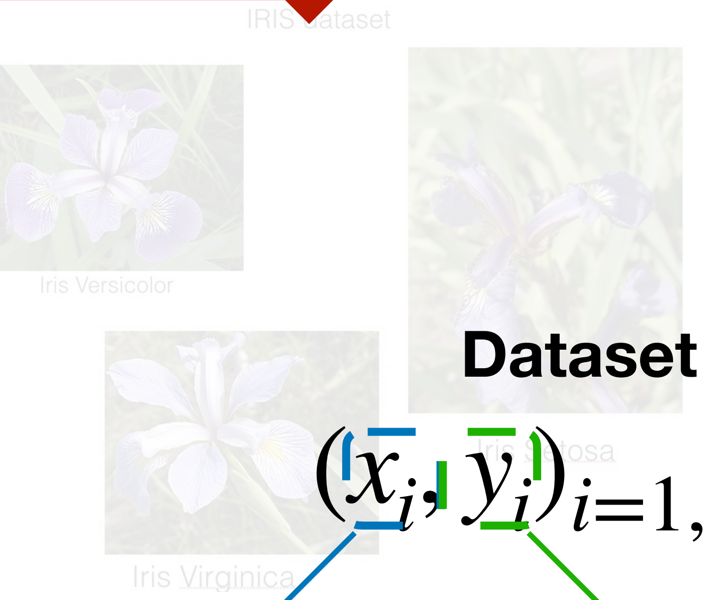
Dataset

$$(x_i, y_i)_{i=1,\ldots,n}$$

Input features

Input class

Quantum machine learning



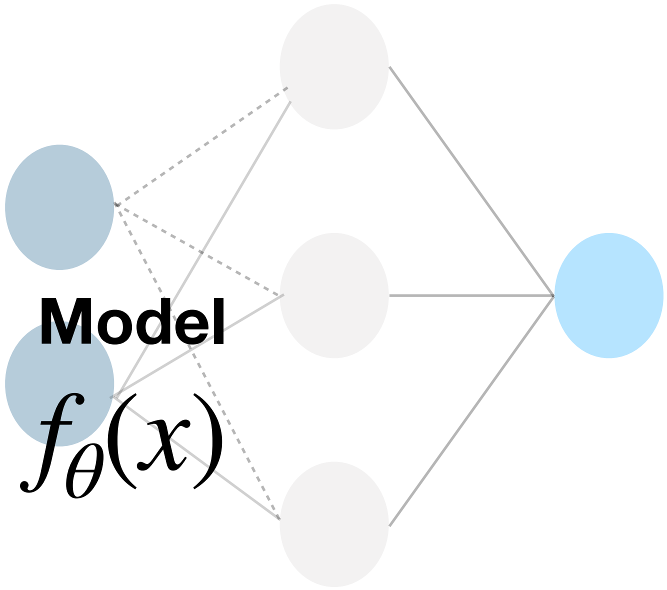
Dataset

$$(\boxed{x_i}, \boxed{y_i})_{i=1,\ldots,n}$$

Input features

Input class

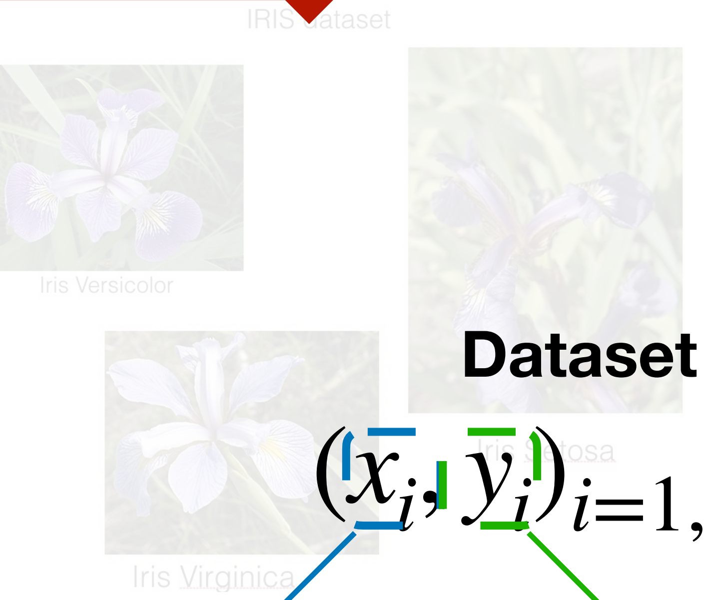
→



Model

$$f_\theta(x)$$

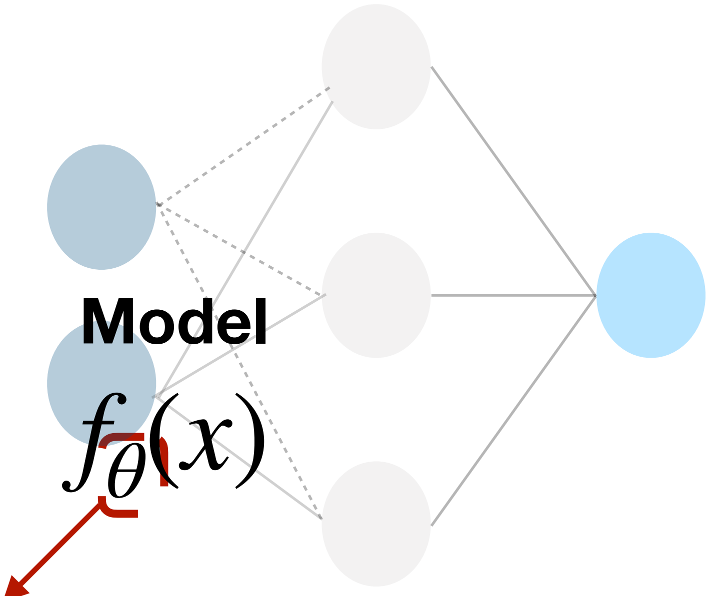
Quantum machine learning



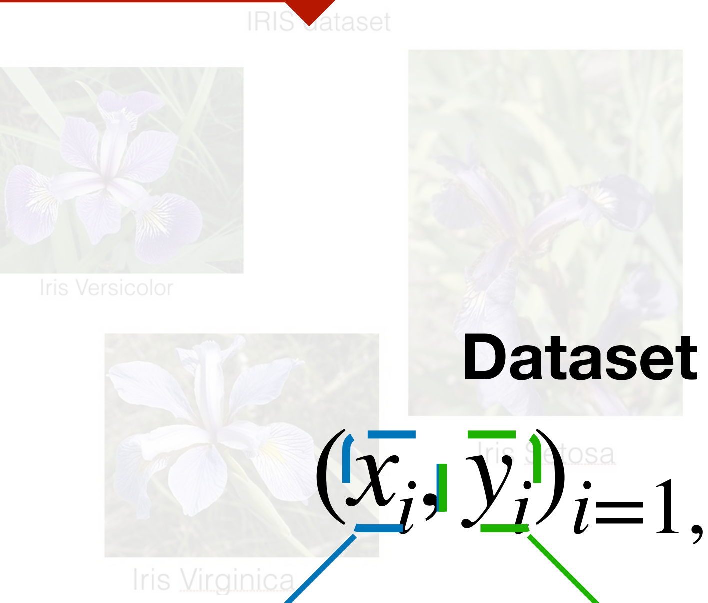
Dataset

$$(\underbrace{x_i}_{\text{Input features}}, \underbrace{y_i}_{\text{Input class}})_{i=1,\ldots,n} \longrightarrow f_{\underbrace{\theta}_{\text{Model parameters}}}(x)$$

Model



Quantum machine learning

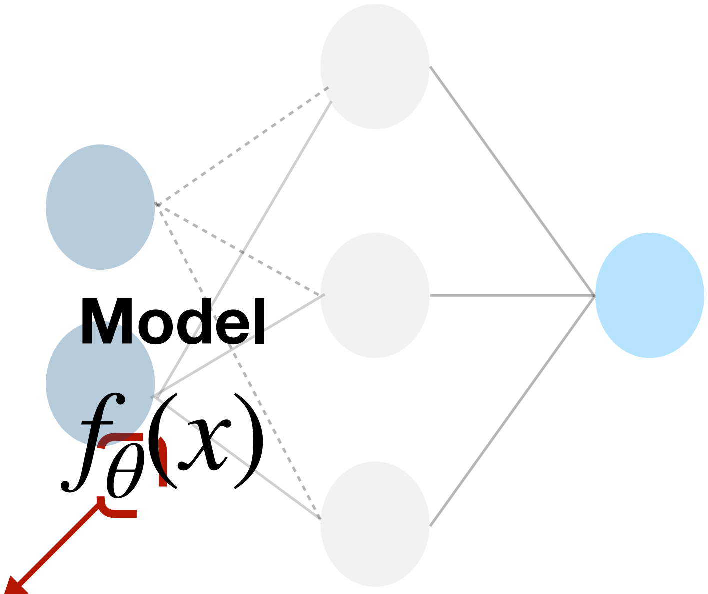


Dataset

$$(x_i, y_i)_{i=1,\dots,n}$$

Input features

Input class



Model

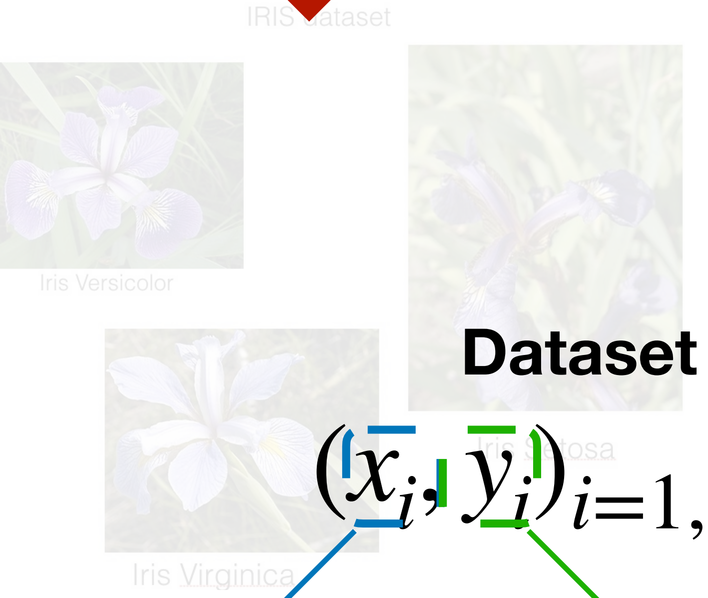
$$f_\theta(x)$$

Model parameters

Prediction

$$f_\theta(x_i) = \tilde{y}_i$$

Quantum machine learning

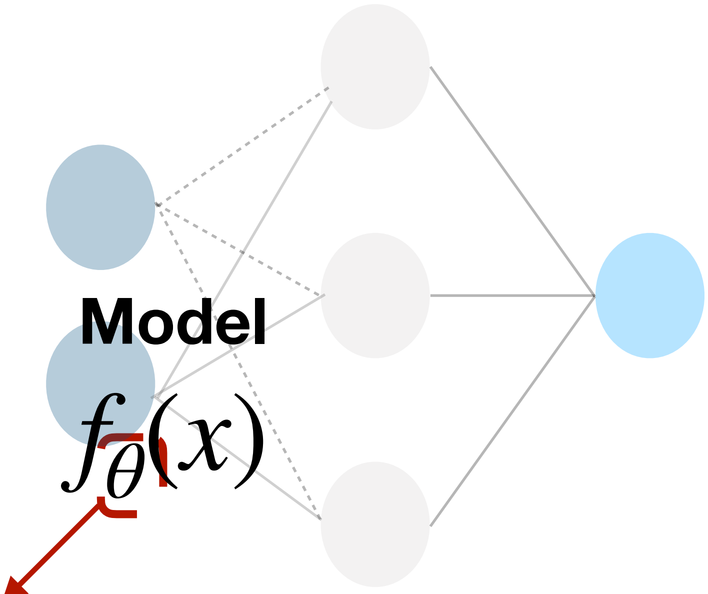


Dataset

$(x_i, y_i)_{i=1,\ldots,n}$

Input features

Input class



Model

$f_\theta(x)$

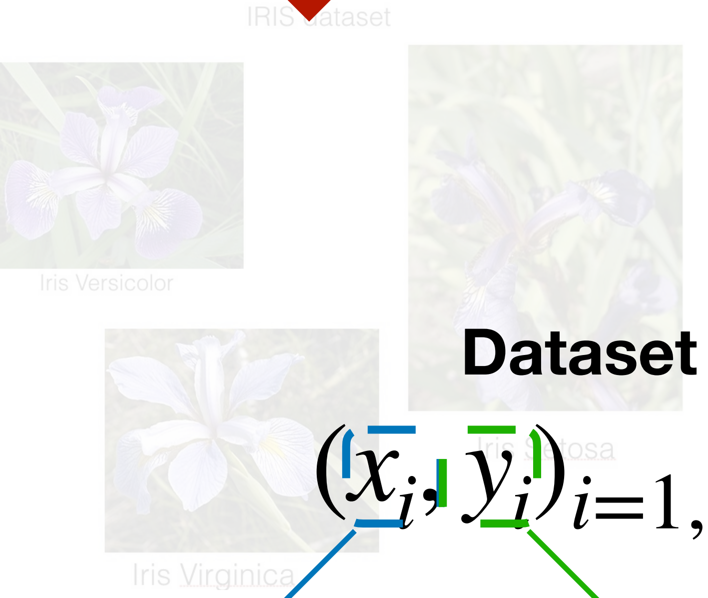
Model parameters

Update θ minimising loss function

Prediction

$f_\theta(x_i) = \tilde{y}_i$

Quantum machine learning



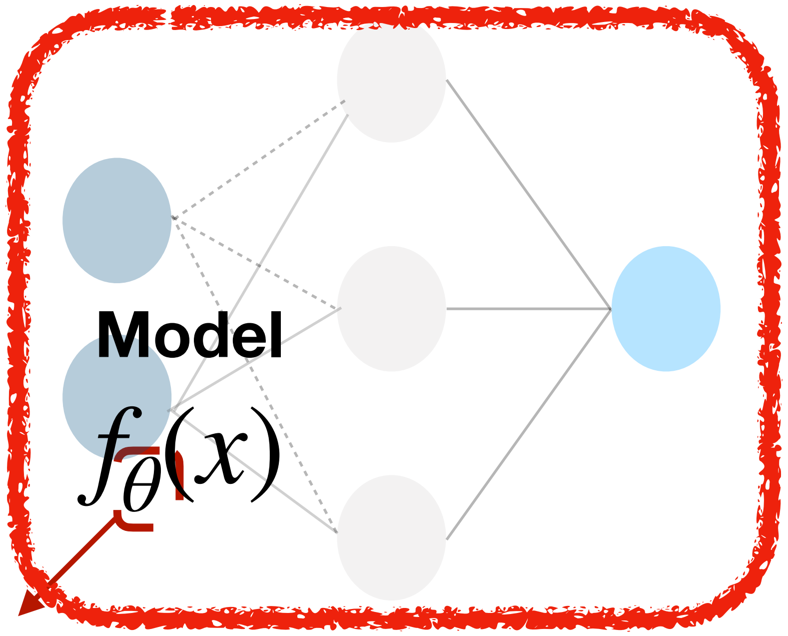
Dataset

$$(\underbrace{x_i}, \underbrace{y_i})_{i=1,\dots,n}$$

Input features

Input class

The model is
QUANTUM



Model

$$f_\theta(x)$$

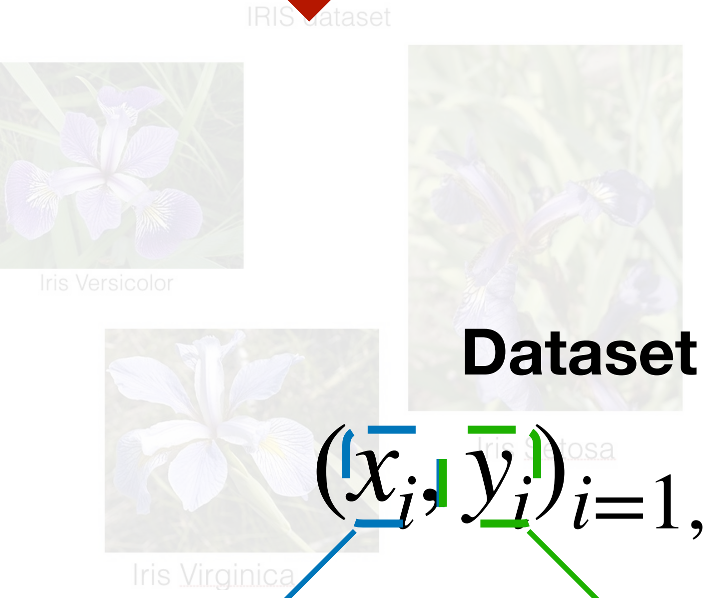
Model
parameters

Update θ
minimising
loss function

Prediction

$$f_\theta(x_i) = \tilde{y}_i$$

Quantum machine learning



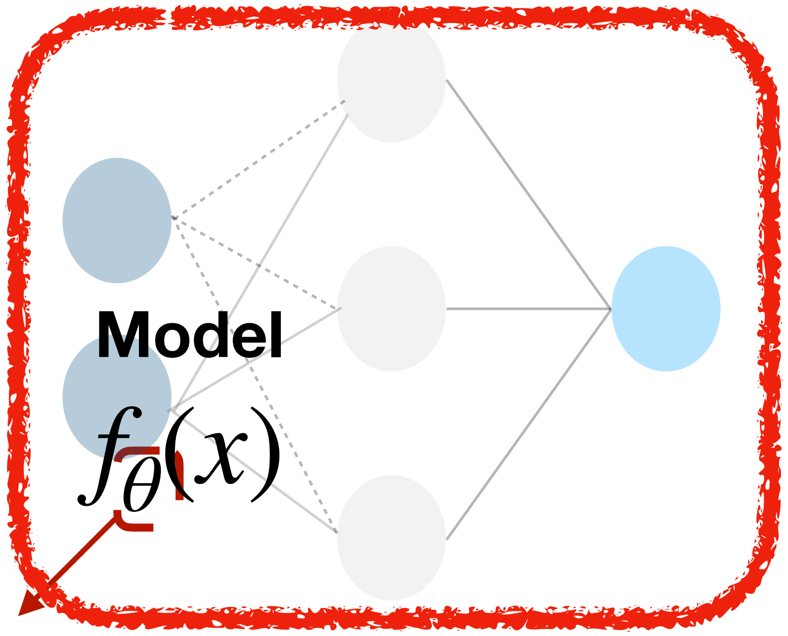
Dataset

$$(x_i, y_i)_{i=1,\dots,n}$$

Input features

Input class

The model is **QUANTUM**



Model

$$f_\theta(x)$$

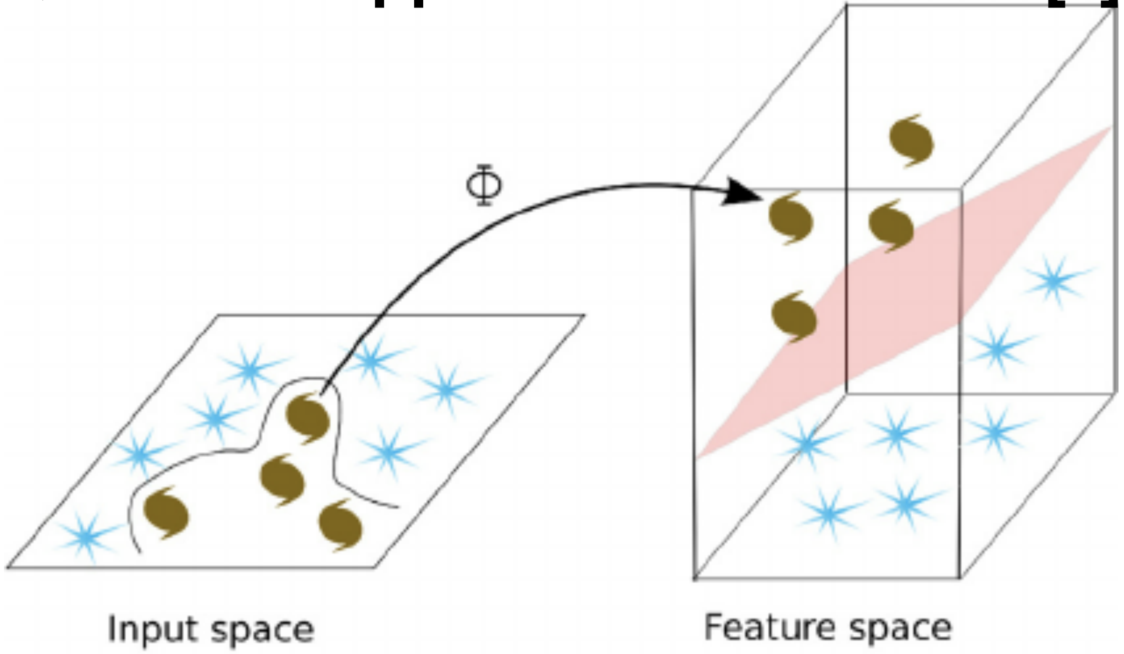
Model parameters

Update θ minimising loss function

Prediction

$$f_\theta(x_i) = \tilde{y}_i$$

Quantum support vector machine [1]

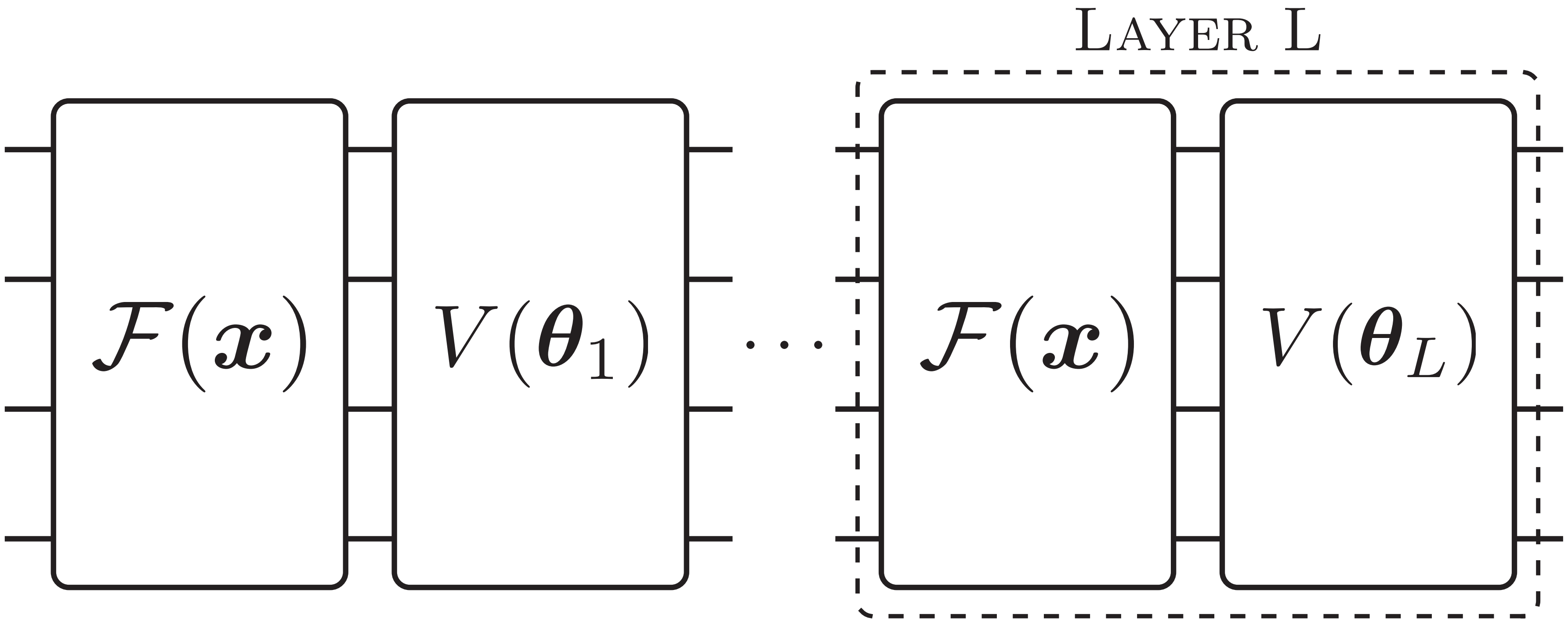


---

[1] Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

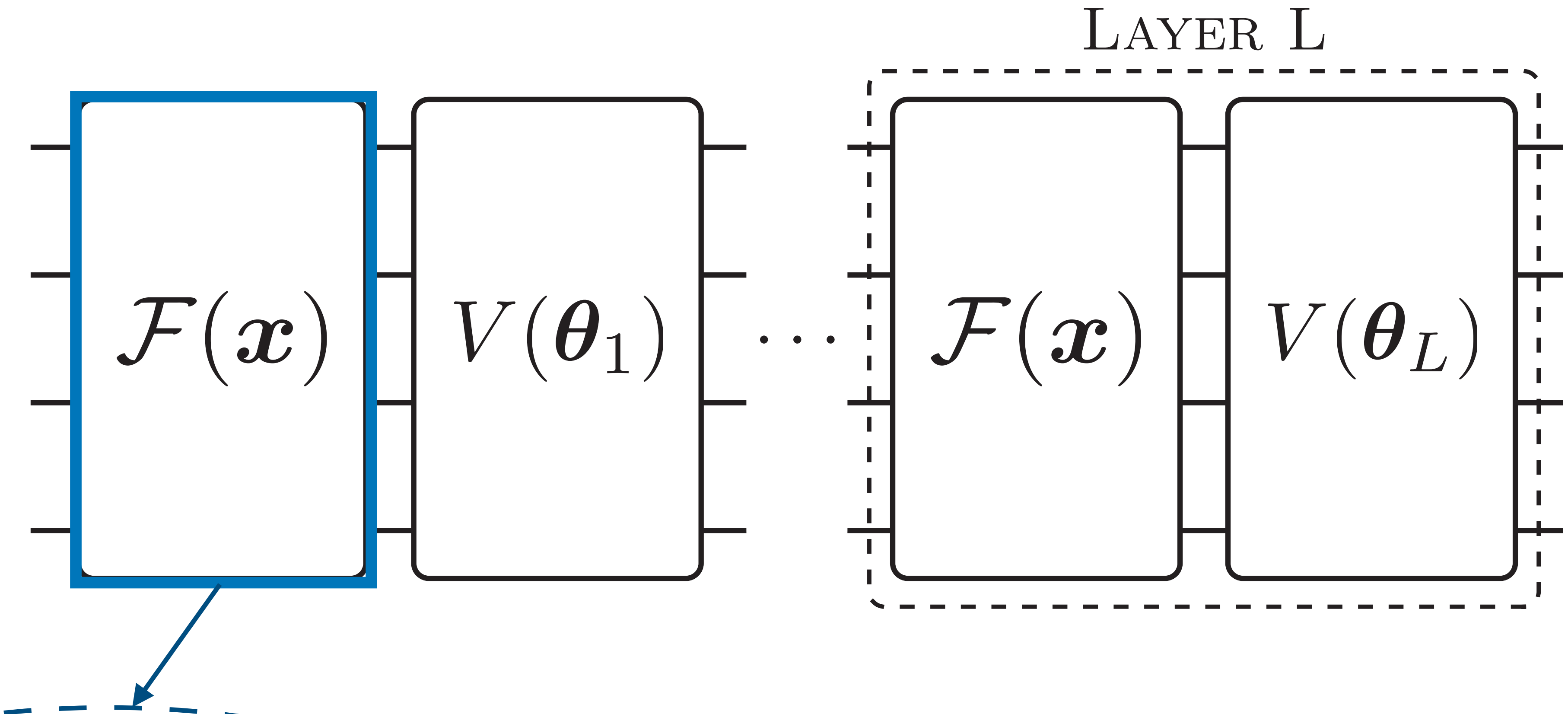
Quantum neural networks

Quantum Neural networks are **variational quantum circuits**.



Quantum neural networks

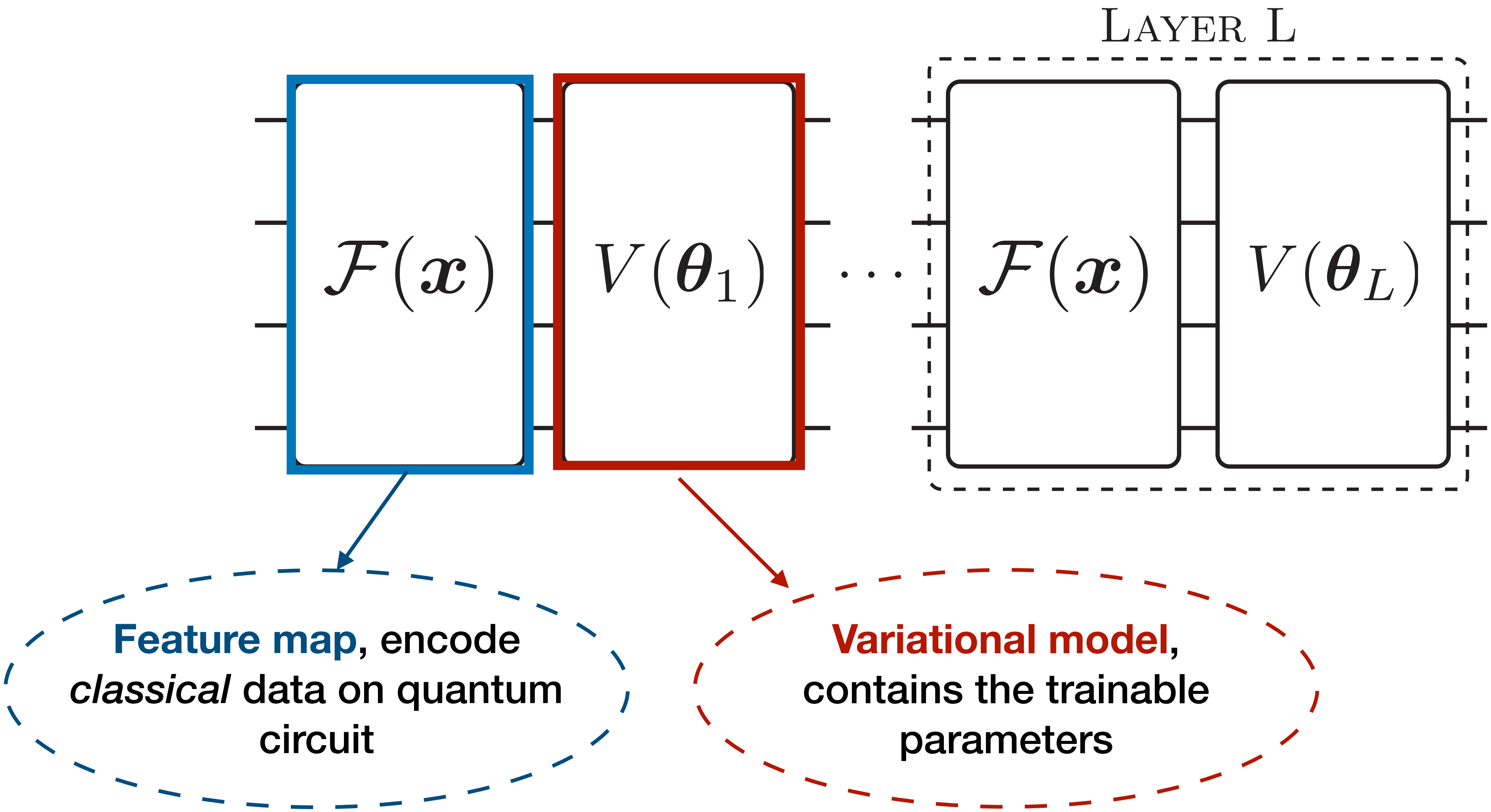
Quantum Neural networks are **variational quantum circuits**.



Feature map, encode *classical* data on quantum circuit

Quantum neural networks

Quantum Neural networks are **variational quantum circuits**.

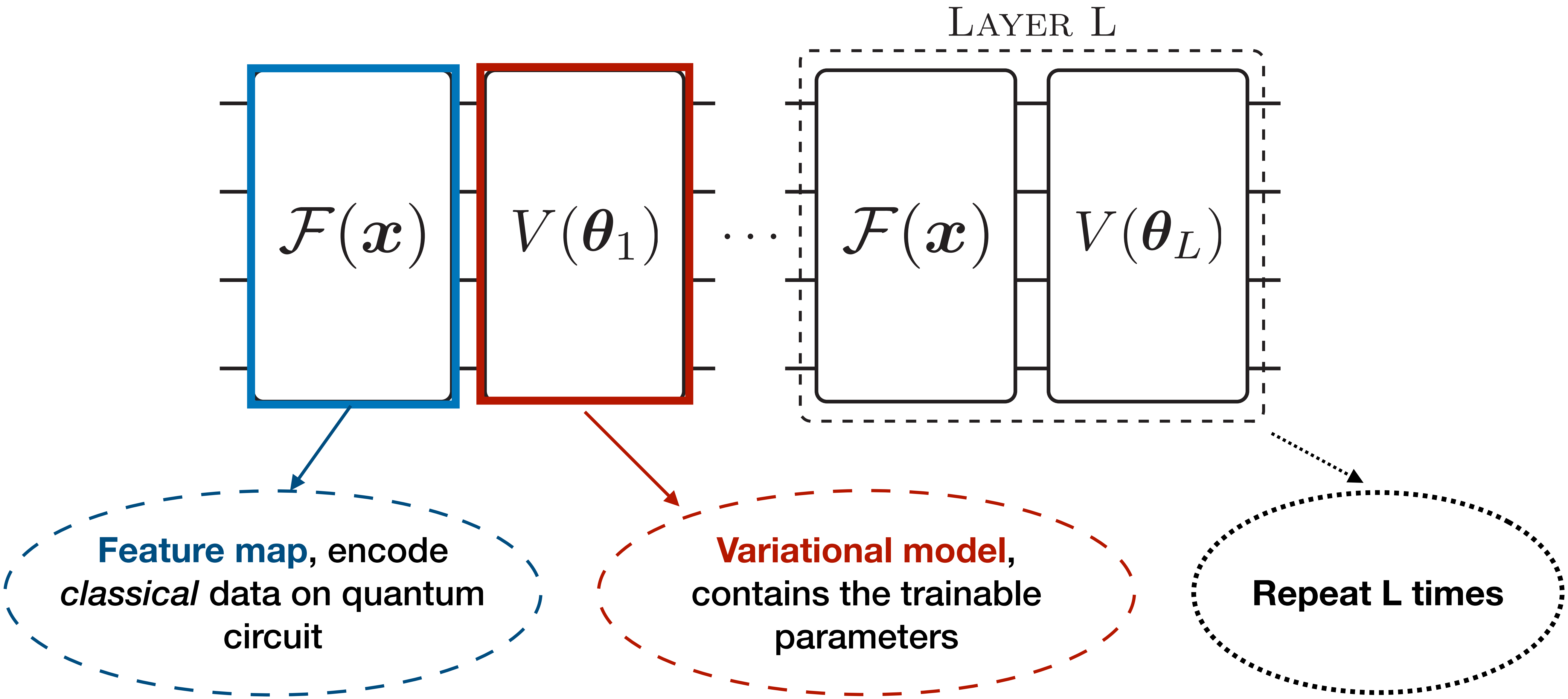


Feature map, encode *classical* data on quantum circuit

Variational model, contains the trainable parameters

Quantum neural networks

Quantum Neural networks are **variational quantum circuits**.



Feature map, encode *classical* data on quantum circuit

Variational model, contains the trainable parameters

Repeat L times

Entanglement: bless or curse?

Quantum neural networks promise to be better than classical NN [3]

---

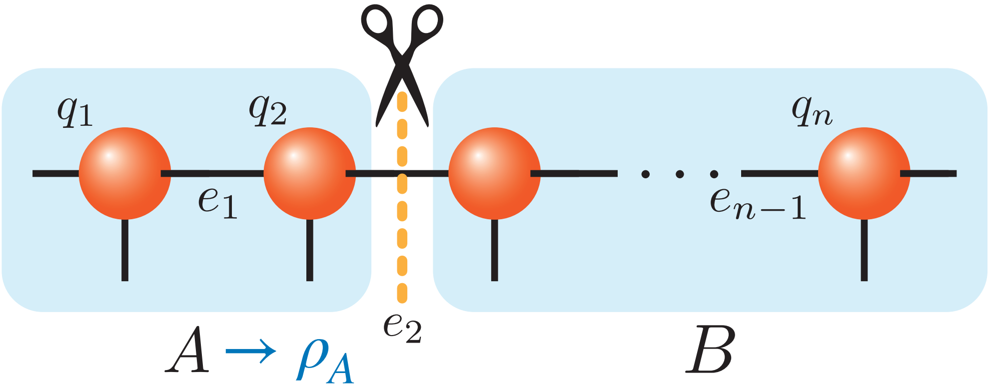
[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

[4] Marrero, Carlos Ortiz, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus." *PRX Quantum* 2.4 (2021): 040316.

Entanglement: bless or curse?

Quantum neural networks promise to be better than classical NN [3]

Entanglement



Von Neumann entanglement entropy between bipartitions A and B (across link e_i):

$$S(e_i) = -\mathbf{Tr}\left[\rho_A \log \rho_A\right]$$

---

[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

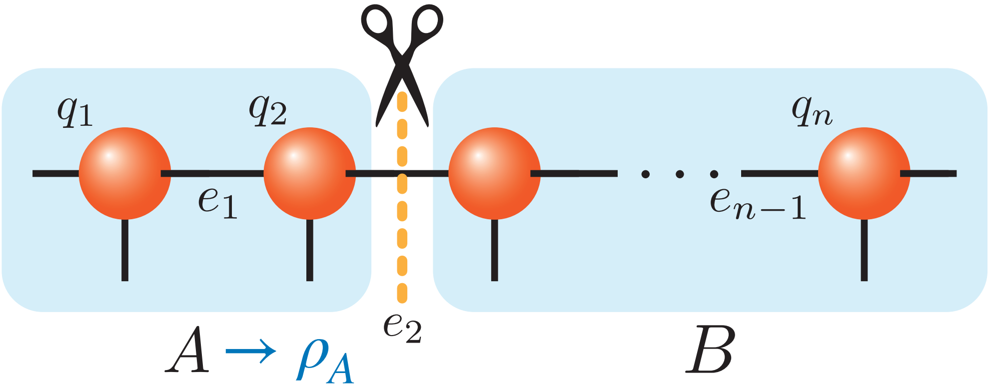
[4] Marrero, Carlos Ortiz, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus." *PRX Quantum* 2.4 (2021): 040316.

Entanglement: bless or curse?

Random states
⇒ highly entangled!
Haar-random state

Quantum neural networks promise to be better than classical NN [3]

Entanglement



Von Neumann entanglement entropy between bipartitions A and B (across link e_i):

$$S(e_i) = -\,\mathbf{Tr}\left[\rho_A \log \rho_A\right]$$

---

[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

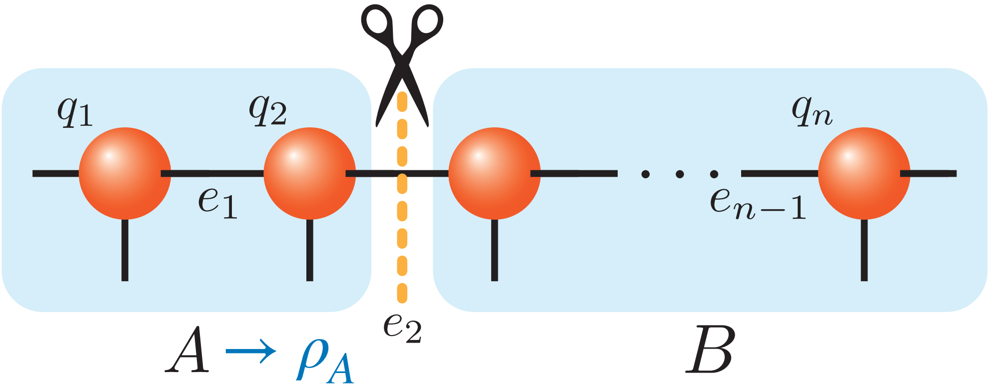
[4] Marrero, Carlos Ortiz, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus." *PRX Quantum* 2.4 (2021): 040316.

Entanglement: bless or curse?

Random states
⇒ highly entangled!
Haar-random state

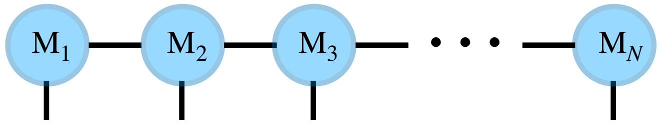
Quantum neural networks promise to be better than classical NN [3]

Entanglement



Von Neumann entanglement entropy between bipartitions A and B (across link e_i):

$$S(e_i) = -\mathbf{Tr}\left[\rho_A \log \rho_A\right]$$



Lowly entangled states ⇒ efficient simulations with tensor networks methods (Quantum matcha tea)

---

[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

[4] Marrero, Carlos Ortiz, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus." *PRX Quantum* 2.4 (2021): 040316.

Entanglement: bless or curse?

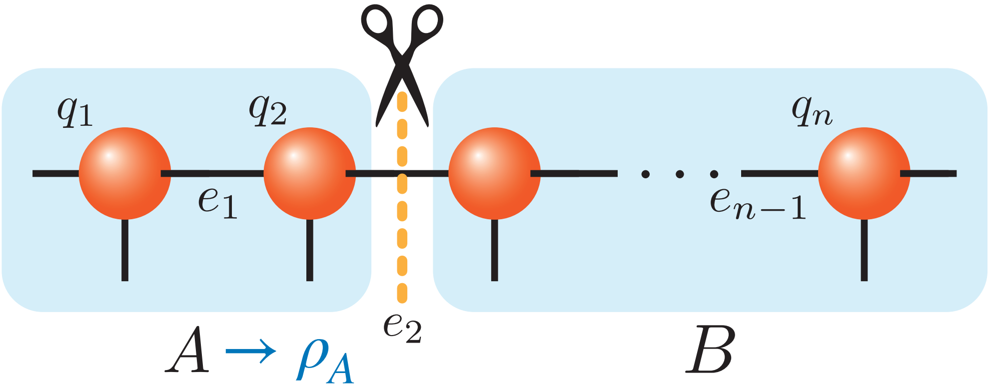
Random states

⇒ highly entangled!

Haar-random state

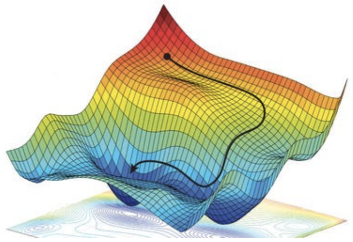
Quantum neural networks promise to be better than classical NN [3]

Entanglement

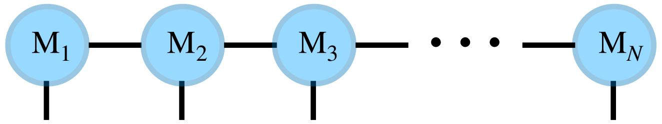


Von Neumann entanglement entropy between bipartitions A and B (across link e_i):

$$S(e_i) = -\,\mathbf{Tr}\left[\rho_A \log \rho_A\right]$$



Highly entangled states ⇒ connected with the emergence of Barren Plateaus [4]



Lowly entangled states ⇒ efficient simulations with tensor networks methods (Quantum matcha tea)

---

[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

[4] Marrero, Carlos Ortiz, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus." *PRX Quantum* 2.4 (2021): 040316.

Tensor network emulator

String compression (simple example):

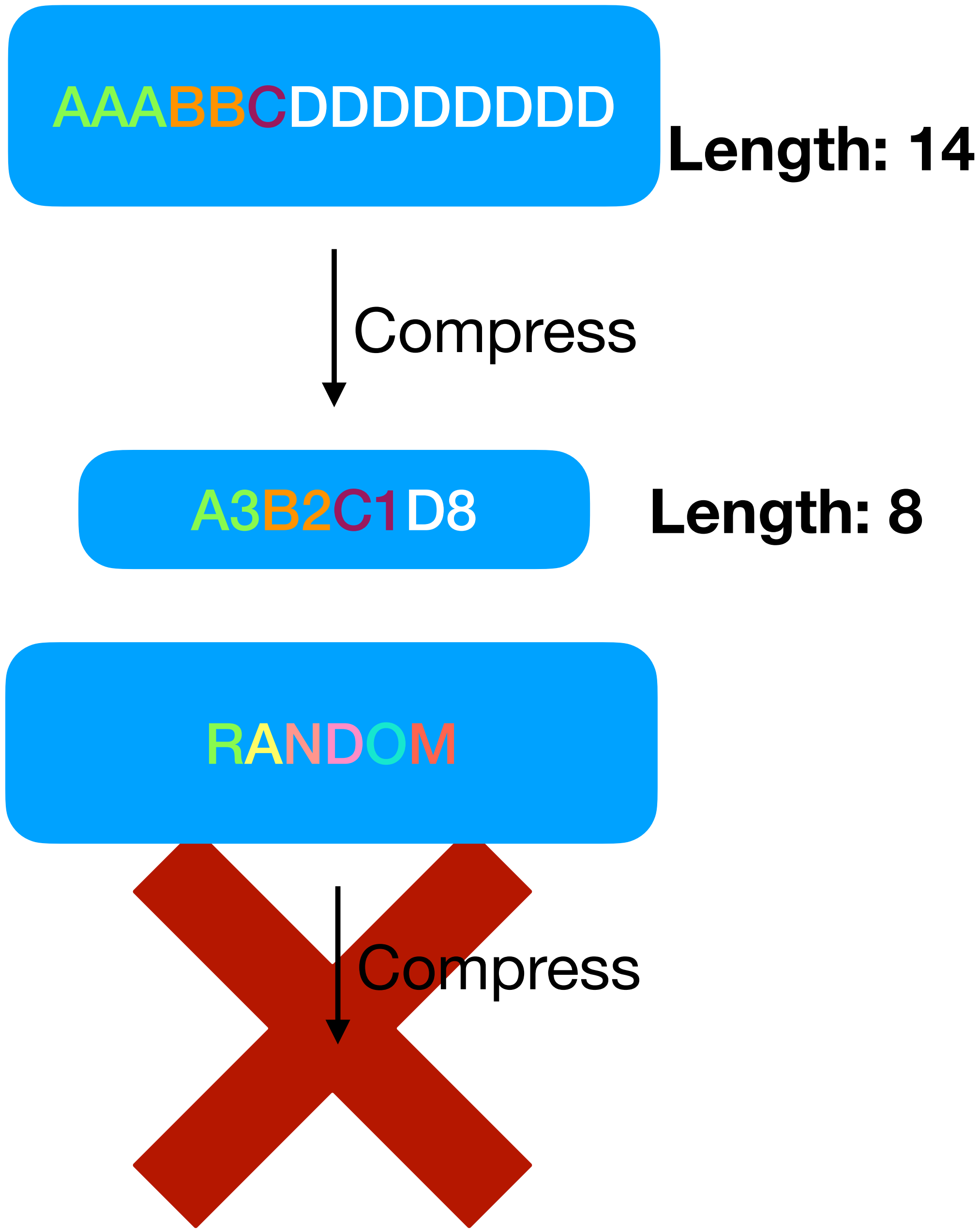
AAABBCDDDDDDDD **Length: 14**

↓ Compress

A3B2C1D8 **Length: 8**

Tensor network emulator

String compression (simple example):



AAABBCDDDDDDDD **Length: 14**

Compress

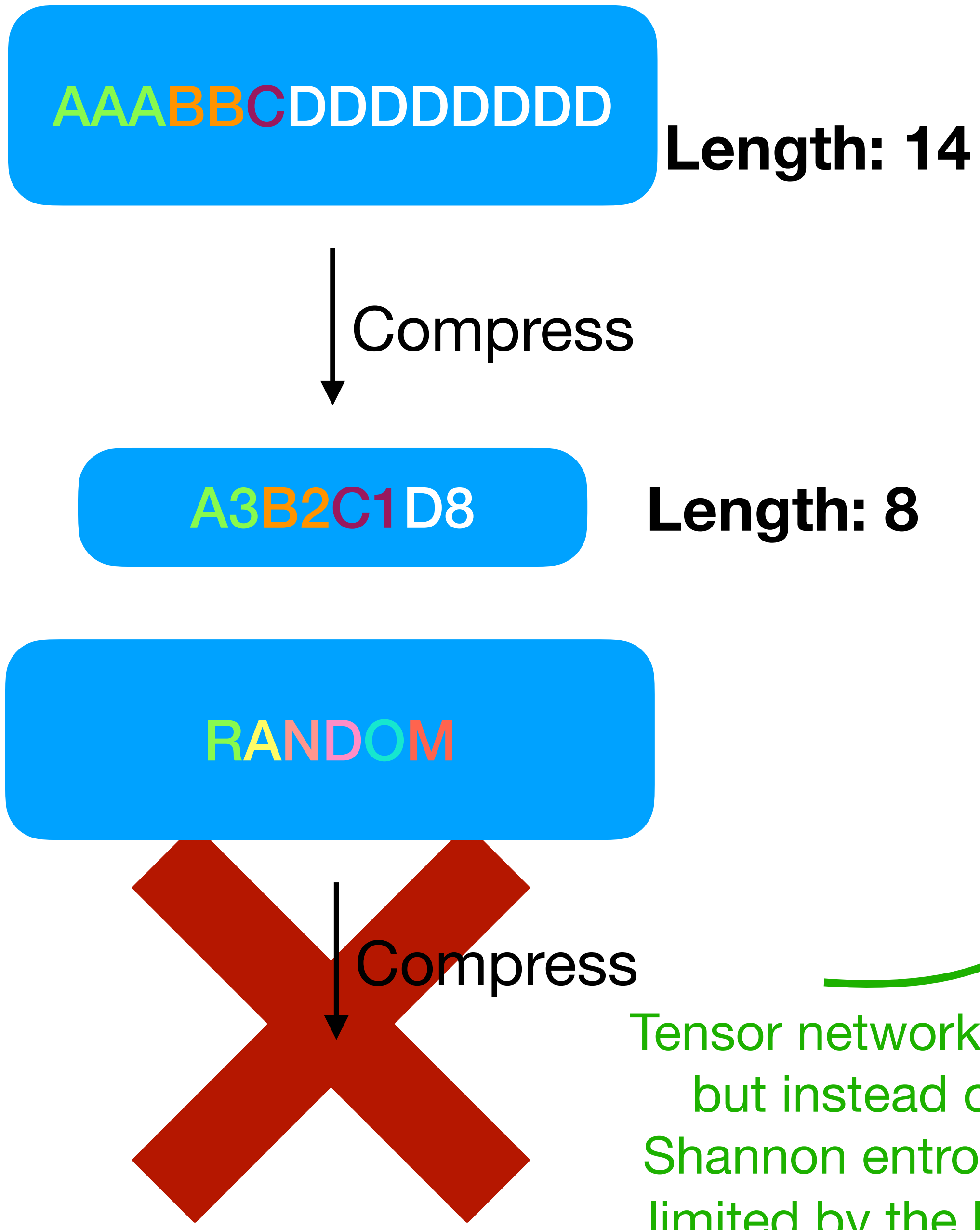
A3B2C1D8 **Length: 8**

RANDOM

Compress

Tensor network emulator

String compression (simple example):



AAABBCDDDDDDDD

Length: 14

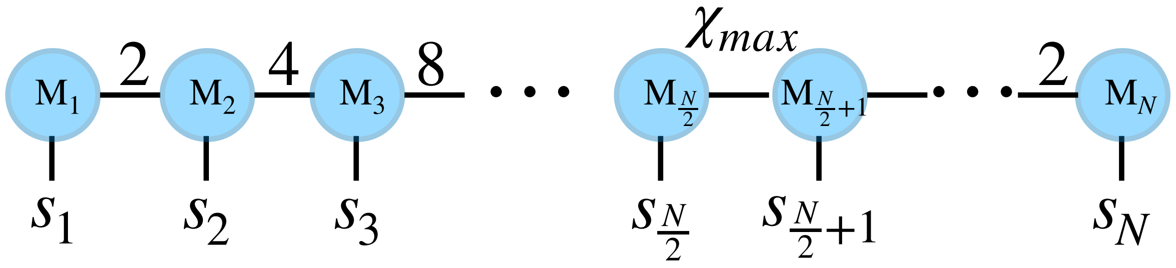
Compress

A3B2C1D8

Length: 8

RANDOM

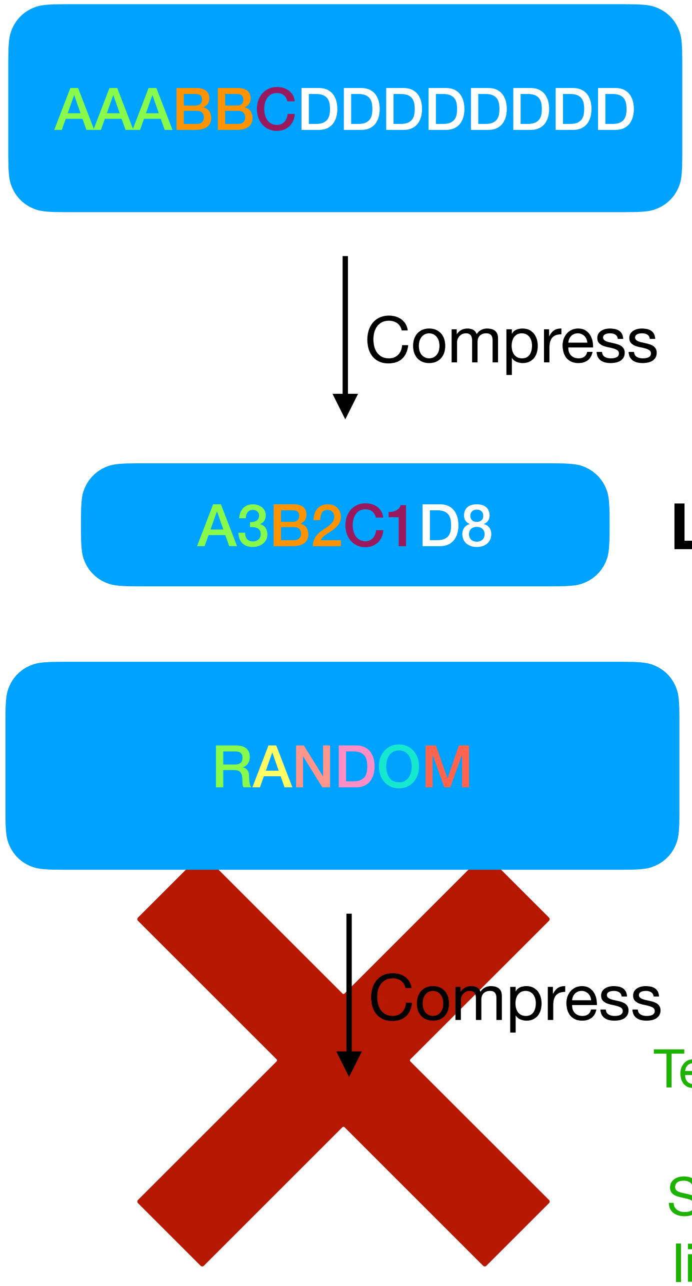
Compress



Tensor networks works in a similar way, but instead of being limited by the Shannon entropy of the string they are limited by the **Entanglement Entropy of the state**

Tensor network emulator

String compression (simple example):

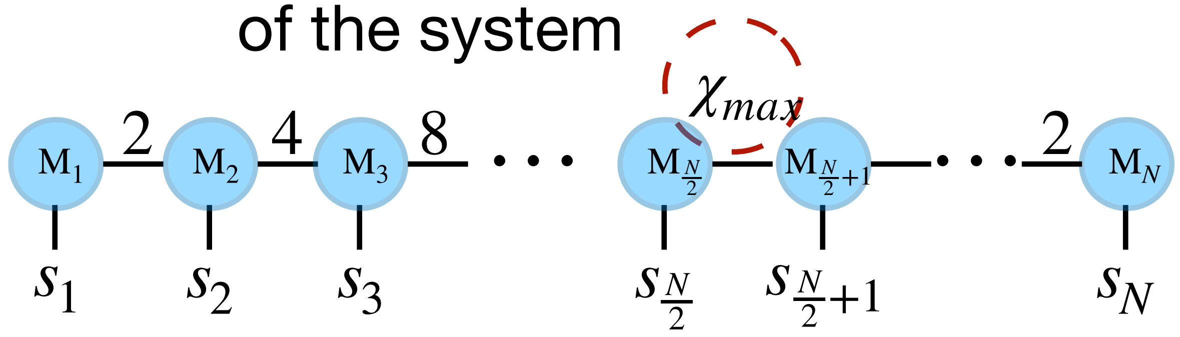


Length: 14

Length: 8

Bond dimension

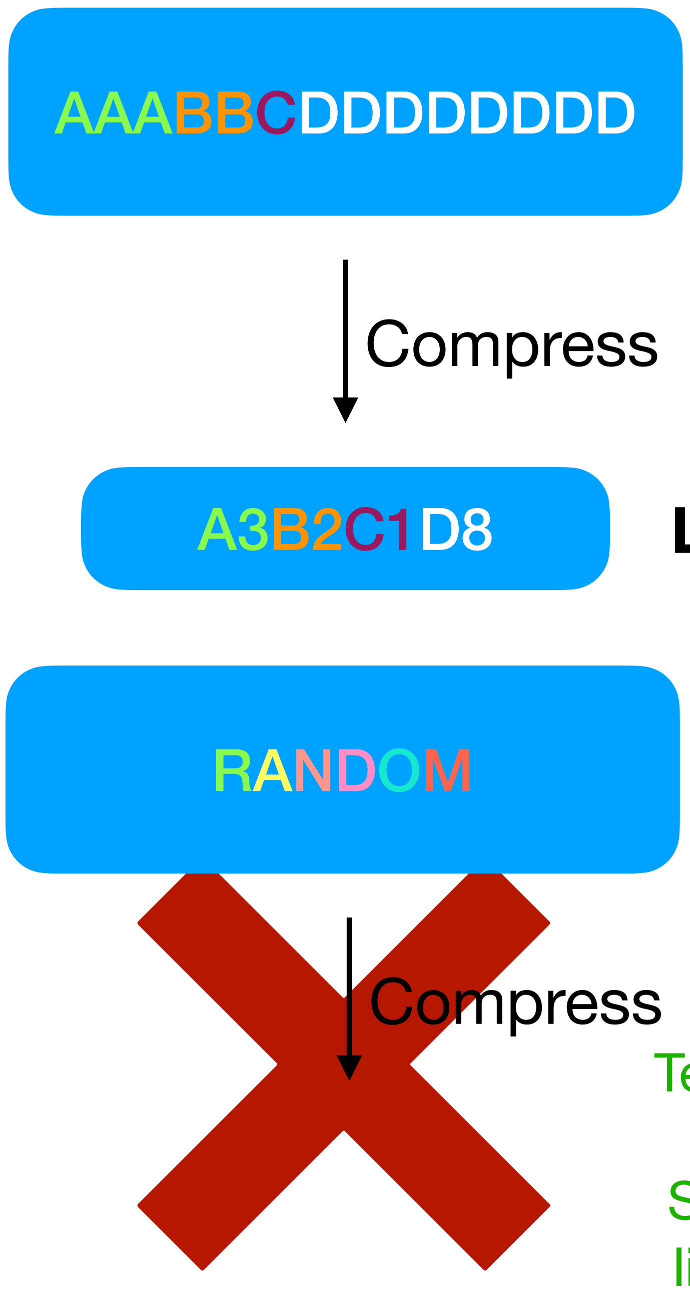
It controls the entanglement of the system



Tensor networks works in a similar way, but instead of being limited by the Shannon entropy of the string they are limited by the **Entanglement Entropy of the state**

Tensor network emulator

String compression (simple example):



AAABBCDDDDDDDD

Length: 14

Compress

A3B2C1D8

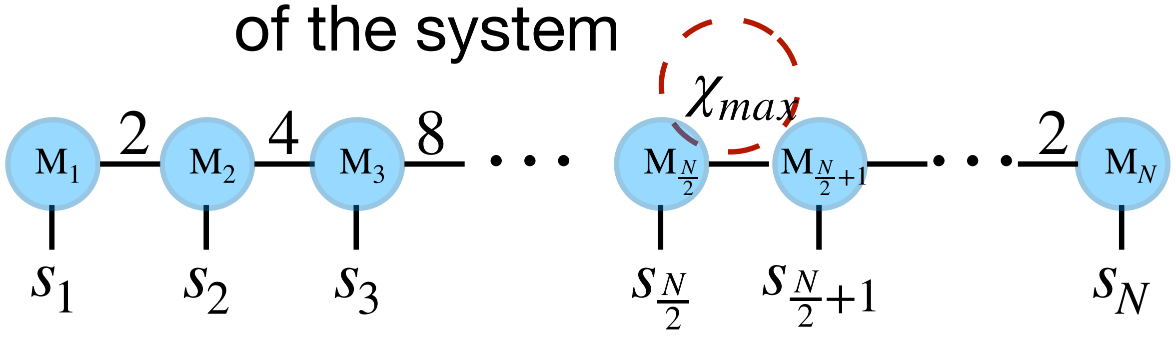
Length: 8

RANDOM

Compress

Bond dimension

It controls the entanglement of the system

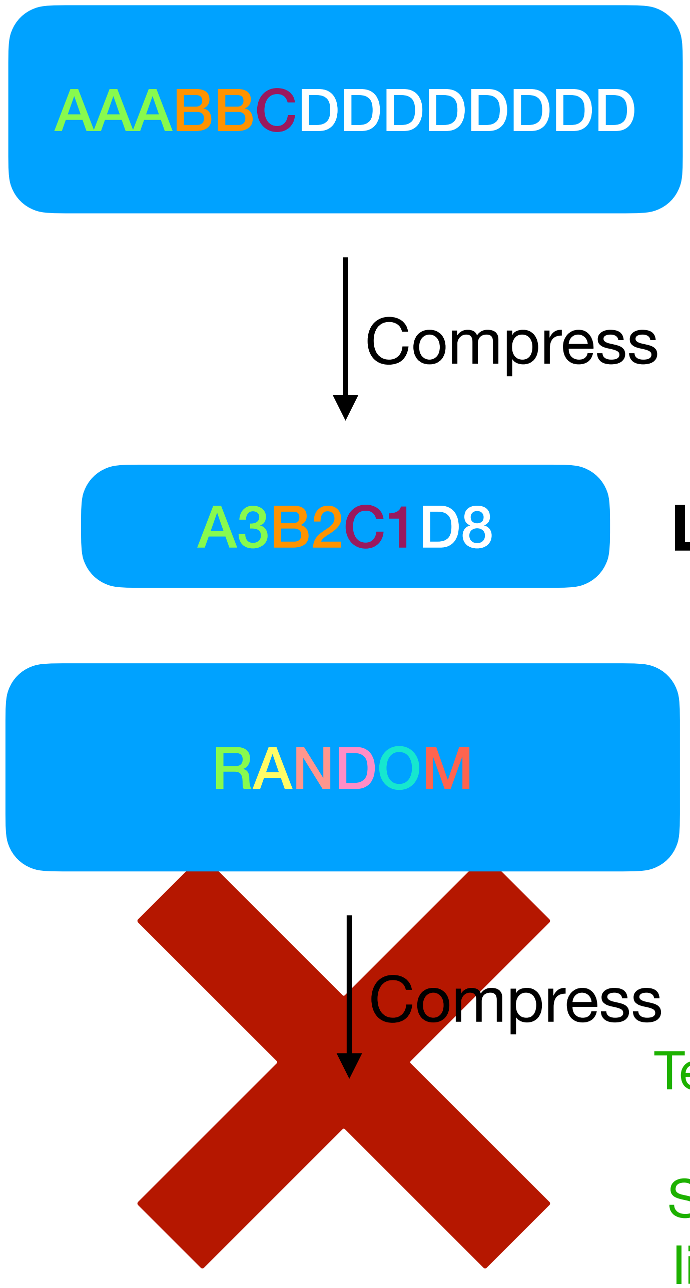


$\chi = 1$ is the Mean Field approximation

Tensor networks works in a similar way, but instead of being limited by the Shannon entropy of the string they are limited by the **Entanglement Entropy of the state**

Tensor network emulator

String compression (simple example):



AAABBCDDDDDDDD

Length: 14

Compress

A3B2C1D8

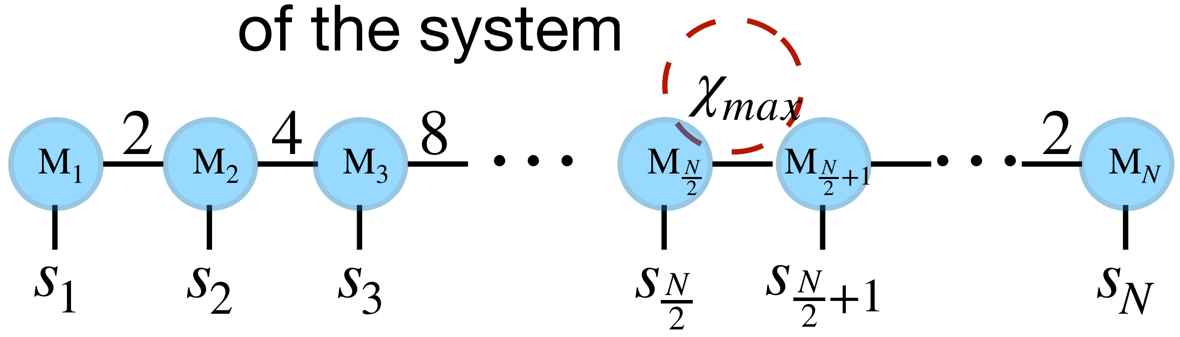
Length: 8

RANDOM

Compress

Bond dimension

It controls the entanglement of the system



$\chi = 1$ is the Mean Field approximation

Quantum matcha tea: efficient tensor network emulator for Quantum Circuits

Tensor networks works in a similar way, but instead of being limited by the Shannon entropy of the string they are limited by the **Entanglement Entropy of the state**

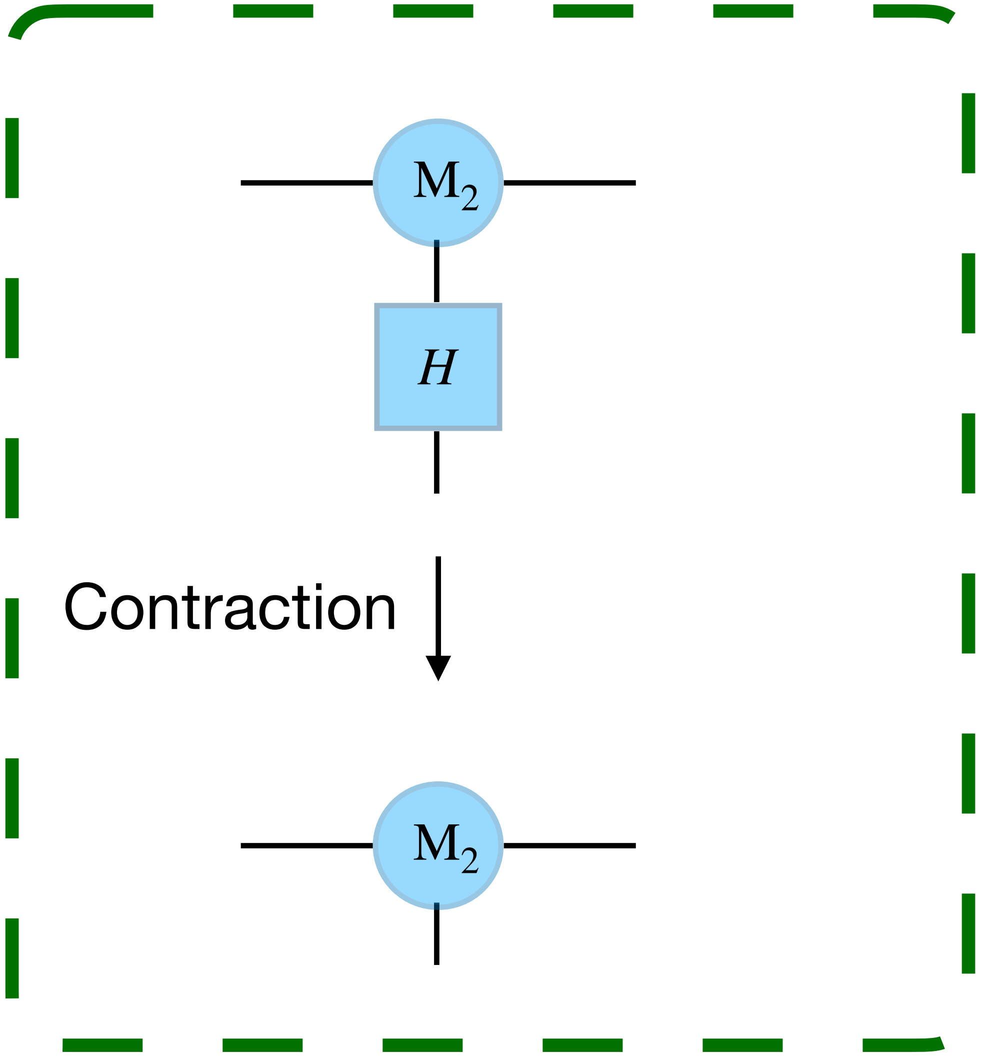
State Evolution

- The initial state can be described by a product state, that can be described exactly by MPS with a bond dimension $\chi = 1.$
- We then apply operators to evolve the state, bringing it into the target state $|\psi\rangle$, as we would do normally with a quantum circuit.

State Evolution

- The initial state can be described by a product state, that can be described exactly by MPS with a bond dimension $\chi = 1.$
- We then apply operators to evolve the state, bringing it into the target state $|\psi\rangle$, as we would do normally with a quantum circuit.

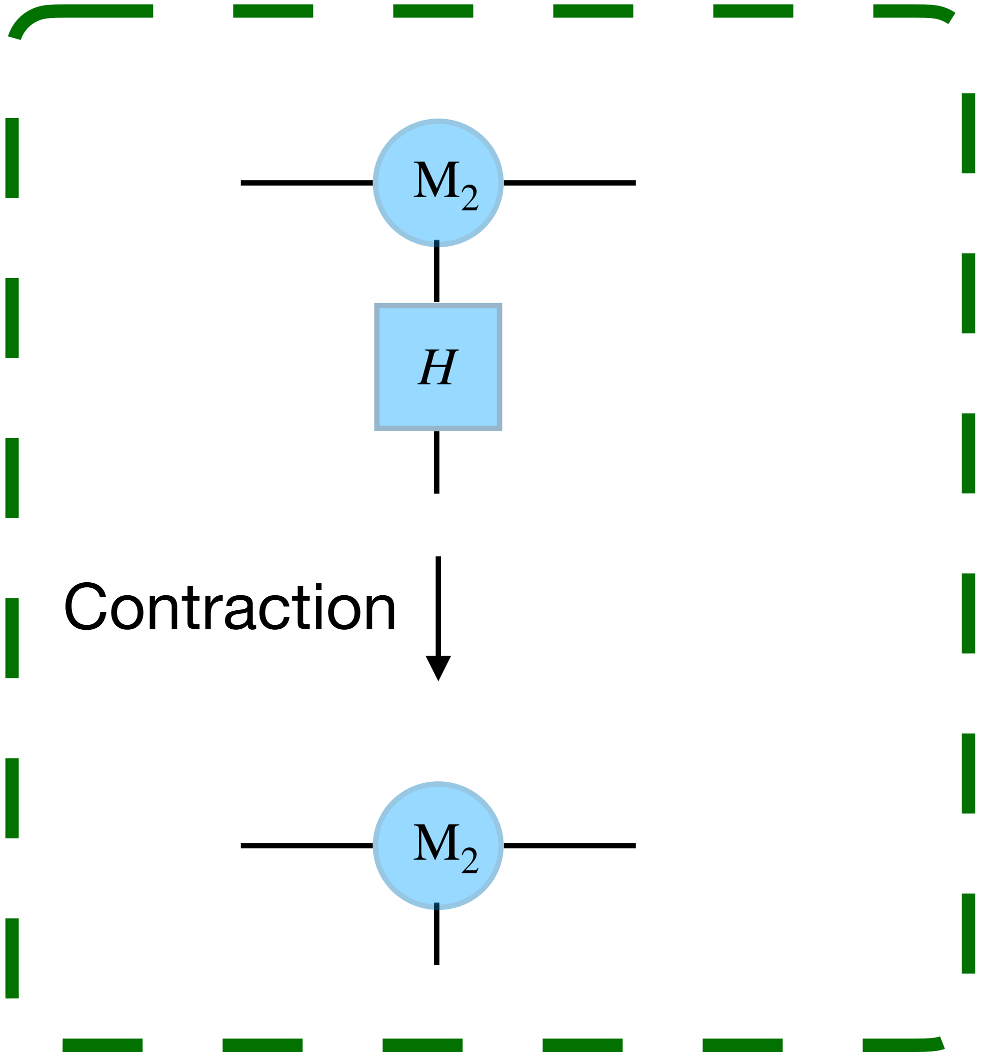
ONE-QUBIT GATE



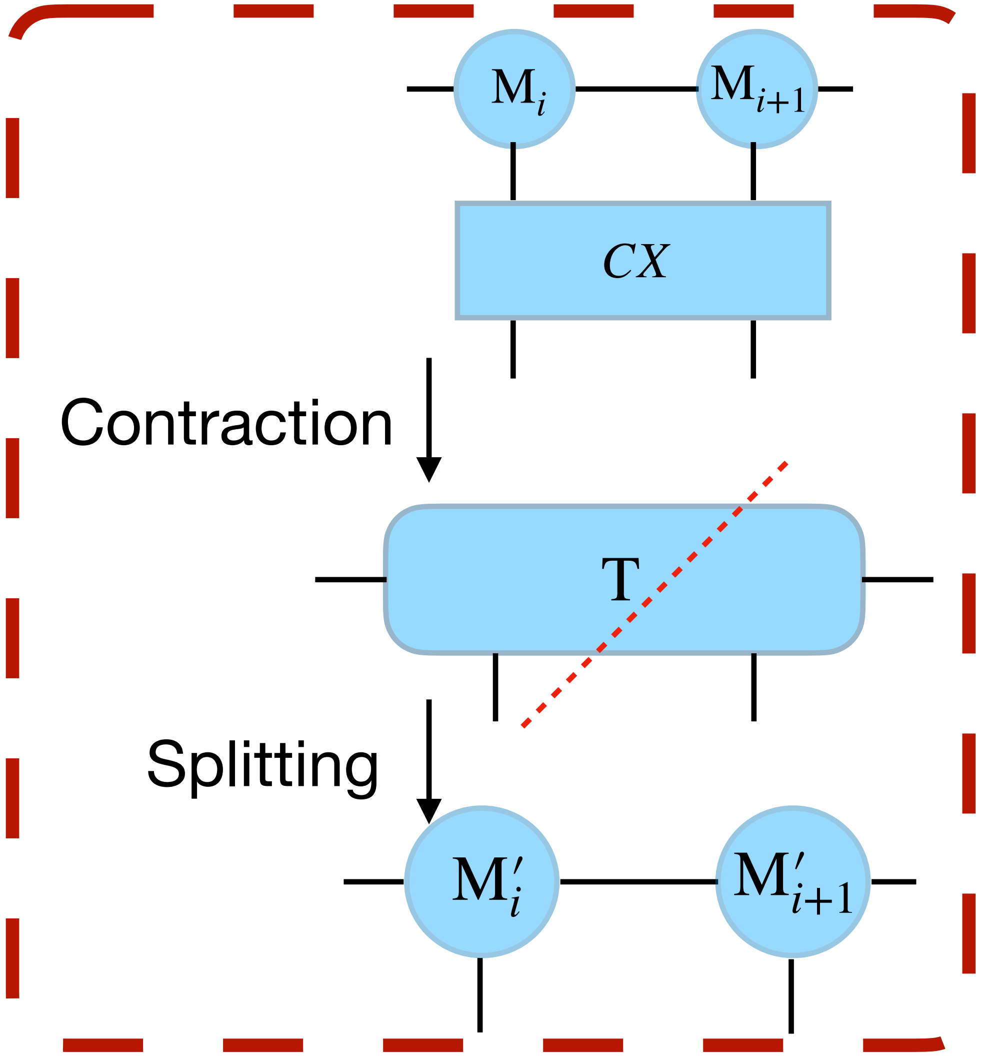
State Evolution

- The initial state can be described by a product state, that can be described exactly by MPS with a bond dimension $\chi = 1$.
- We then apply operators to evolve the state, bringing it into the target state $|\psi\rangle$, as we would do normally with a quantum circuit.

ONE-QUBIT GATE

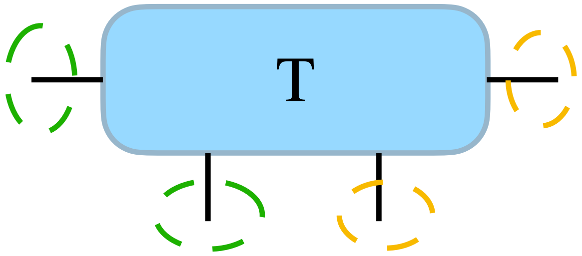


TWO-QUBITS GATE



Singular Values Decomposition

The core of tensor networks algorithms lays in the application of SVDs
to the tensors applying an appropriate approximation on the singular values

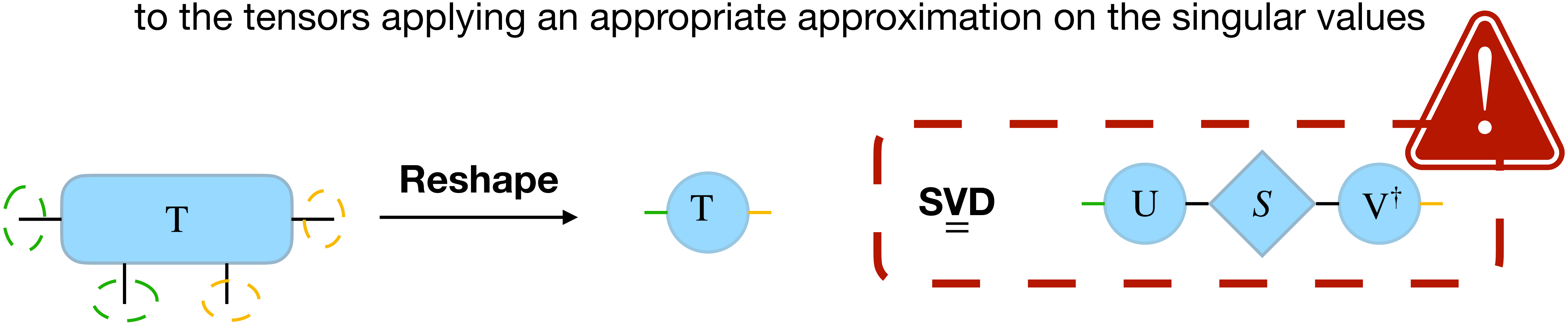


Singular Values Decomposition

The core of tensor networks algorithms lays in the application of SVDs
to the tensors applying an appropriate approximation on the singular values

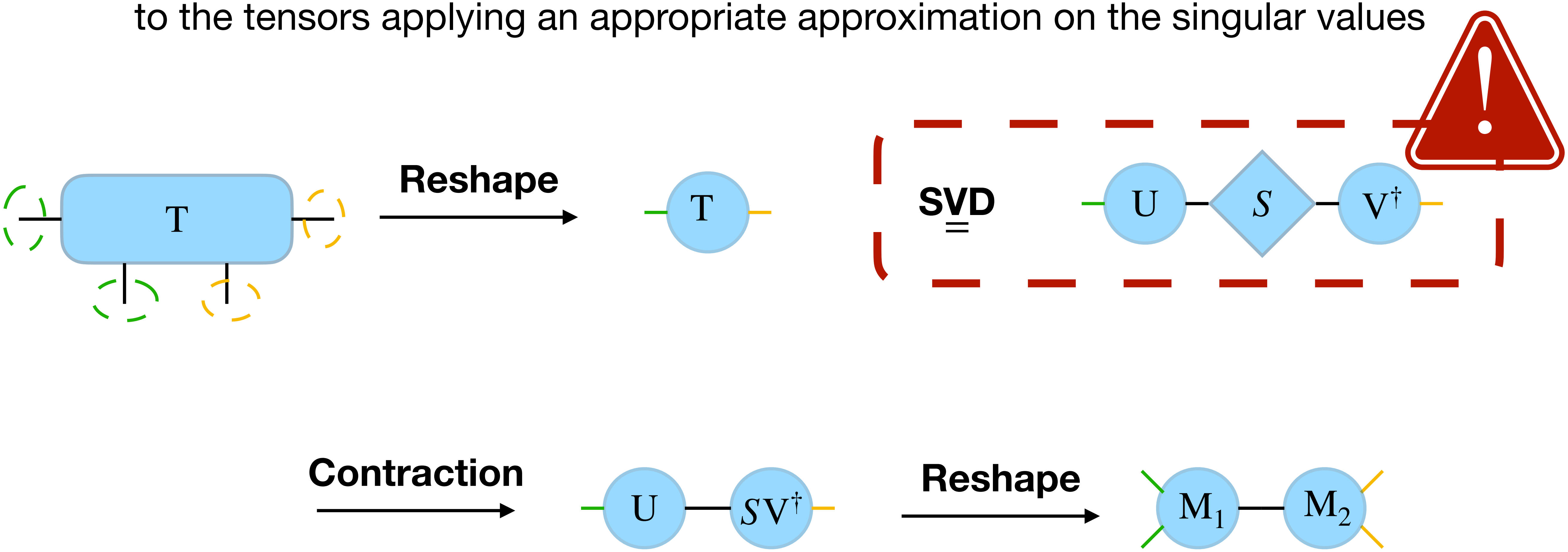
Singular Values Decomposition

The core of tensor networks algorithms lays in the application of SVDs
to the tensors applying an appropriate approximation on the singular values



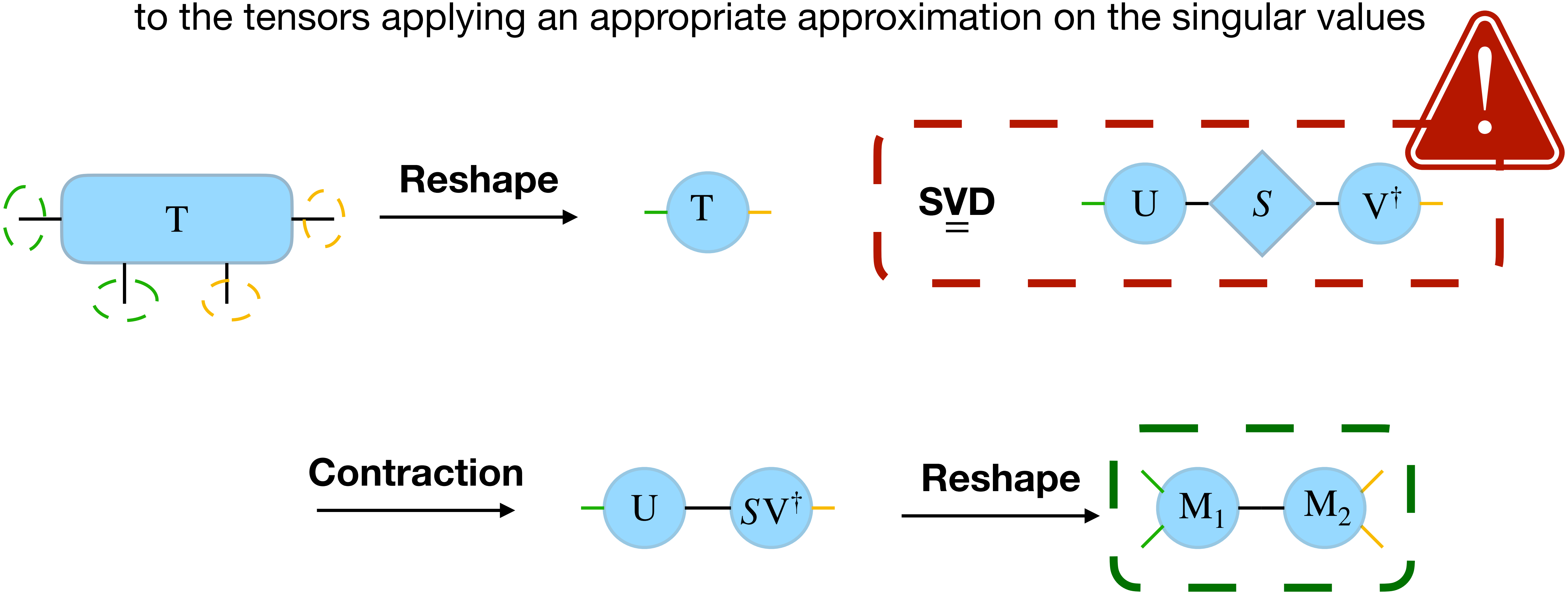
Singular Values Decomposition

The core of tensor networks algorithms lays in the application of SVDs
to the tensors applying an appropriate approximation on the singular values



Singular Values Decomposition

The core of tensor networks algorithms lays in the application of SVDs
to the tensors applying an appropriate approximation on the singular values



We are back to the MPS form

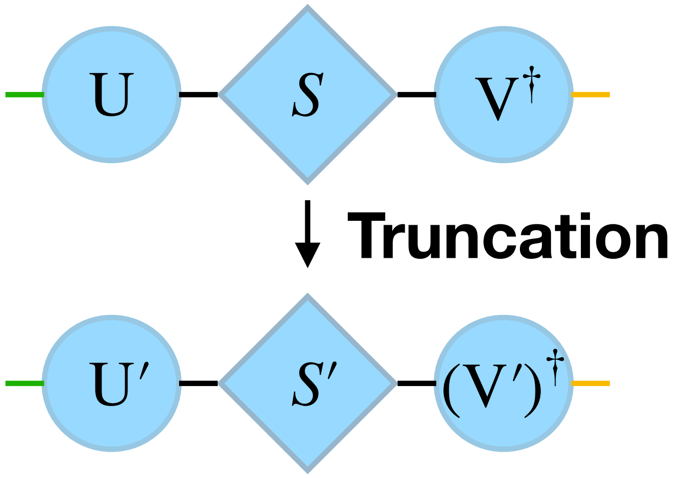
Truncation

This truncation after the SVD is the core of tensor networks algorithms, and enables the efficient compression of information

Truncation

This truncation after the SVD is the core of tensor networks algorithms, and enables the efficient compression of information

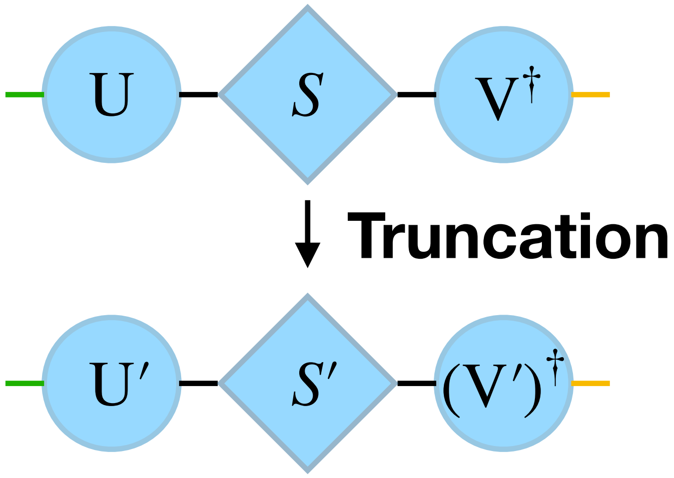
- We call χ_{max} **bond dimension** of the system, and denote with s_1 the greatest eigenvalue of S. Then:



Truncation

This truncation after the SVD is the core of tensor networks algorithms, and enables the efficient compression of information

- We call χ_{max} **bond dimension** of the system, and denote with s_1 the greatest eigenvalue of S. Then:



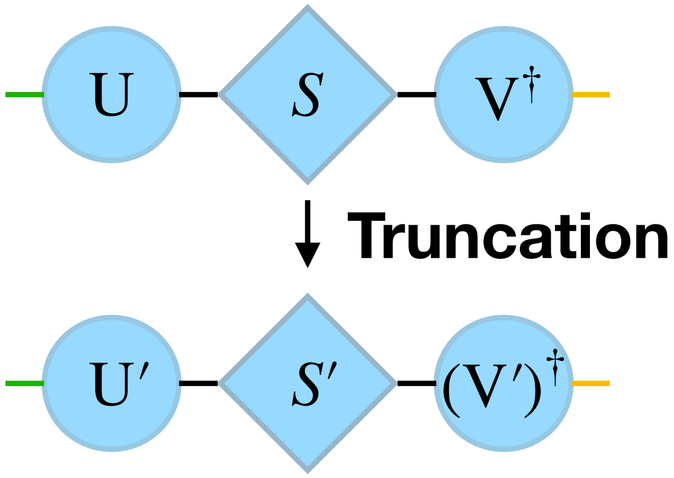
$$S' = \begin{cases} s_i & \textbf{if } \frac{s_i}{s_1} \geq \epsilon \textbf{ and } i \leq \chi_{max} \\ 0 & \textbf{otherwise} \end{cases}$$

We then truncate the 0 term

Truncation

This truncation after the SVD is the core of tensor networks algorithms, and enables the efficient compression of information

- We call χ_{max} **bond dimension** of the system, and denote with s_1 the greatest eigenvalue of S. Then:



$$S' = \begin{cases} s_i & \textbf{if } \frac{s_i}{s_1} \geq \epsilon \textbf{ and } i \leq \chi_{max} \\ 0 & \textbf{otherwise} \end{cases}$$

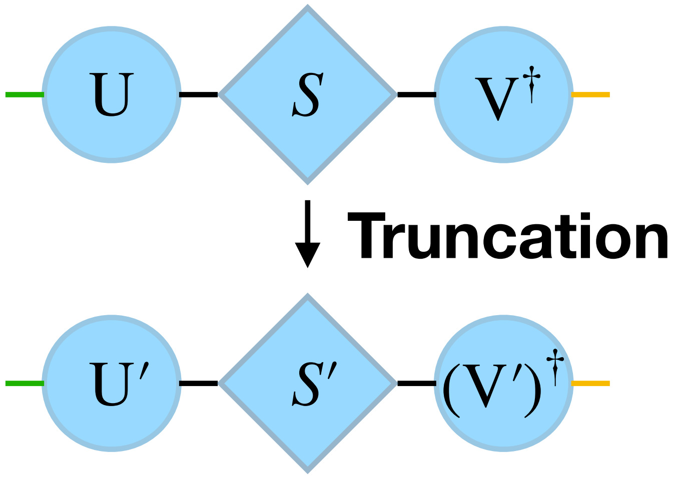
We then truncate the 0 term

We keep the eigenvalues only if they are **big enough**. In this way, we are neglecting the sub-leading term for the state description.

Truncation

This truncation after the SVD is the core of tensor networks algorithms, and enables the efficient compression of information

- We call χ_{max} **bond dimension** of the system, and denote with s_1 the greatest eigenvalue of S. Then:



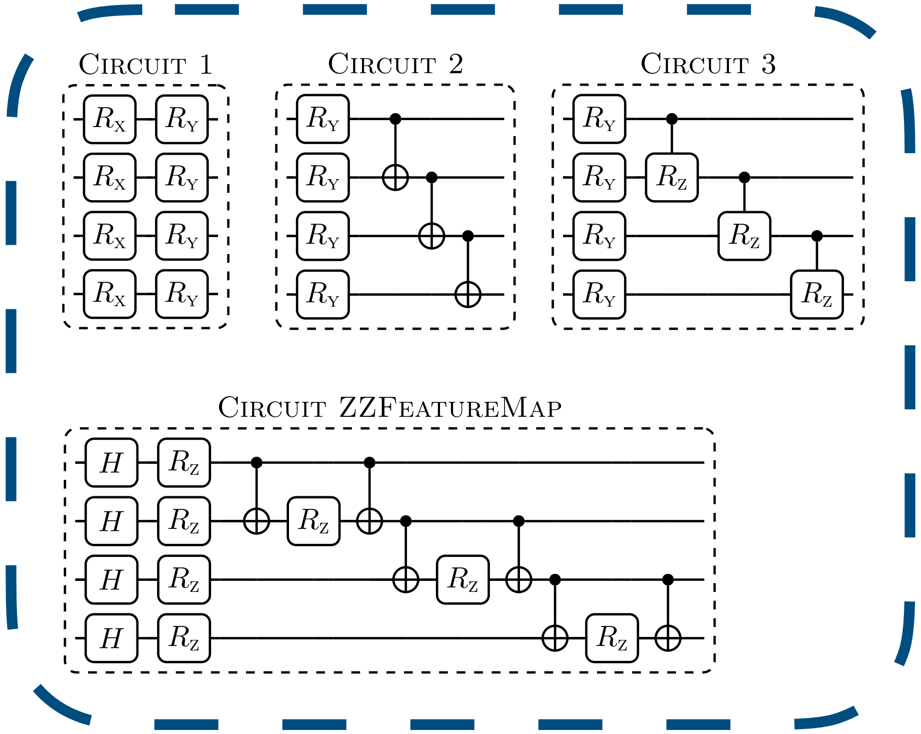
$$S' = \begin{cases} s_i & \textbf{if } \frac{s_i}{s_1} \geq \epsilon \textbf{ and } i \leq \chi_{max} \\ 0 & \textbf{otherwise} \end{cases}$$

We then truncate the 0 term

We keep the eigenvalues only if they are **big enough**. In this way, we are neglecting the sub-leading term for the state description.

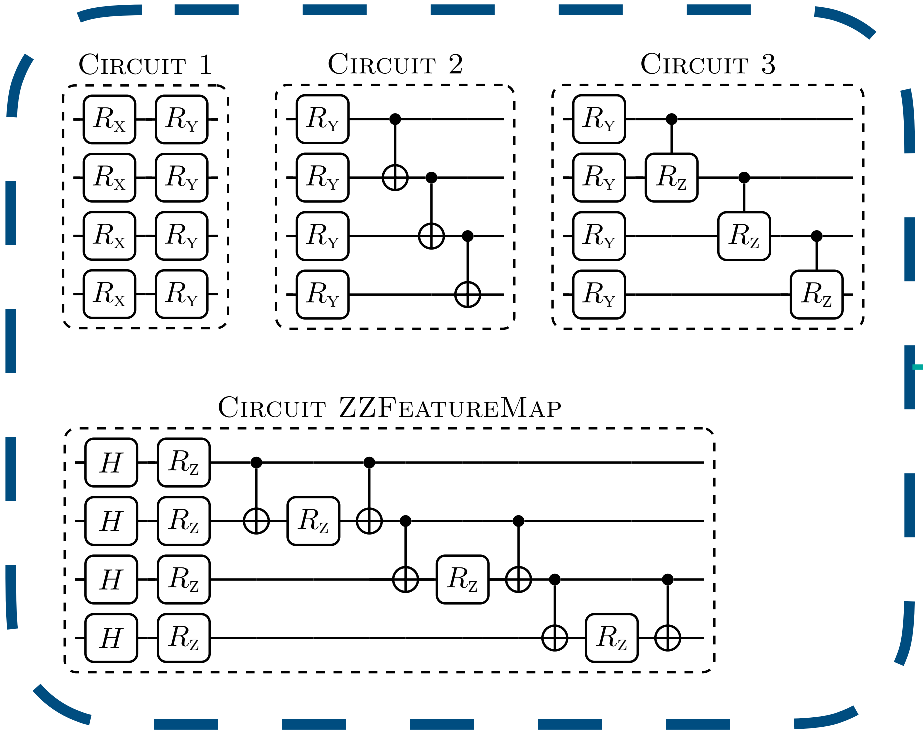
We keep only the **first highest** χ_{max} eigenvalues. In this way, we keep the quantum state manageable even for big number of qubits. However, this may be a strong approximation.

Methods



Pick a feature map $\mathscr{F}(x)$
and a variational ansatz $V(\theta)$

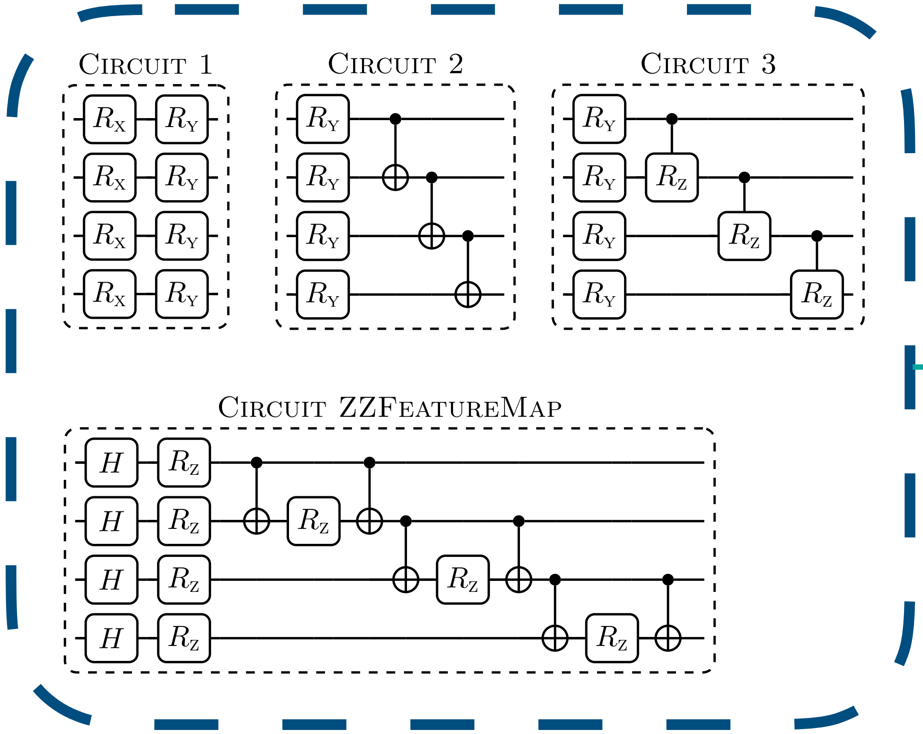
Methods



Select number of layers L

Pick a feature map $\mathscr{F}(x)$ and a variational ansatz $V(\theta)$

Methods

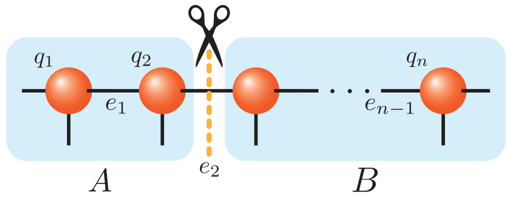


Pick a feature map $\mathscr{F}(x)$
and a variational ansatz $V(\theta)$

Select number
of layers L

Sample parameters
$x_i, \theta_i \sim \text{Unif}(0,\pi)$

If L is big enough
we create a
Haar-random state

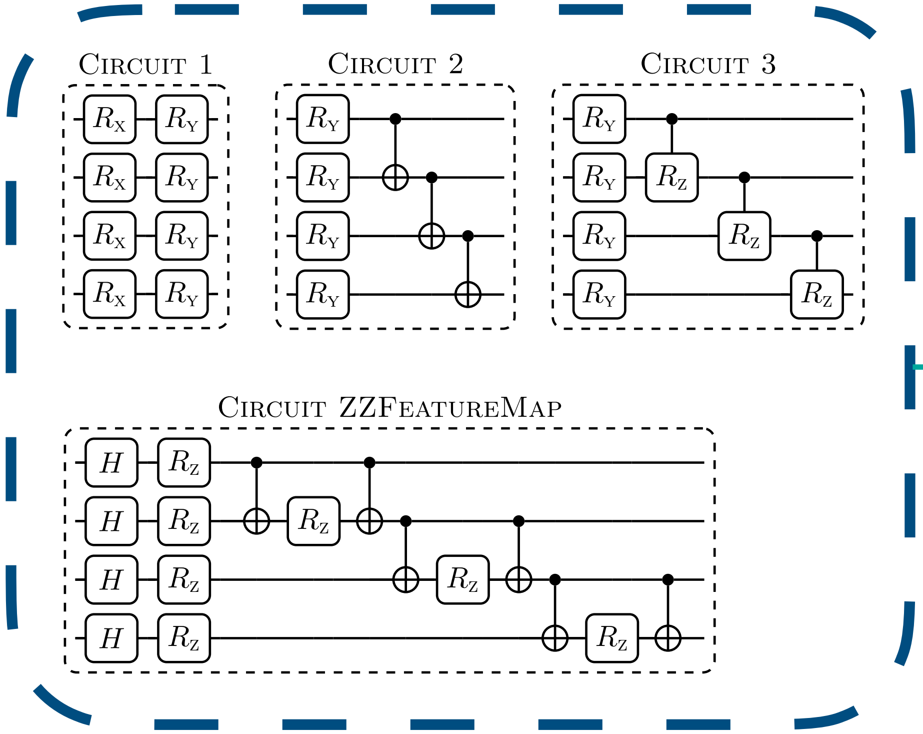


Simulate the evolution
and compute final state

Compute entanglement
on each edge e_i

Repeat M times and take the
average over the realisations

Methods

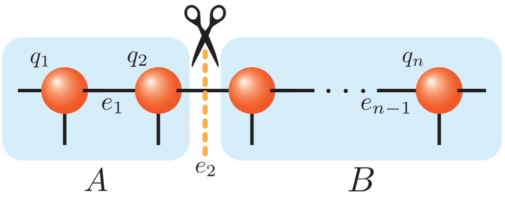


Pick a feature map $\mathscr{F}(x)$
and a variational ansatz $V(\theta)$

Select number
of layers L

Sample parameters
$x_i, \theta_i \sim$ Unif$(0,\pi)$

If L is big enough
we create a
Haar-random state



Simulate the evolution
and compute final state

Compute entanglement
on each edge e_i

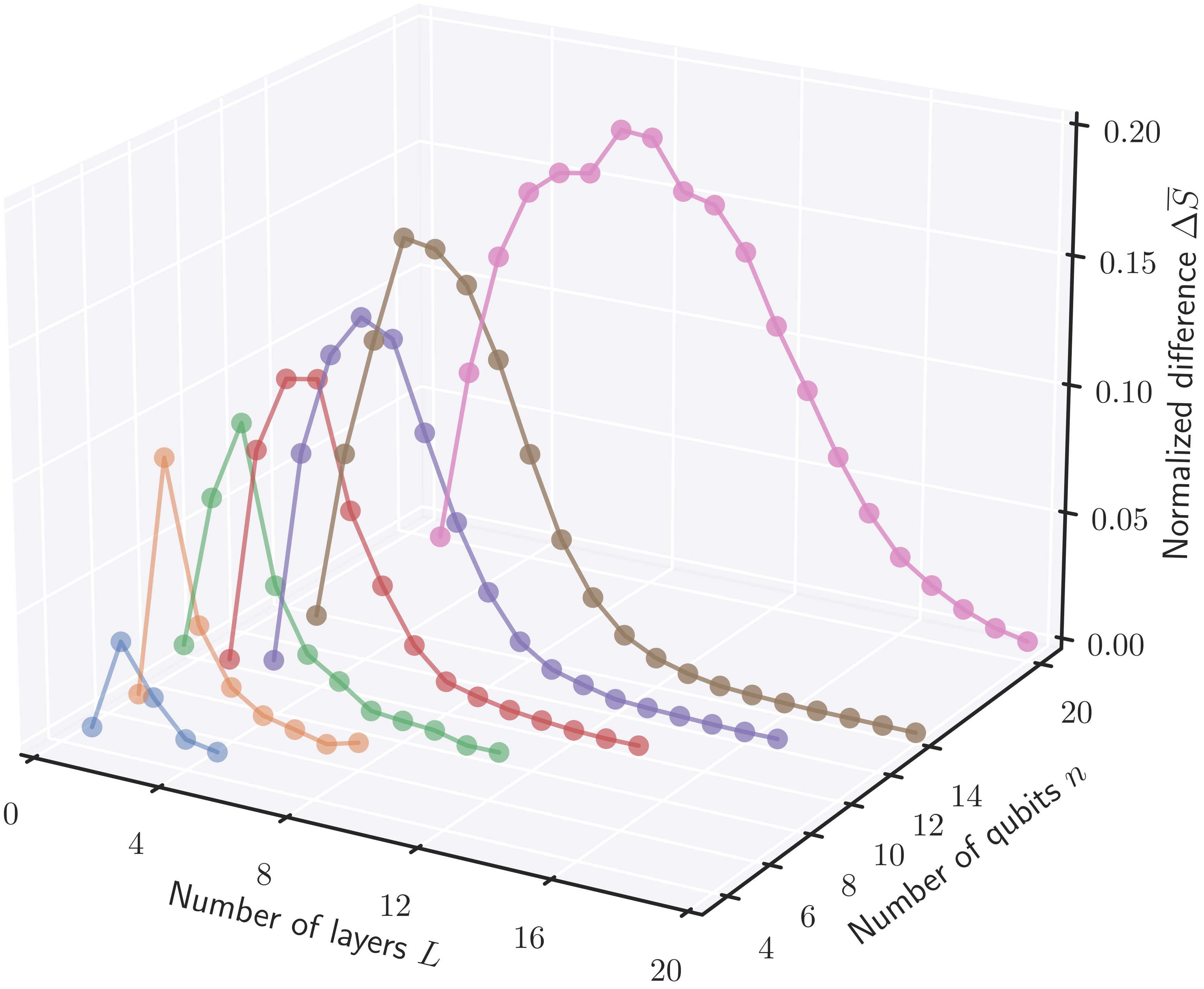
Repeat M times and take the
average over the realisations

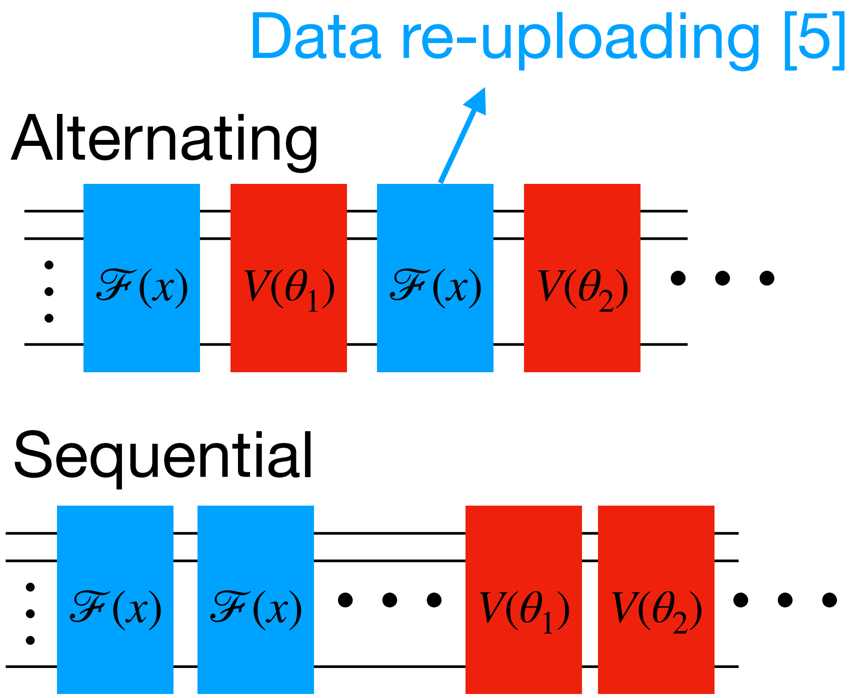
Qiskit up to ~ 16 qubits
with state vector

Efficient simulation

Quantum matcha tea
Tensor network emulator
up to 50 qubits

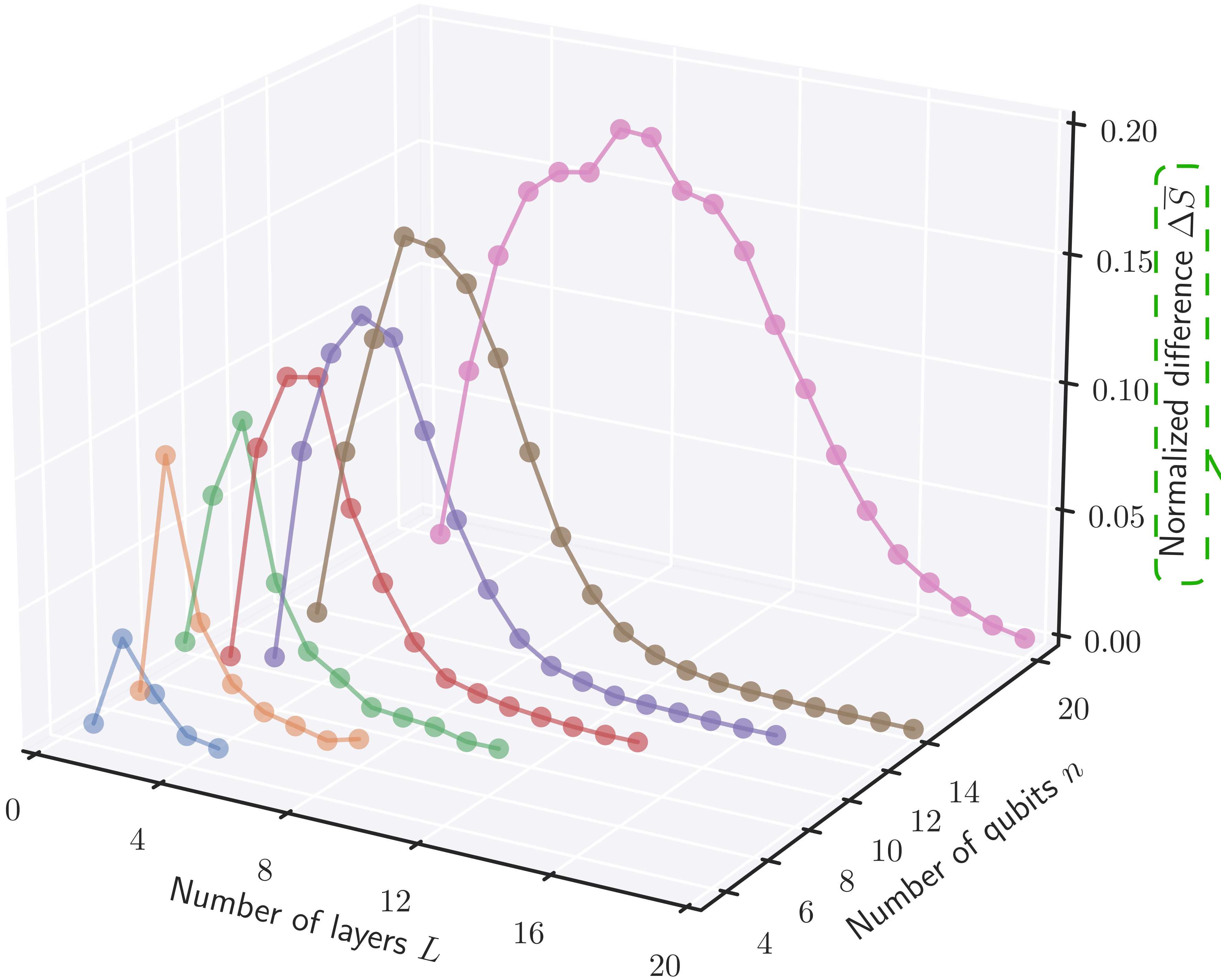
Alternating VS sequential



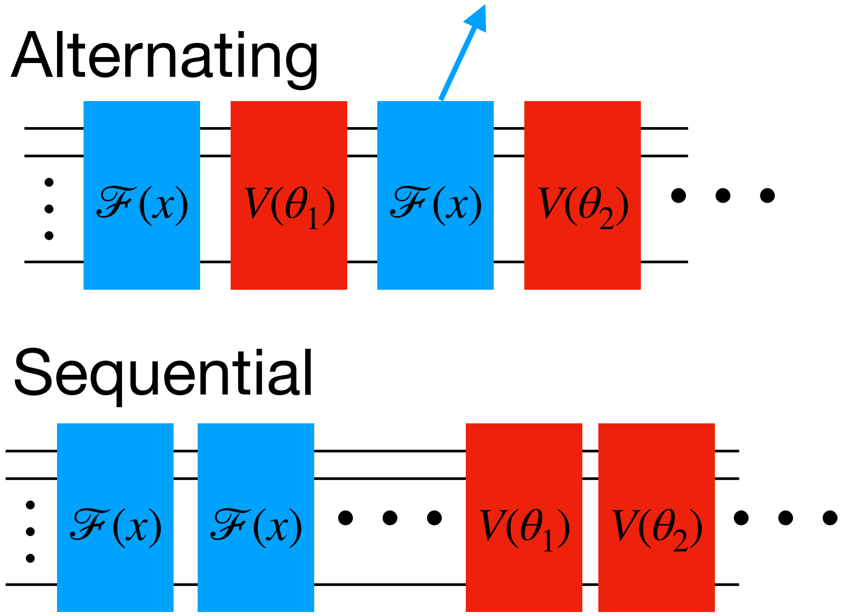


[5] Pérez-Salinas, Adrián, et al. "Data re-uploading for a universal quantum classifier." *Quantum* 4 (2020): 226.

Alternating VS sequential



Data re-uploading [5]

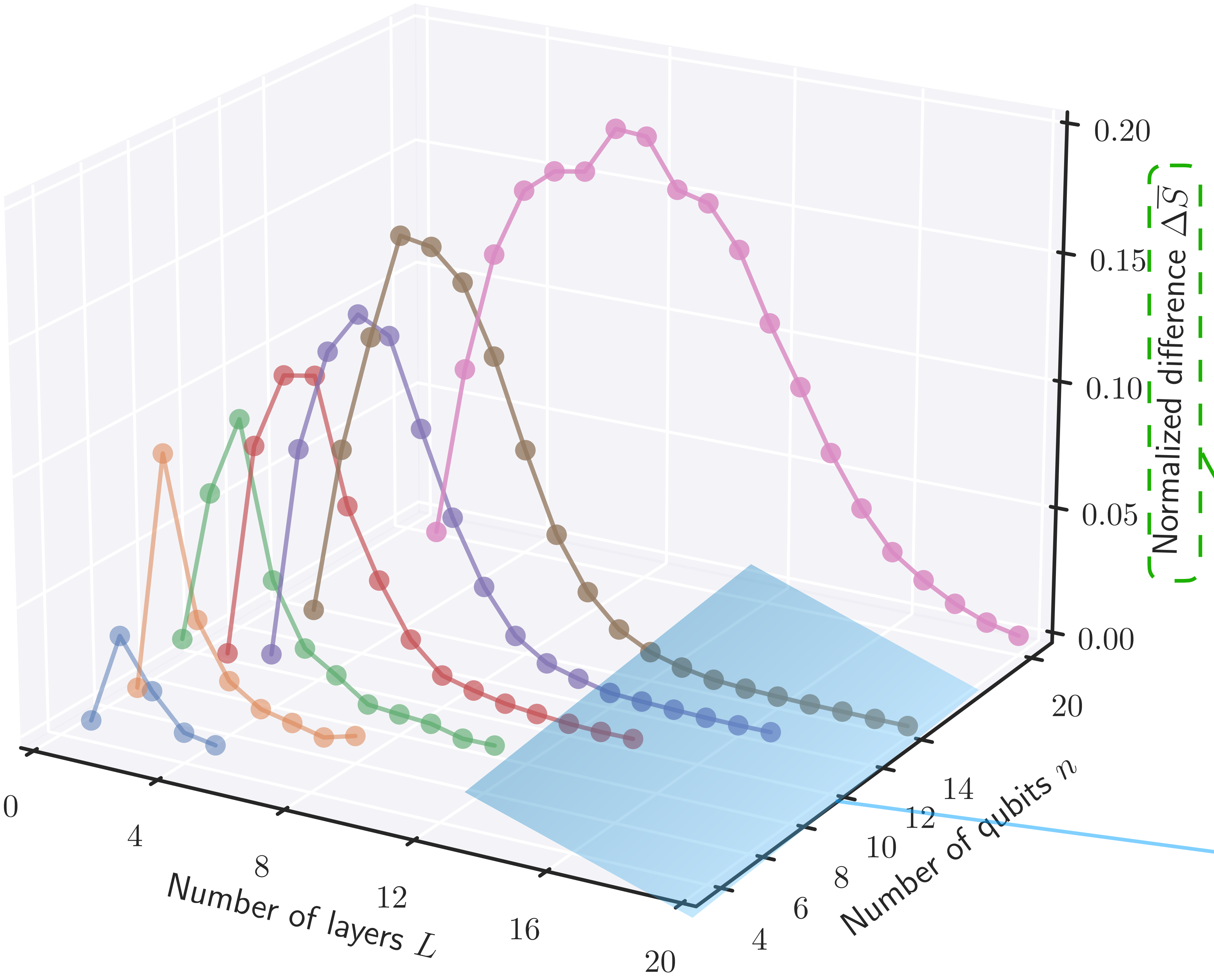


$$\Delta\bar{S} = \frac{S^{alt} - S^{seq}}{(S^{alt} + S^{seq})/2}$$

---

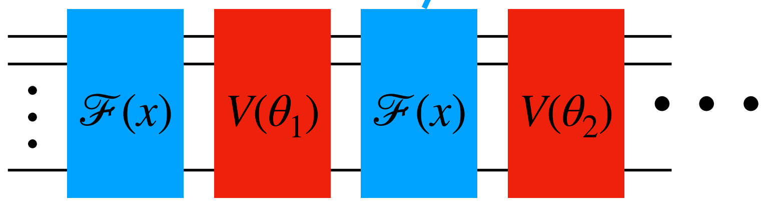
[5] Pérez-Salinas, Adrián, et al. "Data re-uploading for a universal quantum classifier." *Quantum* 4 (2020): 226.

Alternating VS sequential

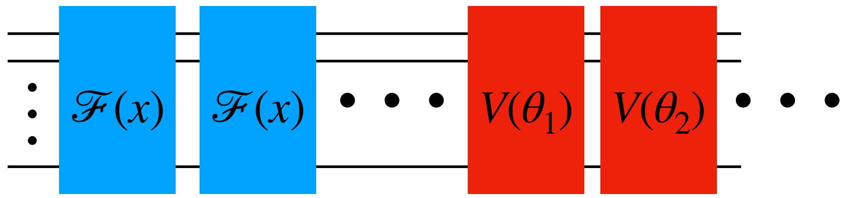


Data re-uploading [5]

Alternating



Sequential



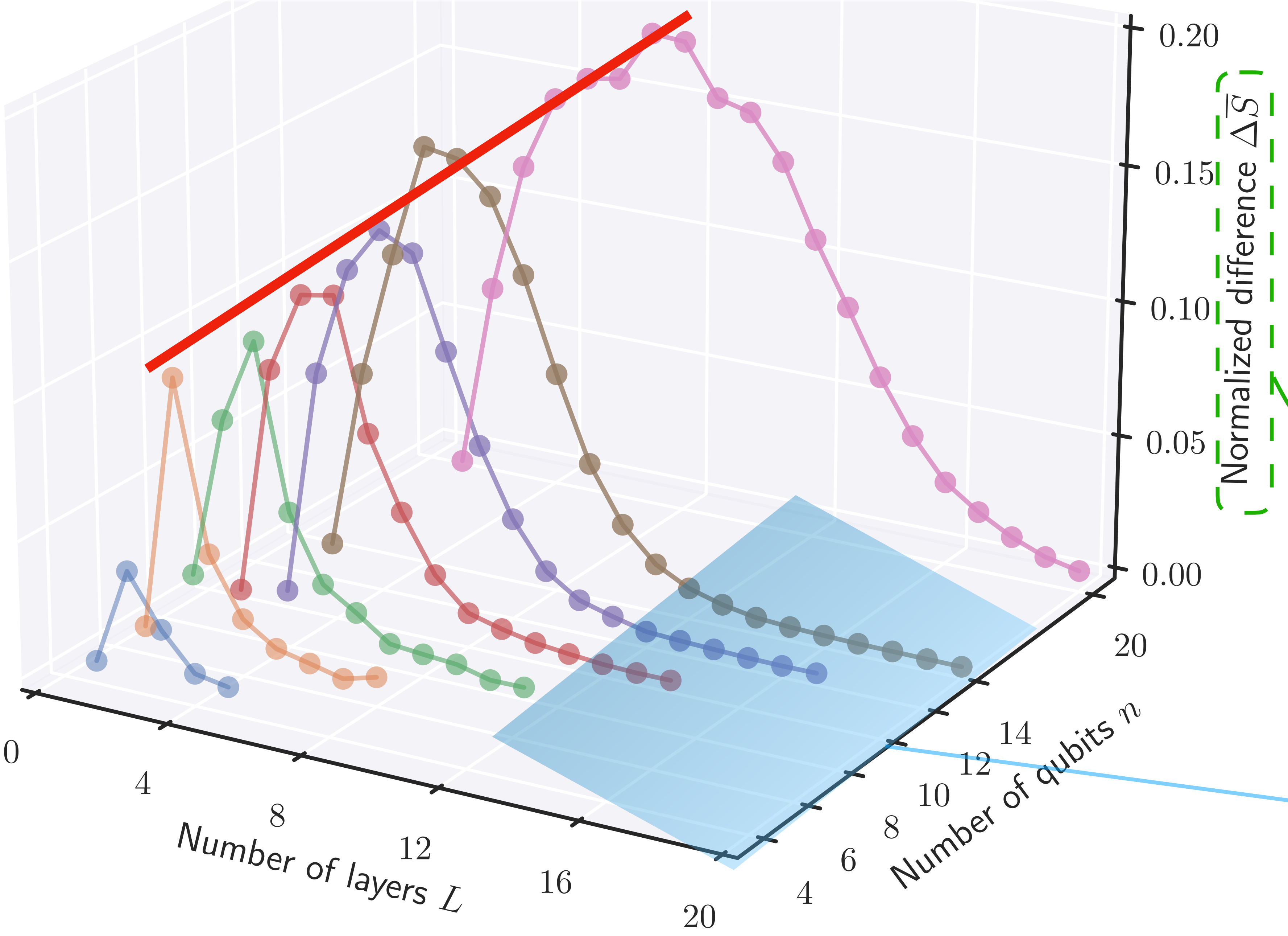
$$\Delta\bar{S} = \frac{S^{alt} - S^{seq}}{(S^{alt} + S^{seq})/2}$$

Both tends towards Haar-random

[5] Pérez-Salinas, Adrián, et al. "Data re-uploading for a universal quantum classifier." *Quantum* 4 (2020): 226.

Alternating VS sequential

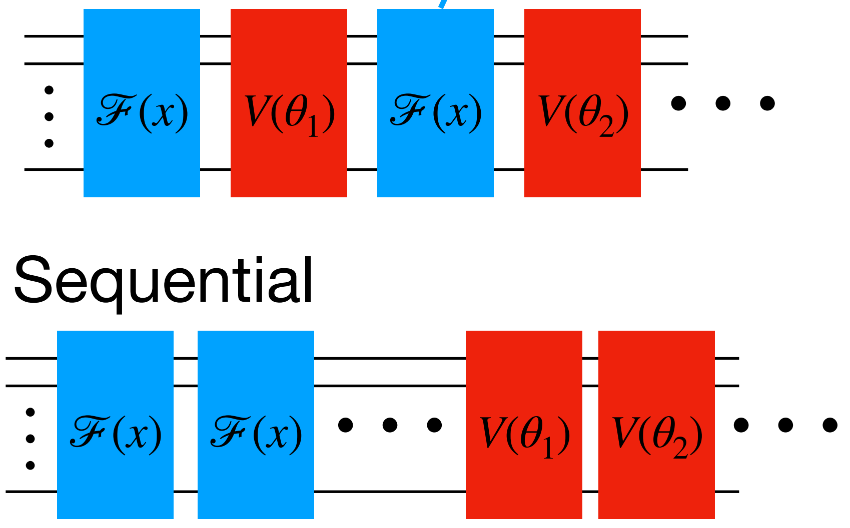
Significative difference that increases with the number of qubits n, up to almost $20\ \%$ for $n = 20$



Data re-uploading [5]

Alternating

Sequential

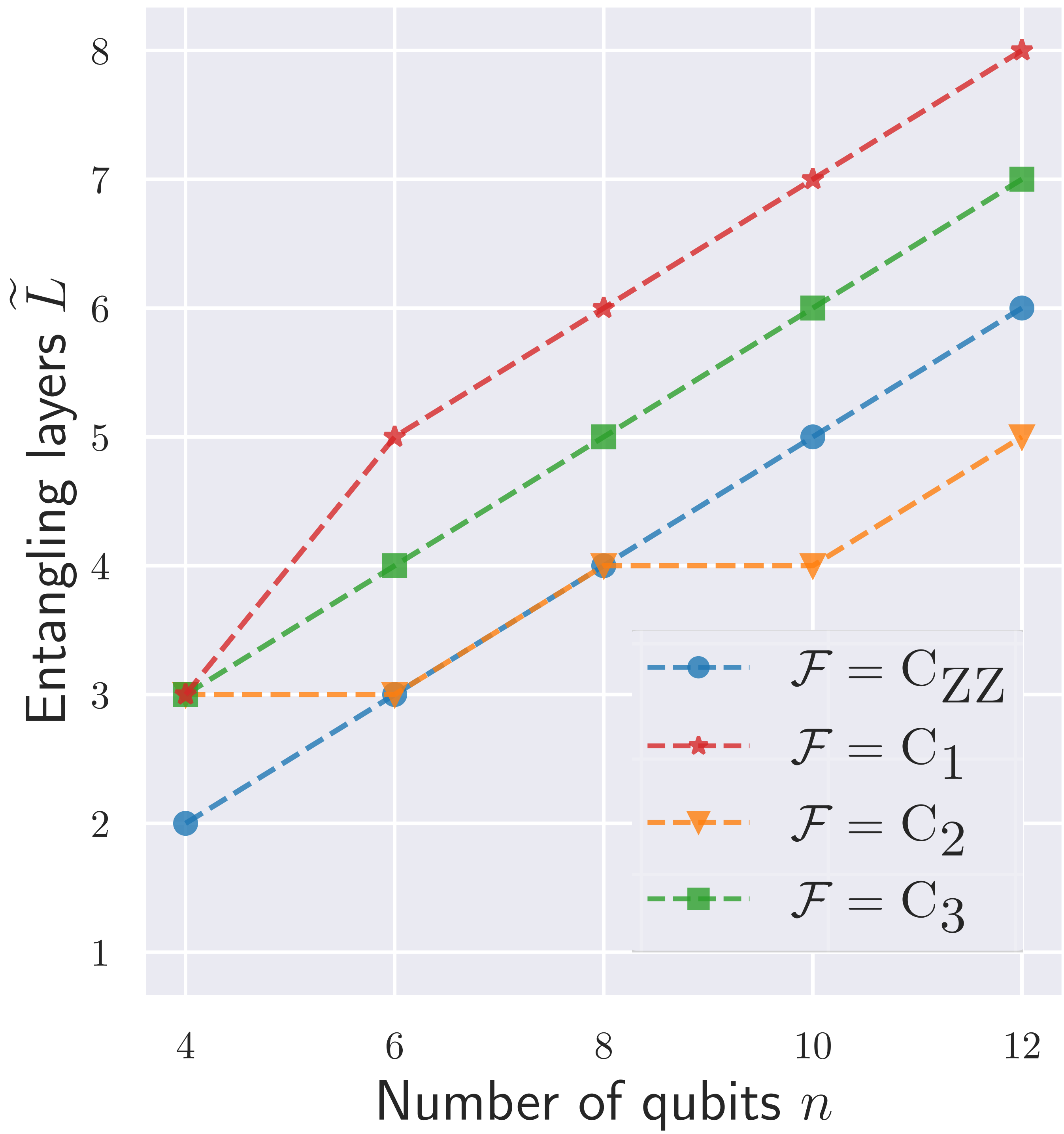


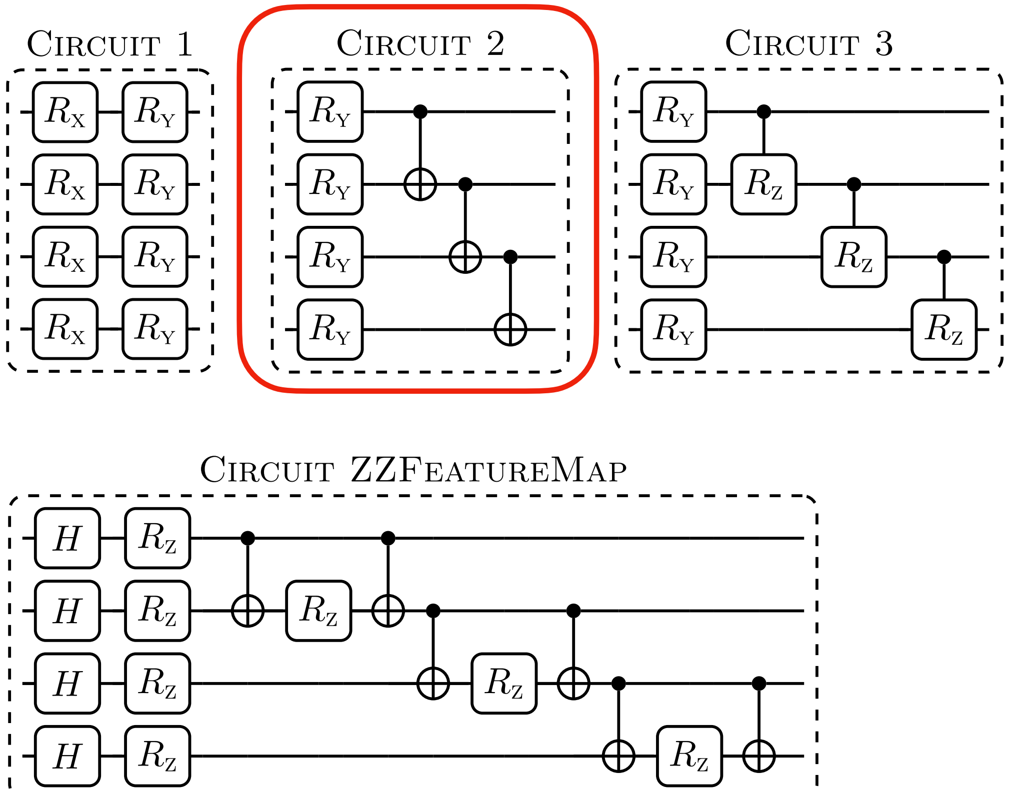
$$\Delta\bar{S} = \frac{S^{alt} - S^{seq}}{(S^{alt} + S^{seq})/2}$$

Both tends towards Haar-random

[5] Pérez-Salinas, Adrián, et al. "Data re-uploading for a universal quantum classifier." *Quantum* 4 (2020): 226.

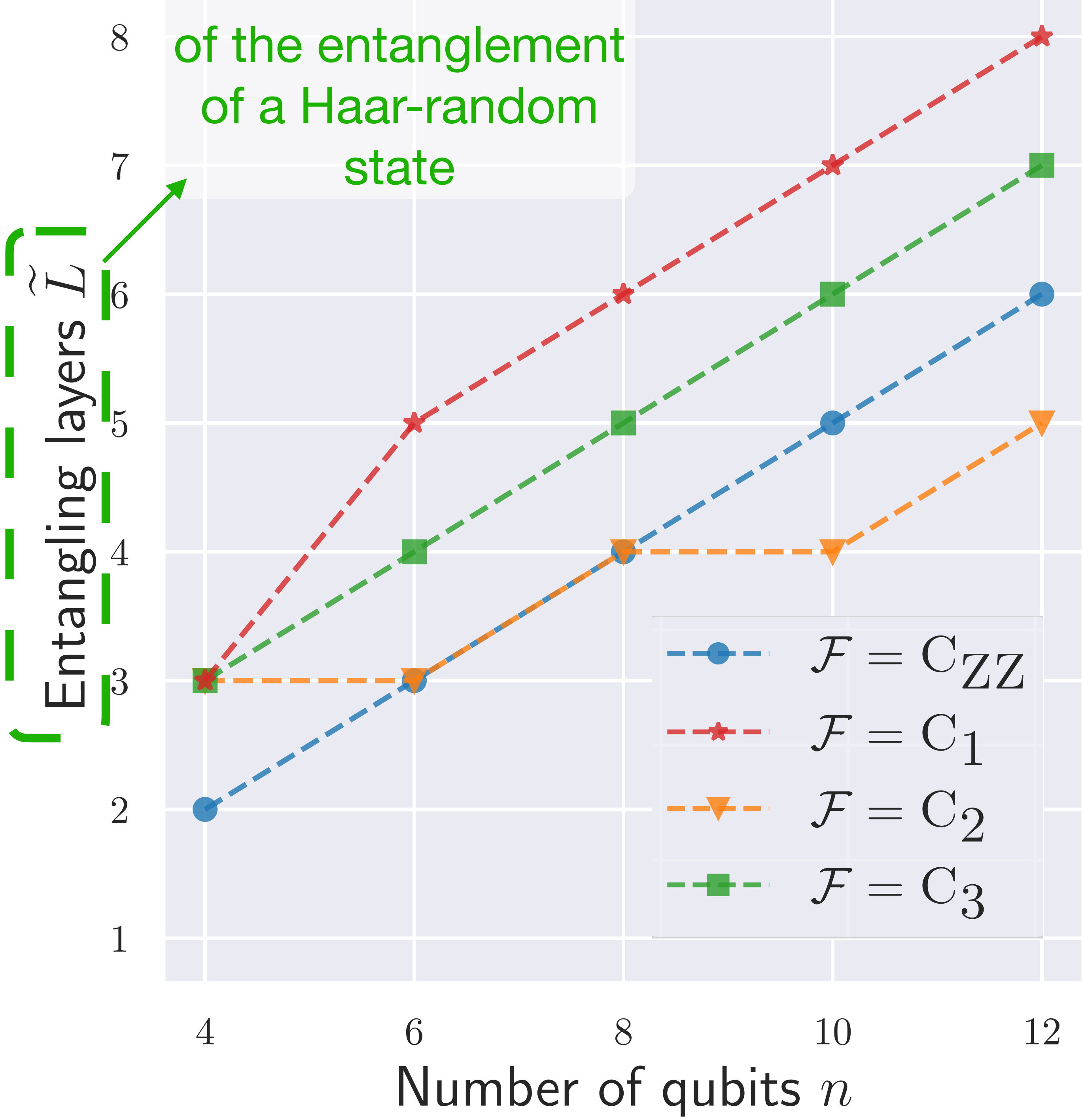
Entanglement scaling with depth





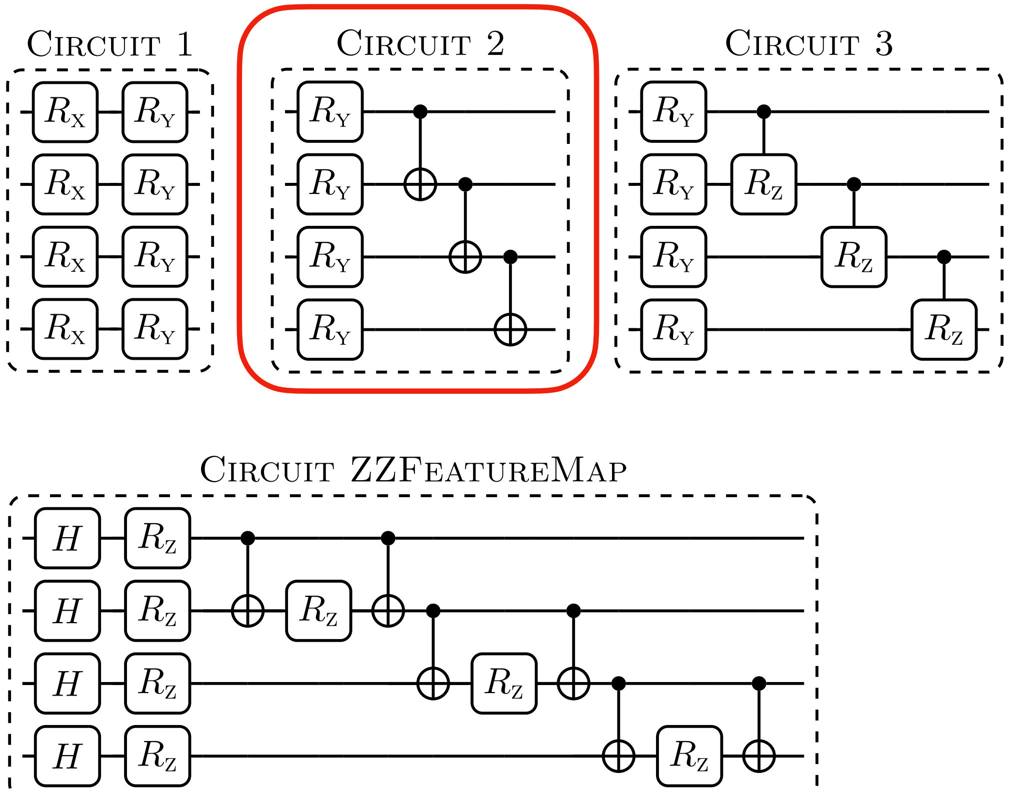
C_2 is always used as variational ansatz

Entanglement scaling with depth

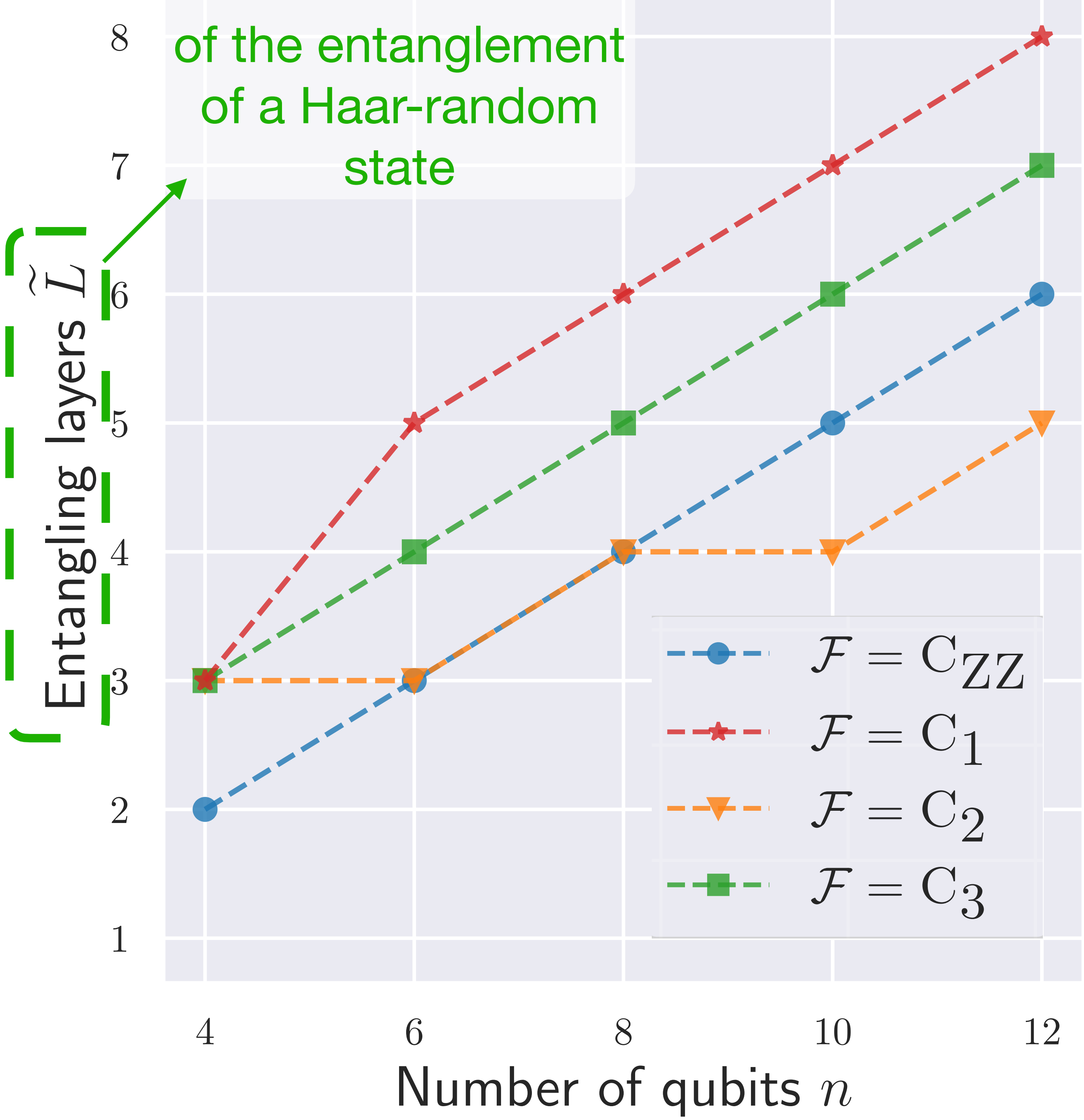


Layers needed to reach $90\,\%$ of the entanglement of a Haar-random state

C_2 is always used as variational ansatz



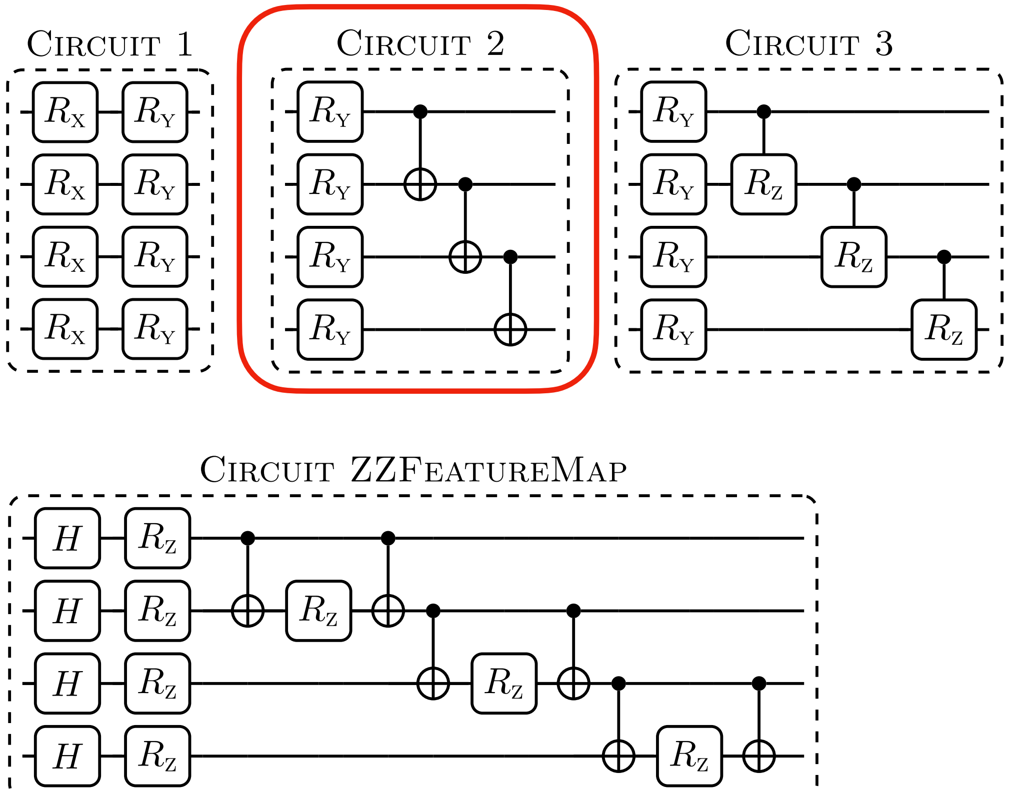
Entanglement scaling with depth



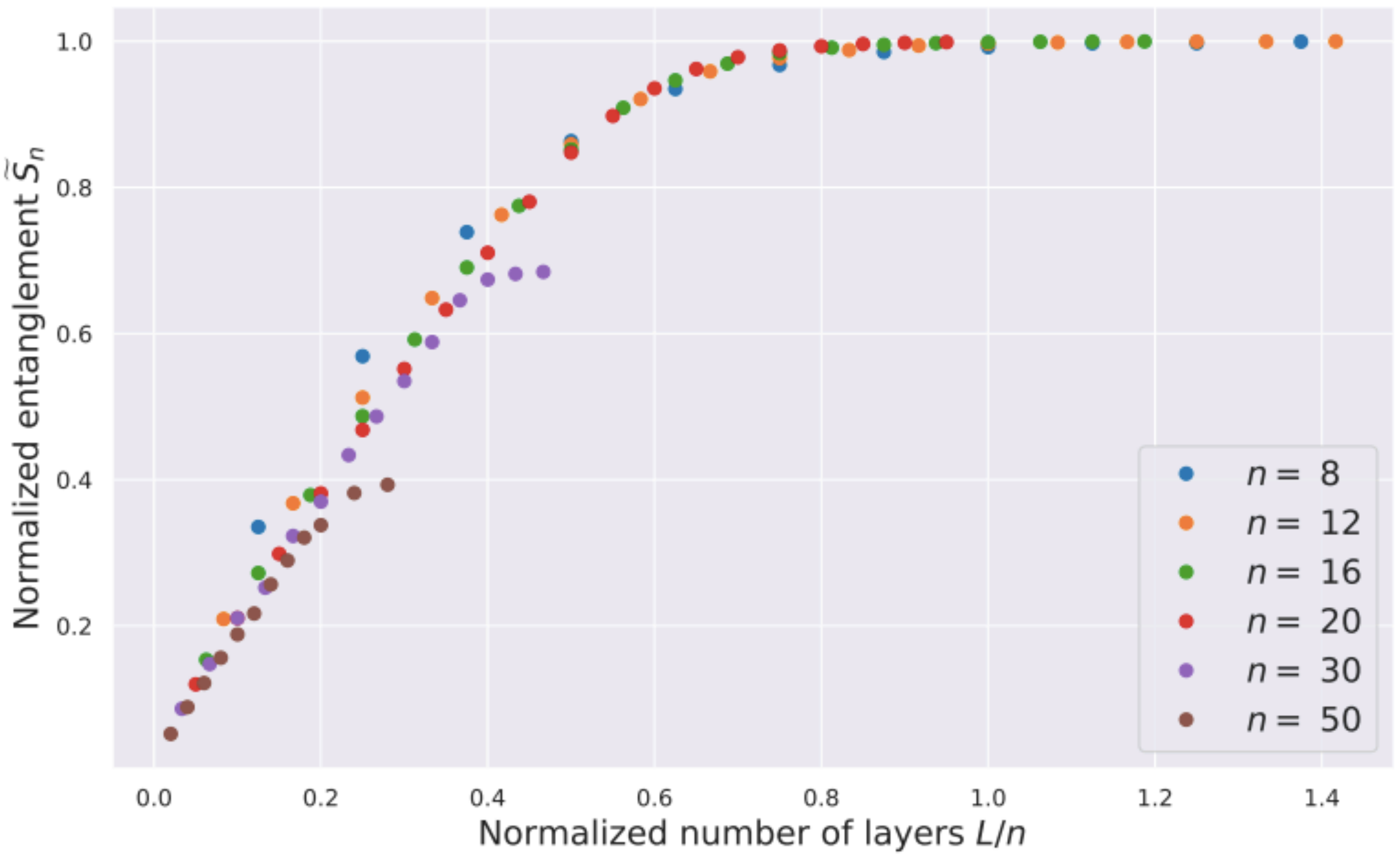
Layers needed
to reach $90\,\%$
of the entanglement
of a Haar-random
state

Number of layers
scales **linearly** with
number of qubits n
to reach Haar-random state

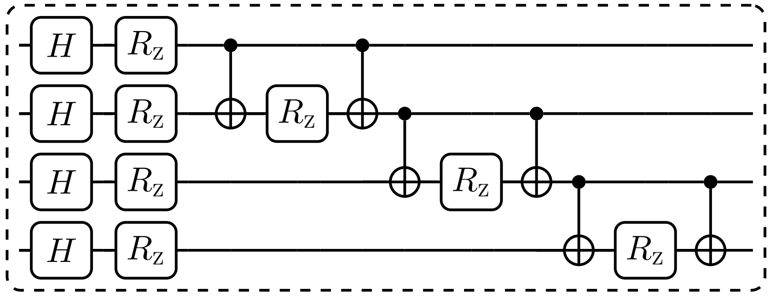
C_2 is always used as
variational ansatz



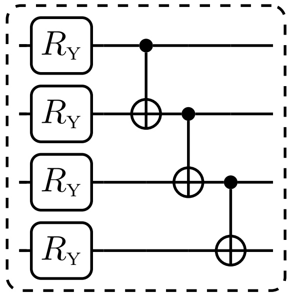
Entangling speed



Circuit ZZFeatureMap



Circuit 2

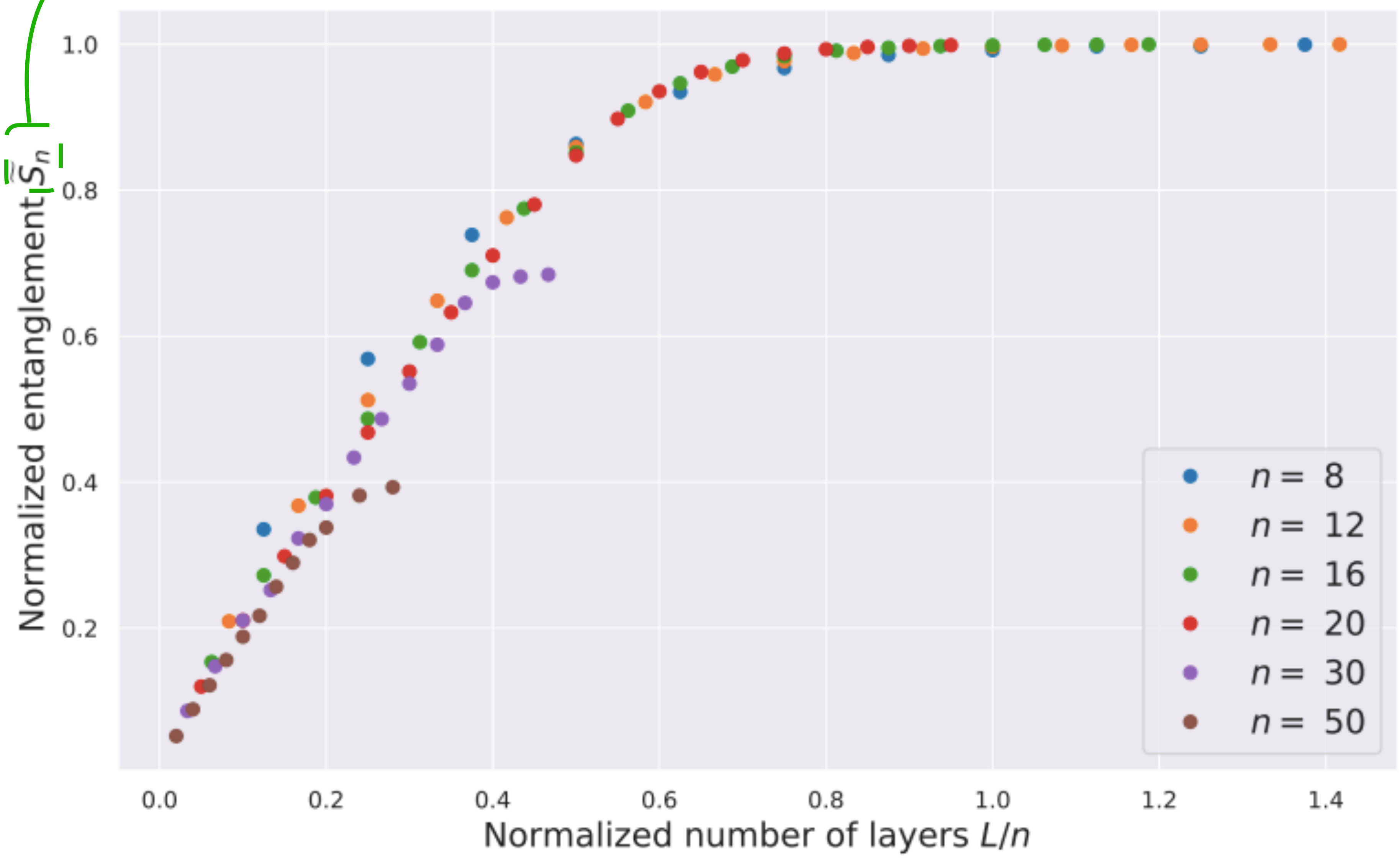


Feature map	Variational Ansatz	Entangling speed
C_{zz}	C_2	(1.8 ± 0.1)
C_{zz}	C_3	(0.59 ± 0.02)
C_1	C_3	(0.316 ± 0.006)

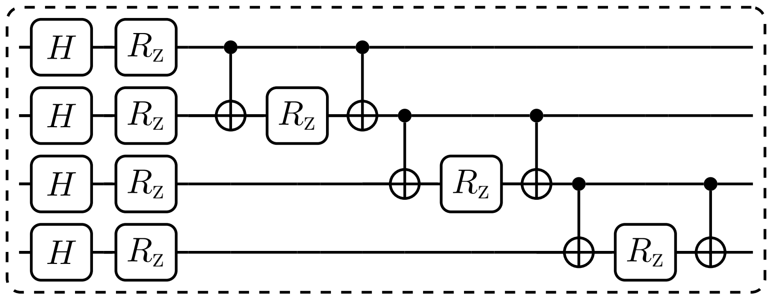
Entangling speed

$$\widetilde{S}_n = \frac{\max_{e_i}\left[S(e_i)\right]}{S^{Haar}_{n,max}}$$

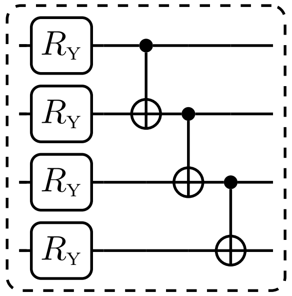
Maximum entanglement of the chain normalised by the maximum entanglement of a Haar-random state



Circuit ZZFeatureMap



Circuit 2

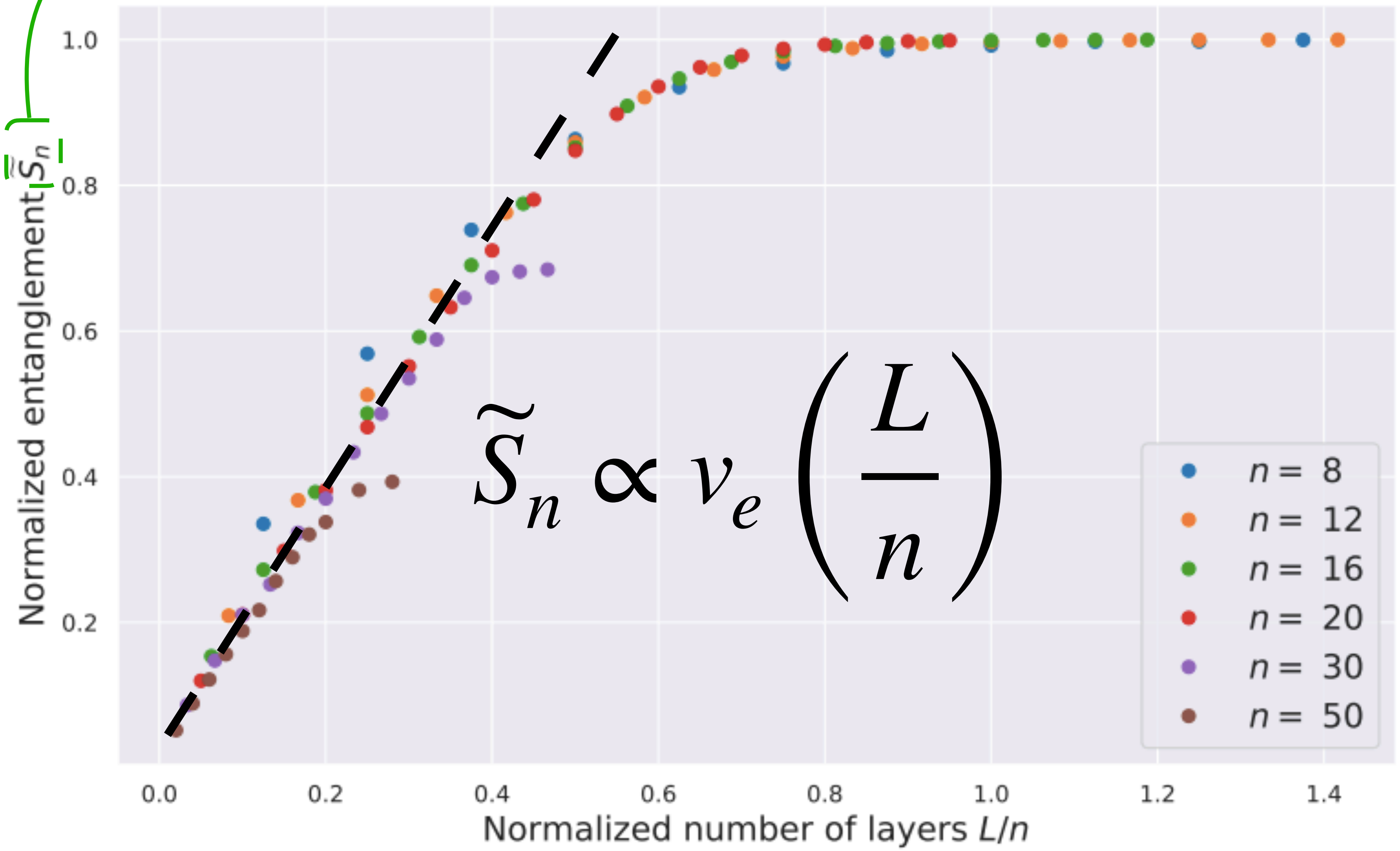


Feature map	Variational Ansatz	Entangling speed
C_{zz}	C_2	(1.8 ± 0.1)
C_{zz}	C_3	(0.59 ± 0.02)
C_1	C_3	(0.316 ± 0.006)

Entangling speed

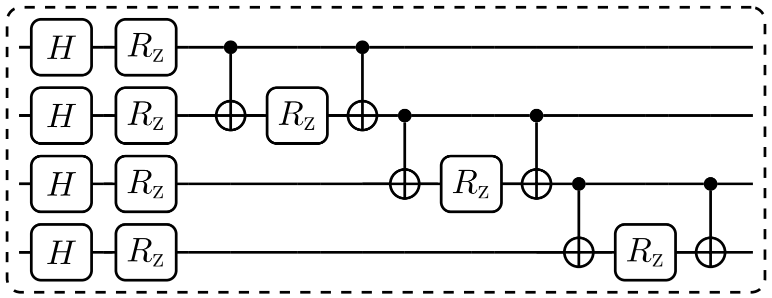
$$\widetilde{S}_n = \frac{\mathbf{max}_{e_i}\left[S(e_i)\right]}{S^{Haar}_{n,max}}$$

Maximum entanglement of the chain normalised by the maximum entanglement of a Haar-random state

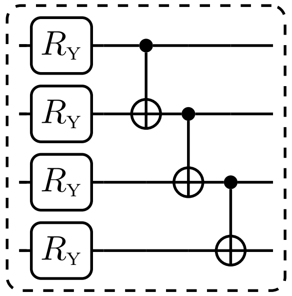


$$\widetilde{S}_n \propto v_e\left(\frac{L}{n}\right)$$

Circuit ZZFeatureMap



Circuit 2

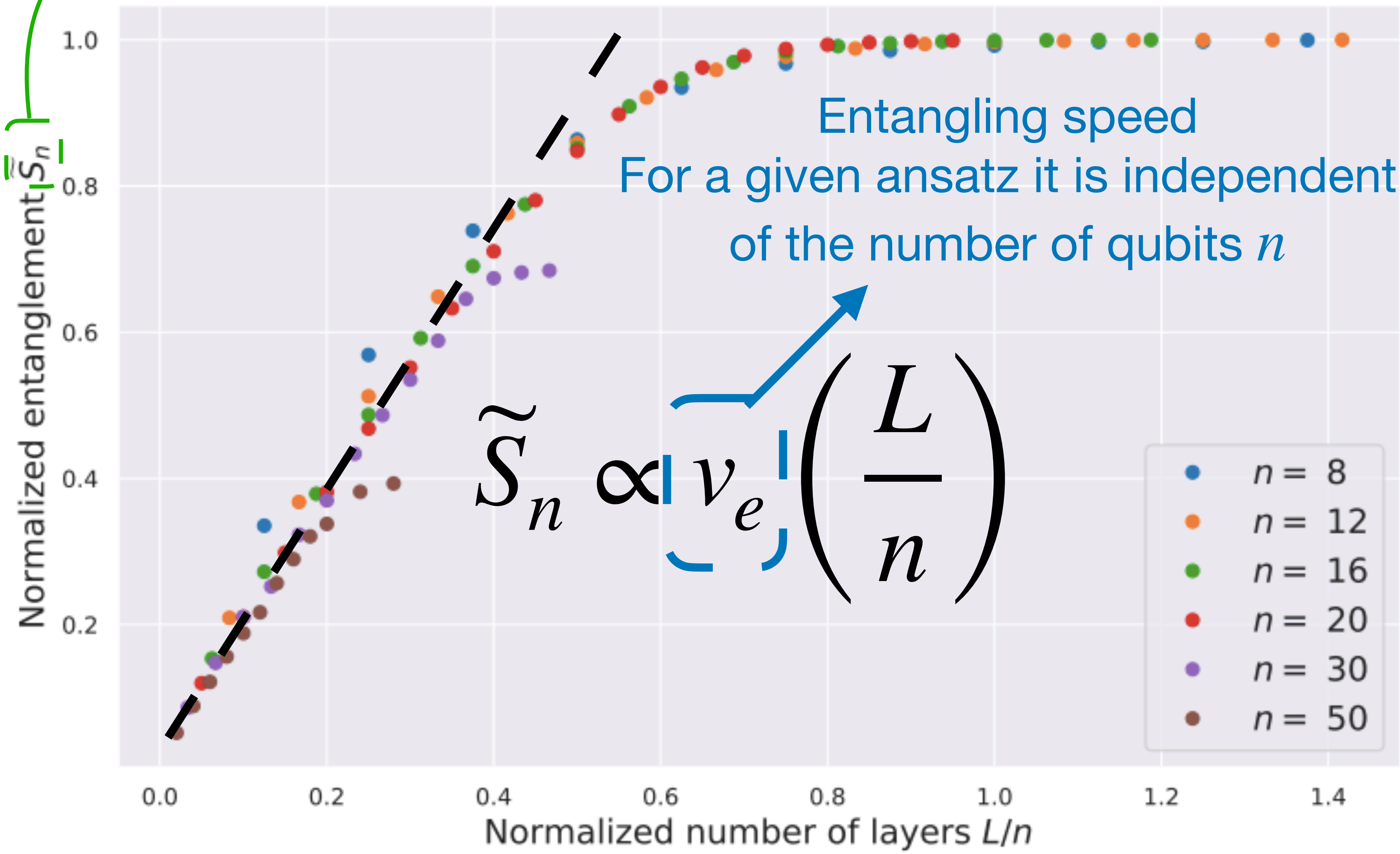


Feature map	**Variational Ansatz**	**Entangling speed**
C_{zz}	C_2	(1.8 ± 0.1)
C_{zz}	C_3	(0.59 ± 0.02)
C_1	C_3	(0.316 ± 0.006)

Entangling speed

$$\widetilde{S}_n = \frac{\max_{e_i}\left[S(e_i)\right]}{S^{Haar}_{n,max}}$$

Maximum entanglement of the chain normalised by the maximum entanglement of a Haar-random state

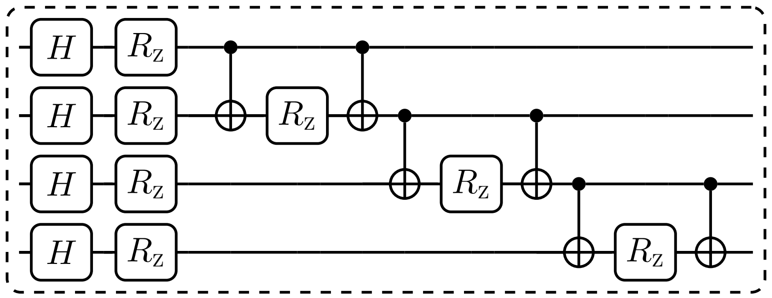


Entangling speed

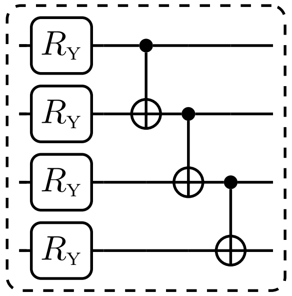
For a given ansatz it is independent of the number of qubits n

$$\widetilde{S}_n \propto v_e\left(\frac{L}{n}\right)$$

Circuit ZZFeatureMap



Circuit 2

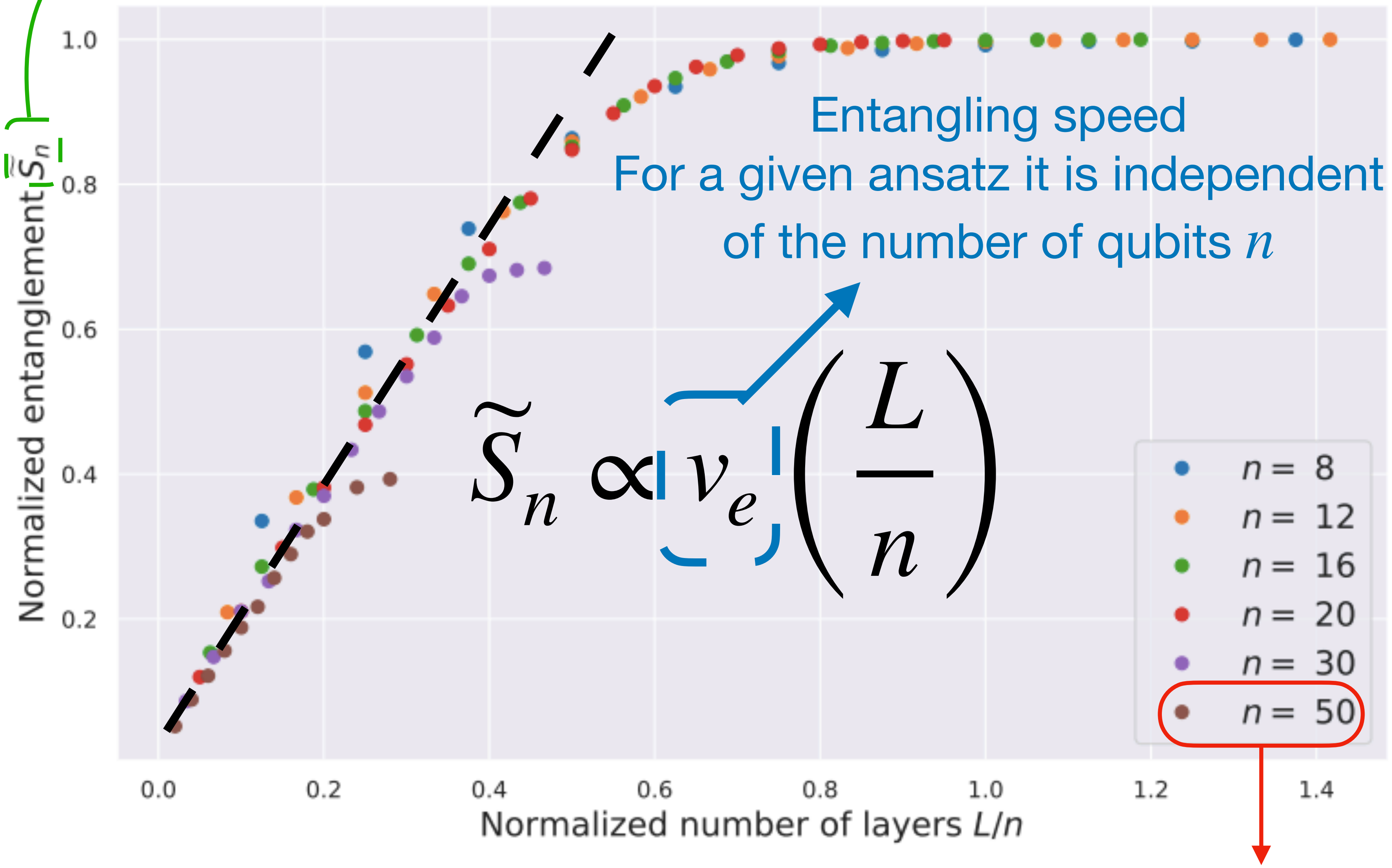


Feature map	Variational Ansatz	Entangling speed
C_{zz}	C_2	(1.8 ± 0.1)
C_{zz}	C_3	(0.59 ± 0.02)
C_1	C_3	(0.316 ± 0.006)

Entangling speed

$$\tilde{S}_n = \frac{\max_{e_i}\left[S(e_i)\right]}{S^{Haar}_{n,max}}$$

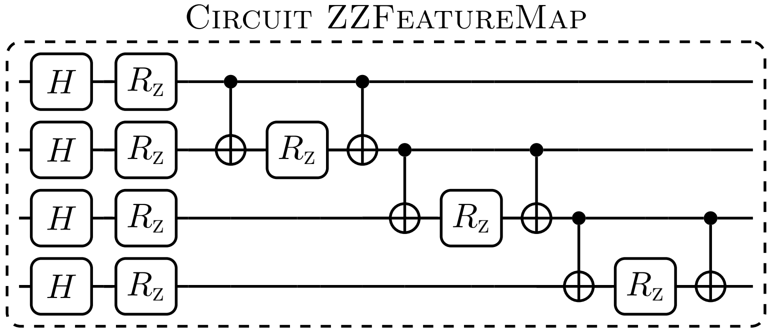
Maximum entanglement of the chain normalised by the maximum entanglement of a Haar-random state

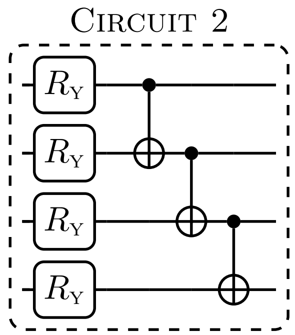


Entangling speed

For a given ansatz it is independent of the number of qubits n

$$\tilde{S}_n \propto v_e \left(\frac{L}{n}\right)$$

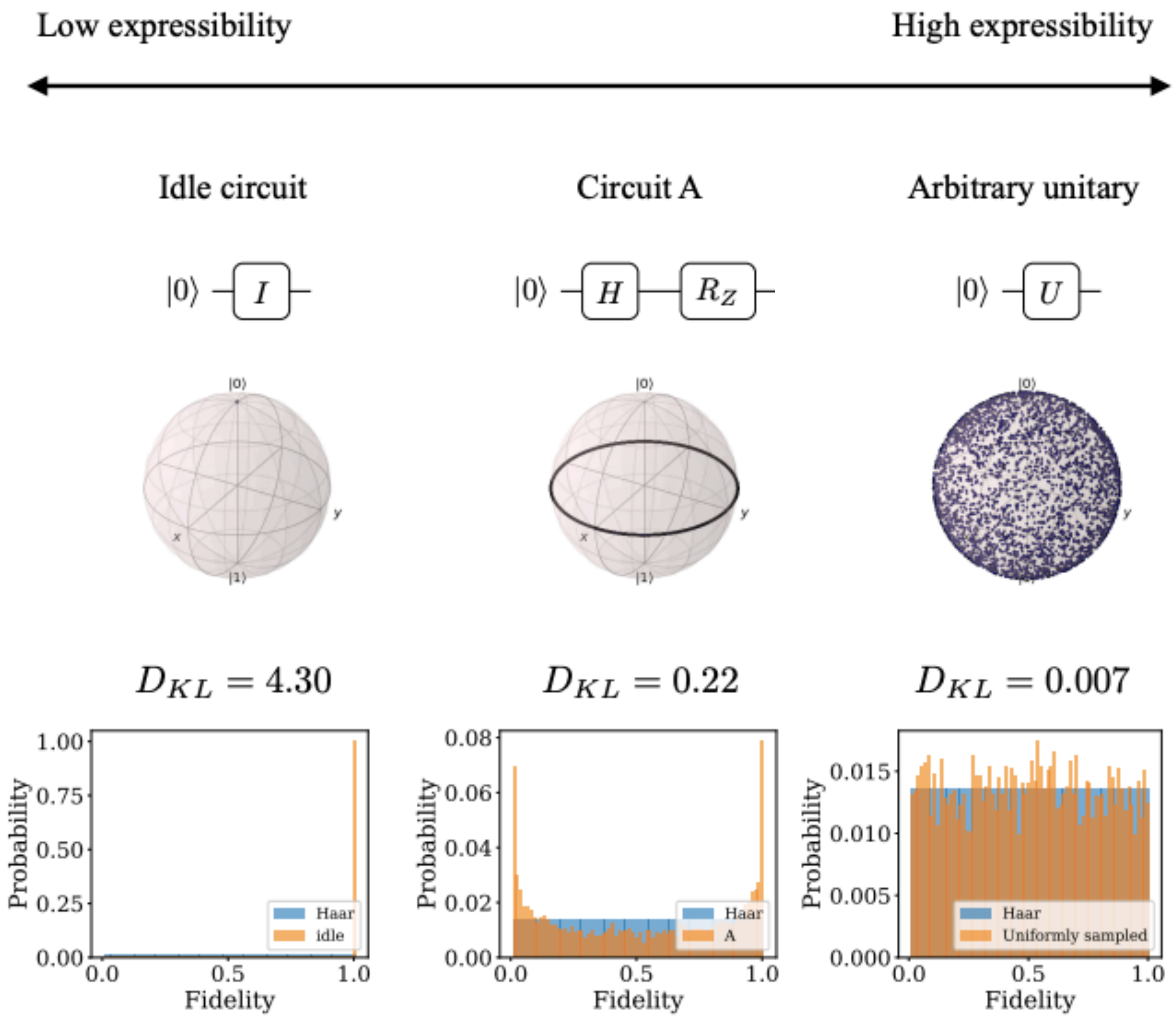




Feature map	**Variational Ansatz**	**Entangling speed**
C_{zz}	C_2	(1.8 ± 0.1)
C_{zz}	C_3	(0.59 ± 0.02)
C_1	C_3	(0.316 ± 0.006)

Using Galileo100 HPC from CINECA

Expressibility



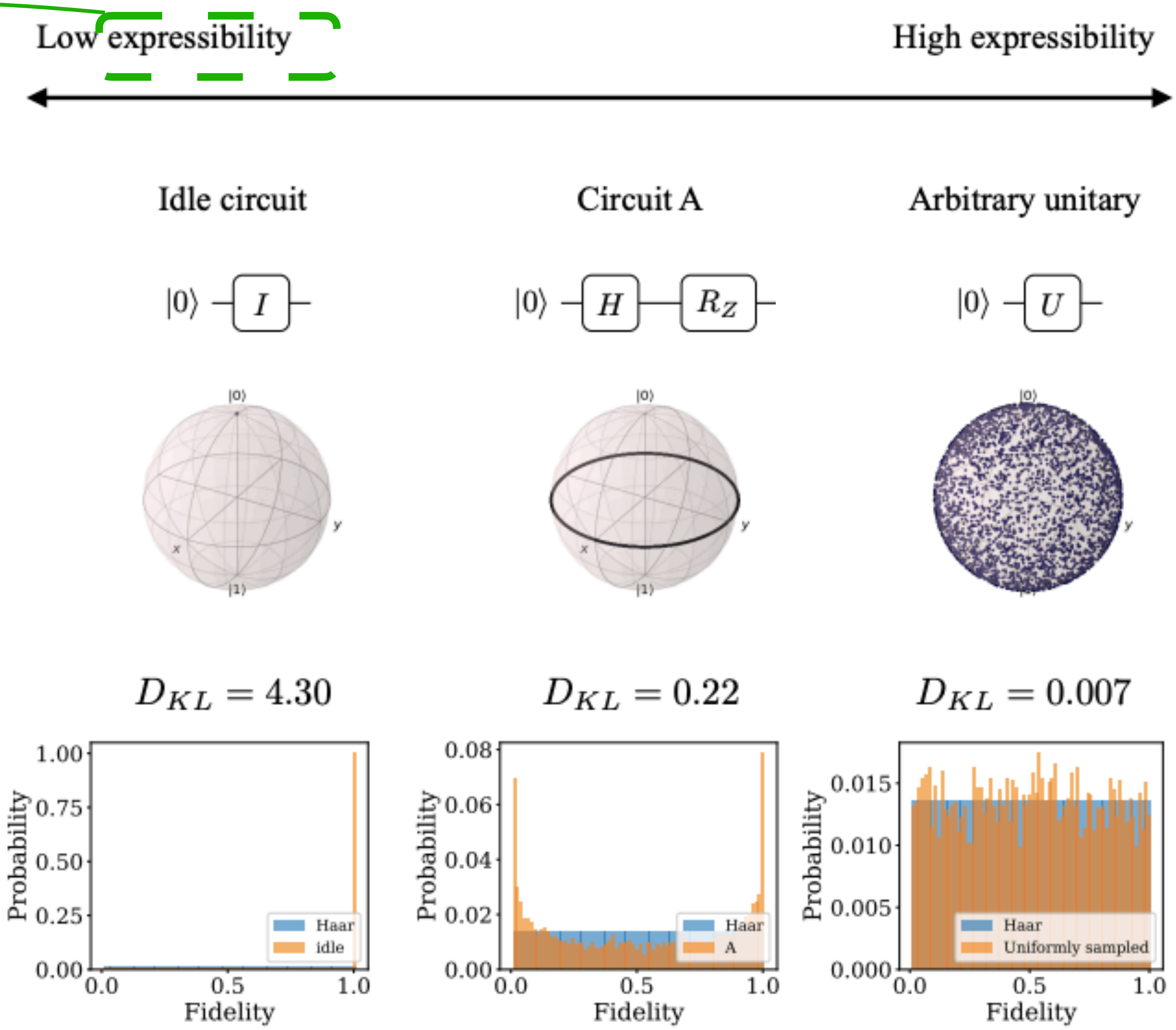
[6] Sim, Sukin, Peter D. Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms." *Advanced Quantum Technologies* 2.12 (2019): 1900070.

Expressibility

Ability to address the full unitary space

Defined as the distance
of the distribution of states
generated by a quantum
circuit from the distribution
of a Haar-random state

$$Expr = D_{KL}\left(\hat{P}_{QNN}(F;\theta)\,||\,P_{Haar}(F)\right)$$



[6] Sim, Sukin, Peter D. Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms." *Advanced Quantum Technologies* 2.12 (2019): 1900070.

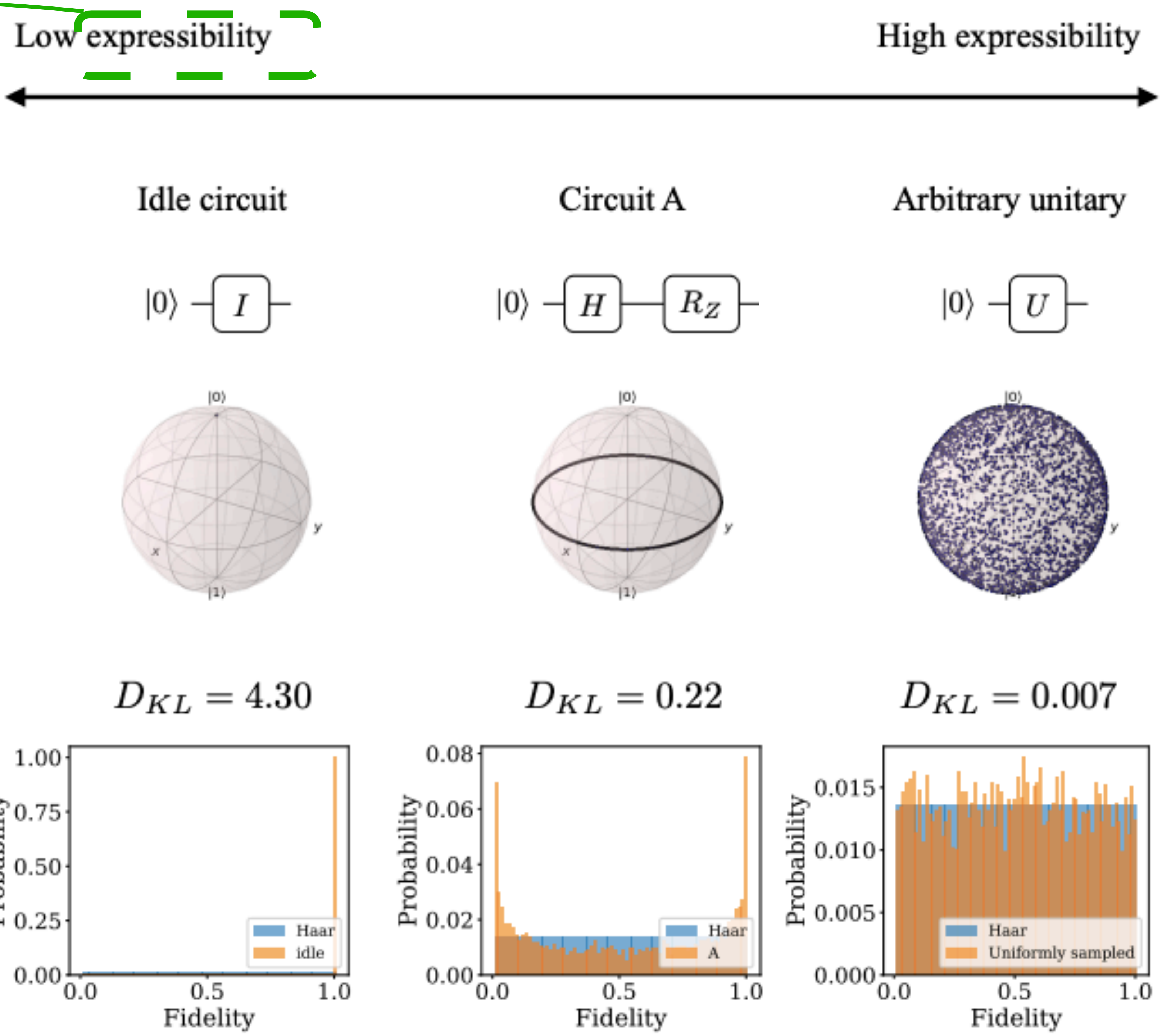
Expressibility

Ability to address the full unitary space

Defined as the distance
of the distribution of states
generated by a quantum
circuit from the distribution
of a Haar-random state

$$Expr = D_{KL}\left(\hat{P}_{QNN}(F;\theta)\,||\,P_{Haar}(F)\right)$$

1. Sample $|\psi_\gamma\rangle, |\psi_\phi\rangle$ from the QNN states
2. Compute the overlap $F = |\langle\psi_\gamma|\psi_\phi\rangle|^2$
3. Repeat many times to obtain statistics



[6] Sim, Sukin, Peter D. Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms." *Advanced Quantum Technologies* 2.12 (2019): 1900070.

Expressibility

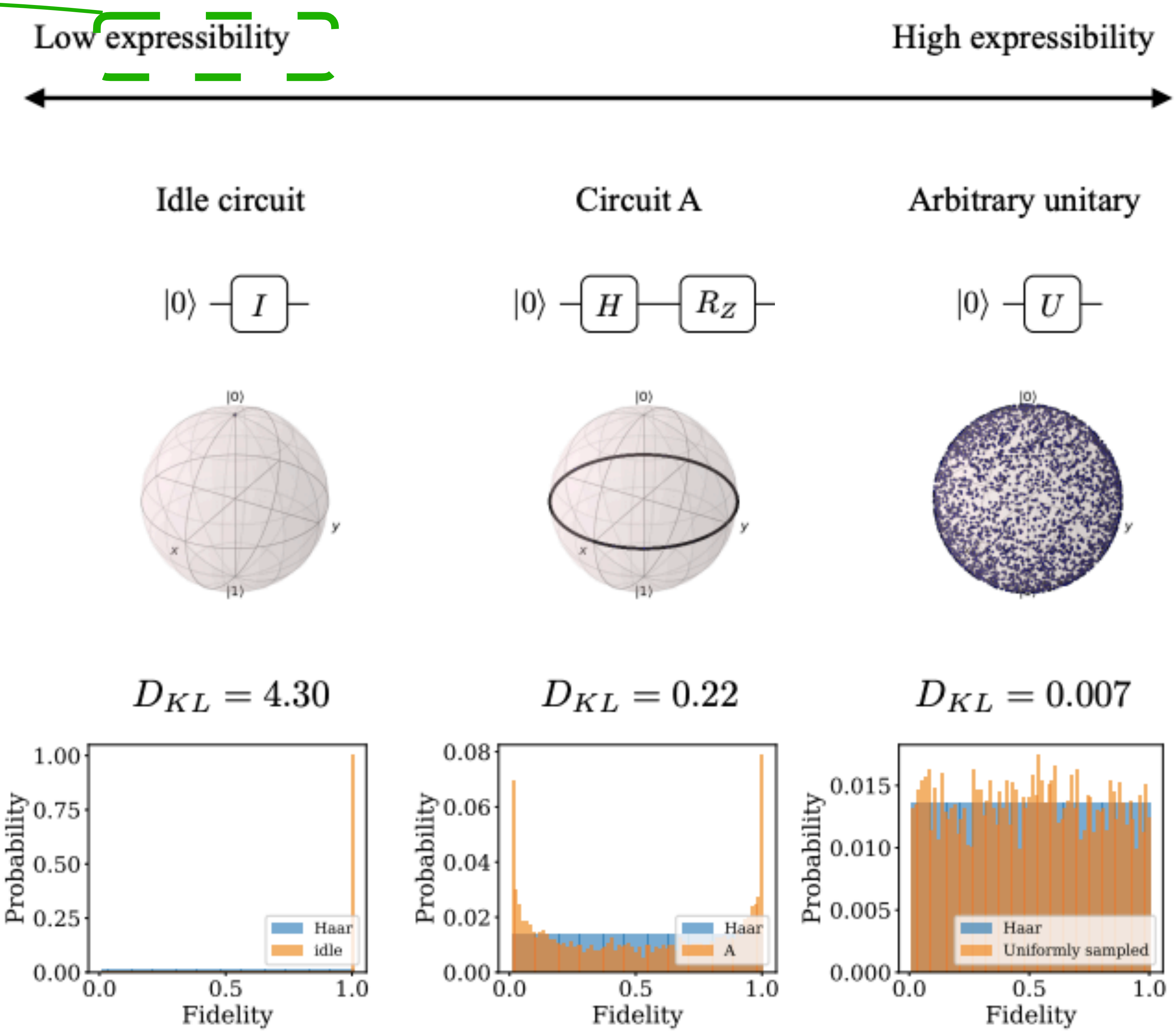
Ability to address the full unitary space

Defined as the distance
of the distribution of states
generated by a quantum
circuit from the distribution
of a Haar-random state

$$Expr = D_{KL}\left(\hat{P}_{QNN}(F;\theta)\,||\,P_{Haar}(F)\right)$$

1. Sample $|\psi_\gamma\rangle, |\psi_\phi\rangle$ from the QNN states
2. Compute the overlap $F = |\langle\psi_\gamma|\psi_\phi\rangle|^2$
3. Repeat many times to obtain statistics

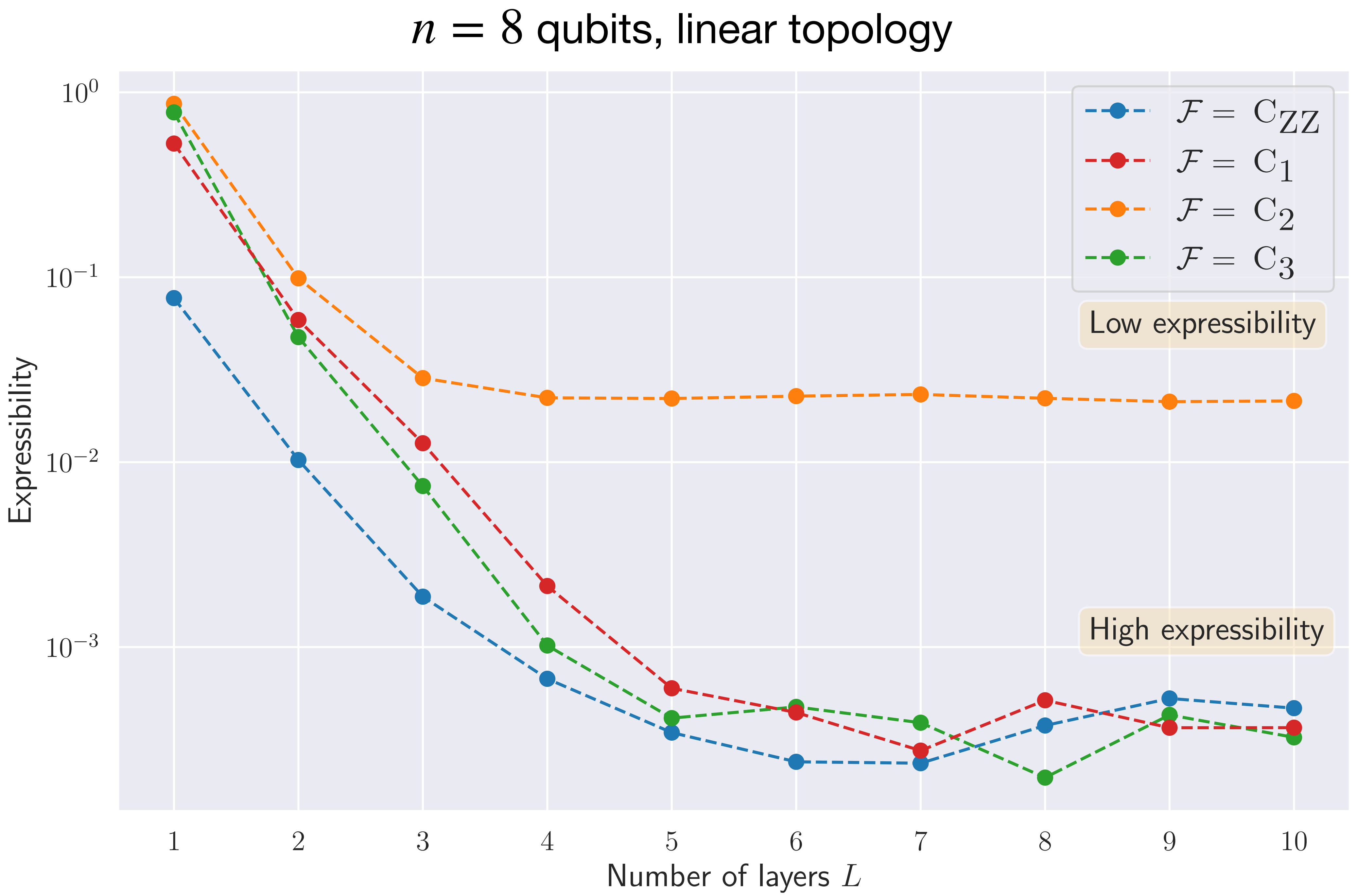
Uniformly distributed fidelity
$\Rightarrow$ address the full space



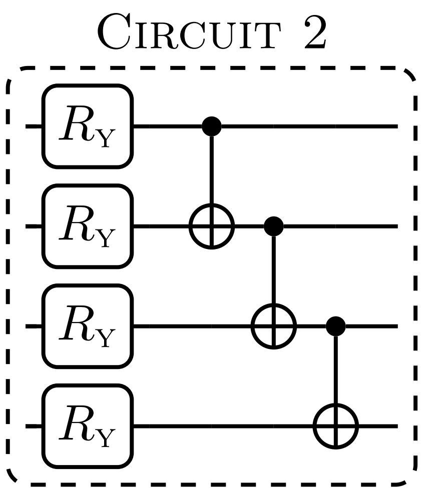
---

[6] Sim, Sukin, Peter D. Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms." *Advanced Quantum Technologies* 2.12 (2019): 1900070.

Expressibility


$n = 8$ qubits, linear topology

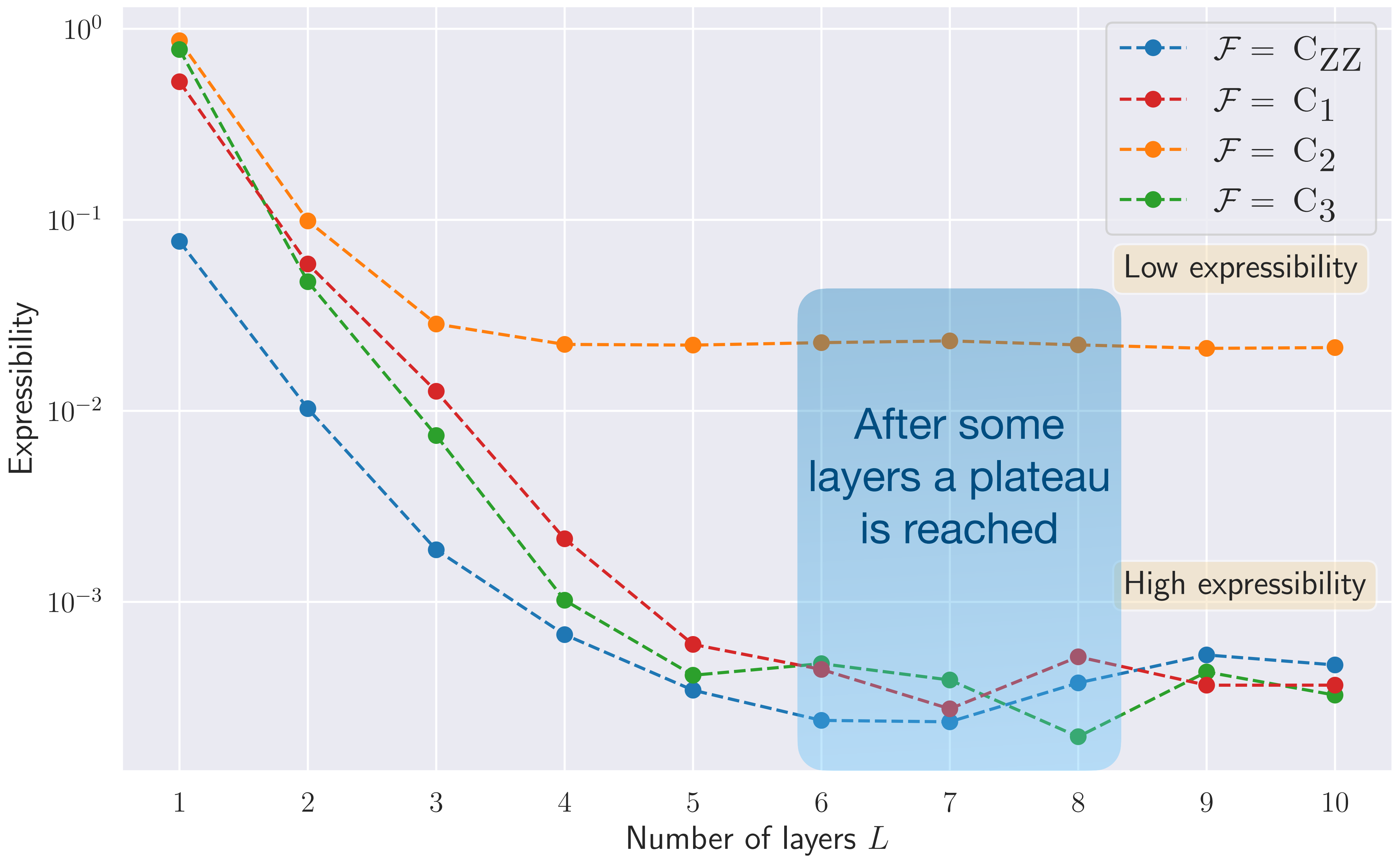
Variational ansatz



[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

Expressibility

$n = 8$ qubits, linear topology



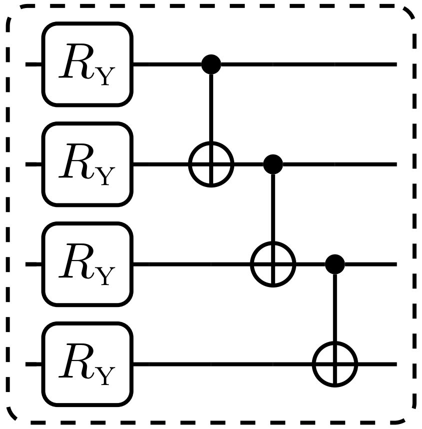
Low expressibility

After some layers a plateau is reached

High expressibility

Variational ansatz

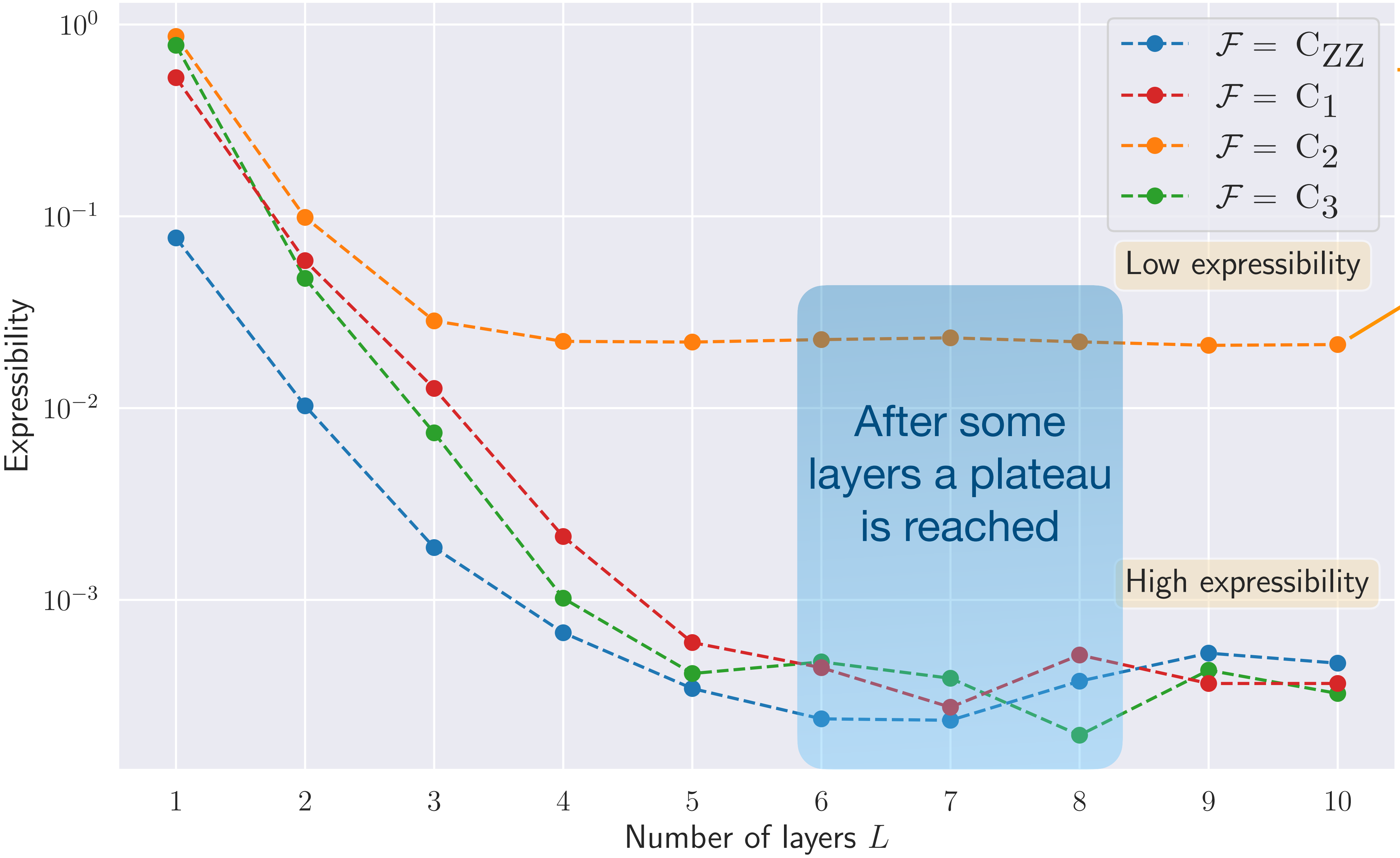
CIRCUIT 2



[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

Expressibility

$n = 8$ qubits, linear topology



Low expressibility

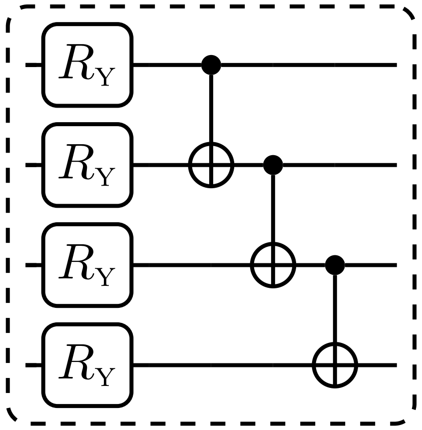
High expressibility

After some layers a plateau is reached

The circuit that builds entanglement faster has the worst expressibility

Variational ansatz

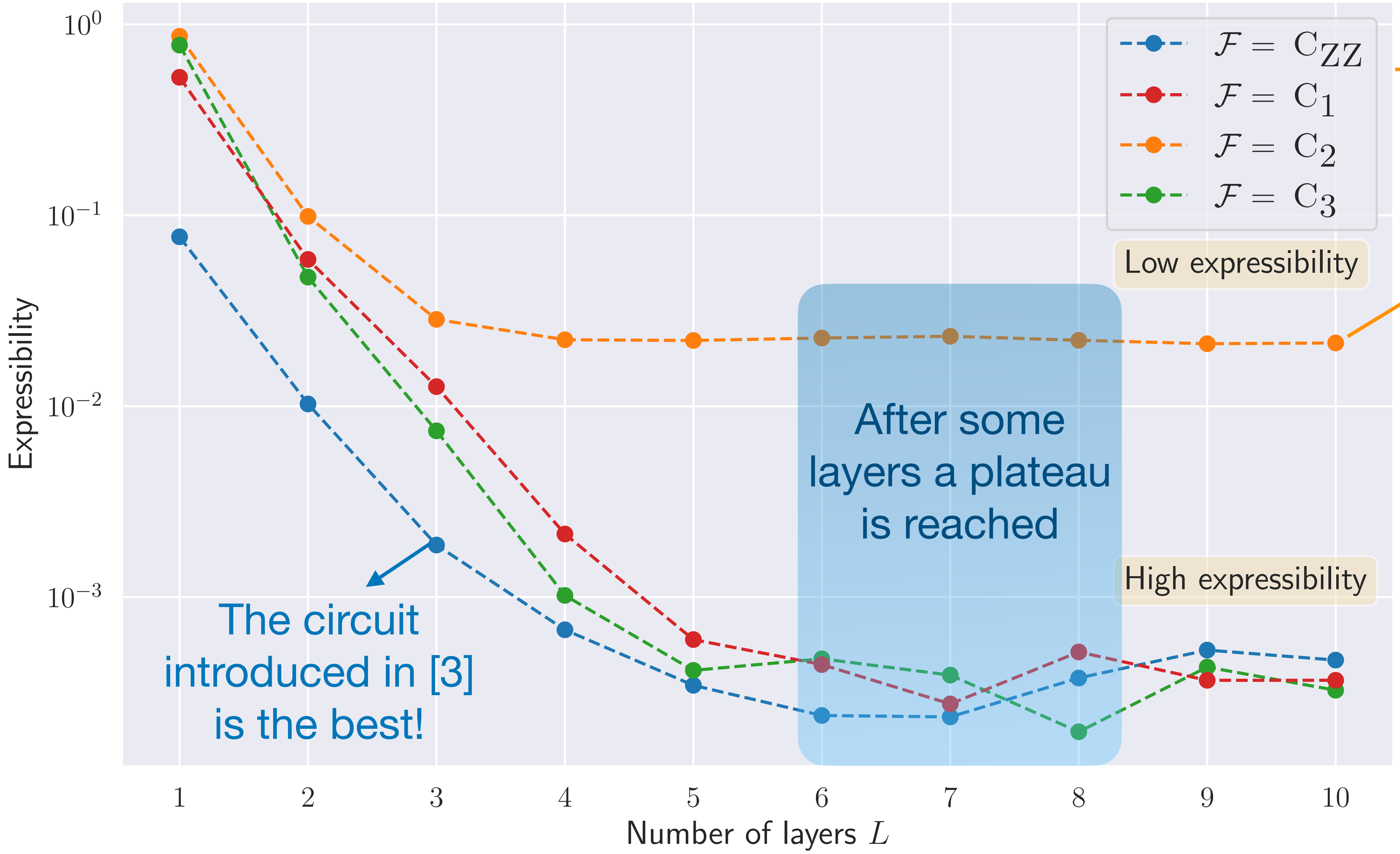
Circuit 2



[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

Expressibility

$n = 8$ qubits, linear topology



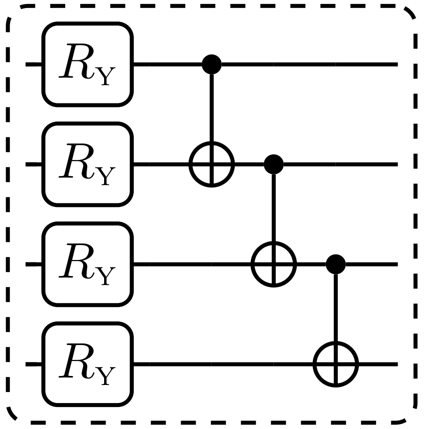
After some layers a plateau is reached

The circuit introduced in [3] is the best!

The circuit that builds entanglement faster has the worst expressibility

Variational ansatz

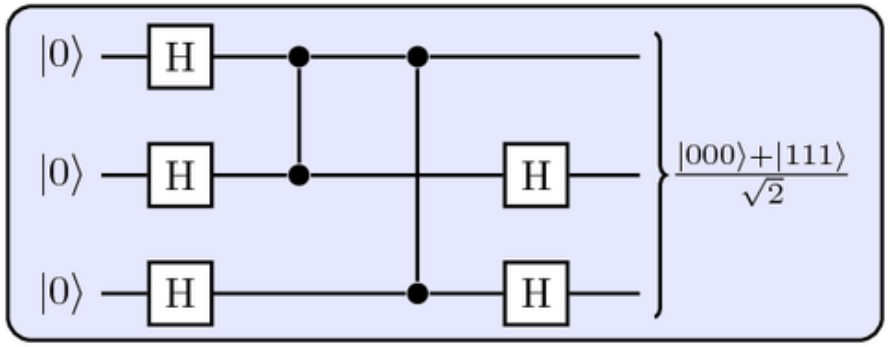
Circuit 2



[3] Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

Quantum Matcha Tea

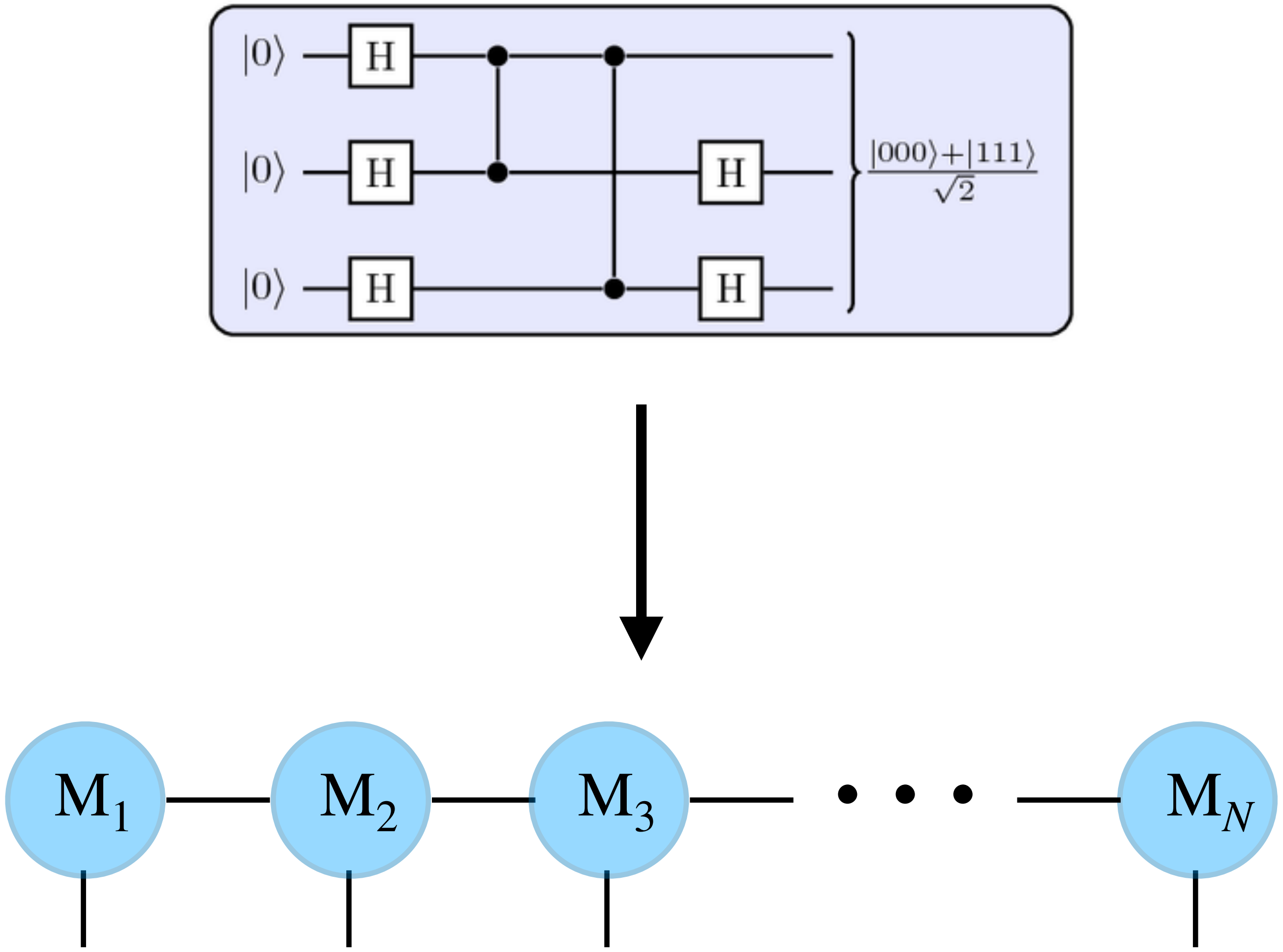
Tensor network emulator



Input:
Quantum circuit

Quantum Matcha Tea

Tensor network emulator

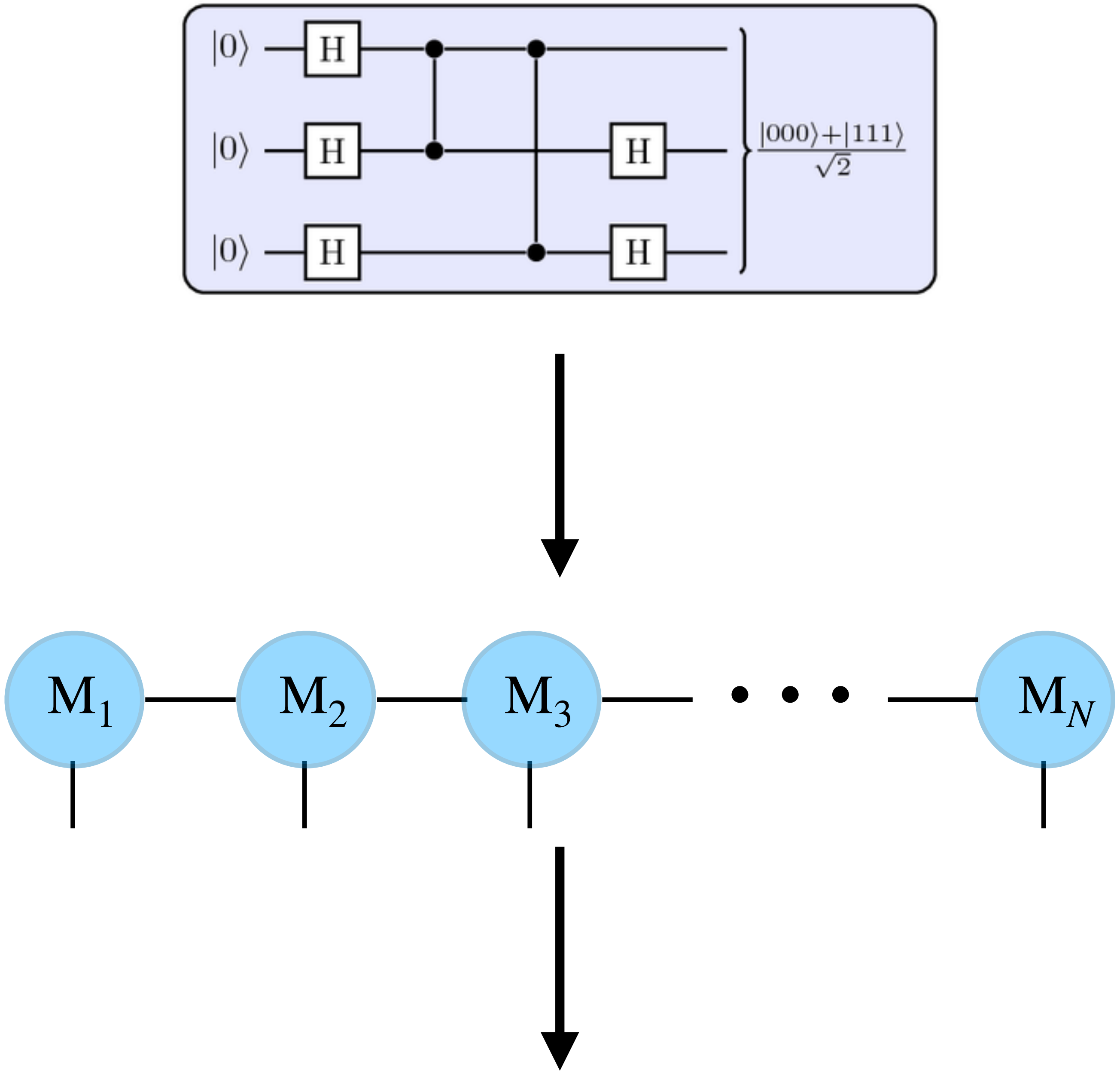


Input:
Quantum circuit

Emulator:
- Python interface
- Fortran and python backend
- MPI enabled for clusters
- GPU backend

Quantum Matcha Tea

Tensor network emulator



$$\langle\psi|A|\psi\rangle$$

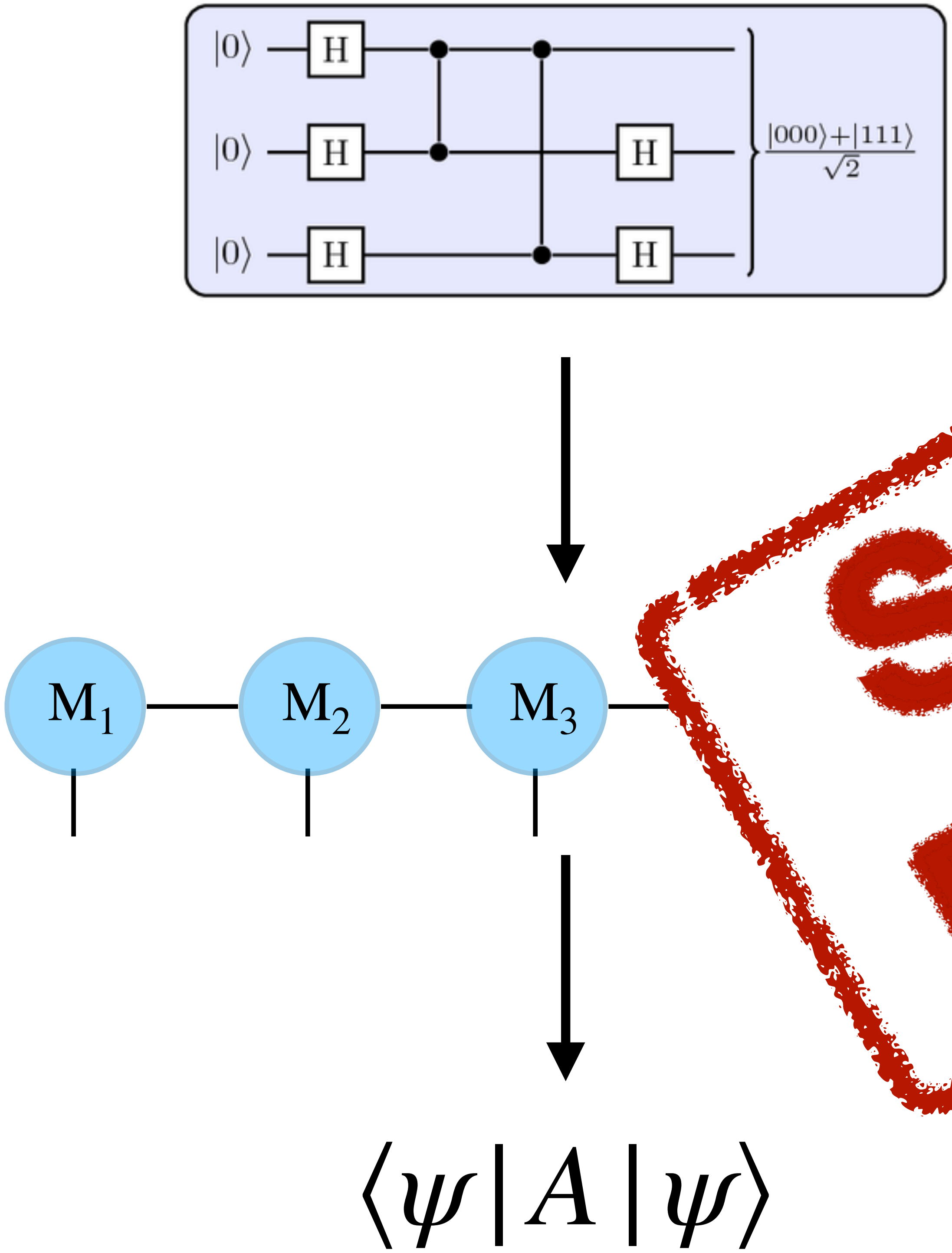
Input:
Quantum circuit

Emulator:
- Python interface
- Fortran and python backend
- MPI enabled for clusters
- GPU backend

Output:
Expectation values of observables

Quantum Matcha Tea

Tensor network emulator



$$\langle\psi|A|\psi\rangle$$

Input:
Quantum circuit

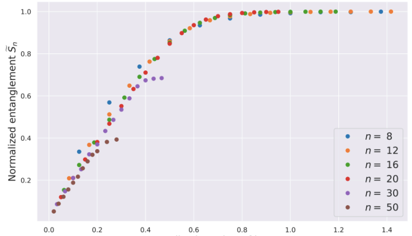
Soon
Public

terface
ran and python backend
MPI enabled for clusters
- GPU backend

Output:
Expectation values of observables

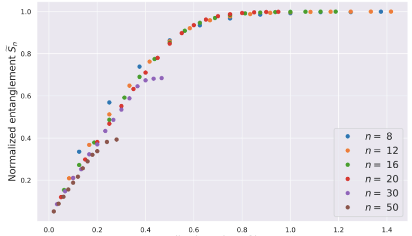
Conclusions and outlook

Entangling speed: characterise the entanglement production of a given QNN architecture

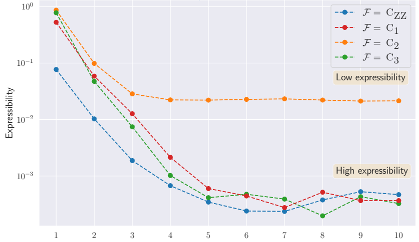


Conclusions and outlook

Entangling speed: characterise the entanglement production of a given QNN architecture

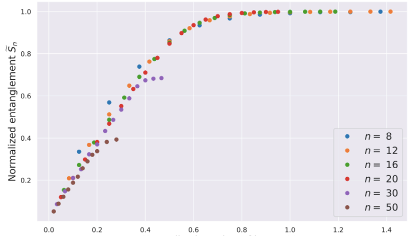


Expressibility: necessity of finding a sweet spot between the entanglement production and the expressibility of the QNN

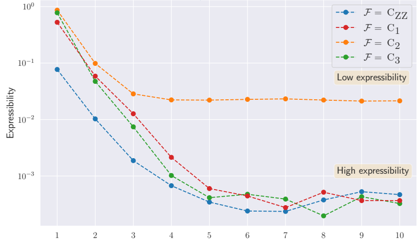


Conclusions and outlook

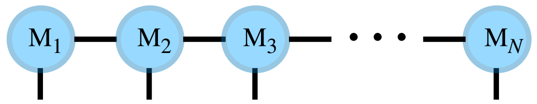
Entangling speed: characterise the entanglement production of a given QNN architecture



Expressibility: necessity of finding a sweet spot between the entanglement production and the expressibility of the QNN

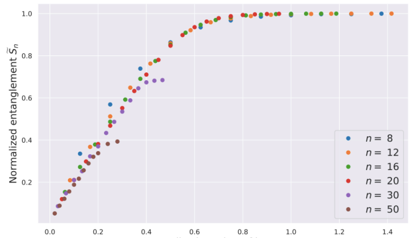


Tensor network methods: use of efficient methods to simulate large system (up to 50 qubits) to investigate the limiting behaviour. **Quantum matcha tea!**

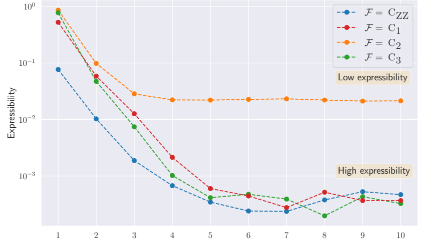


Conclusions and outlook

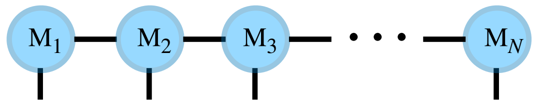
Entangling speed: characterise the entanglement production of a given QNN architecture



Expressibility: necessity of finding a sweet spot between the entanglement production and the expressibility of the QNN



Tensor network methods: use of efficient methods to simulate large system (up to 50 qubits) to investigate the limiting behaviour. **Quantum matcha tea!**

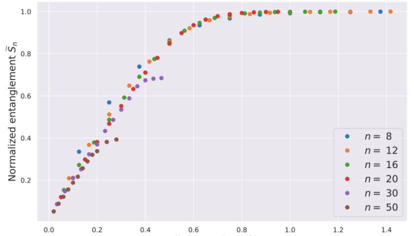


Training: Extend the analysis of the paper to the entanglement produced during the training.

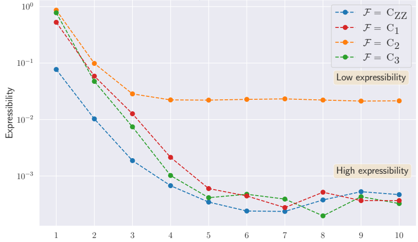
How much entanglement is really needed?

Conclusions and outlook

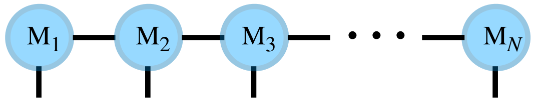
Entangling speed: characterise the entanglement production of a given QNN architecture



Expressibility: necessity of finding a sweet spot between the entanglement production and the expressibility of the QNN



Tensor network methods: use of efficient methods to simulate large system (up to 50 qubits) to investigate the limiting behaviour. **Quantum matcha tea!**



Training: Extend the analysis of the paper to the entanglement produced during the training.
How much entanglement is really needed?

Using different distribution: how is the entanglement changing when we draw the random parameters from a gaussian distribution? Can we observe a “critical” behavior?

Dipartimento
di Fisica
e Astronomia
Galileo Galilei

UNIVERSITÀ DI PAVIA
Dipartimento di Fisica

INFN
Istituto Nazionale di Fisica Nucleare

CINECA

Stefano Mangini

Simone Montangero

Chiara Macchiavello

Riccardo Mengoni

Thank you
for your attention

References

(1) Ballarin, Marco, et al. "Entanglement entropy production in Quantum Neural Networks." *arXiv preprint arXiv:2206.02474* (2022).

(2) Havlíček, Vojtěch, et al. "Supervised learning with quantum-enhanced feature spaces." *Nature* 567.7747 (2019): 209-212.

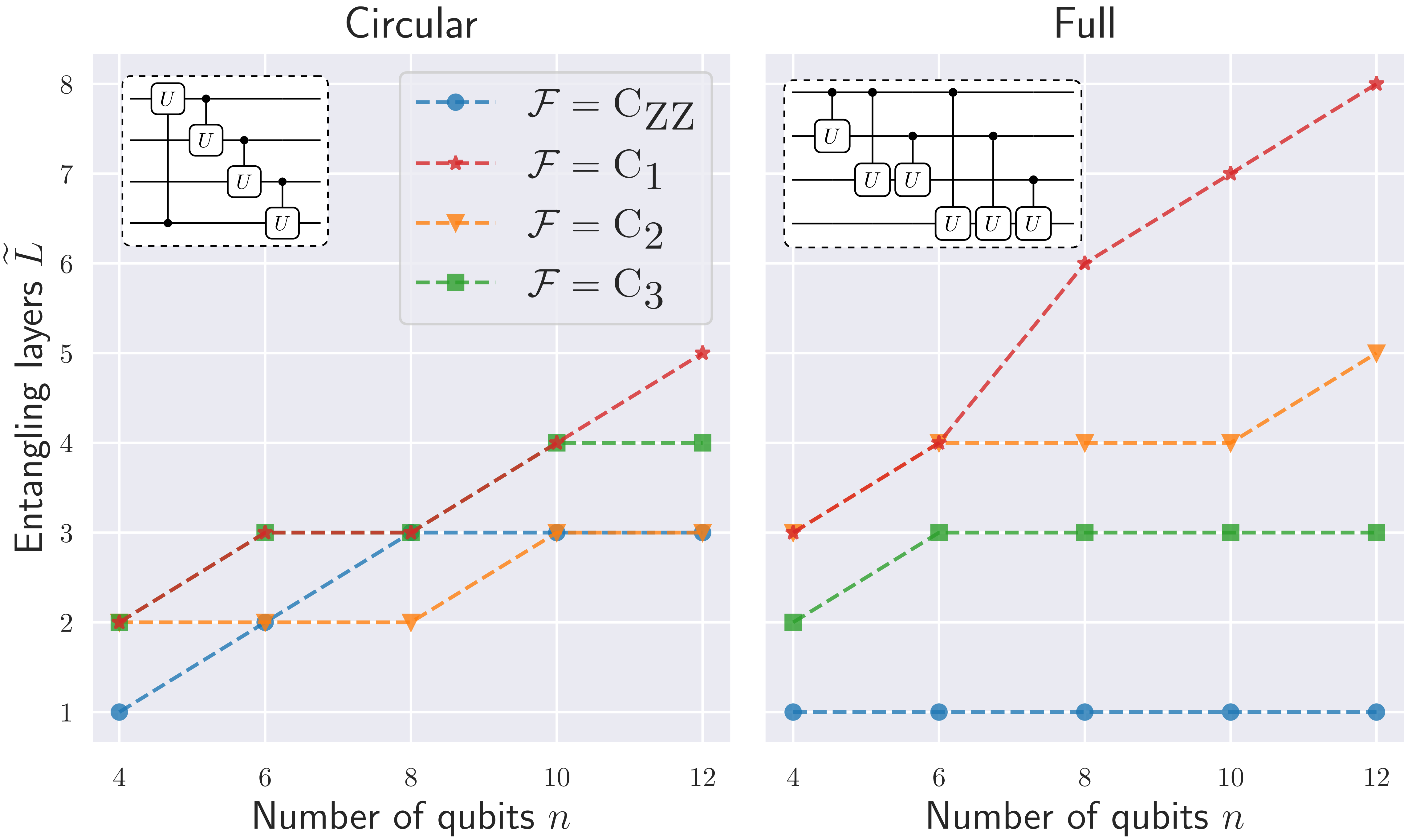
(3) Abbas, Amira, et al. "The power of quantum neural networks." *Nature Computational Science* 1.6 (2021): 403-409.

(4) Marrero, Carlos Ortiz, Mária Kieferová, and Nathan Wiebe. "Entanglement-induced barren plateaus." *PRX Quantum* 2.4 (2021): 040316.

(5) Pérez-Salinas, Adrián, et al. "Data re-uploading for a universal quantum classifier." *Quantum* 4 (2020): 226.

(6) Sim, Sukin, Peter D. Johnson, and Alán Aspuru-Guzik. "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms." *Advanced Quantum Technologies* 2.12 (2019): 1900070.

Entanglement scaling with depth



As expected, entanglement increases much faster in these cases!