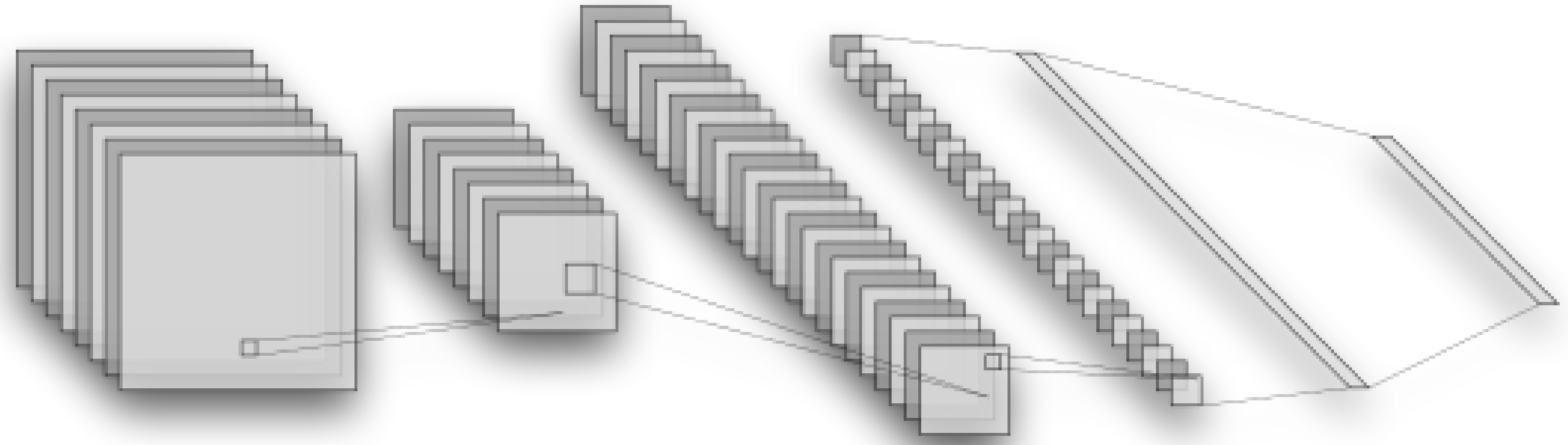
Theory-driven quantum machine learning for HEP

Jack Y. Araz
Institute For Particle Physics Phenomenology
Durham University

AI & Quantum Information Applications in Fundamental Physics
February 16th, 2023

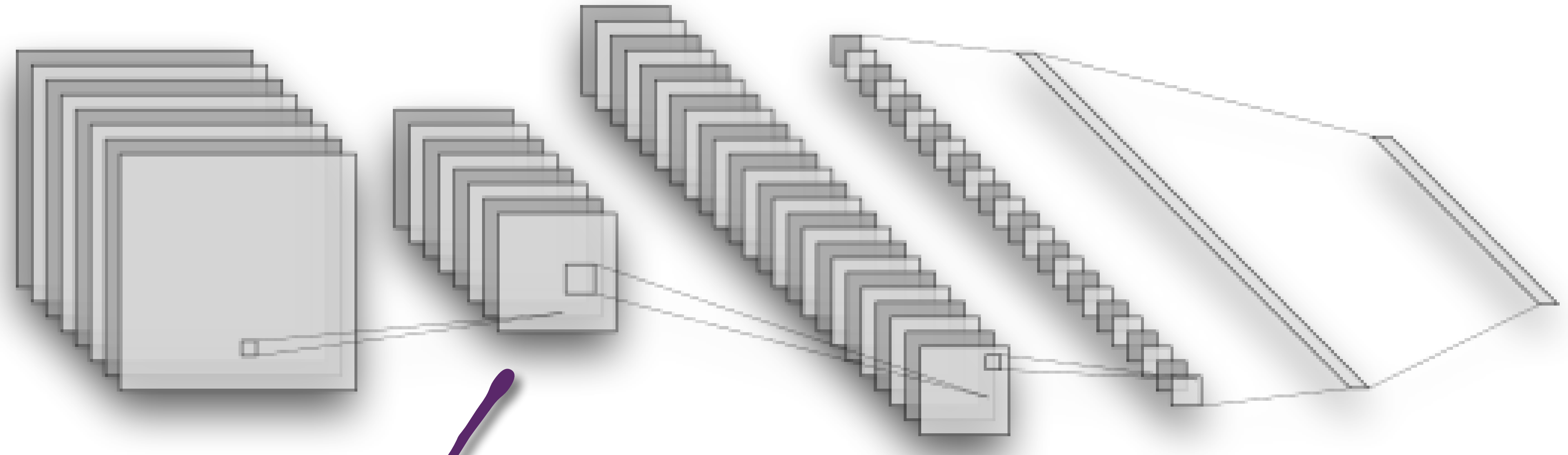
Pre-introduction: What does ML learn?



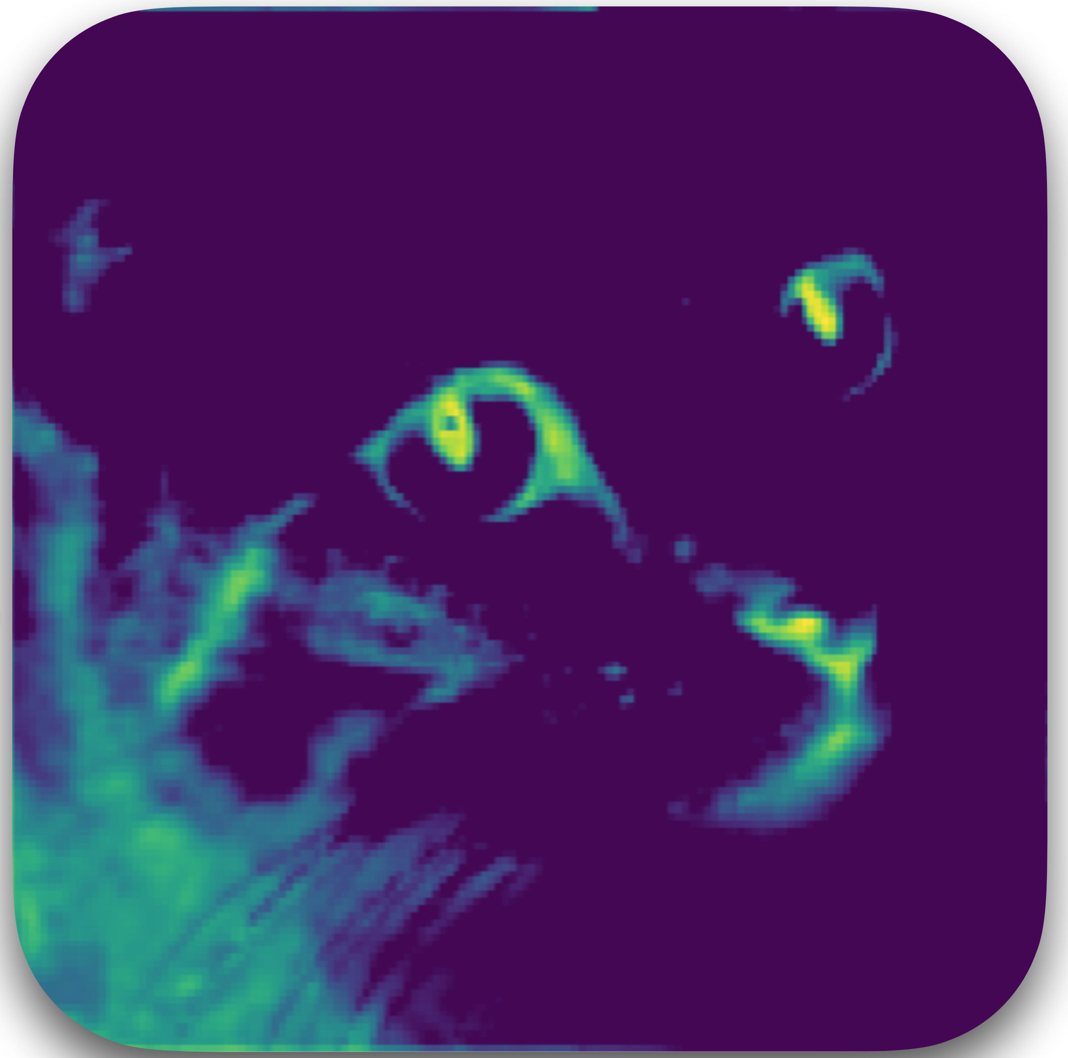
$$f(\mathbf{x};\theta)$$

From Monsters Inc.

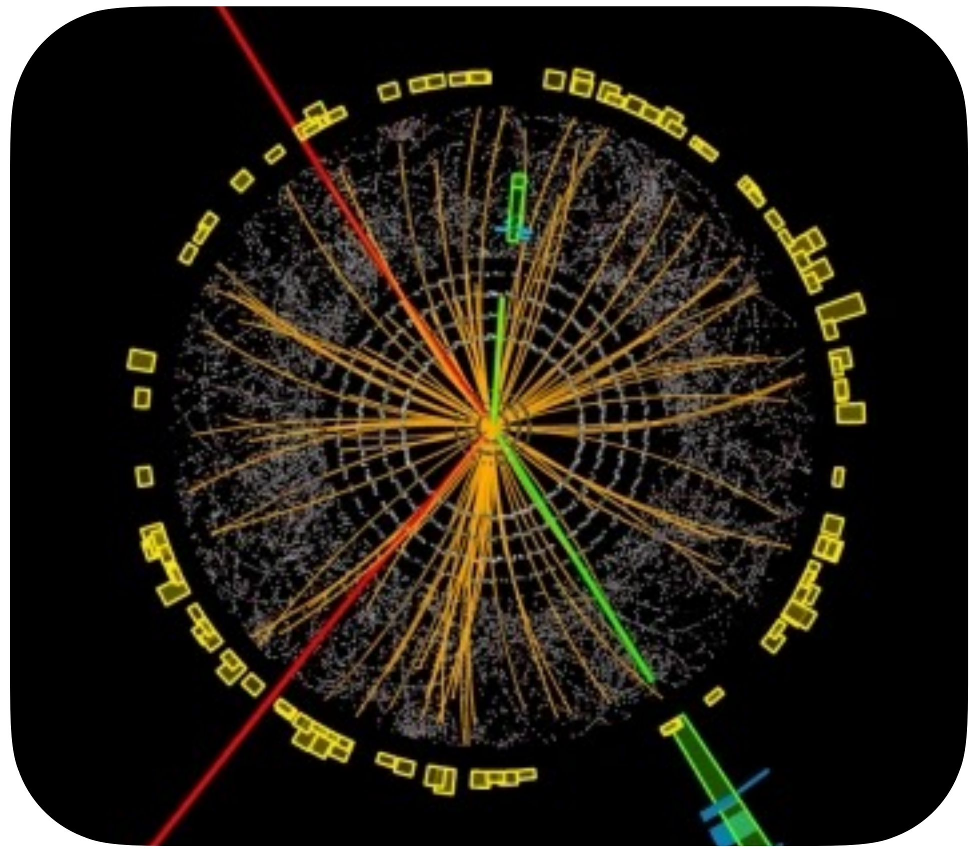
Pre-introduction: What does ML learn?



From Monsters Inc.

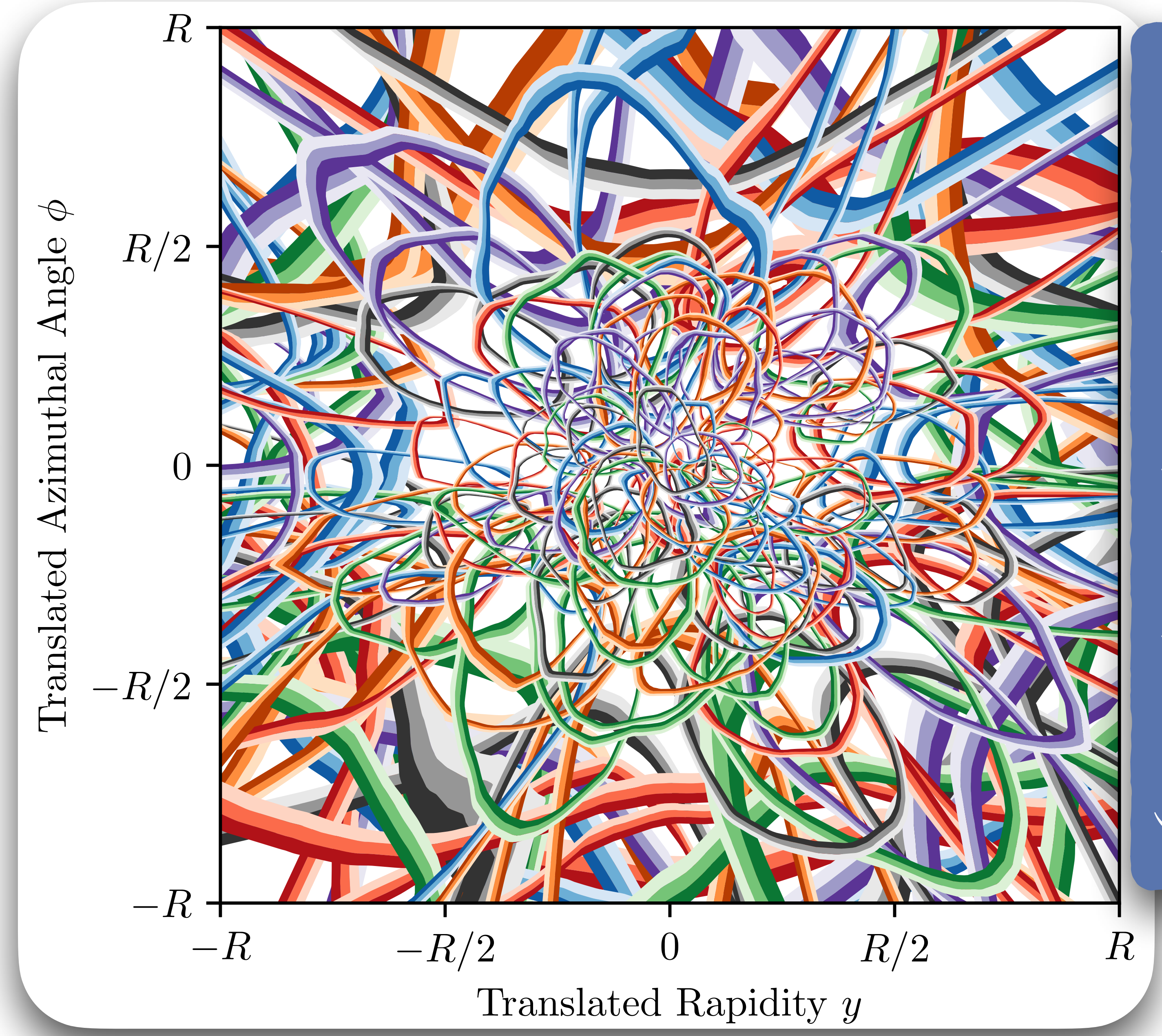


Pre-introduction: What does ML learn?



$f(\mathbf{x}; \theta)$

Gluon shower!



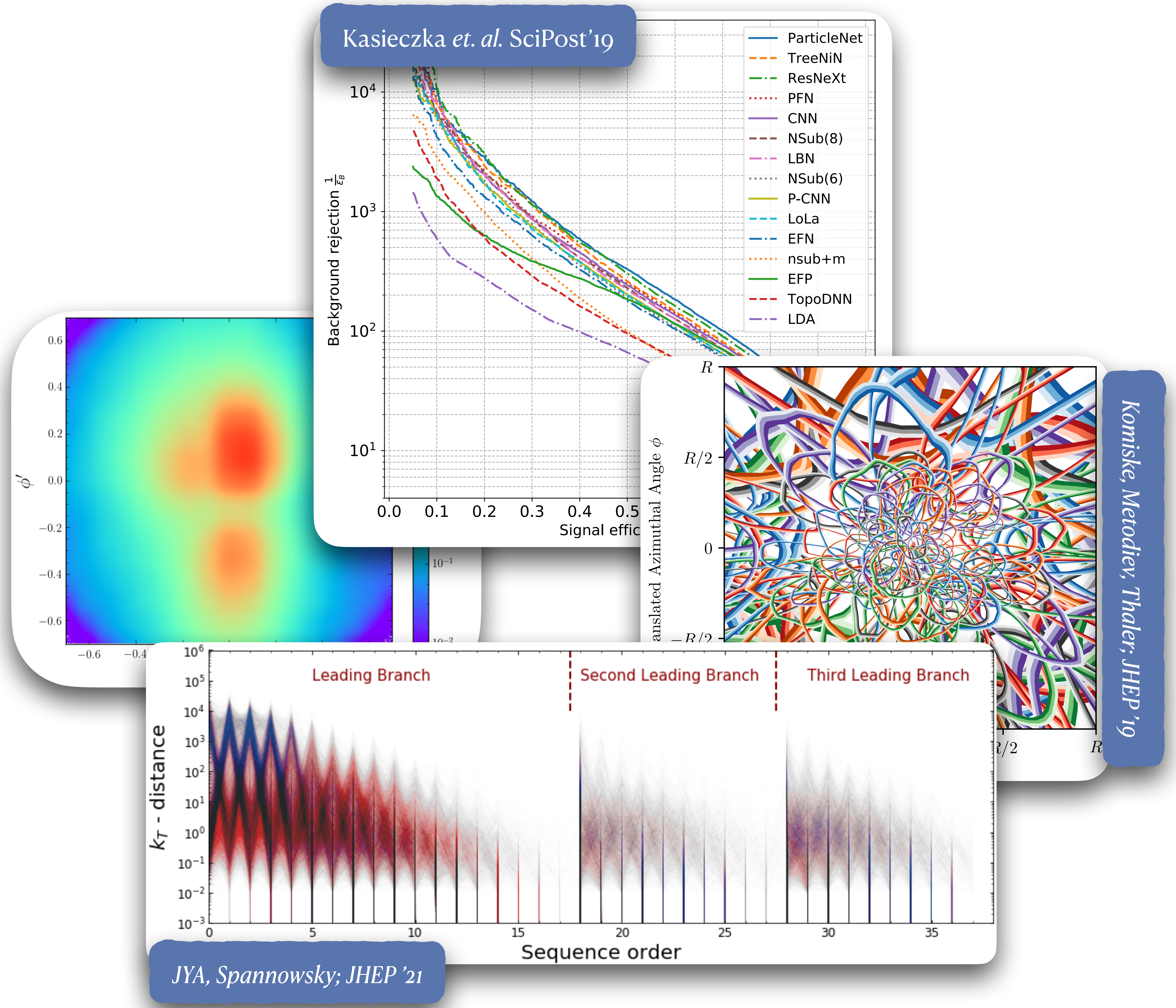
Komiske, Metodiev, Thaler; JHEP '19

Each "ring" corresponds to an output of a filer based on the polynomial jet distribution $y^m\phi^n$.

Sales pitch of the talk!

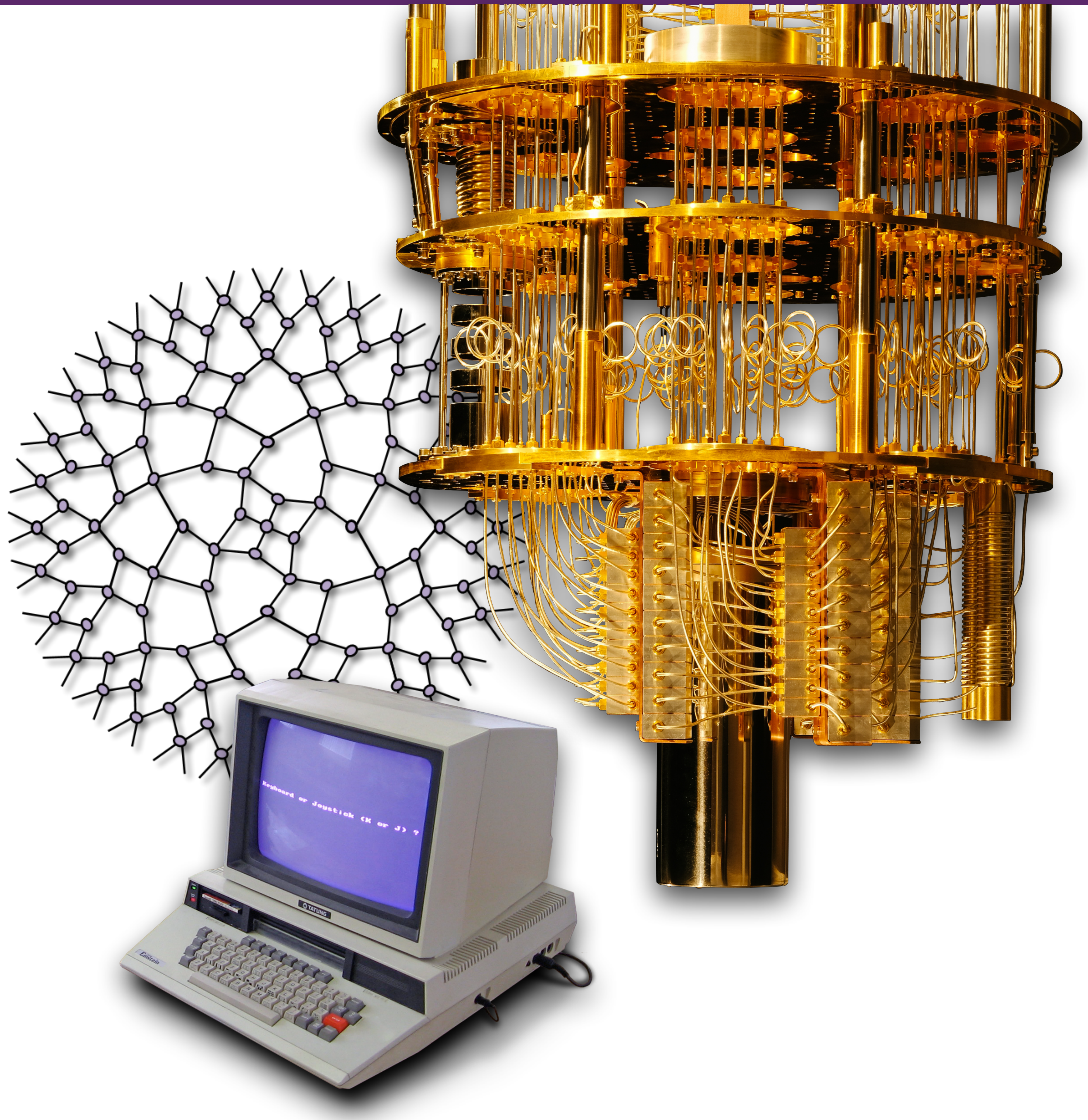
- We more or less know how to get a well-performing Neural Network to classify jets, LHC events, and even cats and dogs…
- What we don't know is what this network learns.
- Can we use Quantum Mechanics to have more insight into the learning process?
  - ✦ What has a model learned?
  - ✦ What is learning?
  - ✦ How do we develop "insightful" algorithms?
  - ✦ How to perform this on a Quantum device?

❖ Can an ML problem be formulated as a quantum many-body system?



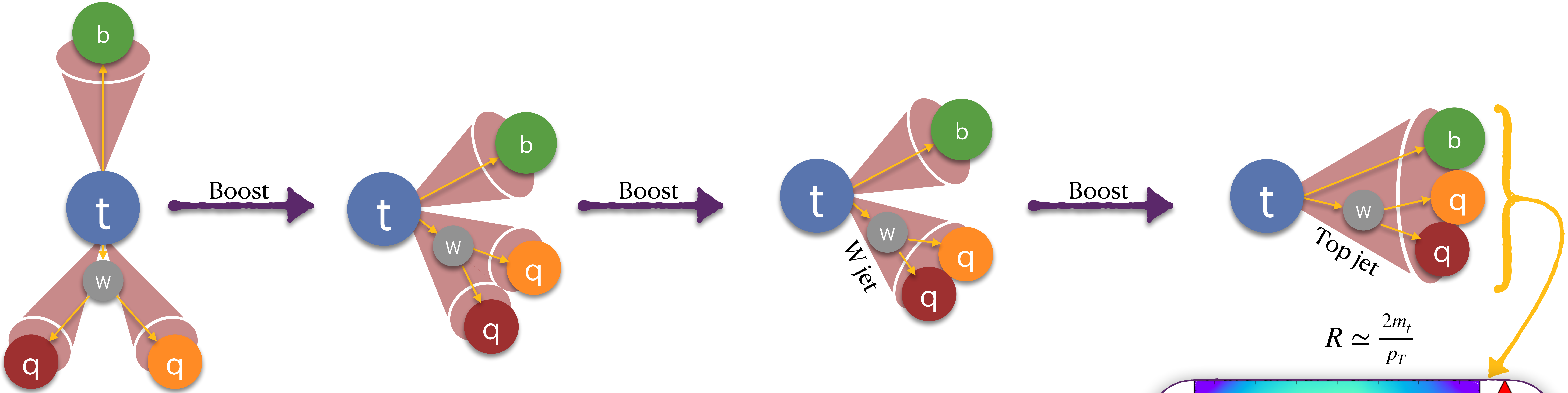
Outline

- Classification as a quantum many-body problem
- Hamiltonian learning for anomaly detection
- Conclusion

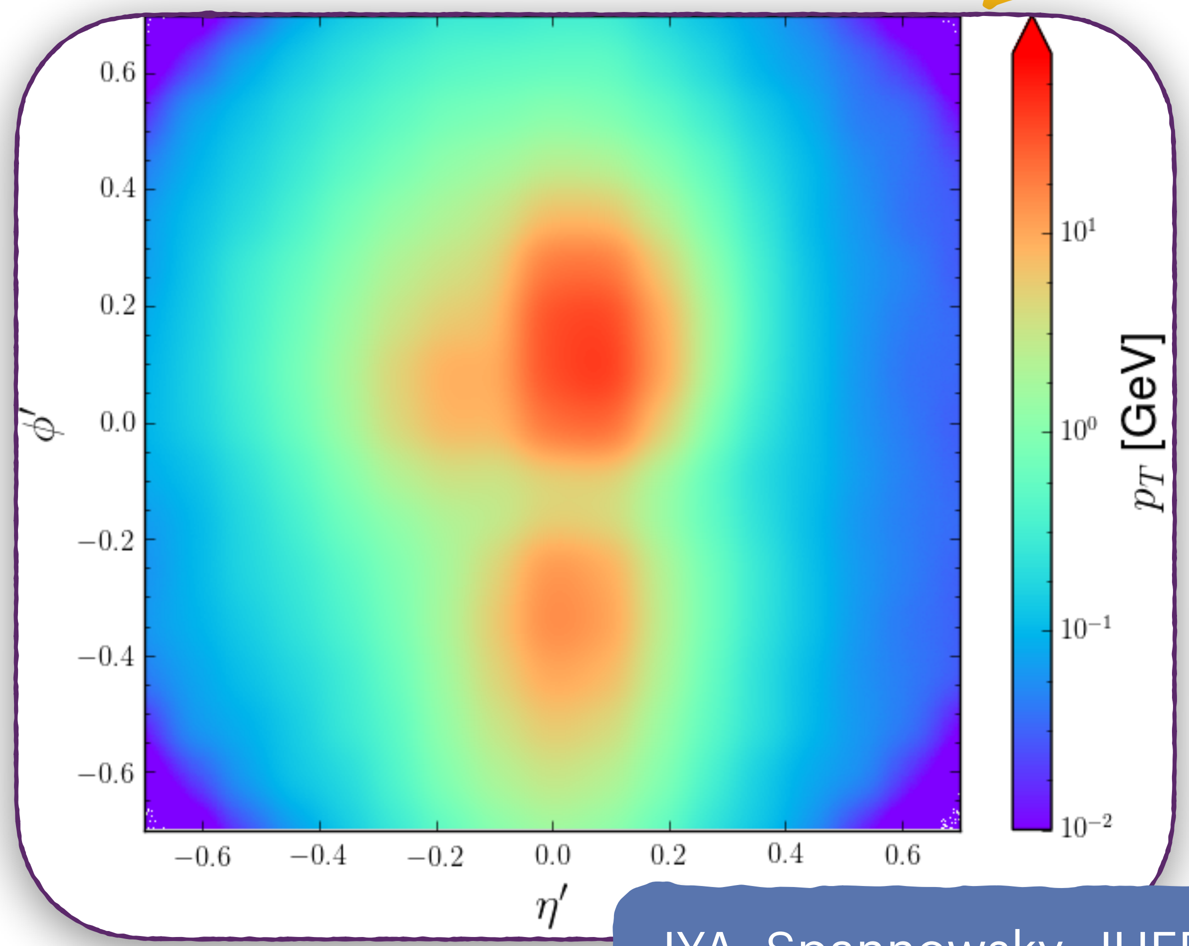


Eduardo from the lobby

Hello world of HEP-ML: Top tagging



$$R \simeq \frac{2m_t}{p_T}$$

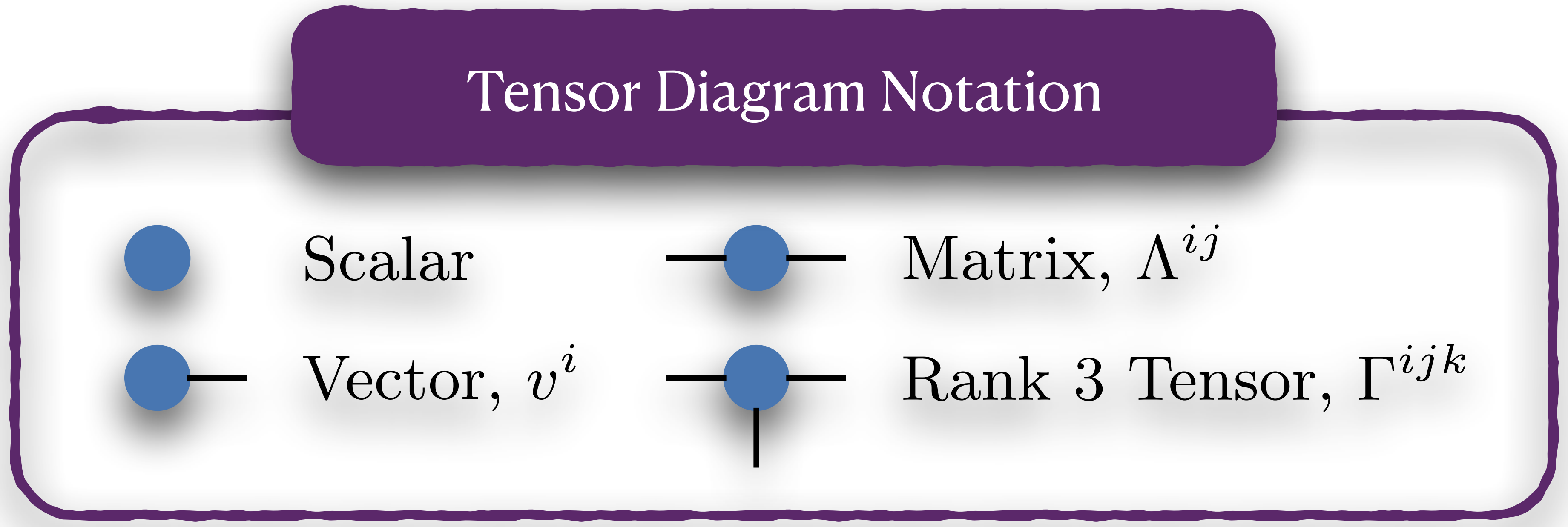


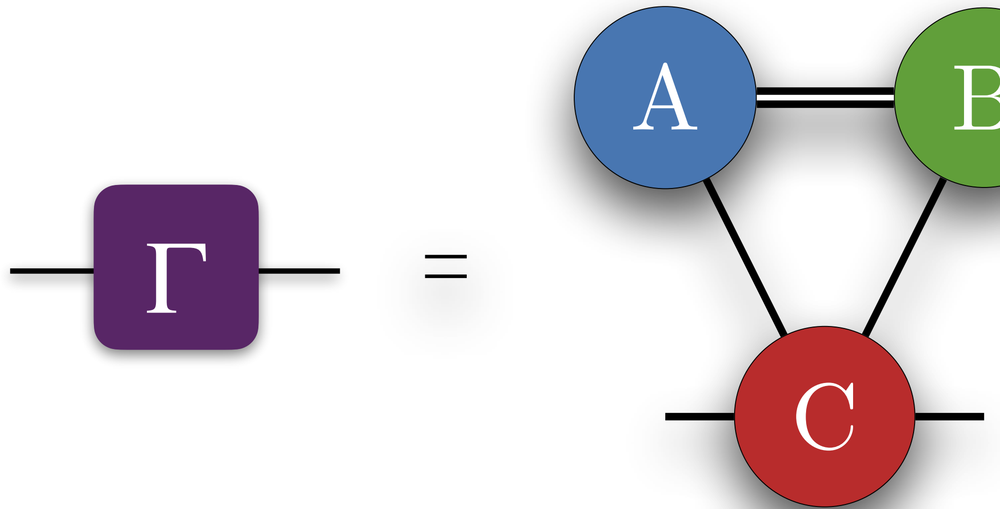
JYA, Spannowsky; JHEP '21

- With the increased boost factor, jets (top decay products) are getting more collimated.
- Hadronic top tagging tools: Mass grooming and filtering, Pruning, Trimming, Soft Drop Tagger, Mass Drop Tagger, HEPTopTagger, Machine Learning

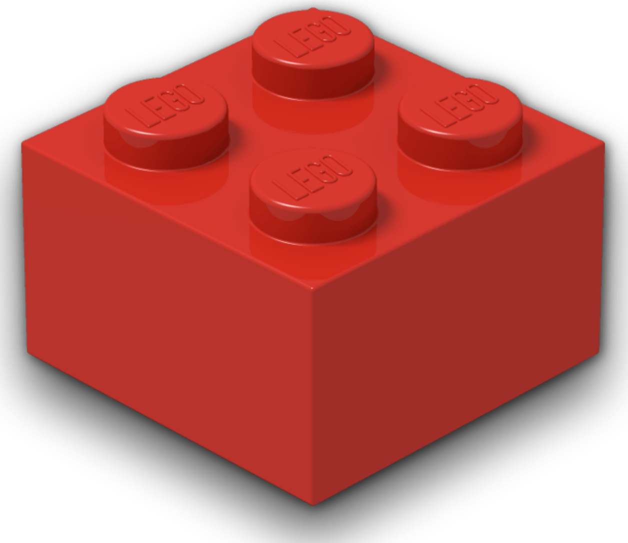
Classification as a quantum many-body problem

Tensor Networks: Origins





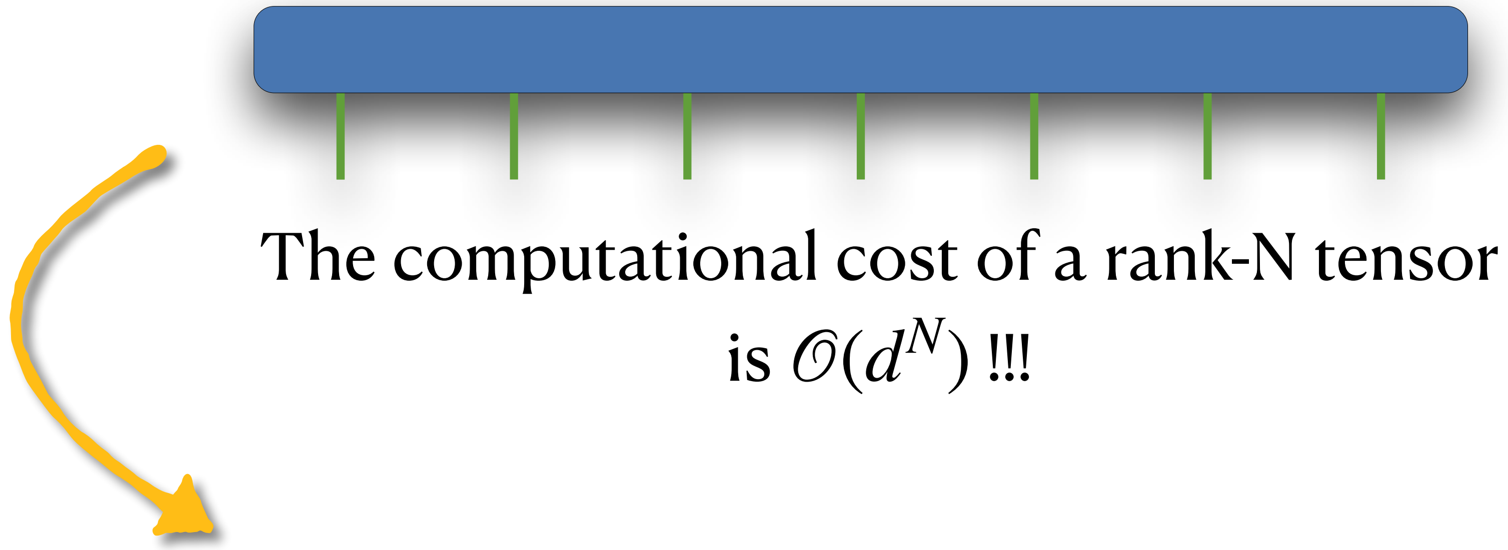
$$\Gamma^{ij} = A^{klm} B^{n}{}_{lm} C^{ij}{}_{kn}$$



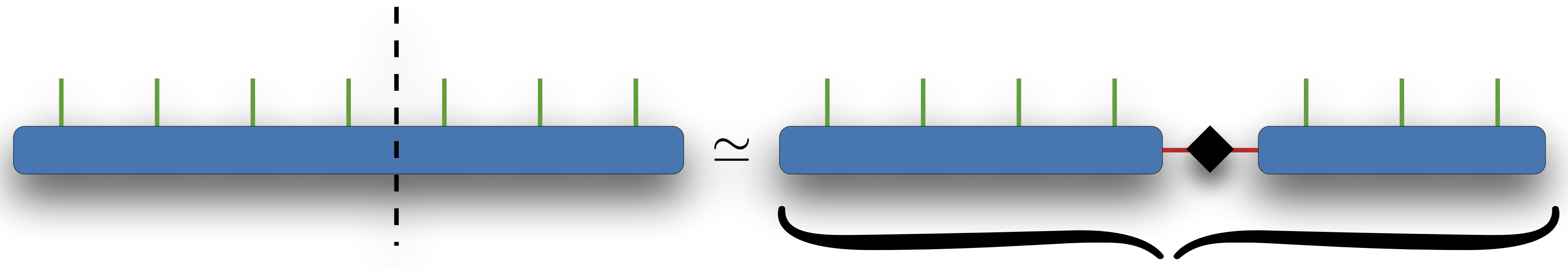
$$= L^{ijkl}{}_{\mu\nu\rho\theta}$$

$$|\Psi\rangle = \sum_{\phi_1,\ldots,\phi_n=0} \mathscr{W}_{\phi_1\ldots\phi_n} |\phi_1\rangle \otimes |\phi_2\rangle \otimes \ldots \otimes |\phi_n\rangle$$

$$\forall |\phi_i\rangle \in \mathscr{H}^{\otimes 2^N} \quad \rightarrow \quad |\phi_i\rangle \in \{ |\uparrow\rangle, |\downarrow\rangle \}$$



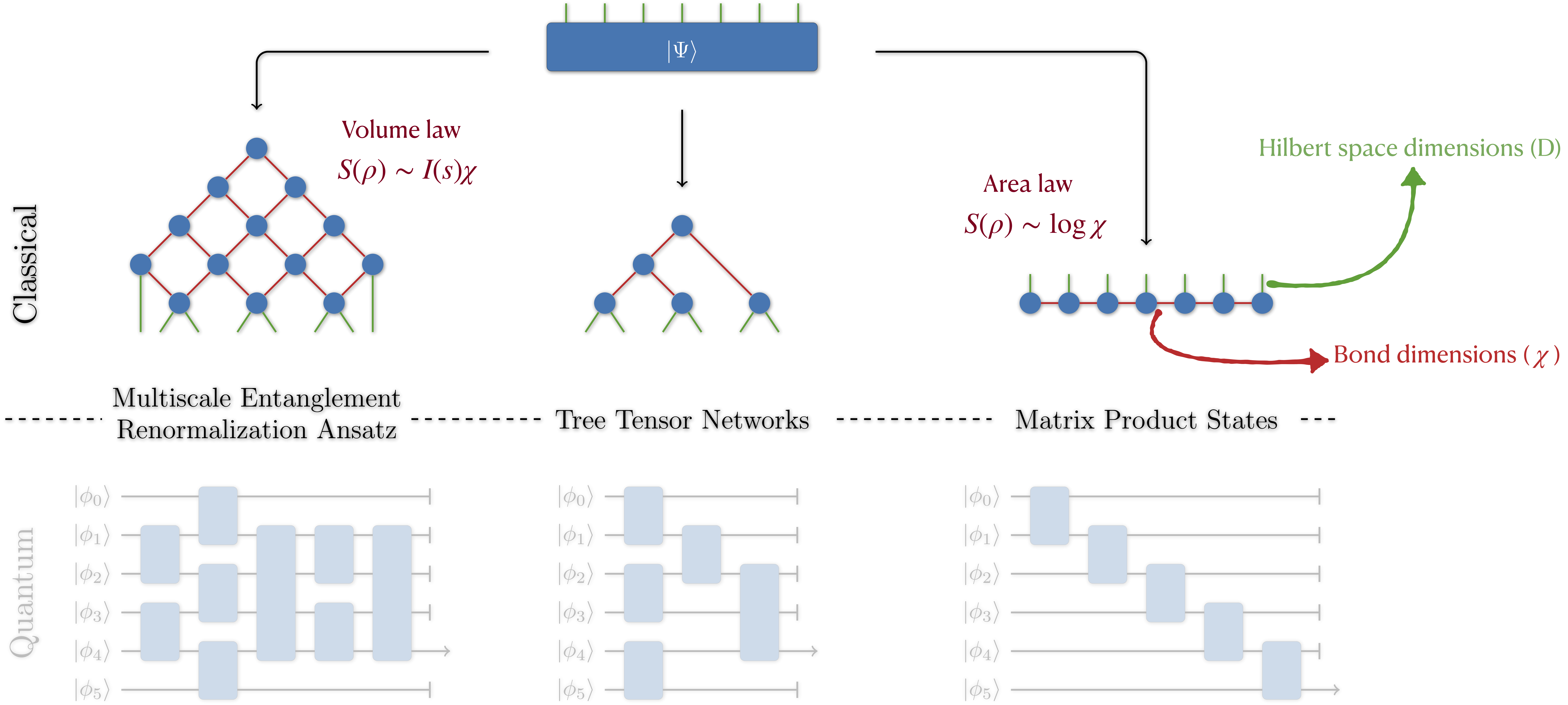
The computational cost of a rank-N tensor is $\mathcal{O}(d^N)$!!!



Computational cost is $\mathcal{O}(d^{N-1}\chi^2)$!!!

Types of Tensor Networks (some of them)

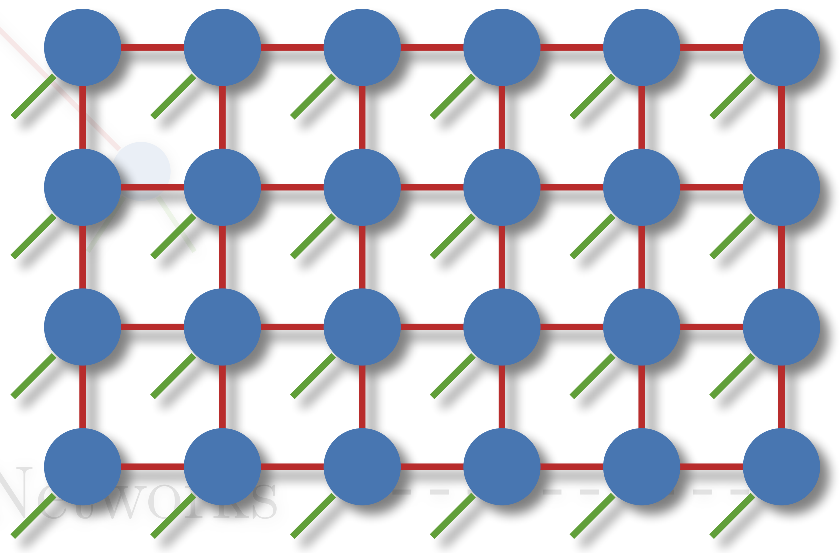
$S(\rho) = \mathrm{Tr}[\rho \log \rho]$



Types of Tensor Networks (some of them)

For discussion session

PEPS is designed for 2D lattices which is more suitable to represent image type of data.



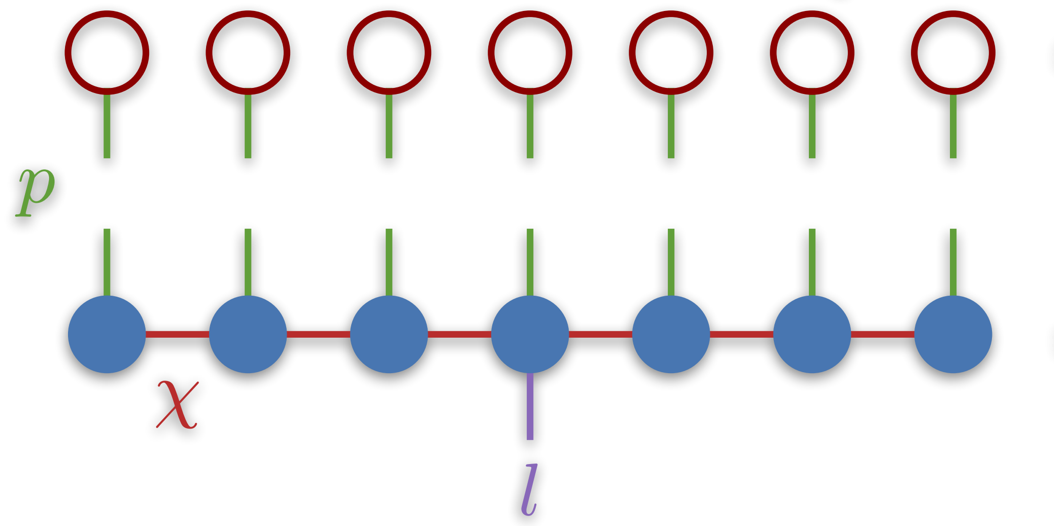
Projected Entangled Pair States

Matrix Product States for Classification

Sub-Outline

- How to embed the data?
- How to form a network?
- How to train the network?

$$\phi^i(\mathbf{x}) = R_{SU(N)}(\tilde{p}^i_T \pi/2)\ |0\rangle$$



$= \Phi^{p_1\cdots p_n}(\mathbf{x})$

$= \mathcal{W}^l{}_{p_1\cdots p_n}$

$:= f^l(\mathbf{x}) = \mathcal{W}^l{}_{p_1\cdots p_n}\ \Phi^{p_1\cdots p_n}(\mathbf{x})$

l

$$p(x^{(i)};\theta) = \left|f^l(x^{(i)})\right|^2$$

No activation function!!
Everything is linear!!

TNs intrinsically have "topological nonlinearity"

Stoudenmire, Schwab 1605.05775

Novikov, Trofimov, Oseledets; 1605.03795

$$\mathscr{L} = \frac{1}{N}\sum_{x\in\mathbf{x}^N} q^{\rm truth}\ \log\left(p(x^{(i)};\theta)\right)$$

Or anything else you like to minimise...

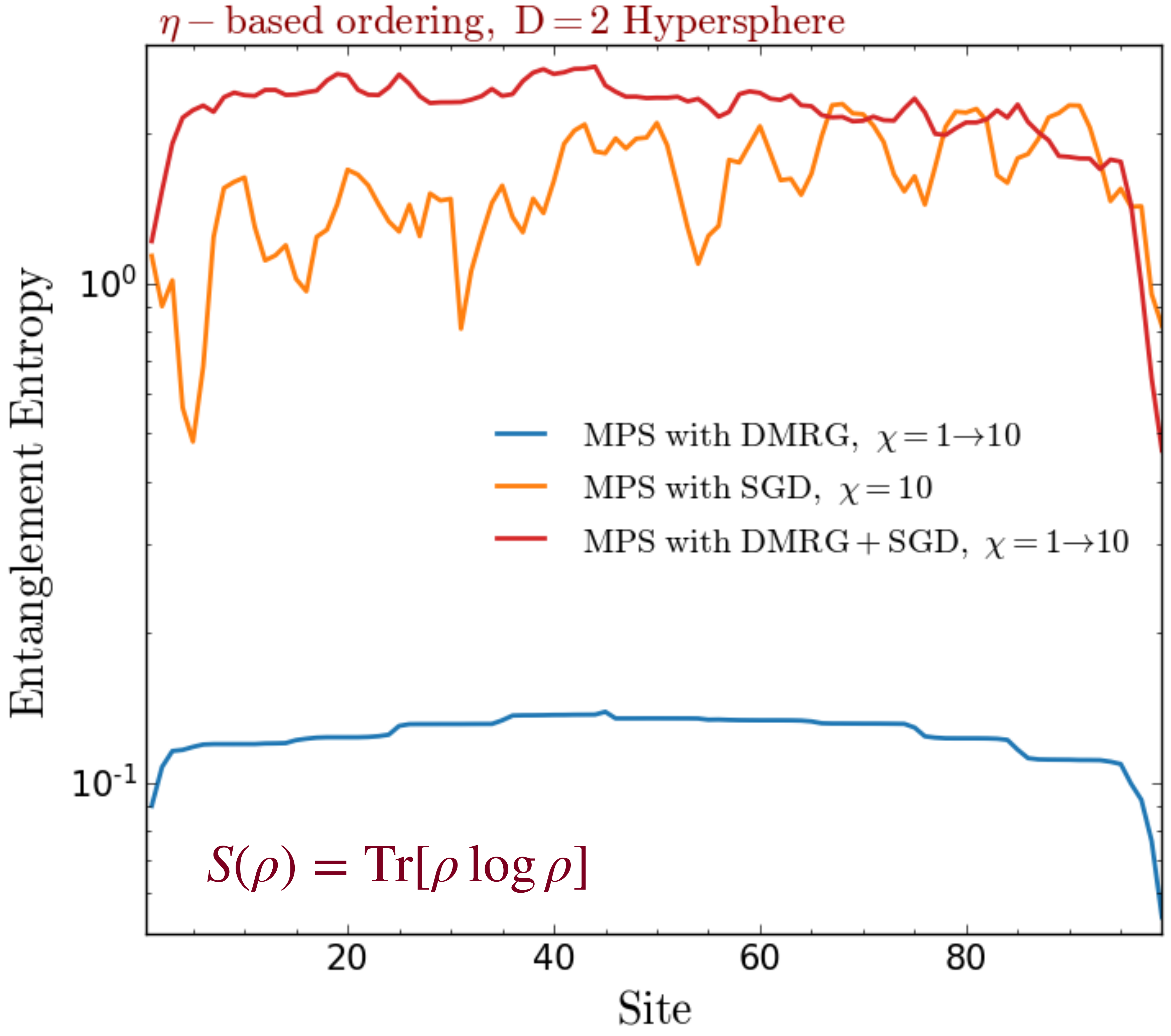
Traditionally NNs are trained with SGD, but MPS is trained with Density Matrix Renormalisation Group Algorithm

JYA, Spannowsky; JHEP '21, arXiv: 2106.08334

Why finding a quantitative measure is important?

JYA, Spannowsky; JHEP '21, arXiv: 2106.08334

- A 50% reduction in the number of pixels used and a 91% reduction in the number of parameters lead to the same classification quality!
- Understanding the network gives the ability to build better training algorithms.
- Scientific data is largely sparse; if we know where the information comes from, we can get rid of large amounts of data.
- Suppress the noise (and for pile-up mitigation to be confirmed)!



Why finding a quantitative measure is important?

Felser, Trenti, Sestini, Gianelle, Zuliani, Lucchessi, Montangero; npj 2021

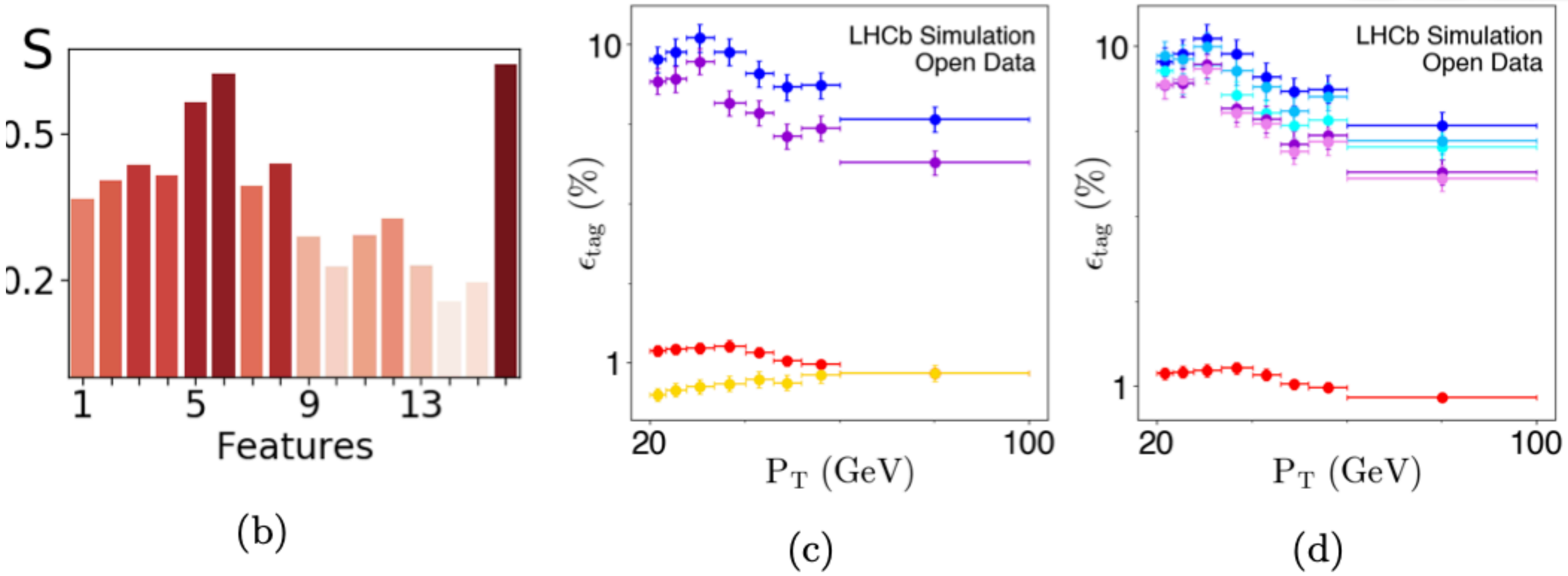
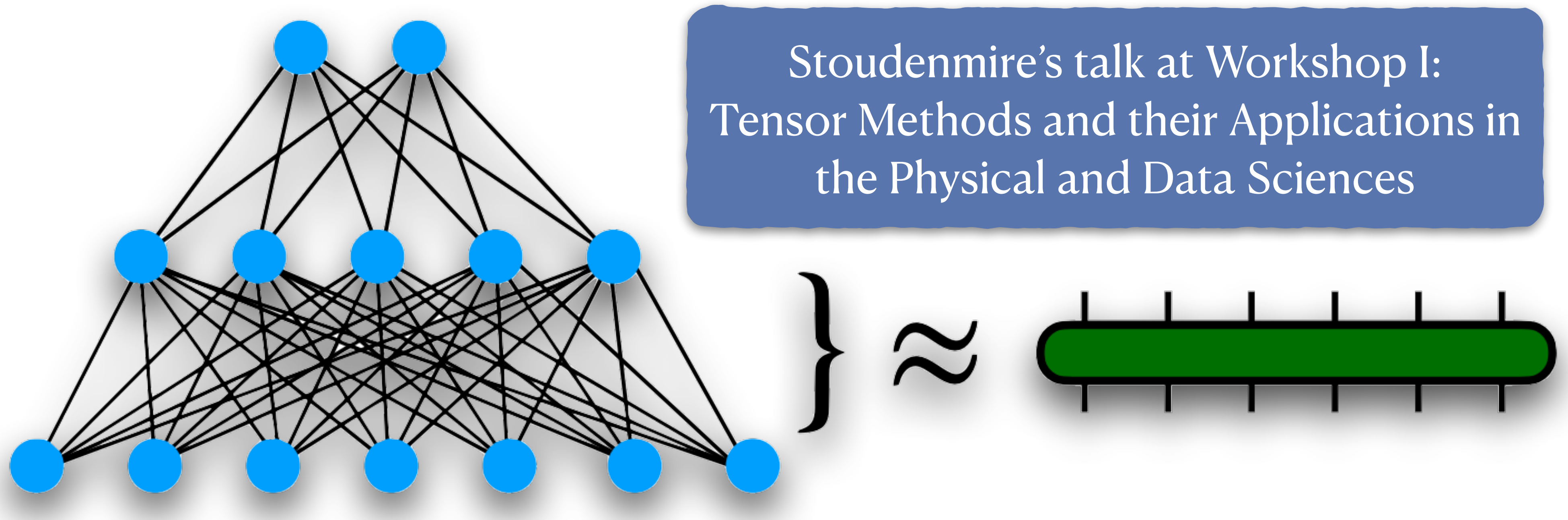


FIG. 3: Exploiting the information provided by the learned TTN classifier: (a) Correlations between the 16 input features (blue for anti-correlated, white for uncorrelated, red for correlated). The numbers indicate q, p_T^{rel}, ΔR of the muon (1-3), kaon (4-6), pion (7-9), electron (10-12), proton (13-15) and the jet charge Q (16). (b) Entropy of each feature as the measure for the information provided for the classification. (c) Tagging power for learning on all features (blue), the best 8 proposed by QuIPS exploiting insights from (a)+(b) (magenta), the worst 8 (yellow) and the muon tagging (red). (d) Tagging power for decreasing bond dimension truncated after training: The complete model (blue shades for $\chi = 100$, $\chi = 50$, $\chi = 5$), for using the QuIPS best 8 features only (violet shades for $\chi = 16$, $\chi = 5$), and the muon tagging (red).

Other applications:

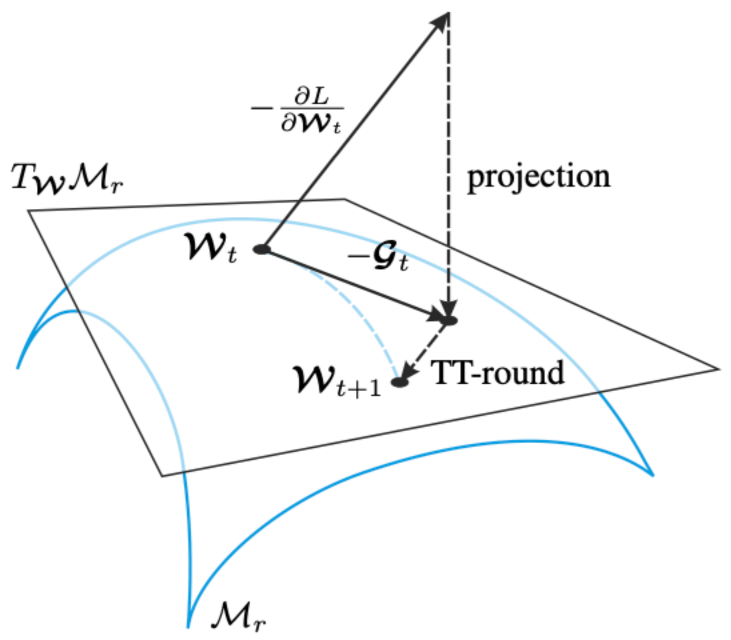
Garipov, Podoprikhin, Novikov, Vetrov
arXiv:1611.03214

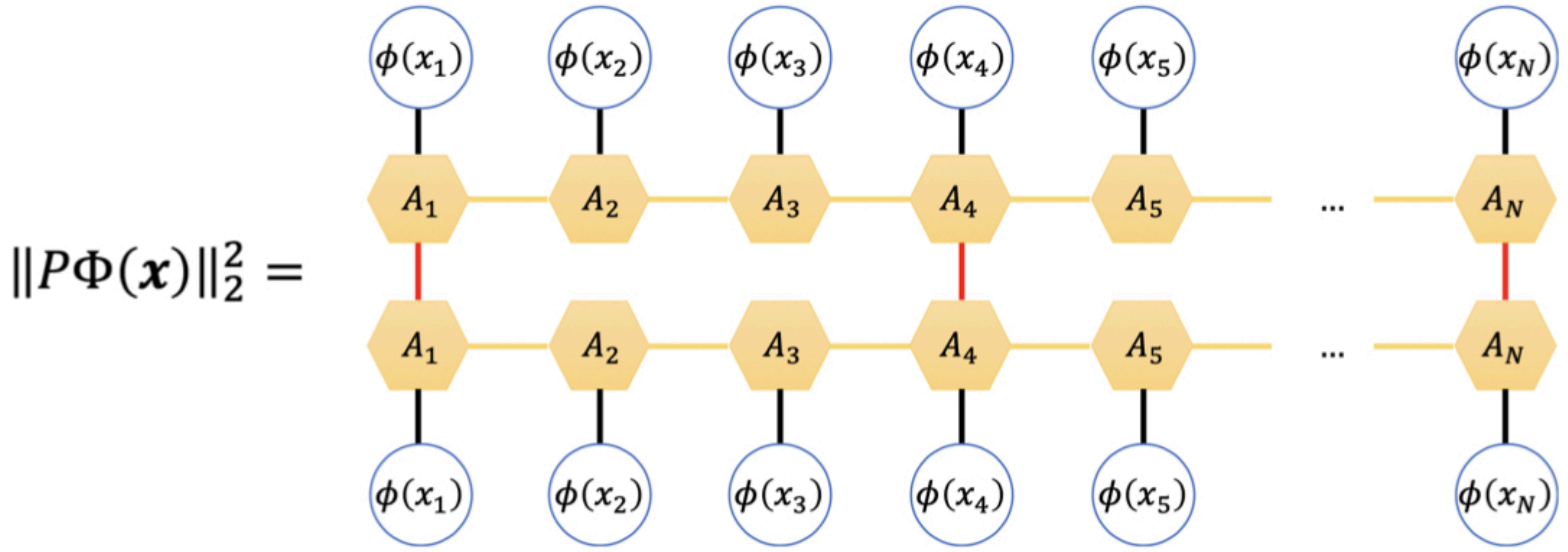
Stoudenmire's talk at Workshop I:
Tensor Methods and their Applications in the Physical and Data Sciences



Polynomial data embedding with Riemannian optimisation

Novikov, Trofimov, Oseledets; arXiv: 1605.03795





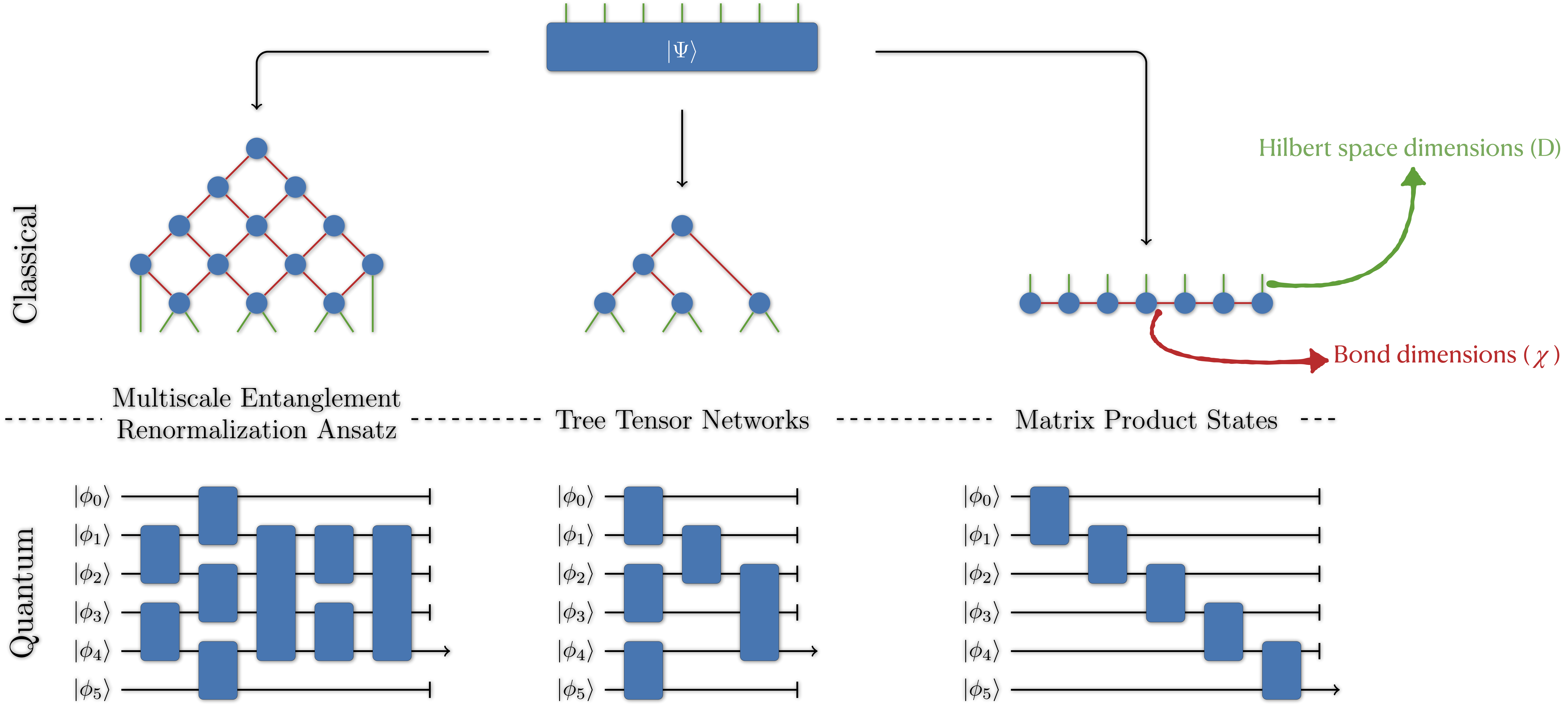
Anomaly detection & generative modelling

Wang, Roberts, Vidal, Leichenauer; arXiv: 2006.02516

Durham University

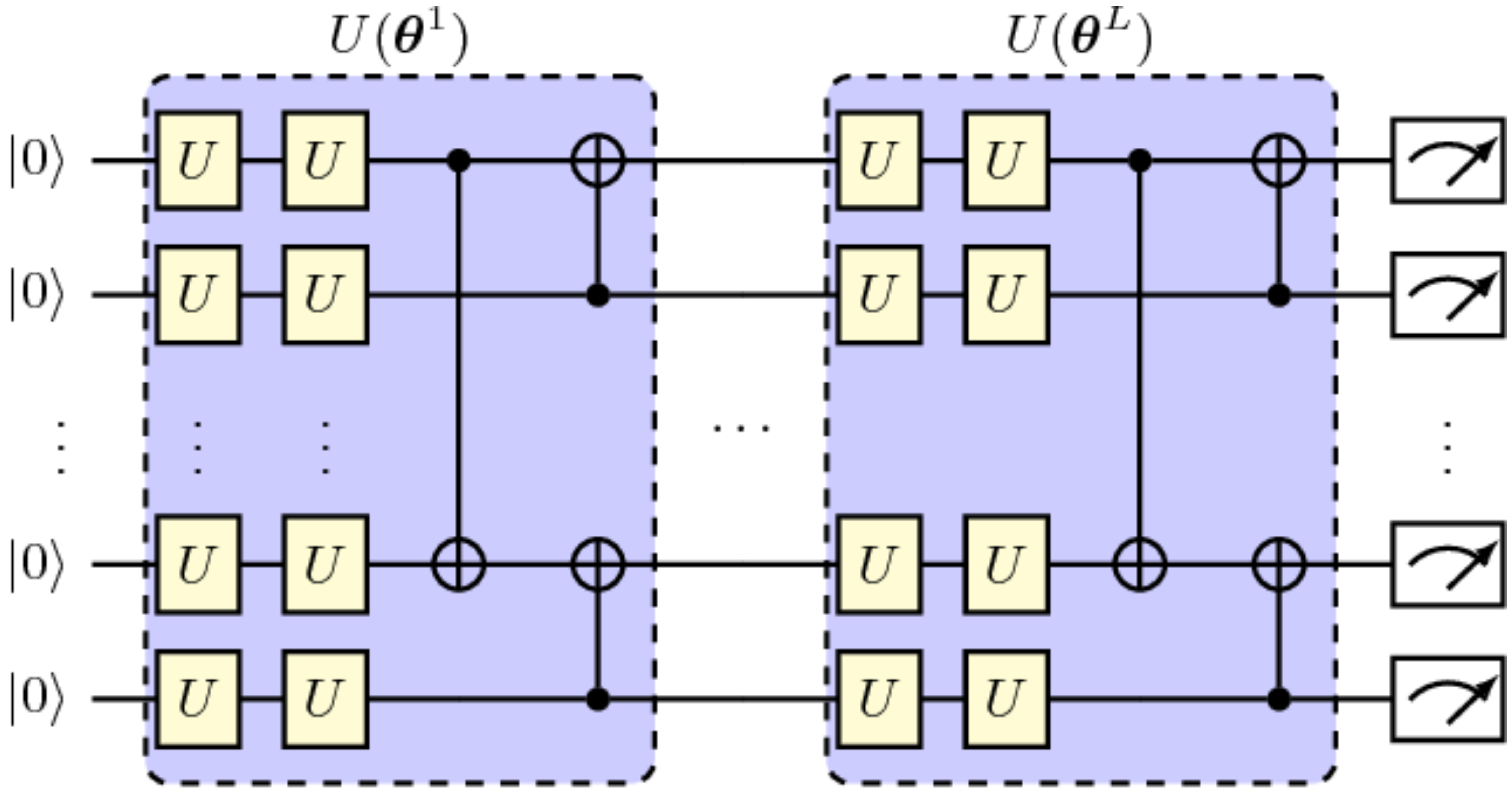
IP3

Types of Tensor Networks (some of them)



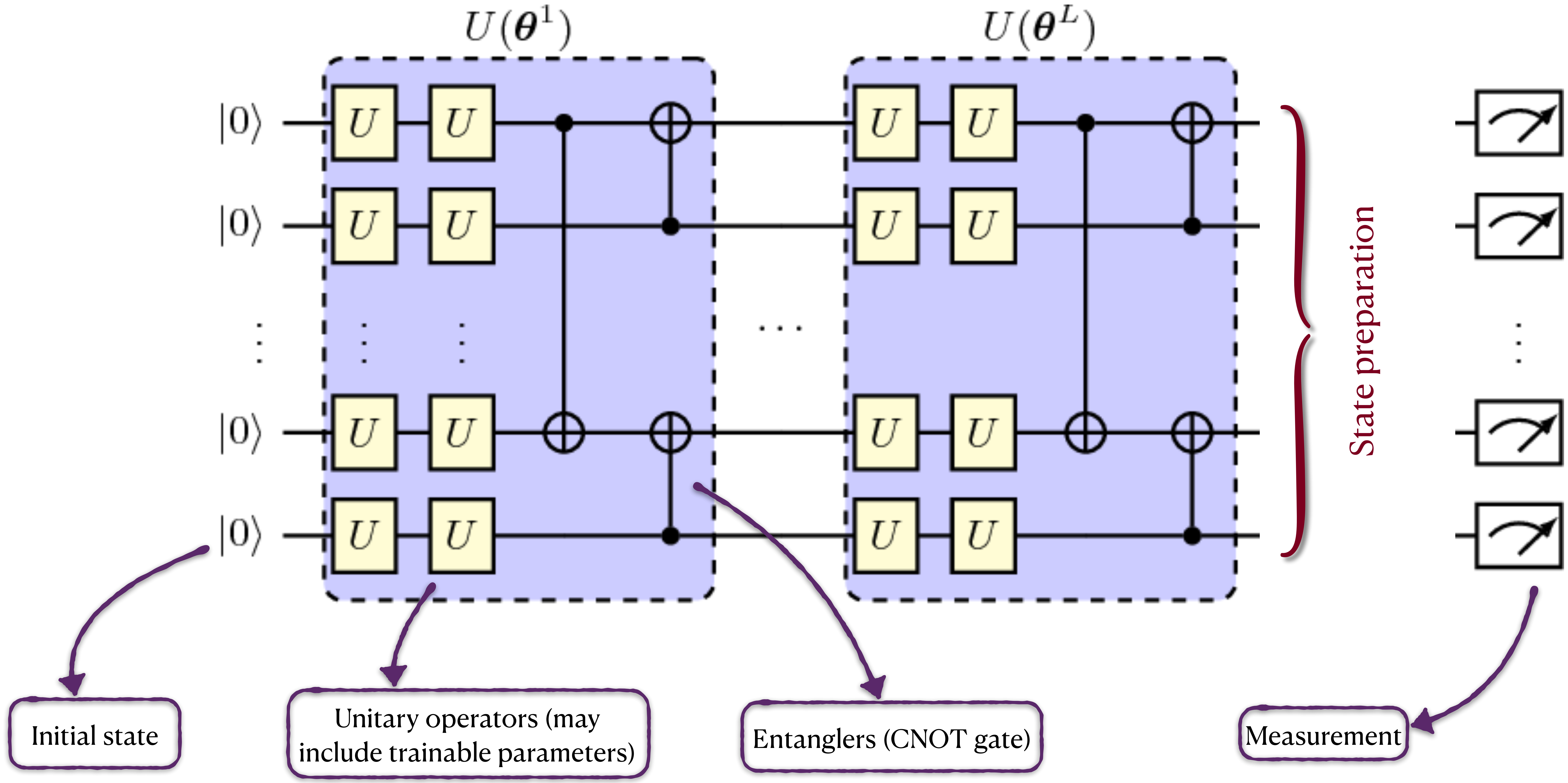
Yet another small intro...

Du, Hsieh, Liu, Tao; Phys. Rev. `20

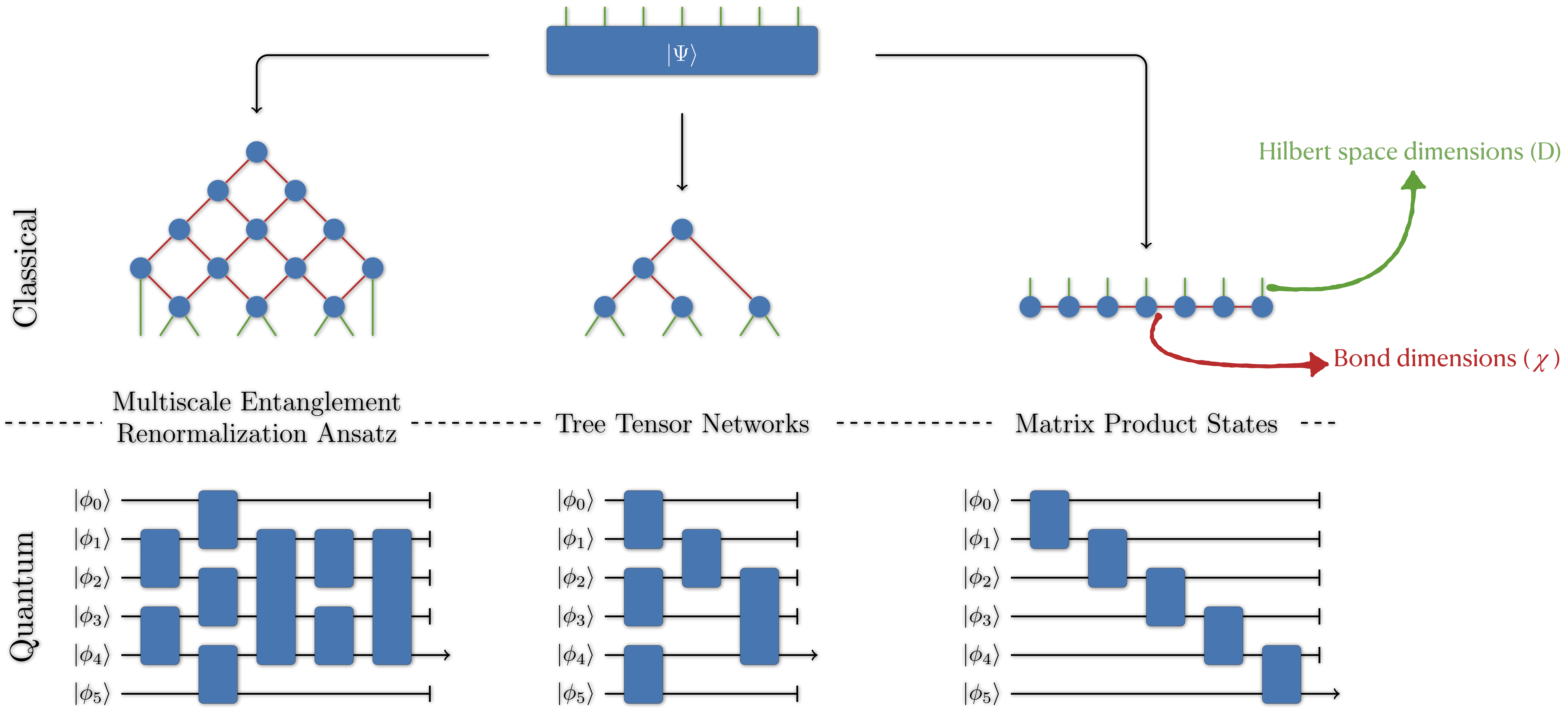


Yet another small intro...

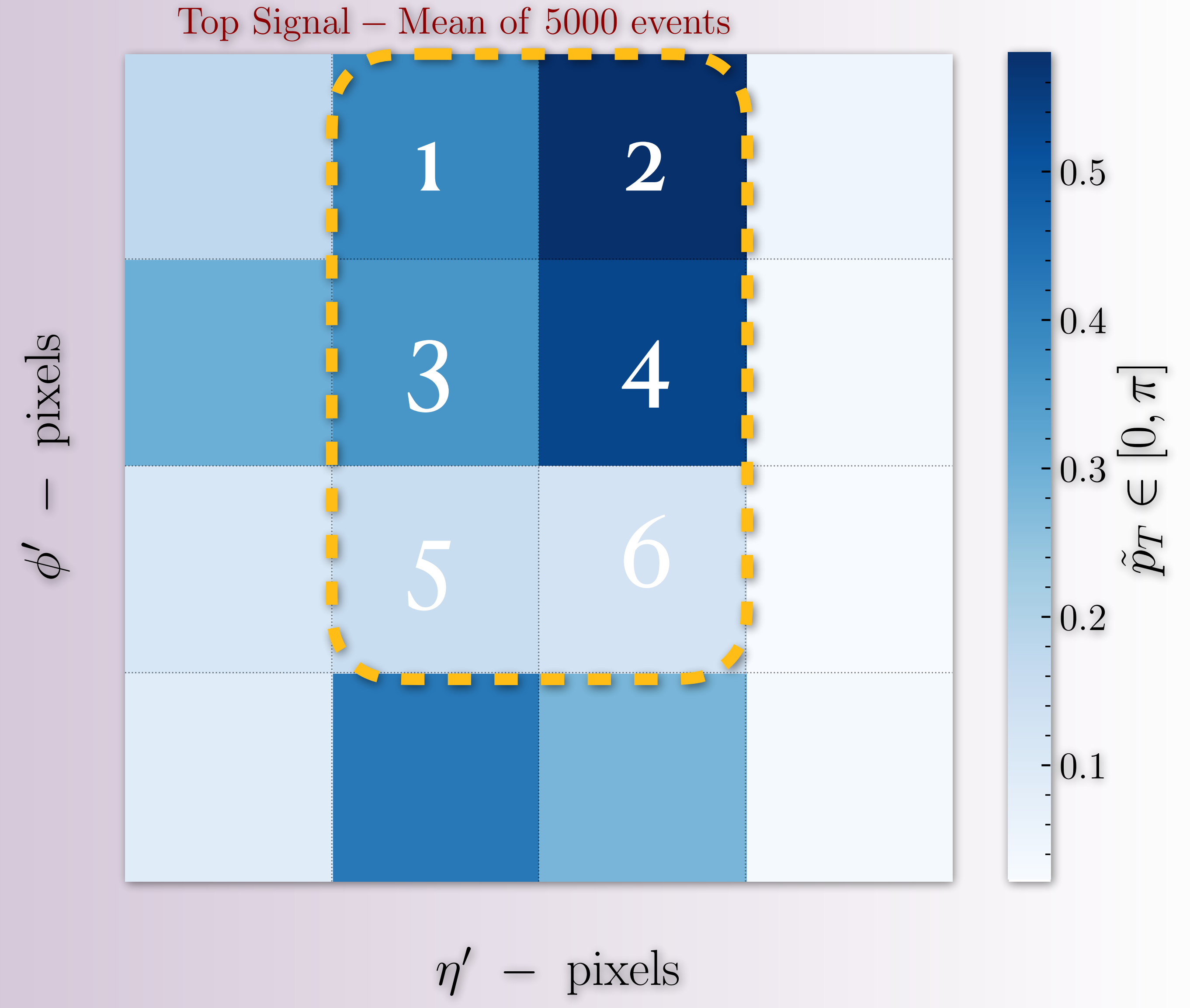
Du, Hsieh, Liu, Tao; Phys. Rev. `20

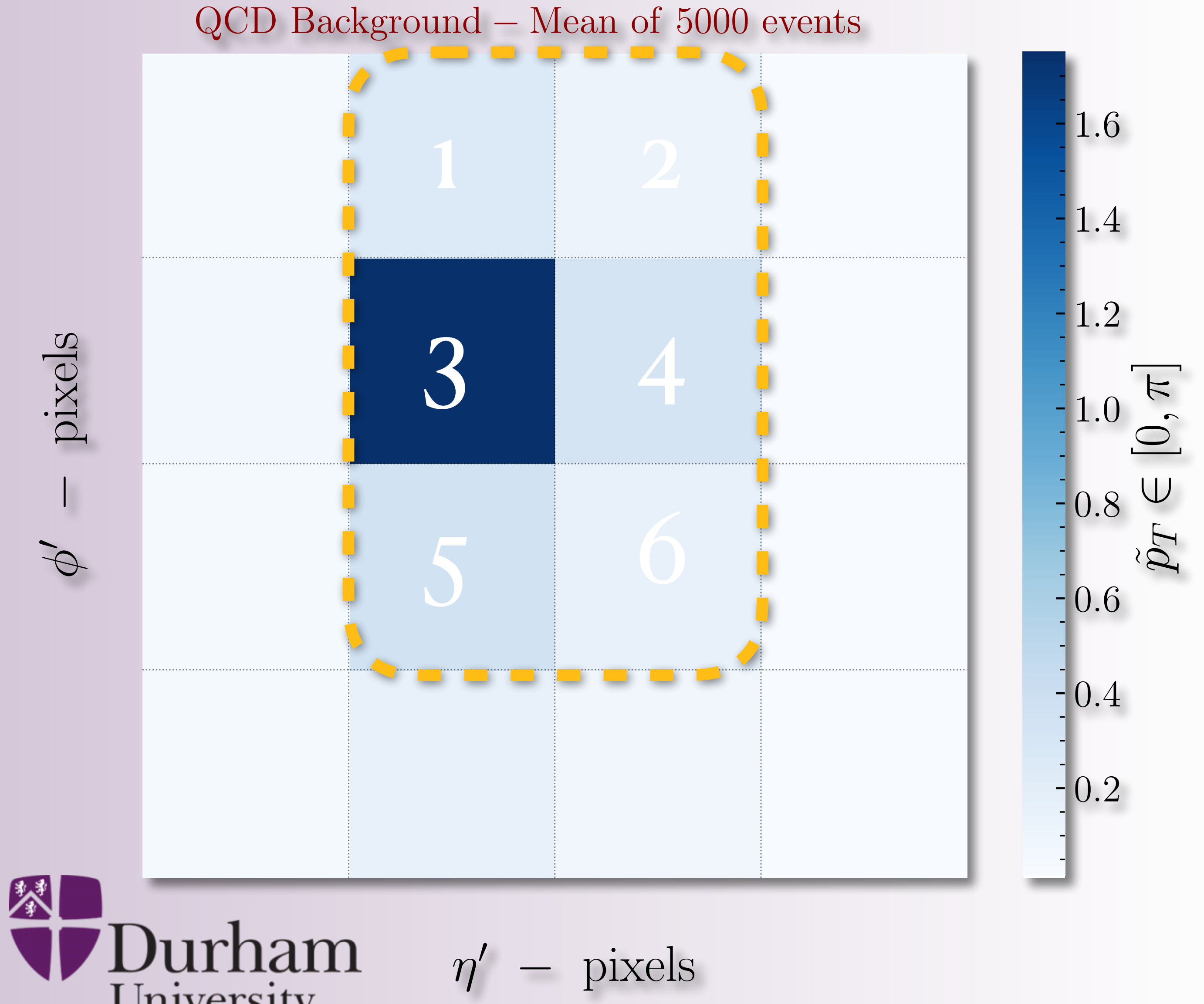


Types of Tensor Networks (some of them)



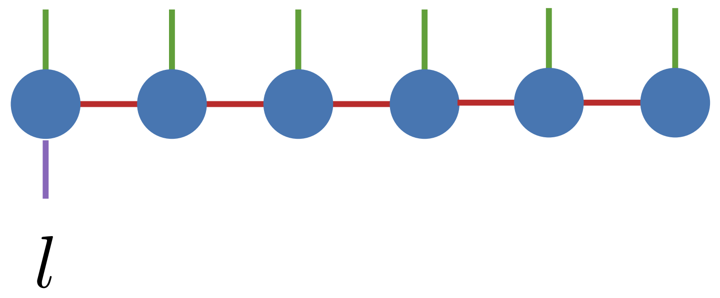
What can we gain if we adopt VQC?

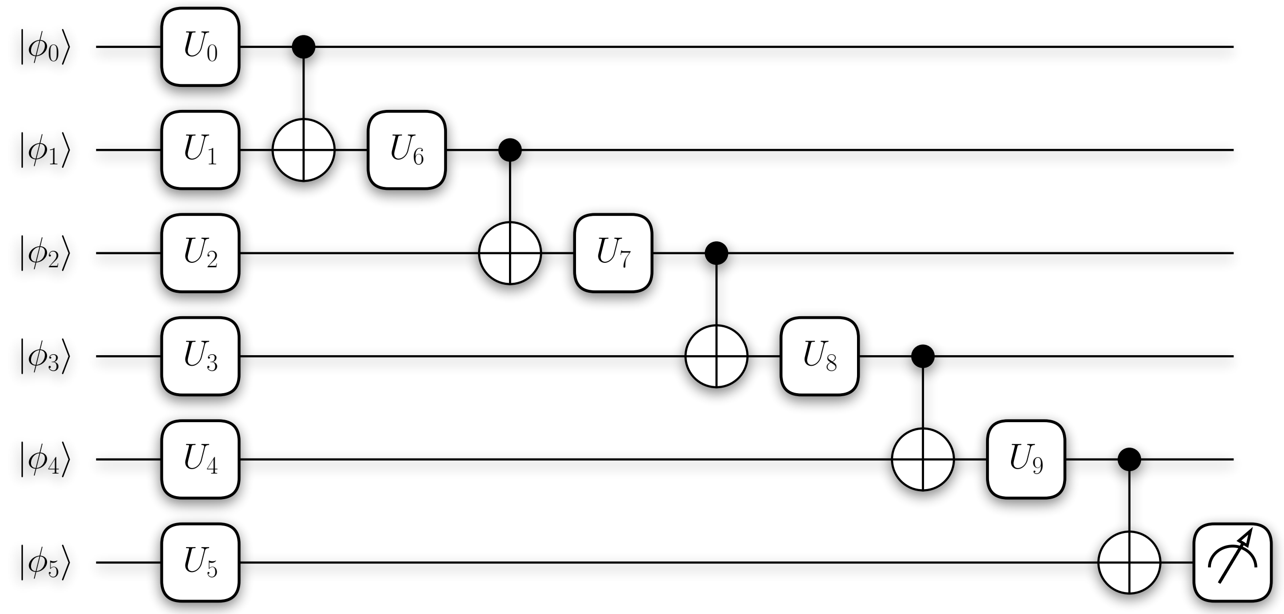




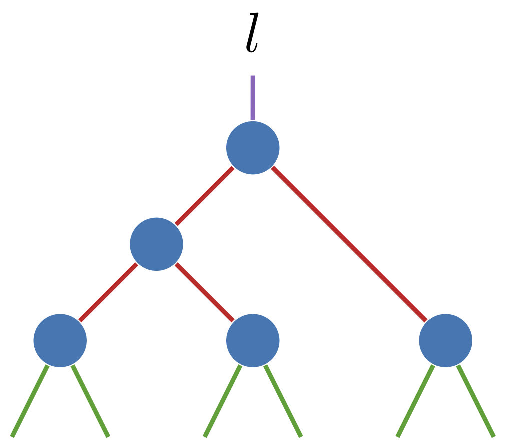
JYA, Spannowsky; PRA '22, arXiv: 2202.10471

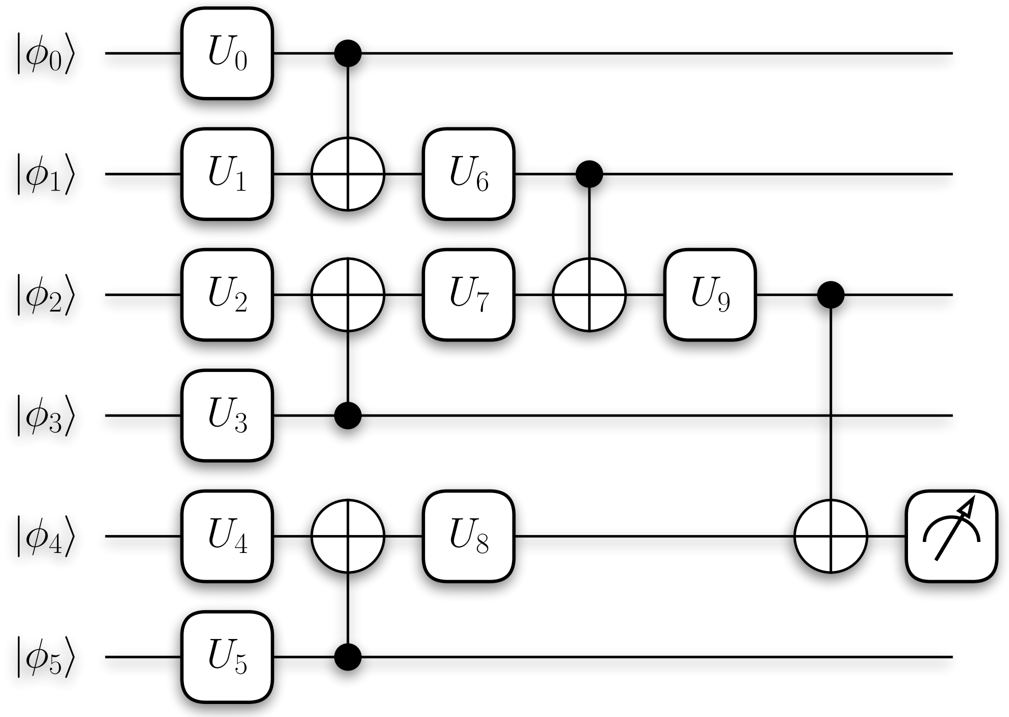
Matrix Product States



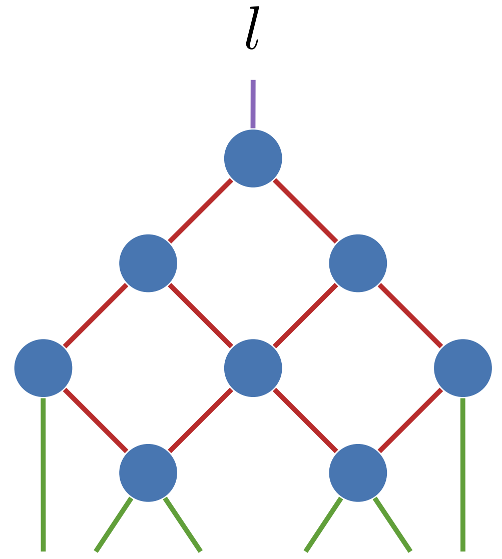


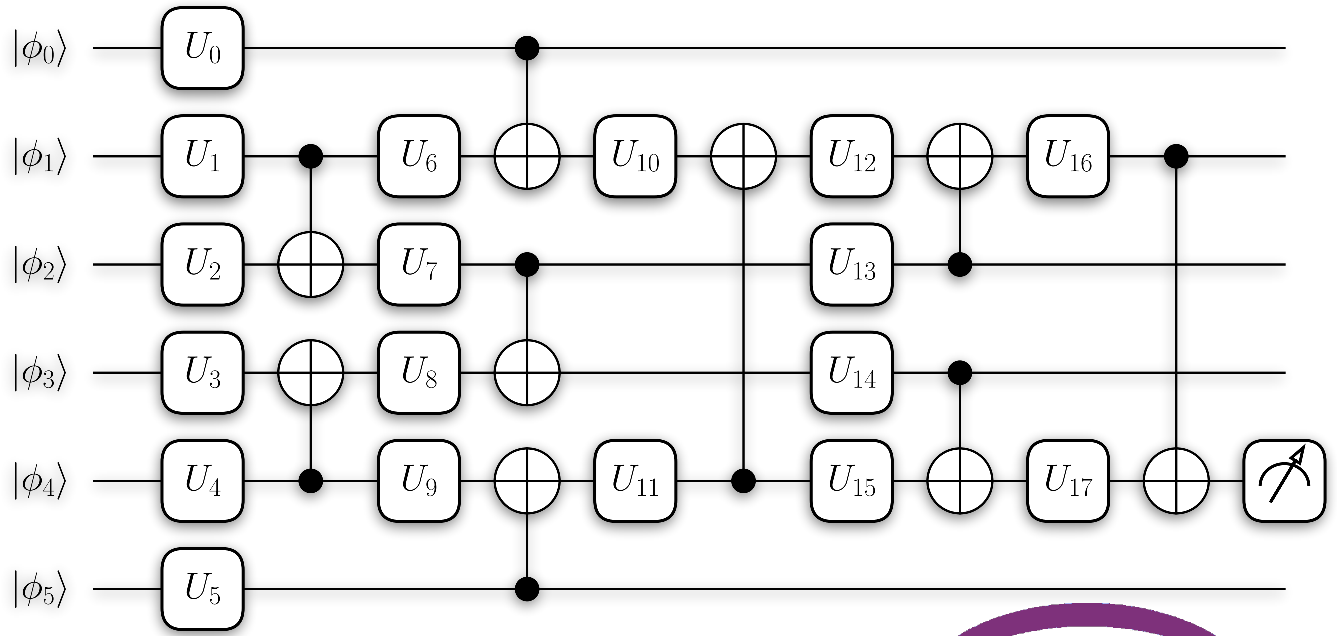
Tree Tensor Networks



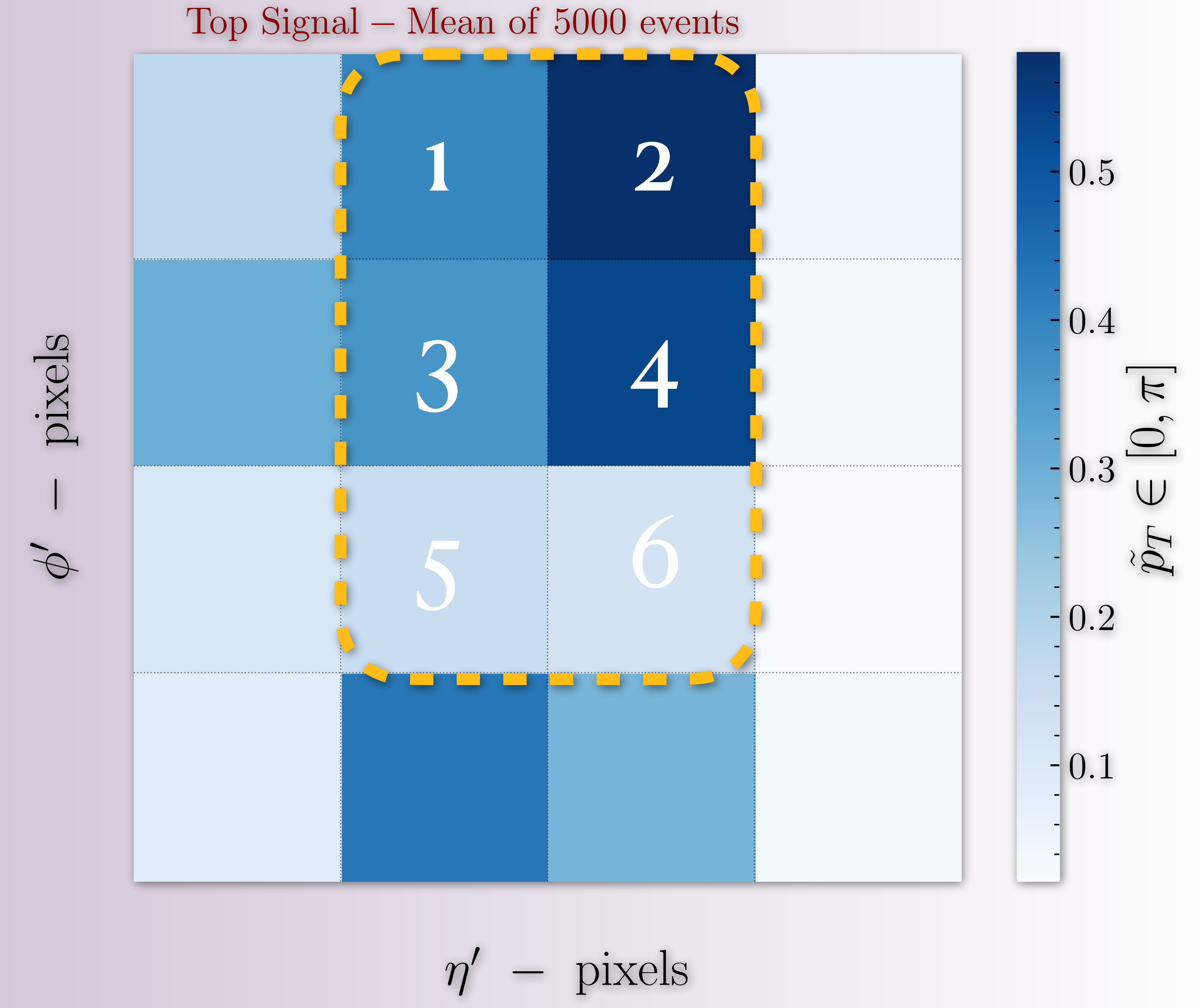


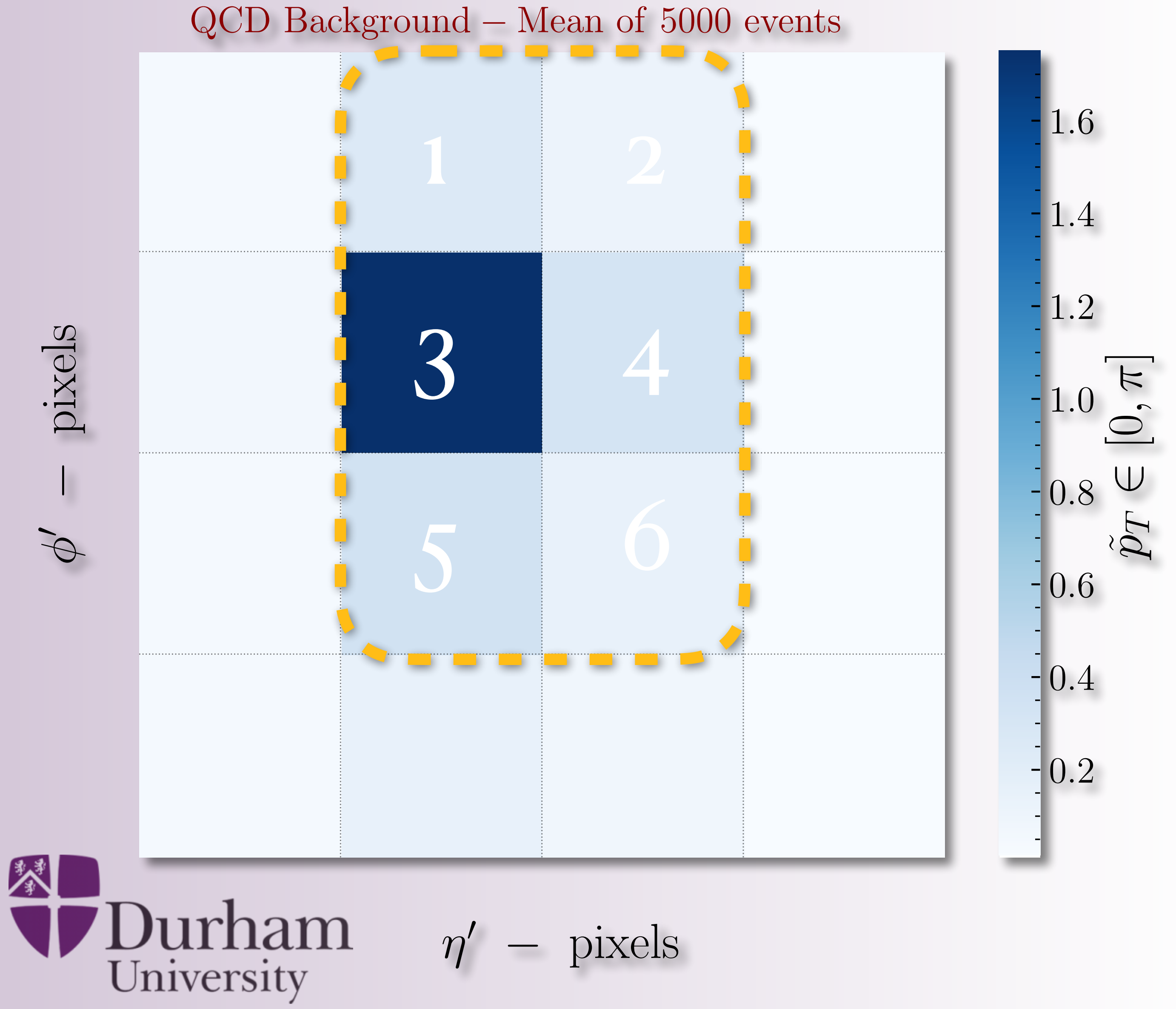
Multiscale Entanglement Renormalisation Ansatz



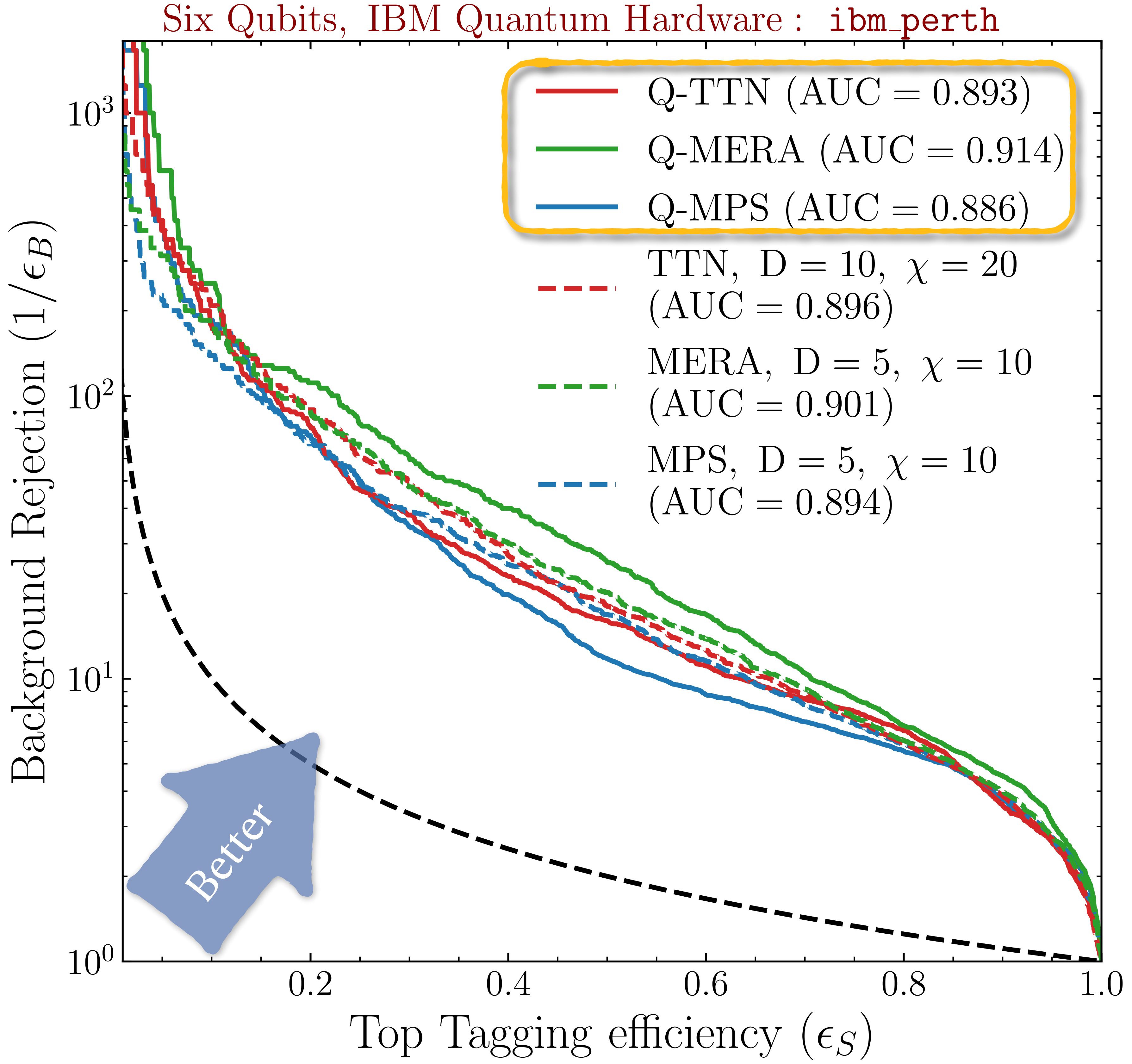


Experimenting with 6-Qubits





JYA, Spannowsky; PRA '22, arXiv: 2202.10471

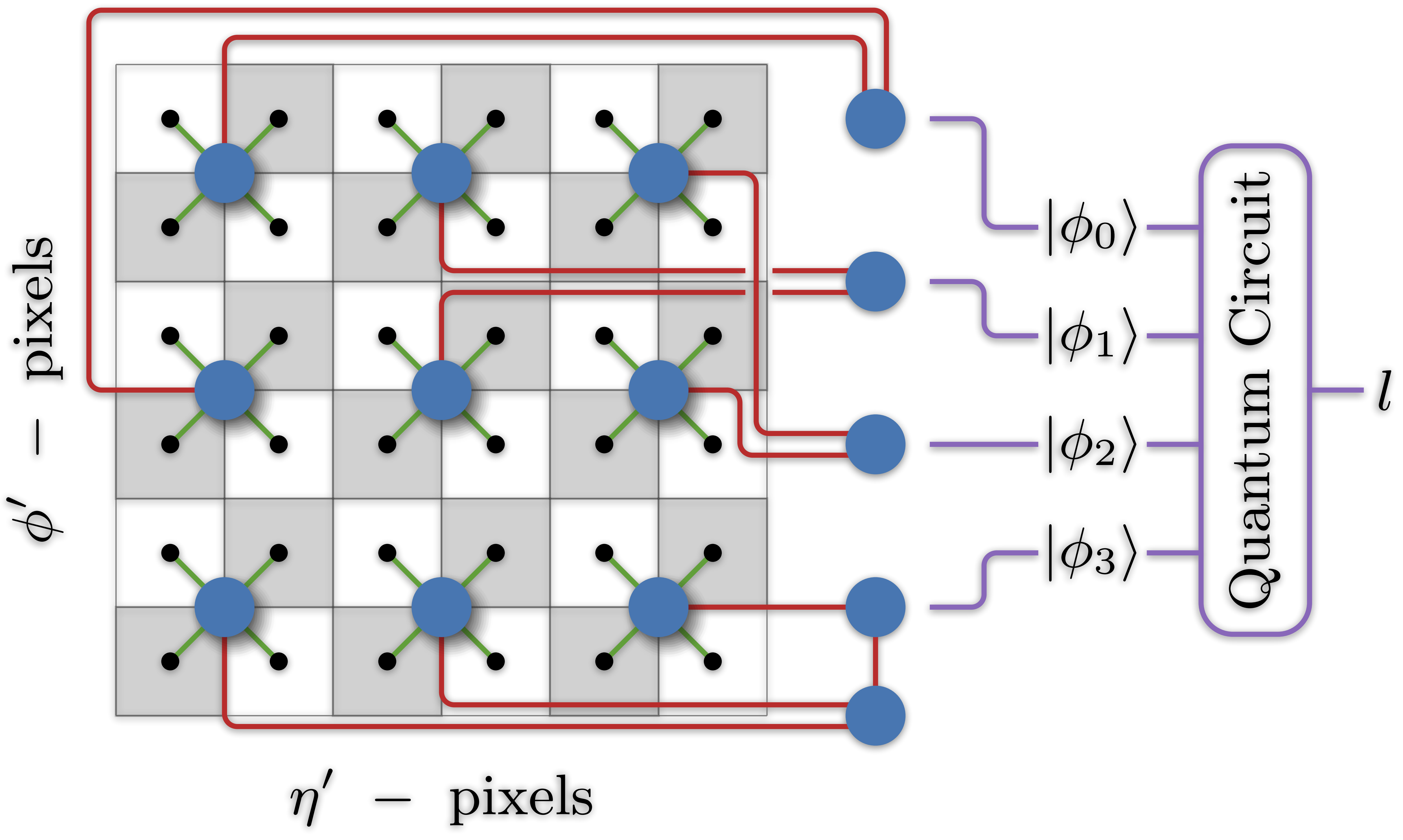


Ansatz	D	χ	# Parameters	AUC
TTN	2	5	235	0.755
	2	10	1320	0.803
	2	20	9040	0.849
	5	10	1950	0.873
	10	20	14800	0.896
MPS	2	5	230	0.811
	2	10	860	0.819
	2	20	3320	0.818
	5	10	2150	0.894
MERA	2	5	1225	0.850
	2	10	13400	0.840
	2	20	181600	0.848
	5	10	18200	0.901
Q-TTN	-	-	9	0.893
Q-MPS	-	-	9	0.886
Q-MERA	-	-	17	0.914

Loss landscape for classical TNs becomes exponentially flat!

Ability to construct dynamic hybrid architectures!

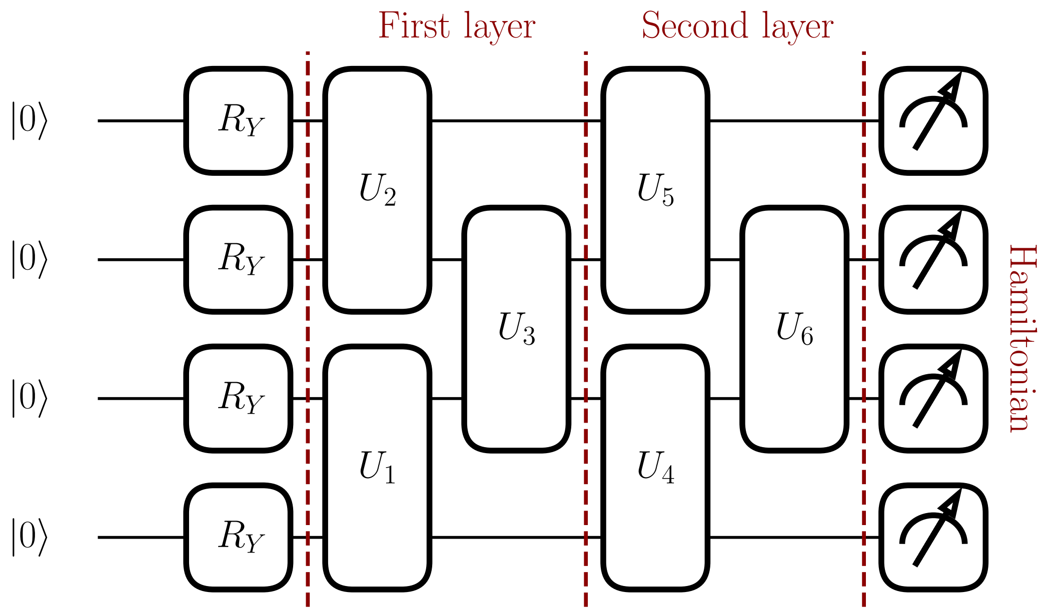
JYA, Spannowsky; PRA '22, arXiv: 2202.10471

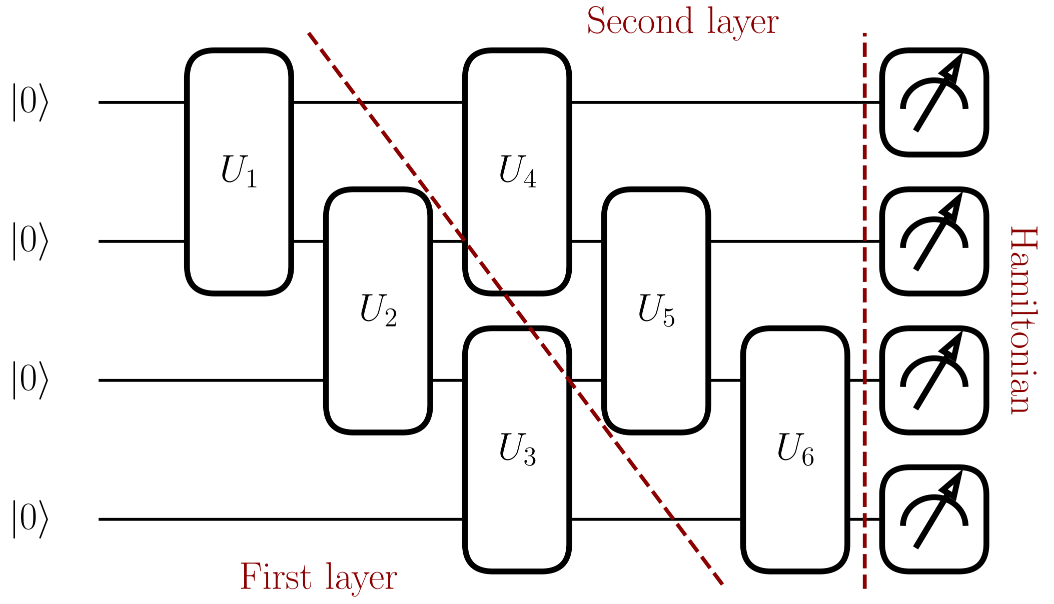


- Near-term quantum devices are quite limited; hence hybrid quantum-classical systems are essential
- Tensor Network nodes can be dynamically converted into qubits, as more will be available in the future!

Classical/Quantum simulation of field theories

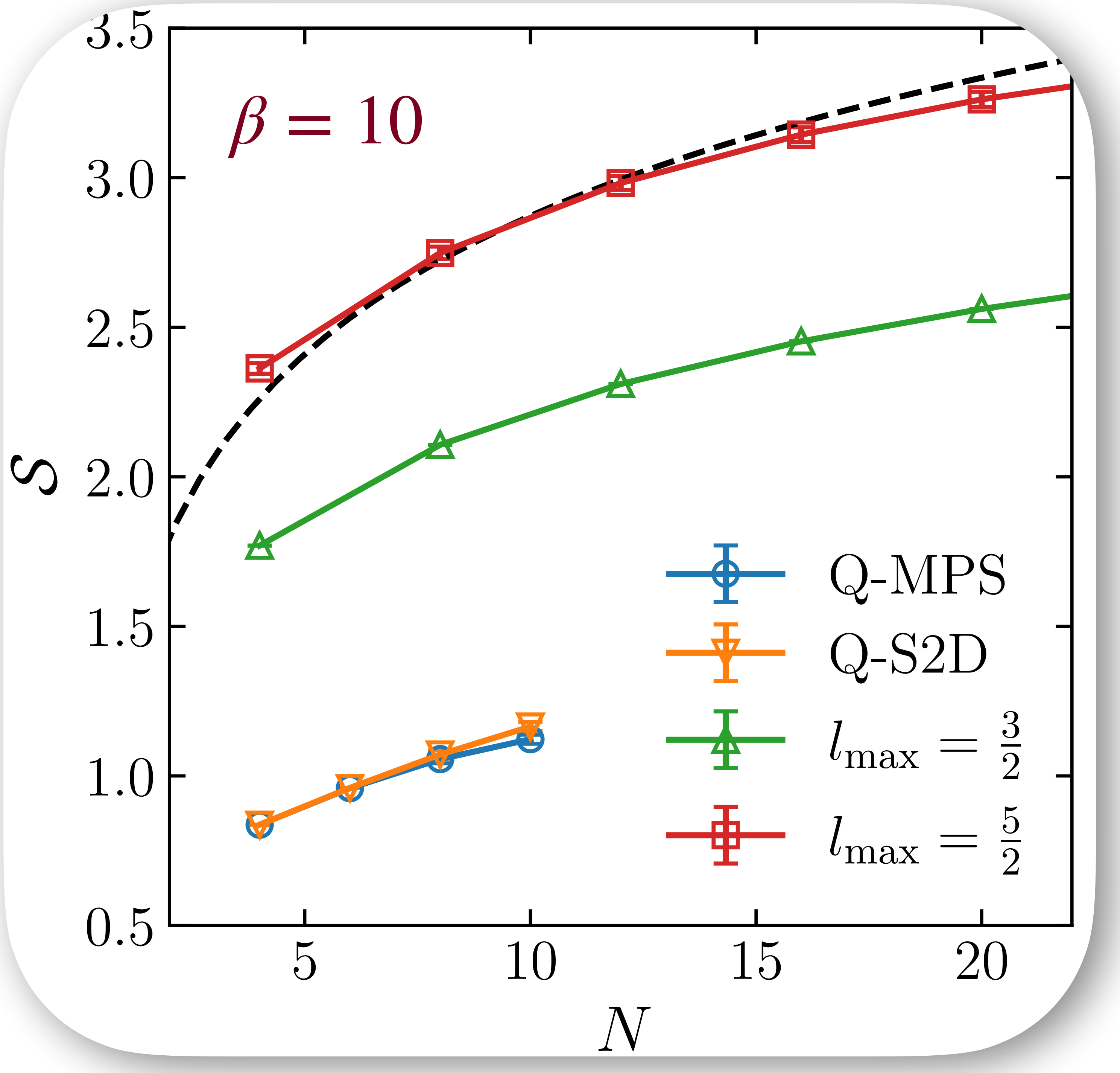
- Both quantum and classical computational techniques have certain limitations for simulating lattice field theories. MC suffers from sign problems, TNs are limited to low dimensional lattices, and near-term noisy quantum circuits are significantly limited in simulating large Hilbert spaces or many sites.





Nonlinear Sigma Model

$$H = \frac{1}{2\beta}\sum_{k}^{N} L_k^2 + \beta \sum_{k}^{N,\,\mathrm{PBC}} \mathbf{n_k n_{k+1}}$$



JYA, Schenk, Spannowsky; arXiv: 2210.03679

Hamiltonian learning for anomaly detection

Jack Y. Araz

Before we dive in...

What is Hamiltonian learning?

What is Variational Quantum Thermaliser?

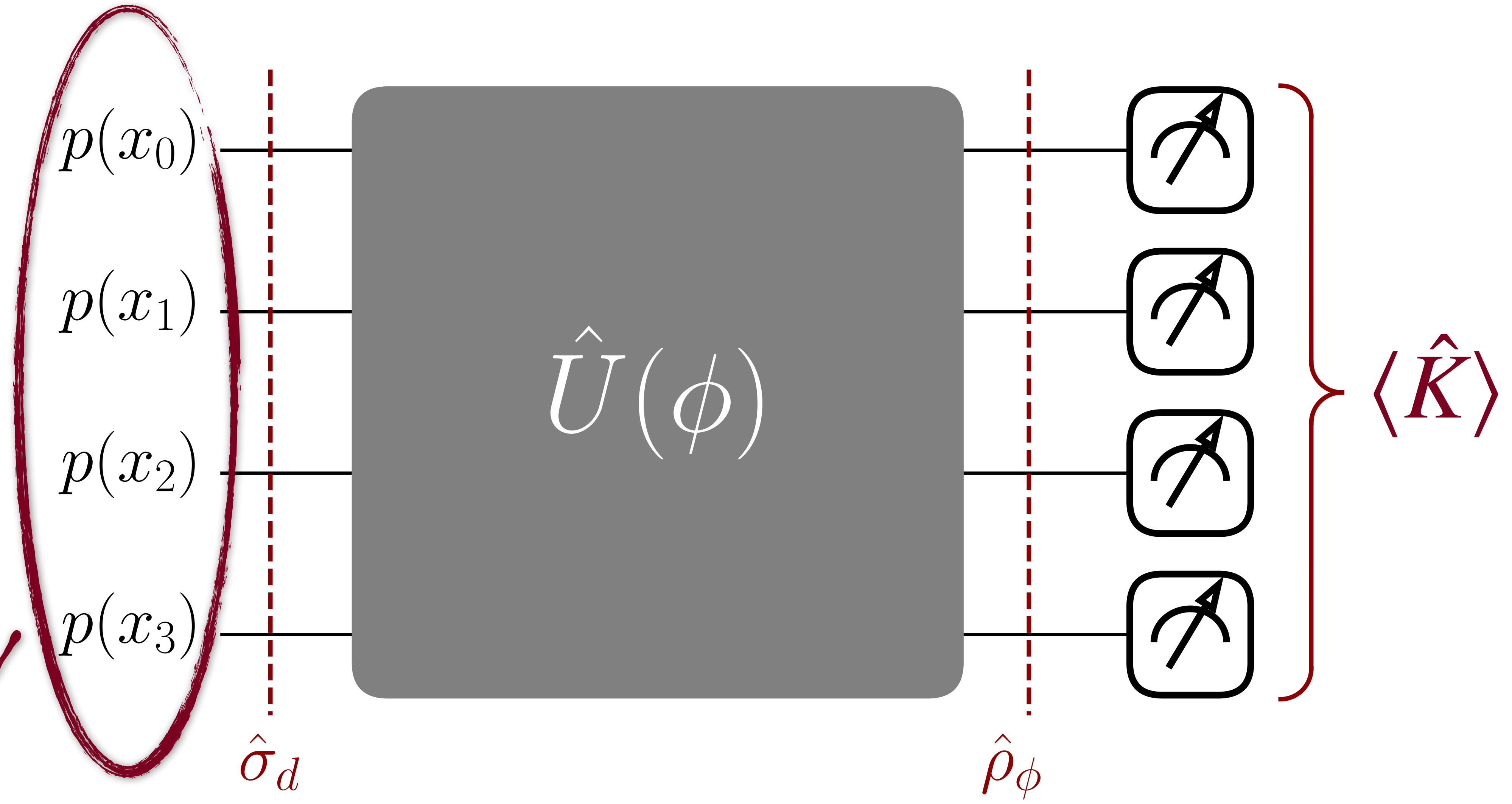
What has Hamiltonian to do with data?

JYA, Spannowsky; arXiv: 2211.03803

Quantum Circuit is a pure-state simulator!

A data point can be represented as a mixed state

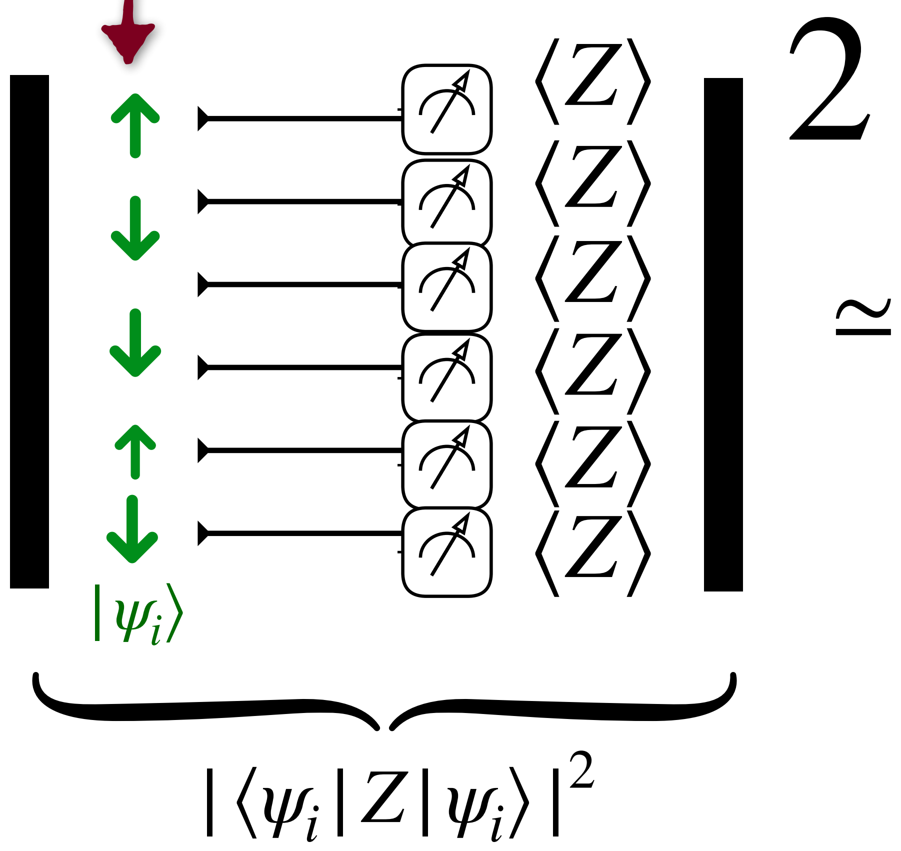
$$\sigma_D = \sum_i p_i |\psi_i\rangle \ , \ |\psi_i\rangle := \text{pure states}$$



Hamiltonian captures the entropic probability distribution of the mixed state.

Bernoulli(p_i)

$$\frac{1}{N_{\text{sample}}} \sum_{\psi_i \in \Psi}^{N_{\text{sample}}}$$



$$\simeq \text{sigmoid(Data)}$$

What type of Hamiltonian can we choose?

- Any field theory Hamiltonian, e.g. Ising model
- A generic Hamiltonian e.g.
  $$\sum_{\langle i,j \rangle} (\alpha_{i,j} \sigma_i^+ \sigma_j^- + \text{h.c.})$$
- But can these options capture the full complexity of the data? Can we get ambitious?

Durham University

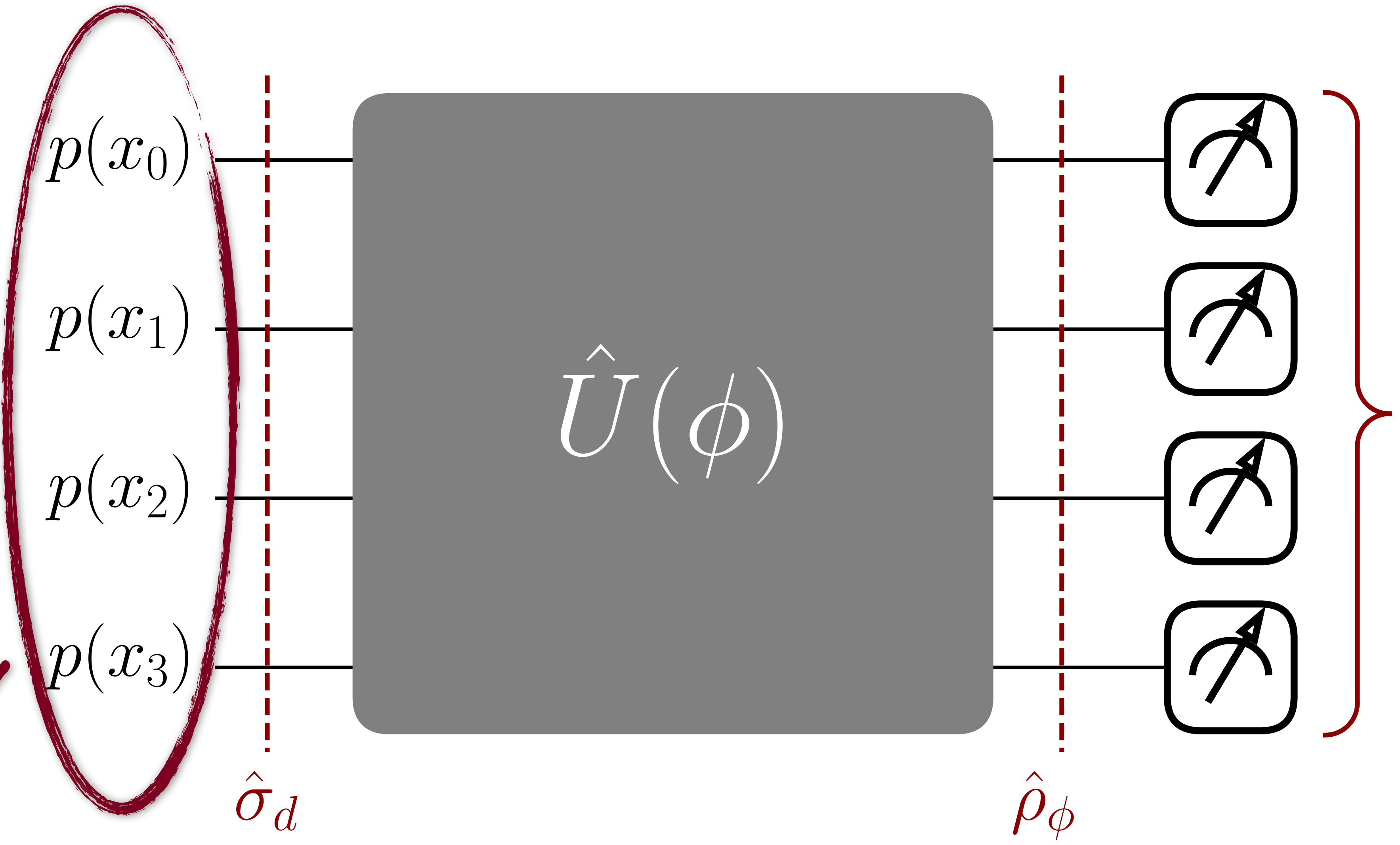
IP3

What has Hamiltonian to do with data?

JYA, Spannowsky; arXiv: 2211.03803

A data point can be represented as a mixed state

$$\sigma_D = \sum_i p_i |\psi_i\rangle \ , \ |\psi_i\rangle := \text{pure states}$$

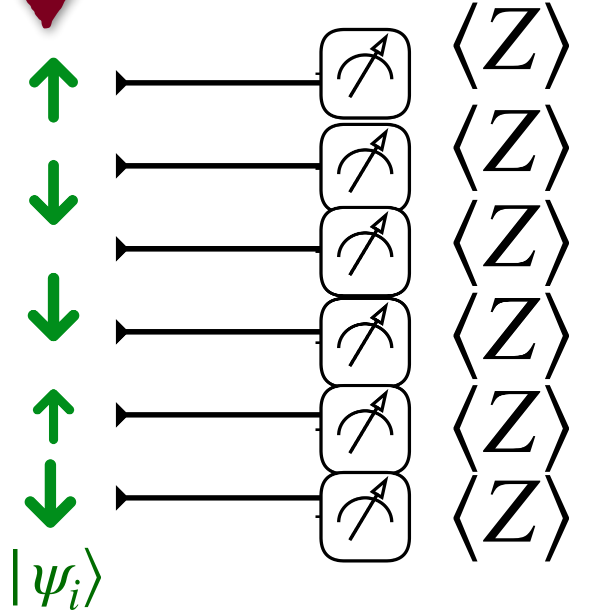


Hamiltonian captures the entropic probability distribution of the mixed state.

$$K_\theta = \sum E_\theta(s) |s\rangle\langle s|$$

$$\hat{\rho}_\phi \to \hat{\rho}_{\theta,\phi} = \frac{1}{Z_\theta} e^{-\beta \hat{K}_\theta}$$

Bernoulli(p_i)



$$\frac{1}{N_{\text{sample}}} \sum_{\psi_i \in \Psi}^{N_{\text{sample}}} \Big| \underbrace{\cdots}_{|\langle \psi_i | Z | \psi_i \rangle|^2} \Big|^2 \simeq \text{sigmoid(Data)}$$

See Gibbs-Delbrück-Moliéve variational principle

$$F = E - TS = -k_\beta T \ln Z_\theta$$

$$\mathcal{L}_{\theta,\phi}(\sigma_D) = \beta \langle \hat{K} \rangle_{\theta,\phi} + k_\beta \ln Z_\theta \ \geq \ S(\sigma_D)$$

$k_\beta :=$ Boltzmann constant

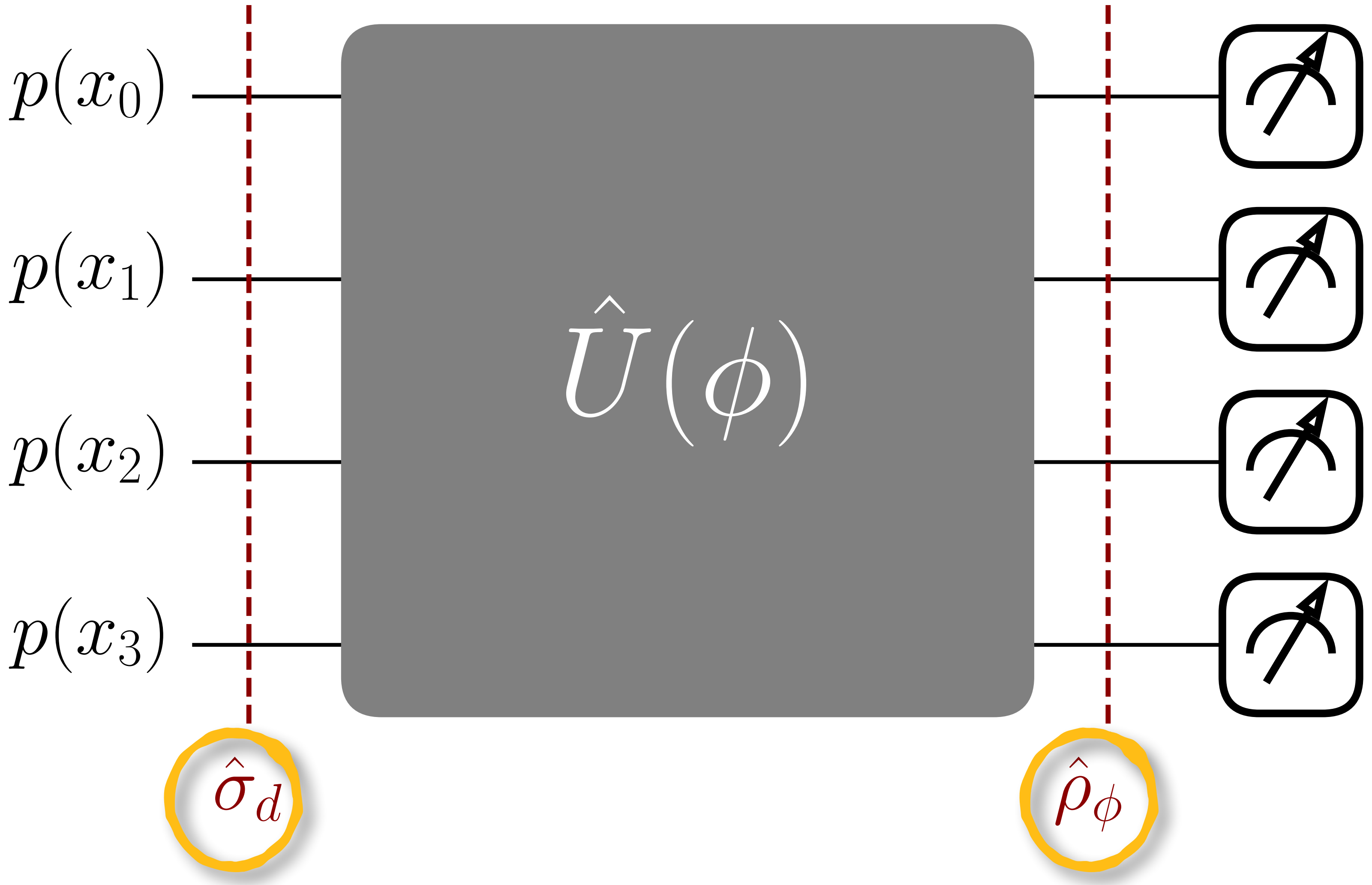
$\beta :=$ Inverse temperature

Durham University

IP3

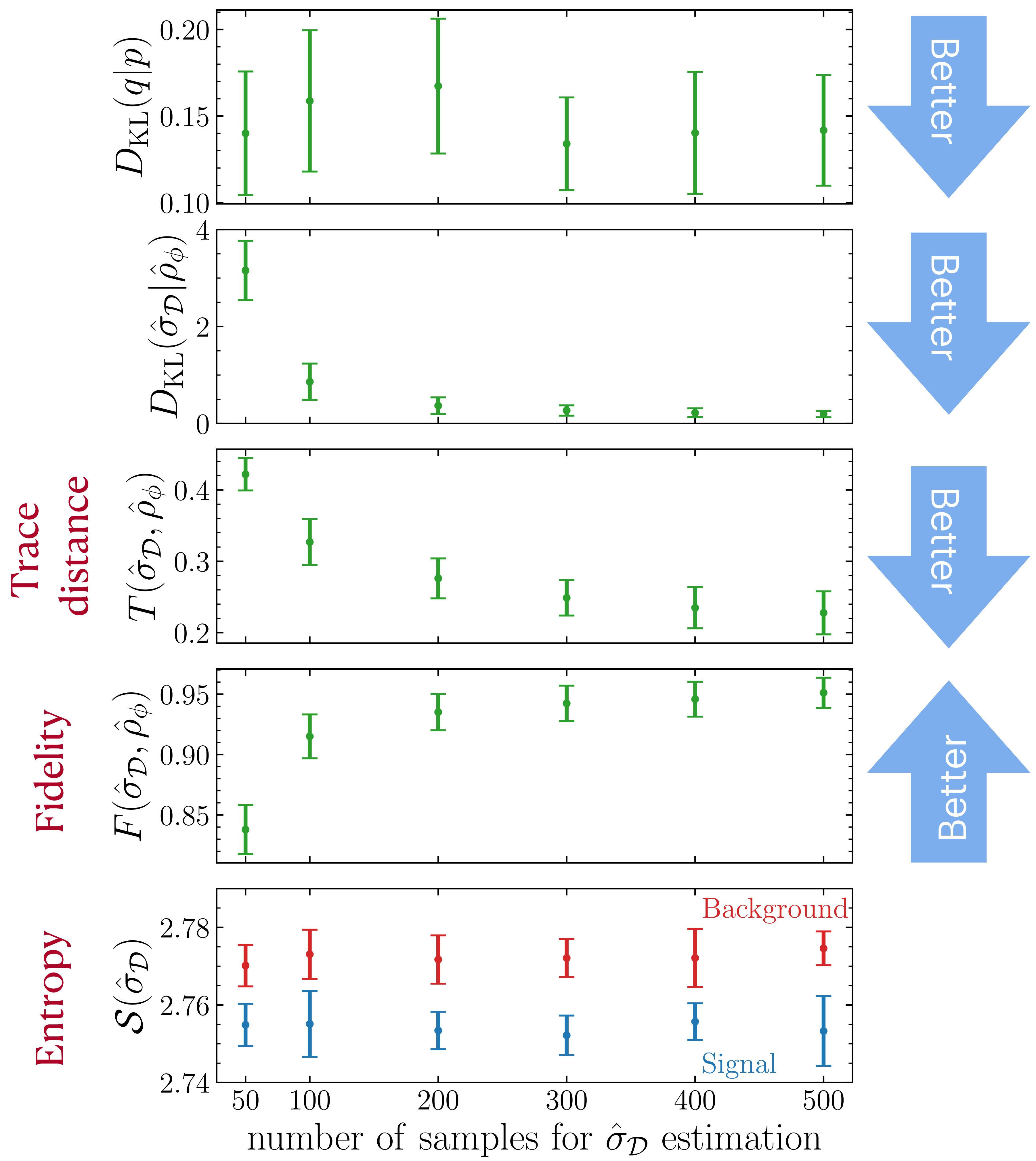
What has Hamiltonian to do with data?

JYA, Spannowsky; arXiv: 2211.03803



$$\mathscr{L}_{\theta,\phi}(\sigma_D) = \langle \hat{K} \rangle_{\theta,\phi} + \ln Z_\theta \;\geq\; S(\sigma_D)$$

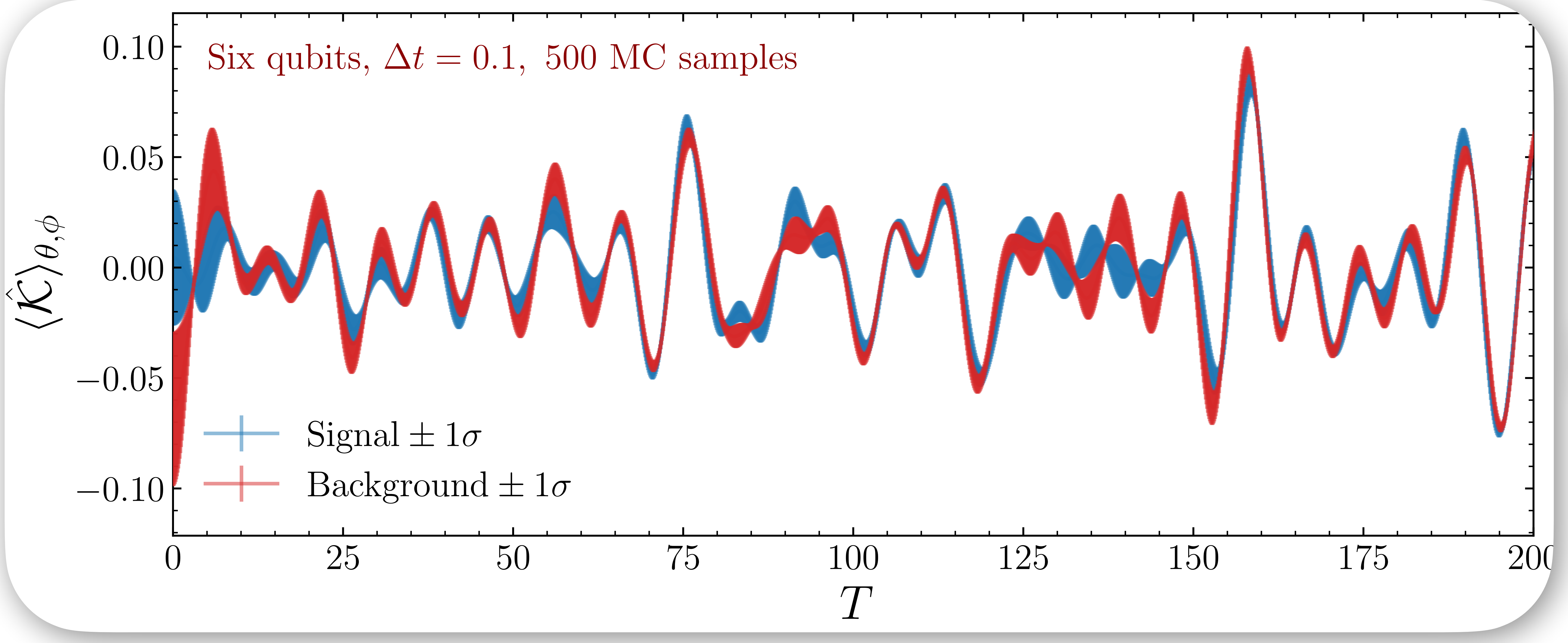
$$\arg\min_{\theta,\phi} \mathscr{L}_{\theta,\phi}(\sigma_D) \simeq S(\sigma_D)$$



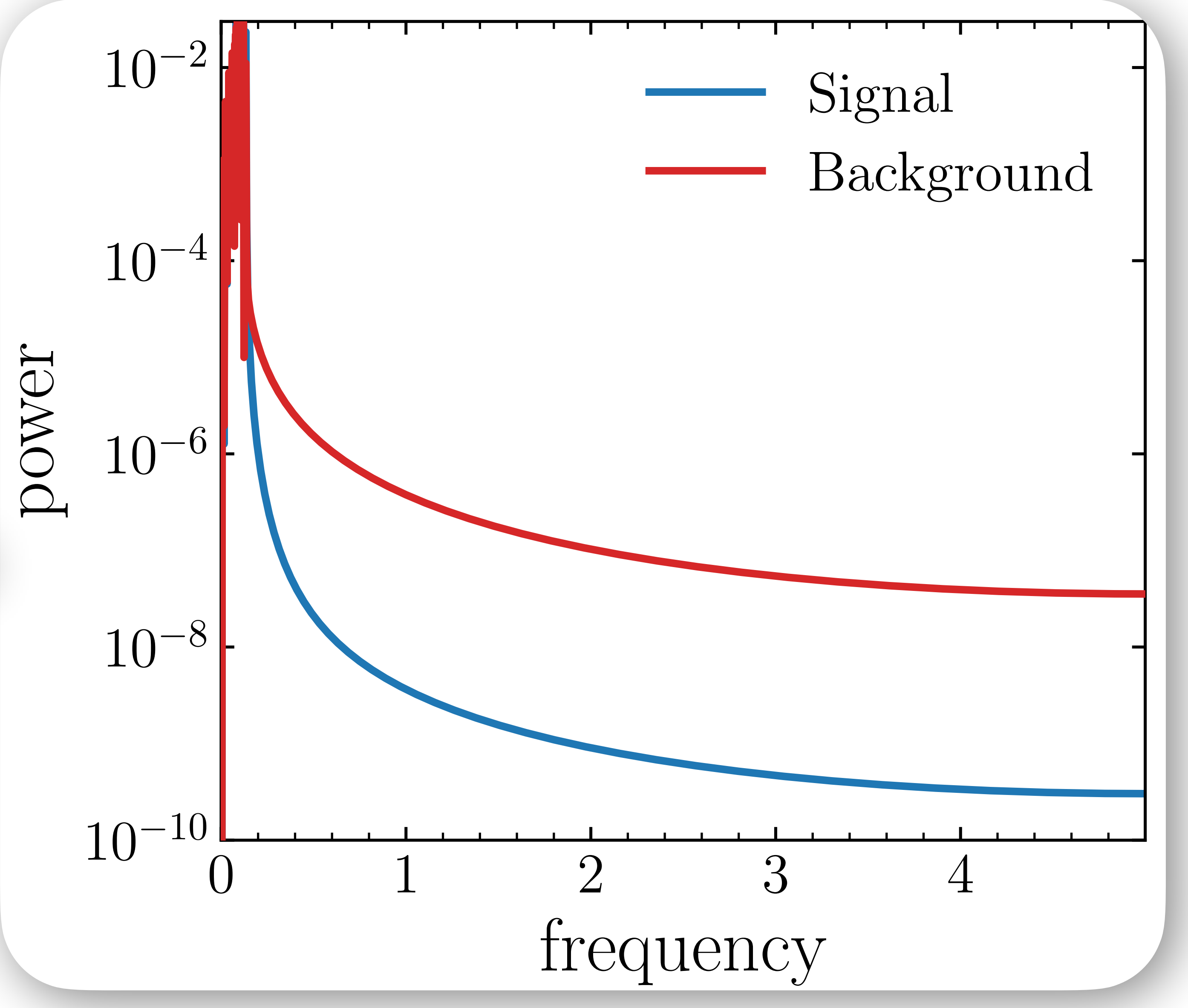
Hamiltonian as a discriminator!

JYA, Spannowsky; arXiv: 2211.03803

Trotter-Suzuki approximation → $e^{-iT\hat{K}_\theta} = \prod^{N} e^{-i\Delta t \hat{K}_\theta}$



Fast Fourier Transform

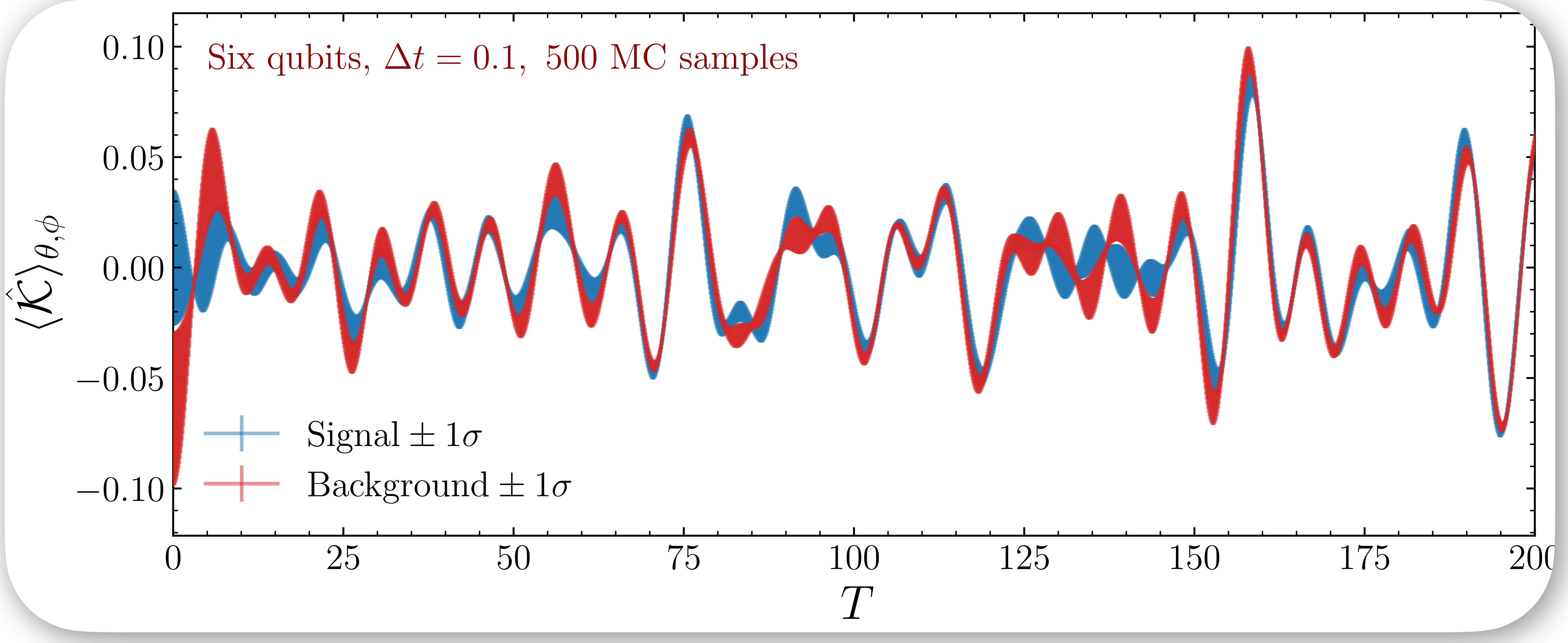


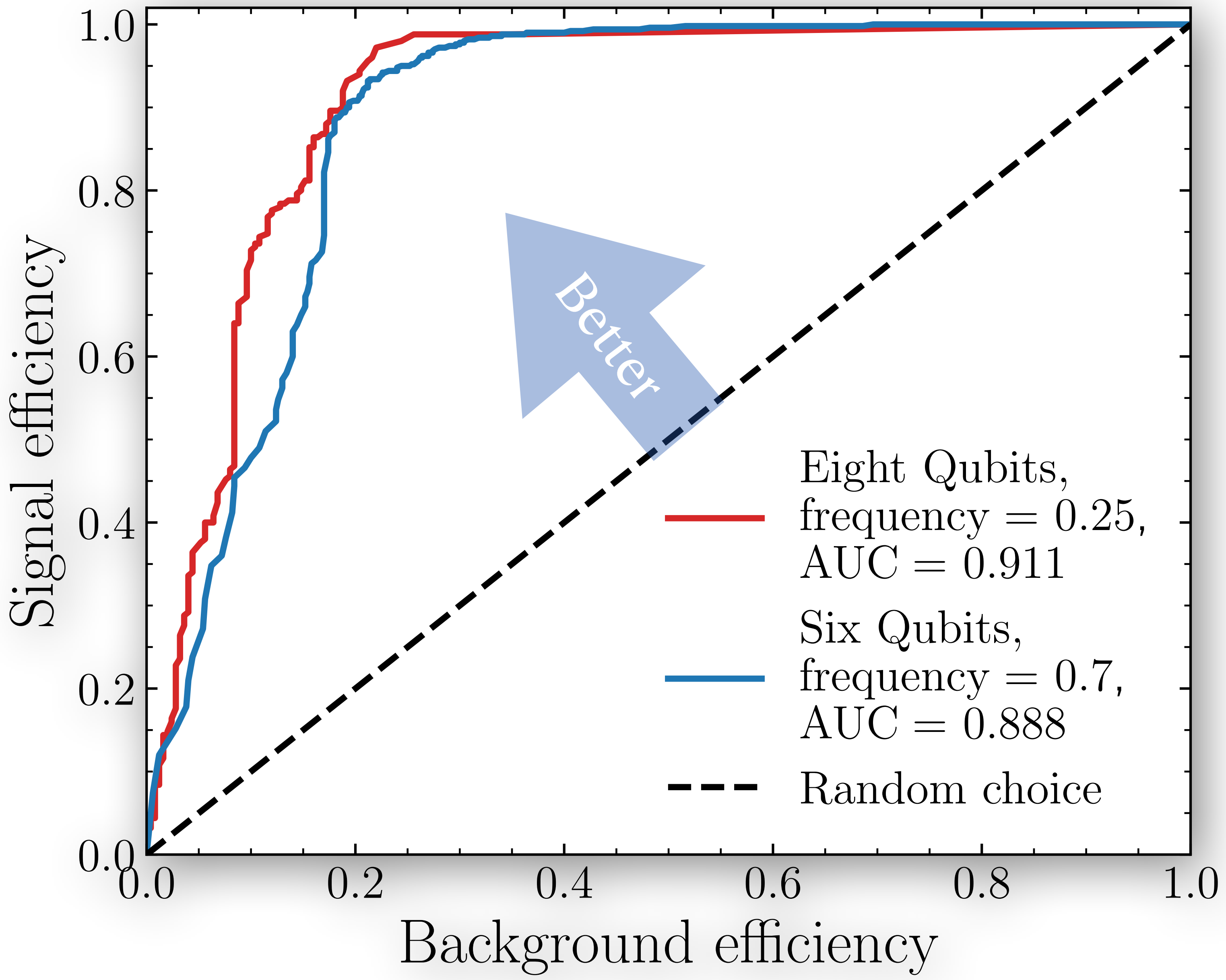
$$\langle \hat{K} \rangle_{\theta,\phi} = \frac{1}{N_{\text{smp}}} \sum_{\sigma}^{N_{\text{smp}}} \langle \sigma_T^i | \hat{K}_\theta | \sigma_T^i \rangle$$

Hamiltonian as a discriminator!

JYA, Spannowsky; arXiv: 2211.03803

Trotter-Suzuki approximation → $e^{-iT\hat{K}_\theta} = \prod^{N} e^{-i\Delta t\hat{K}_\theta}$



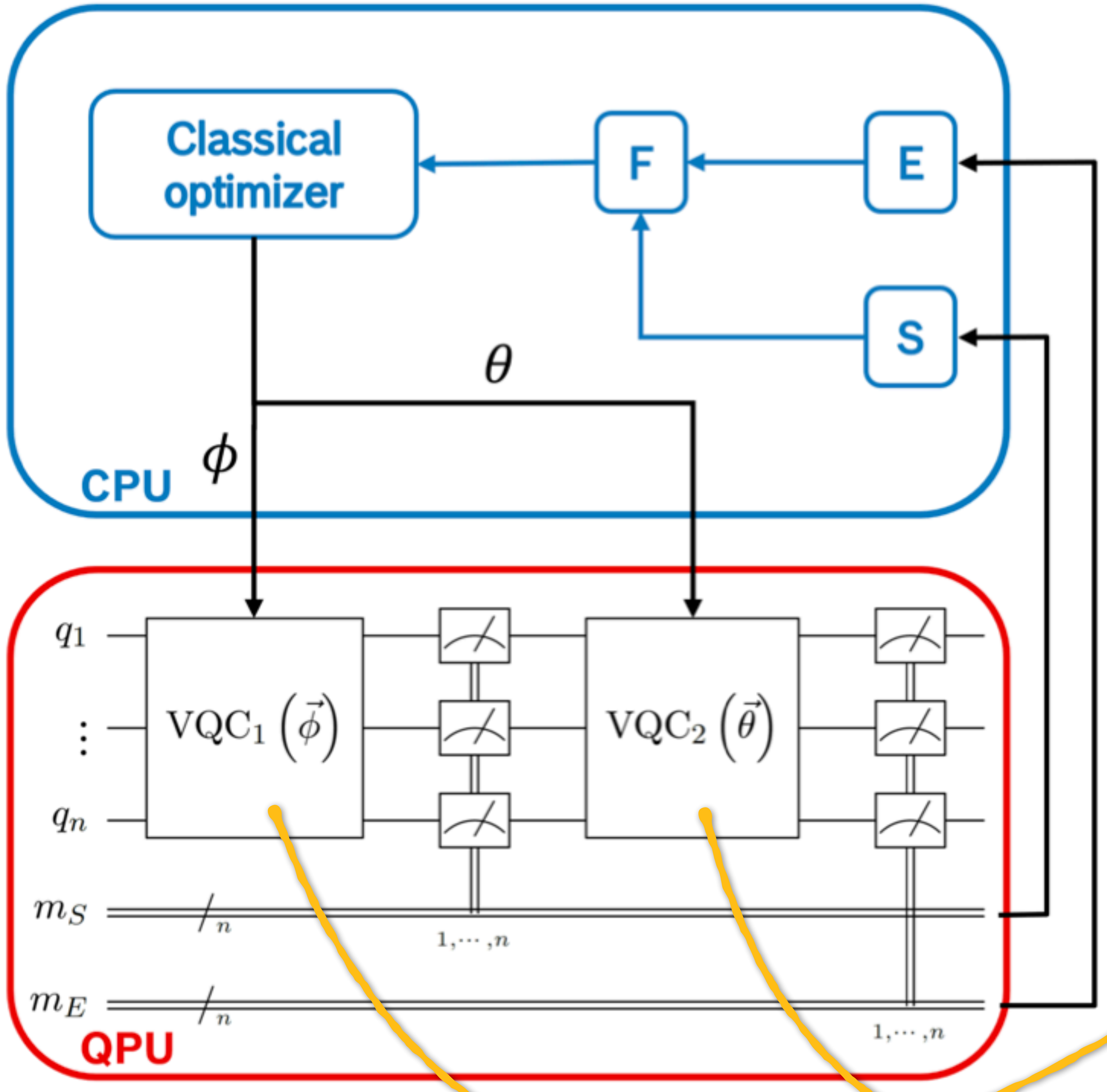


Durham University

IP3

Next step: Q-VQT (physicists are not good at naming things…)

Selisko, Amsler, Hammerschmidt, Drautz, Eckl; arXiv: 2208.07621



Determines Shanon Entropy of each possible state and generates the mixed state for the second circuit. Outputs represent the Gibbs state.

Approximates the thermal state of the Hamiltonian

Conclusion

Durham University

IP3

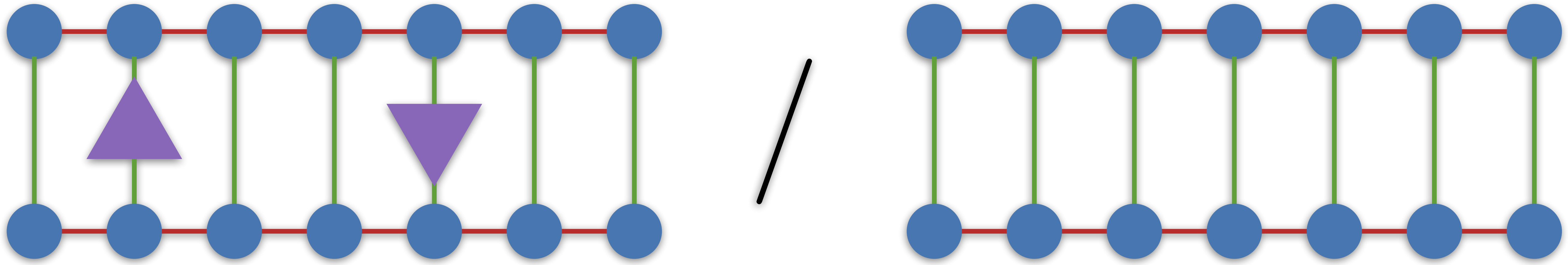
Conclusion

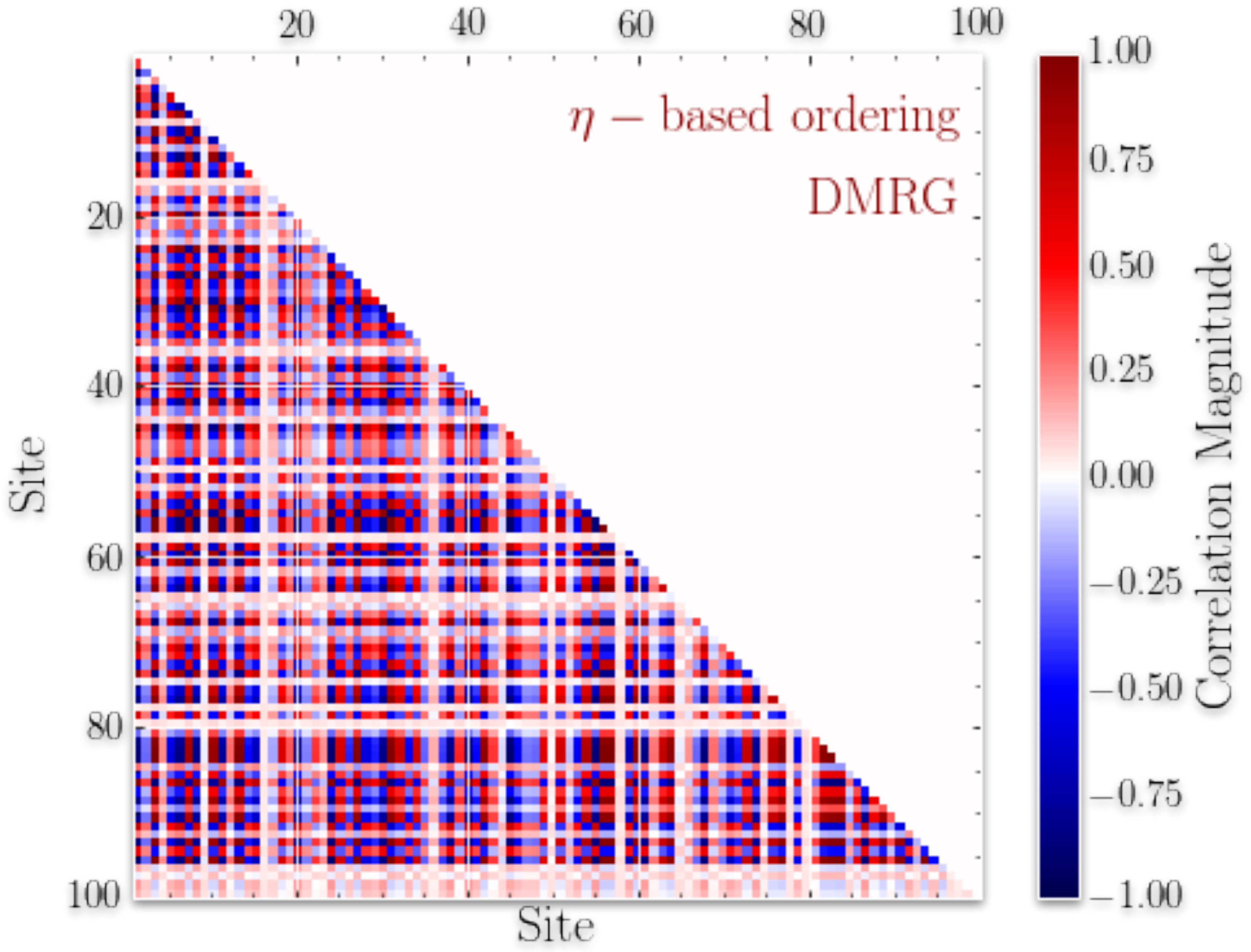
- The name of the game is optimisation. Techniques developed for field theory computations are easily transferable for ML applications!
- Designing quantifiable measures from the ansätze can allow us to improve training procedures and can be used for feature selection. (Thought: maybe helpful to lock on a symmetry during training?)
- Tensor Networks are the tool for the near future to understand quantum computing until the machinery is ready for more significant problems.

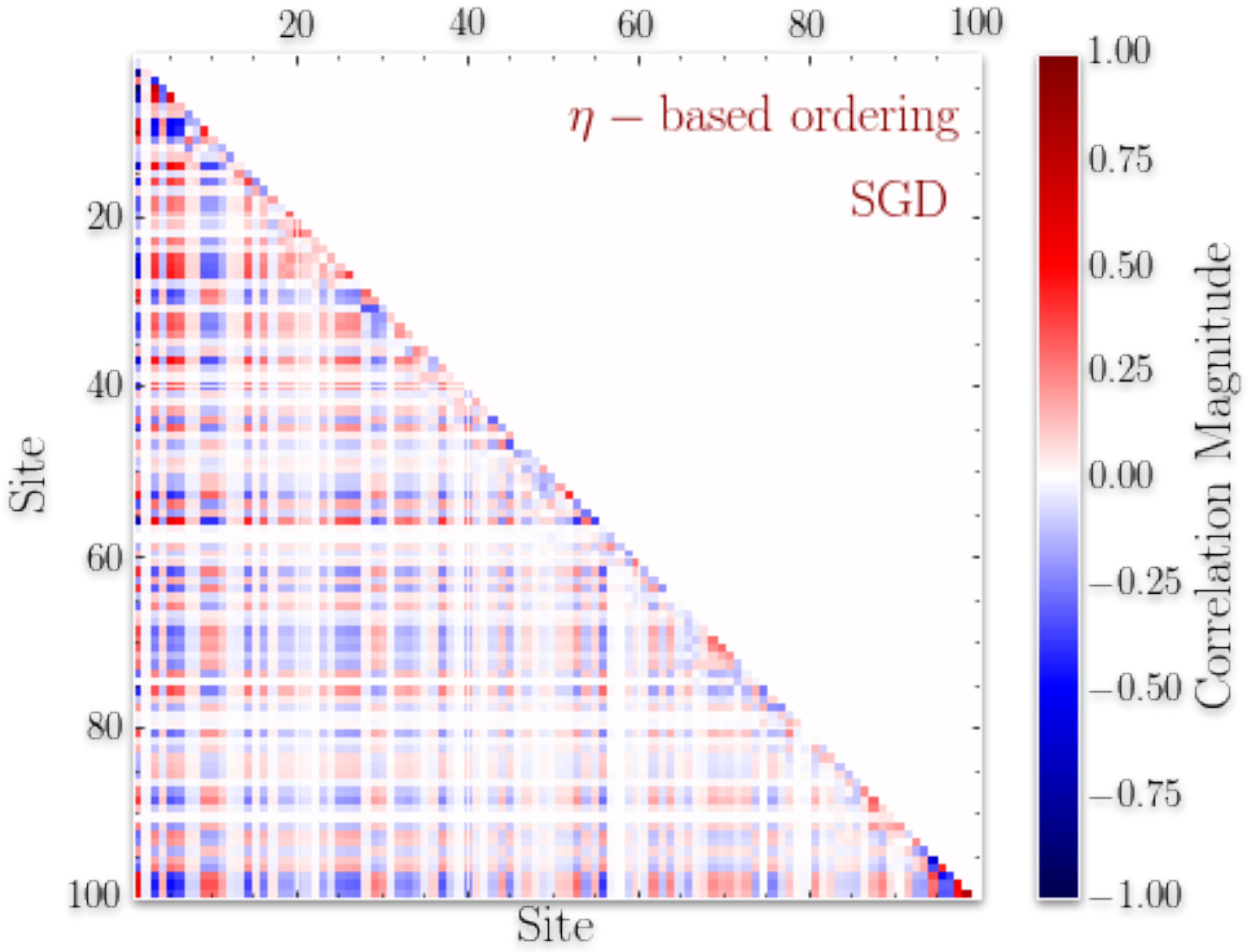
BACKUP

Correlations by SU(2) generators

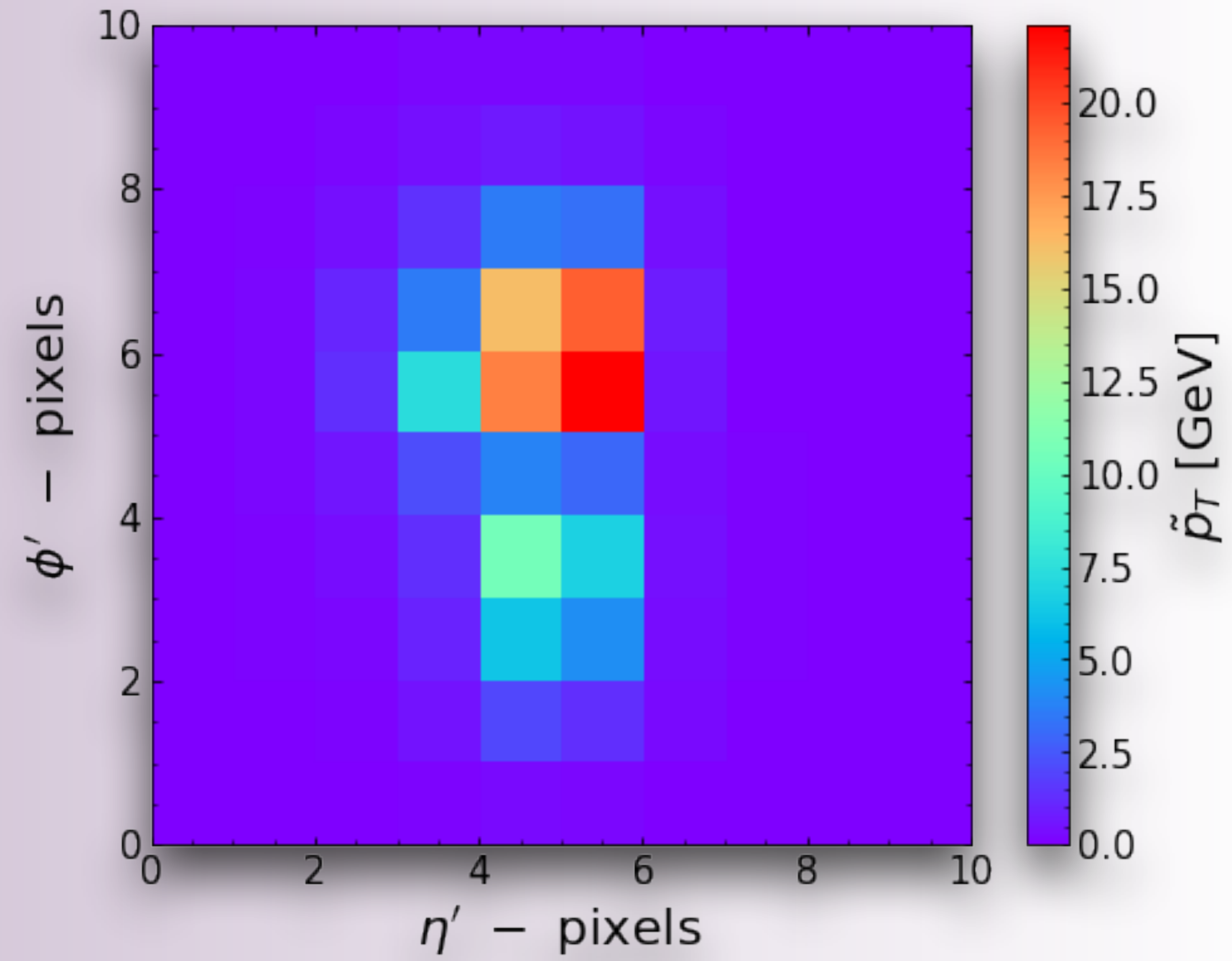
$$\mathcal{C}^l_{ij} = \frac{\langle \mathcal{W}^l | \mathcal{O}_i \mathcal{O}_j^\dagger | \mathcal{W}^l \rangle}{\langle \mathcal{W}^l | \mathcal{W}^l \rangle} =$$

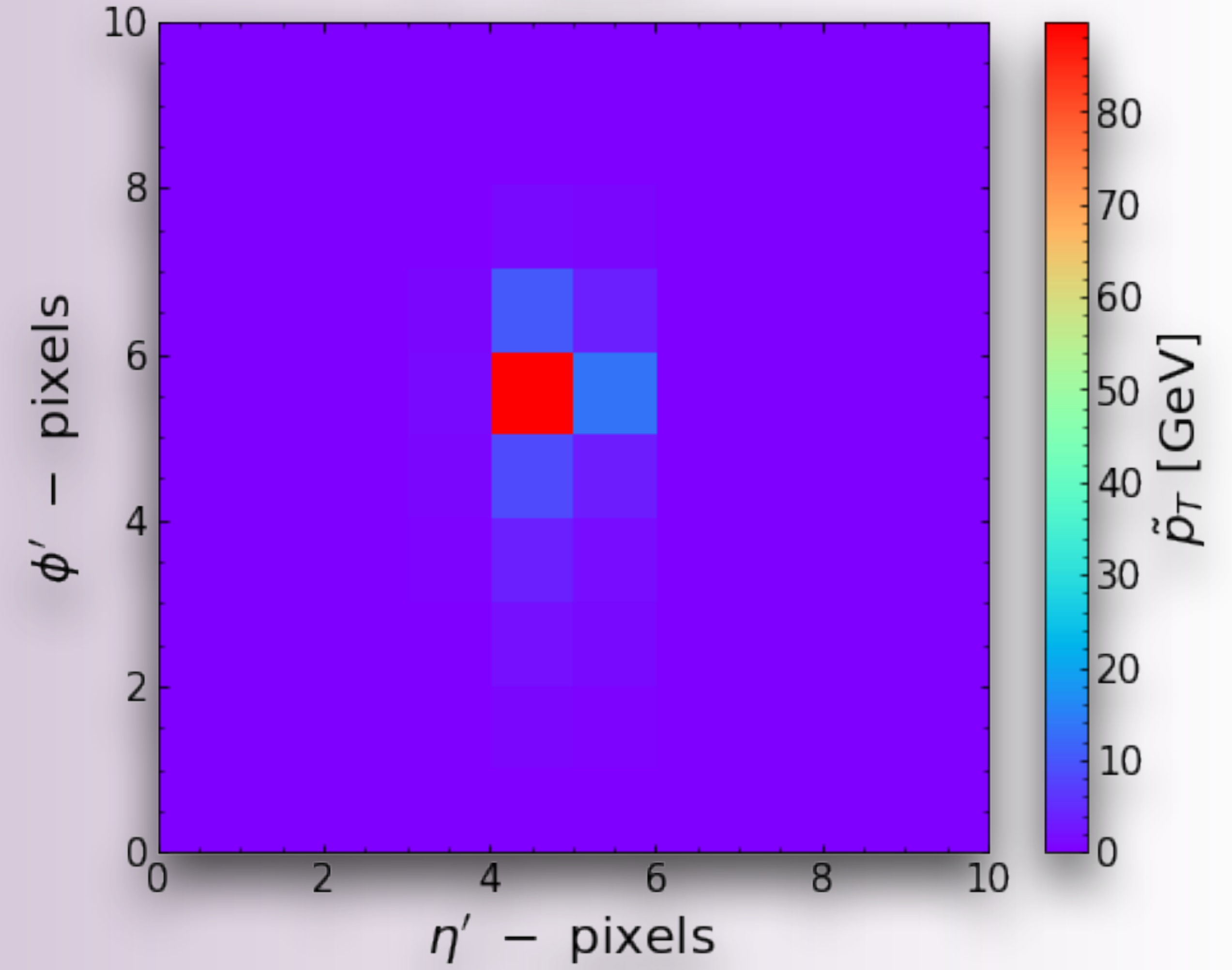






Top Tagging through MPS



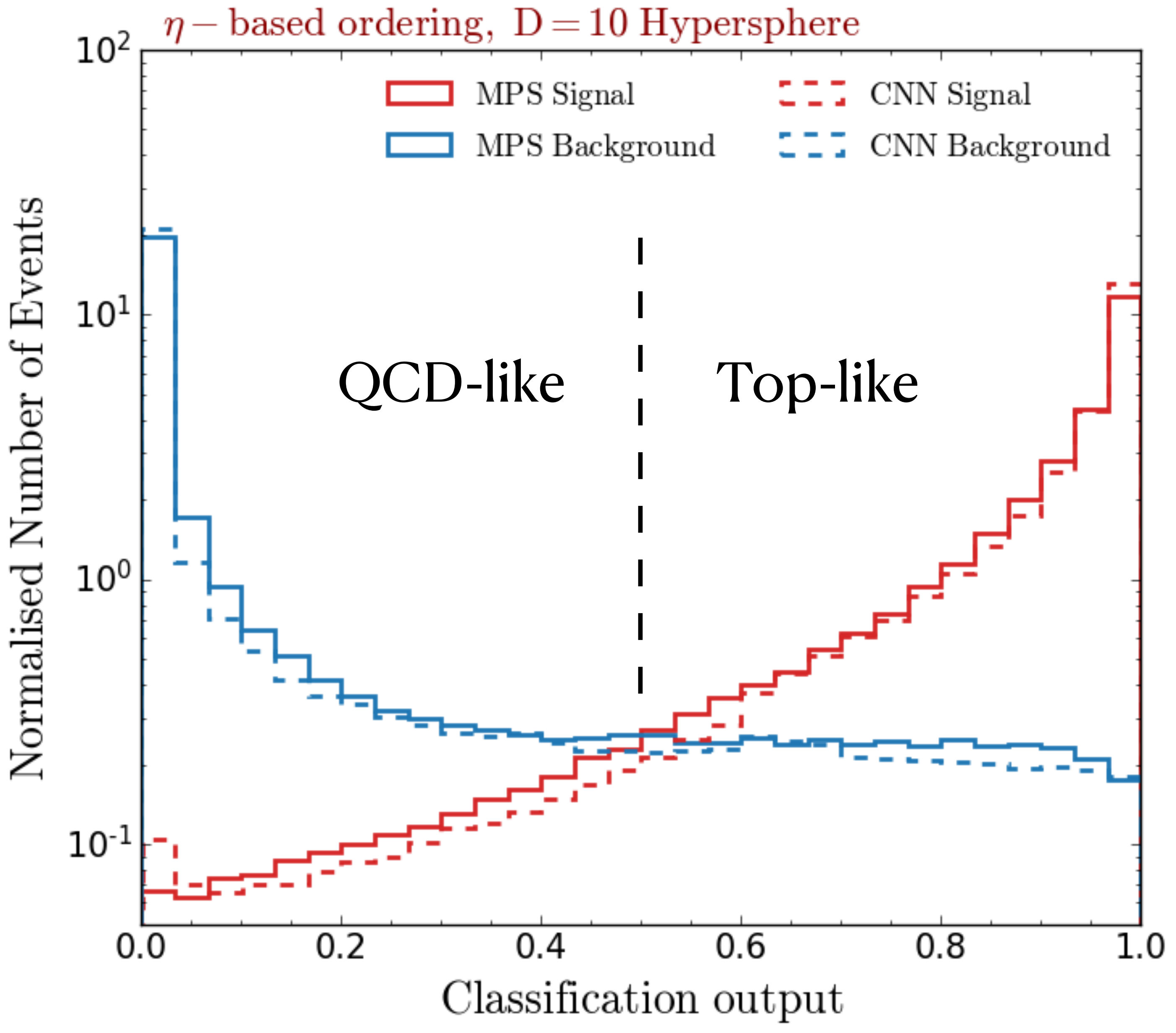


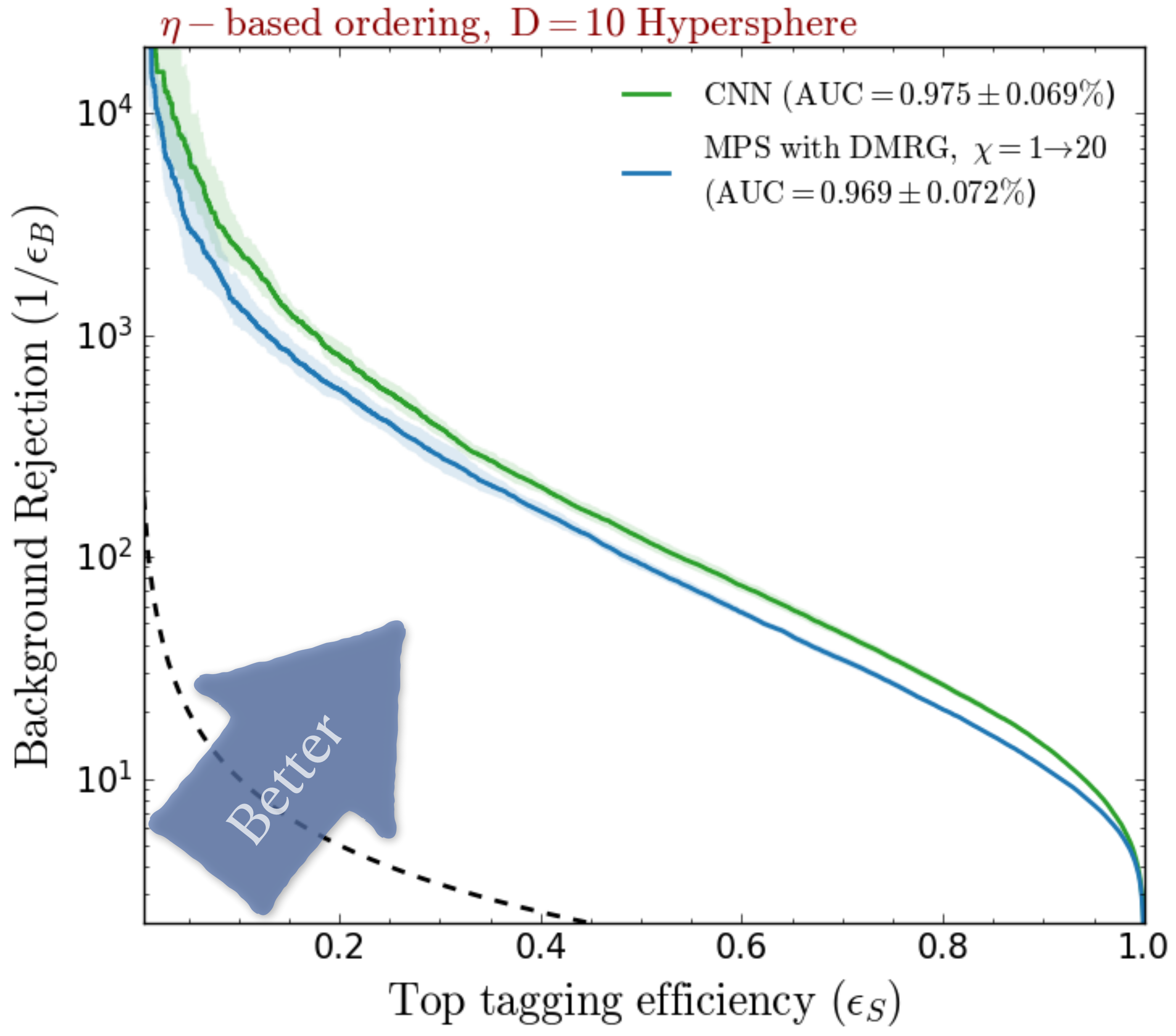
Assumptions & Requirements

- ❖ No prior entanglement/correlation between pixels
- ❖ Network is a Born Machine → square of the wave-function gives the probability of the classification.
- ❖ Maximum bond dimension that network can get is 20.

CNN architecture from:

JYA, Spannowsky; JHEP '21

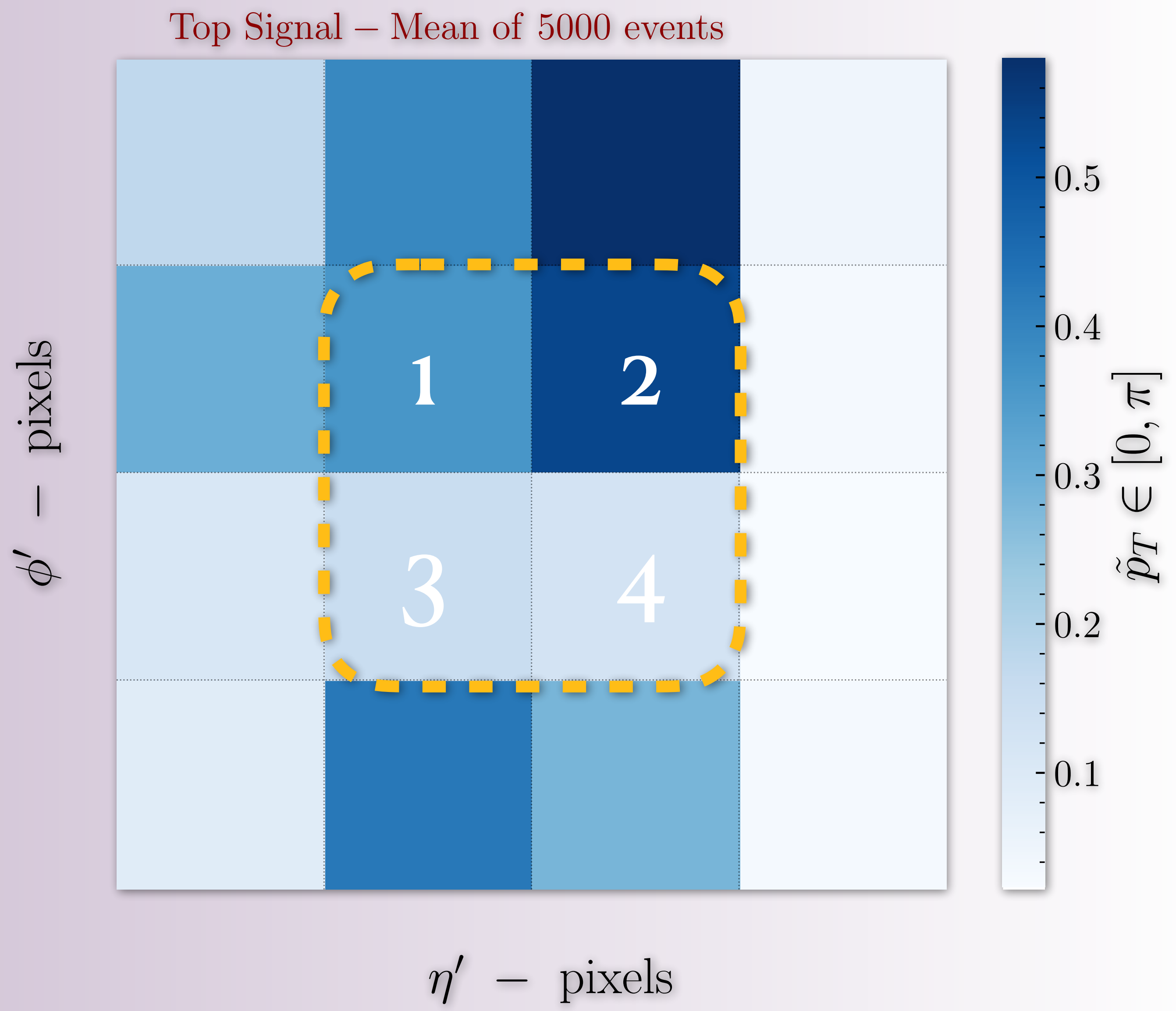


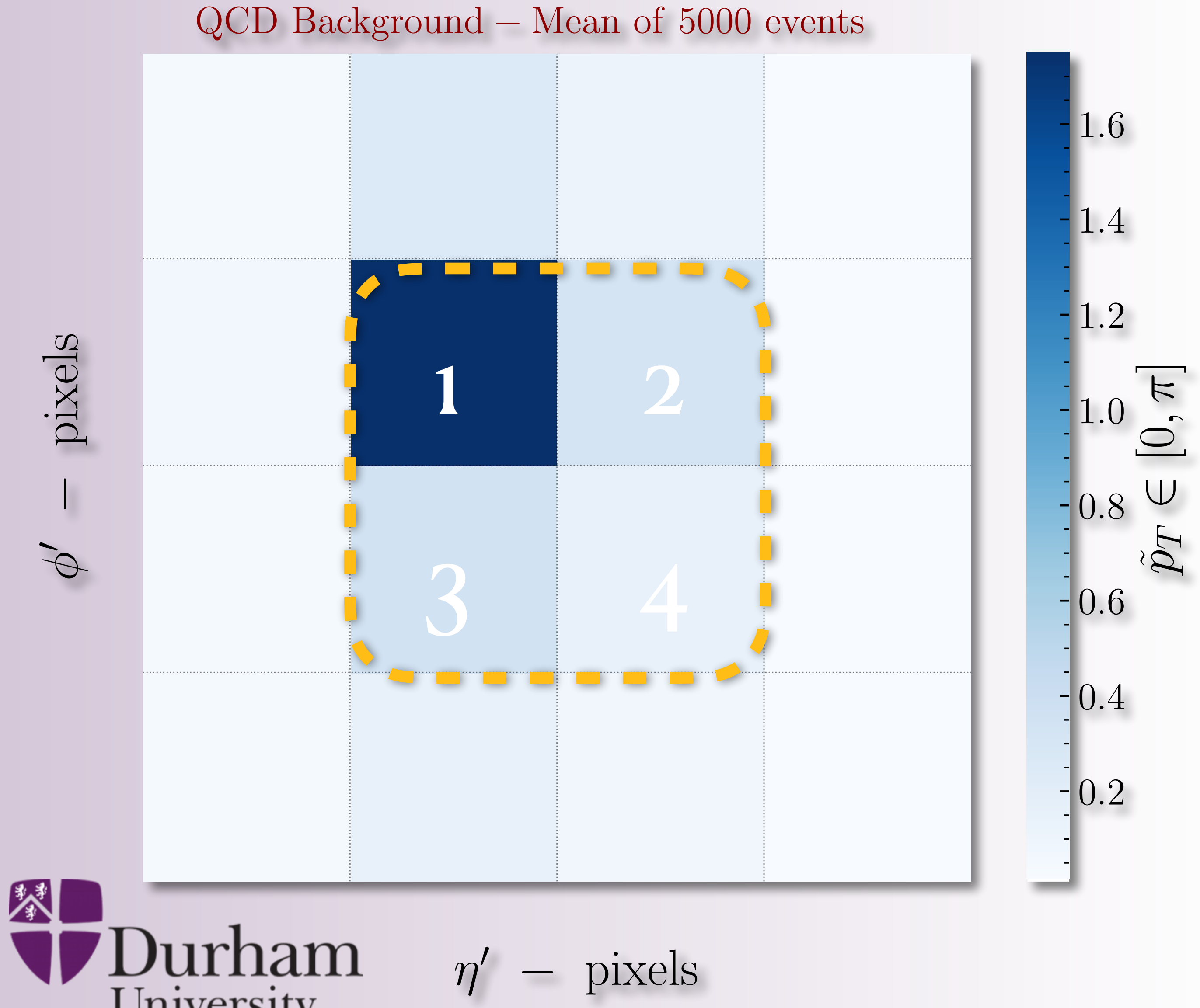


Durham University

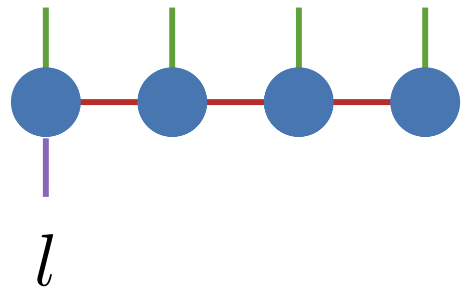
IP3

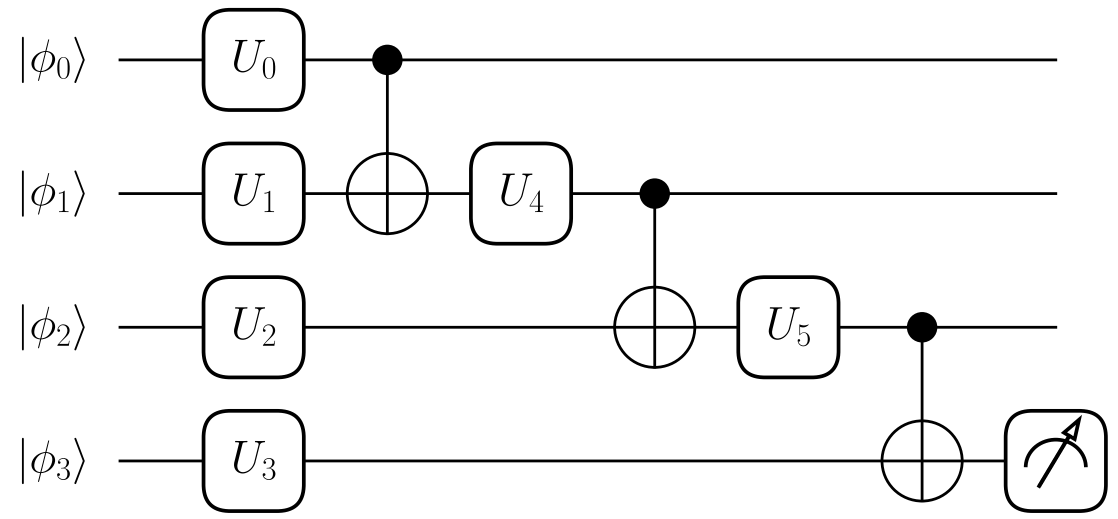
Experimenting with 4-Qubits



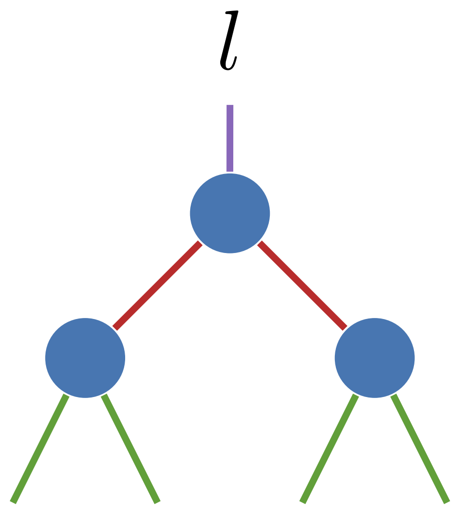


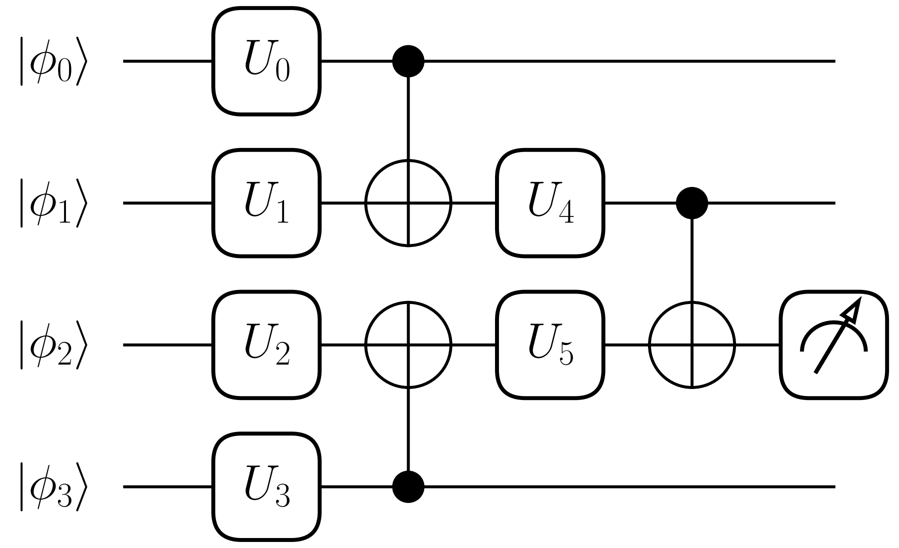
Matrix Product States



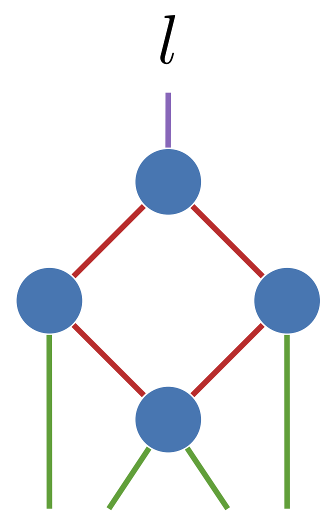


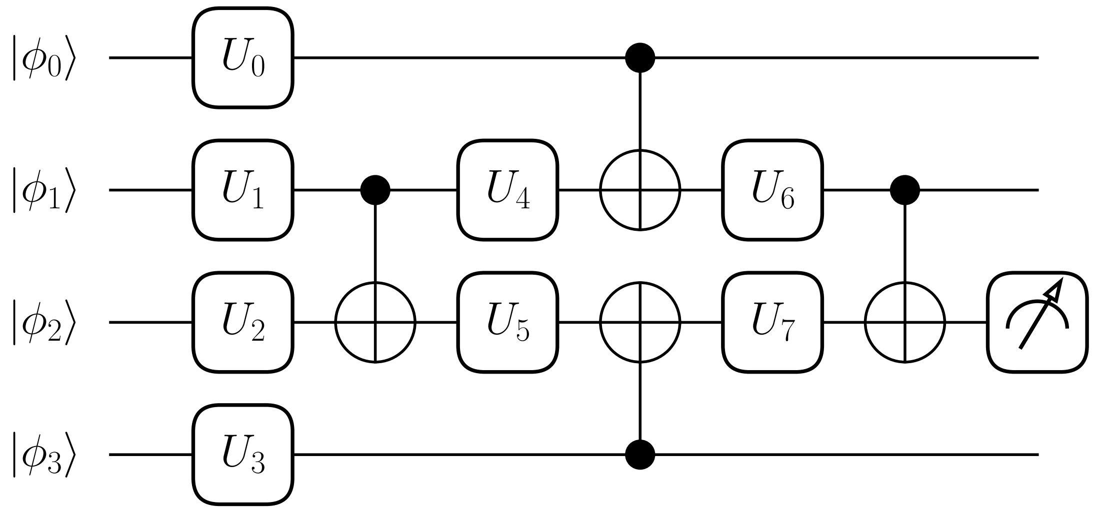
Tree Tensor Networks



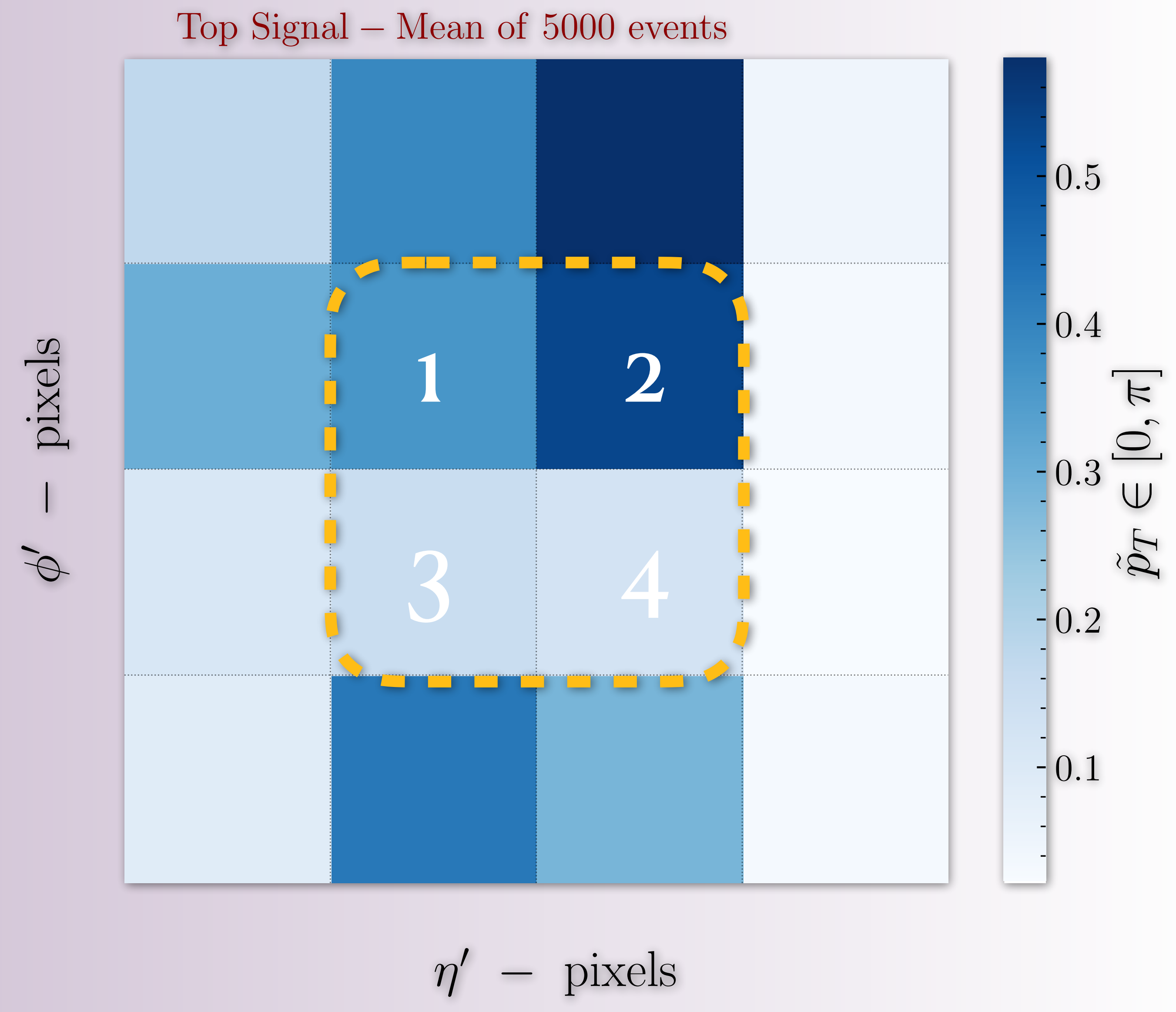


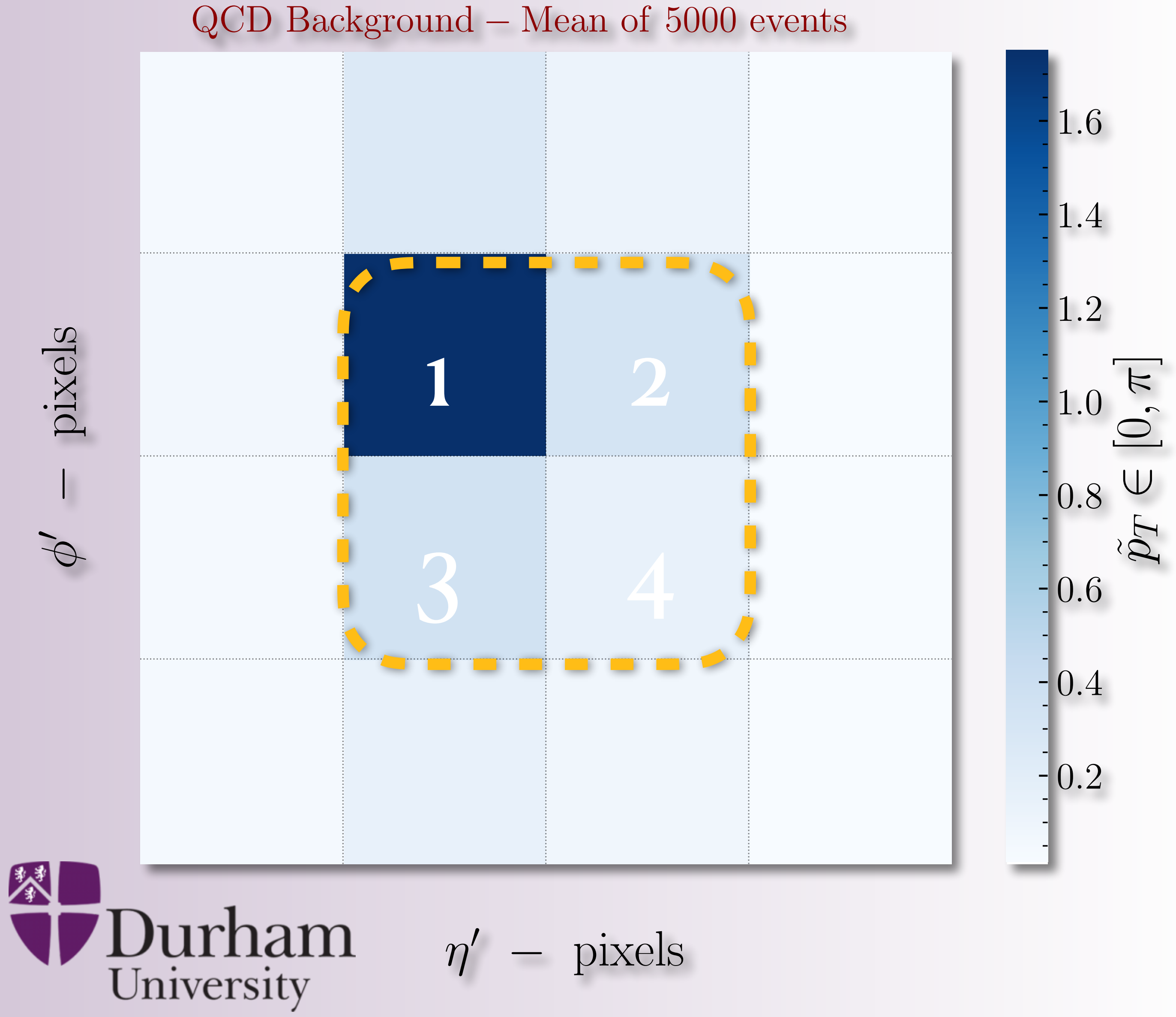
Multiscale Entanglement Renormalisation Ansatz

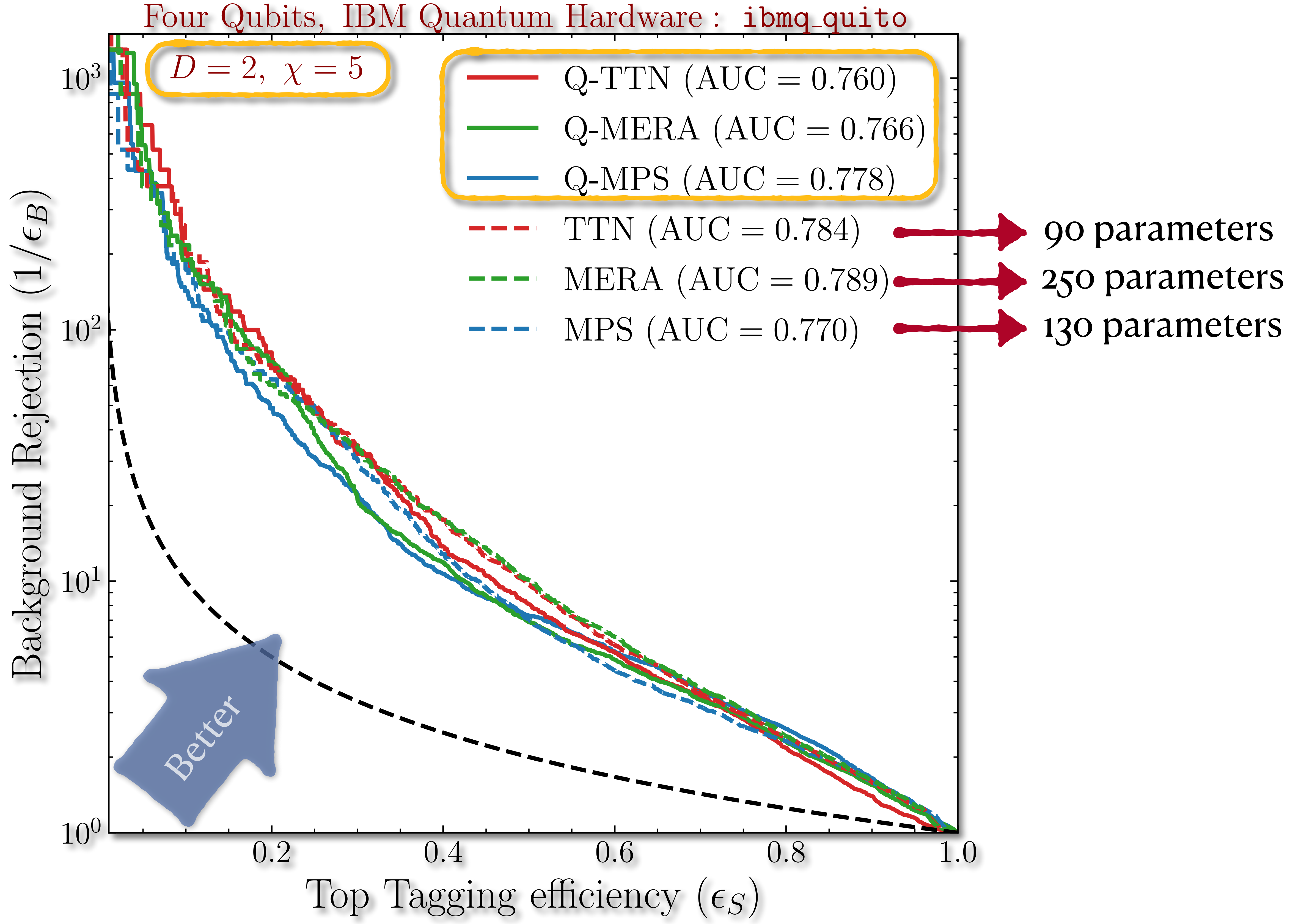




Experimenting with 4-Qubits

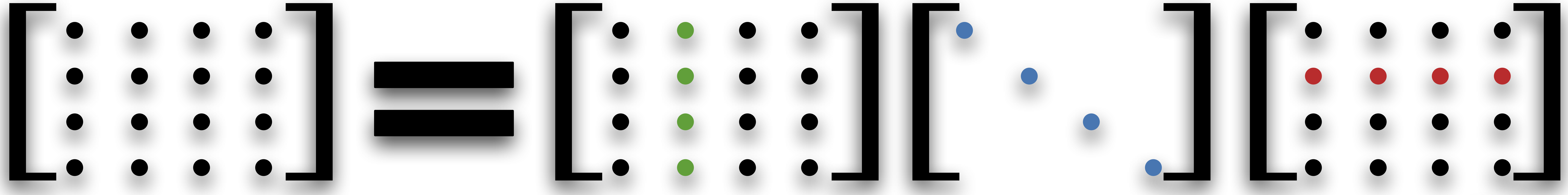






Singular Value Decomposition

$$m \times n \qquad\qquad m \times k \qquad\qquad k \times l \qquad\qquad l \times n$$



$$\mathbf{M} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^{\dagger}$$

Orthogonal singular column vectors

Positive definite singular values in descending order

Orthogonal singular row vectors

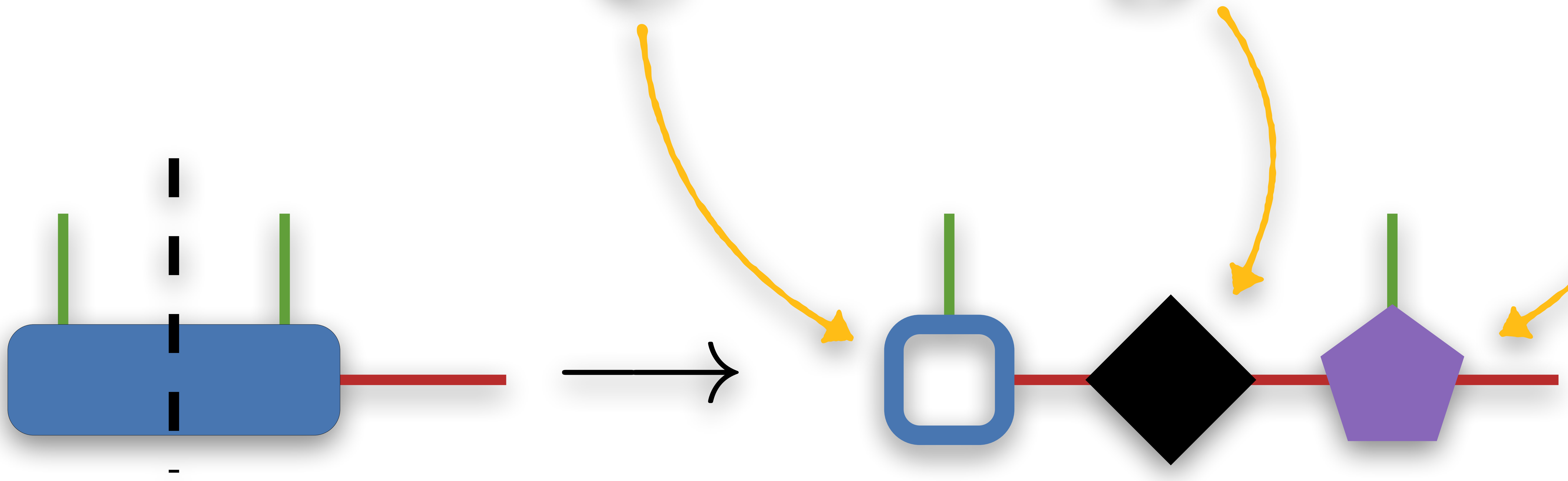
By changing the number of singular values one can change the accuracy of the decomposition!!!

$$\mathbf{S} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$$

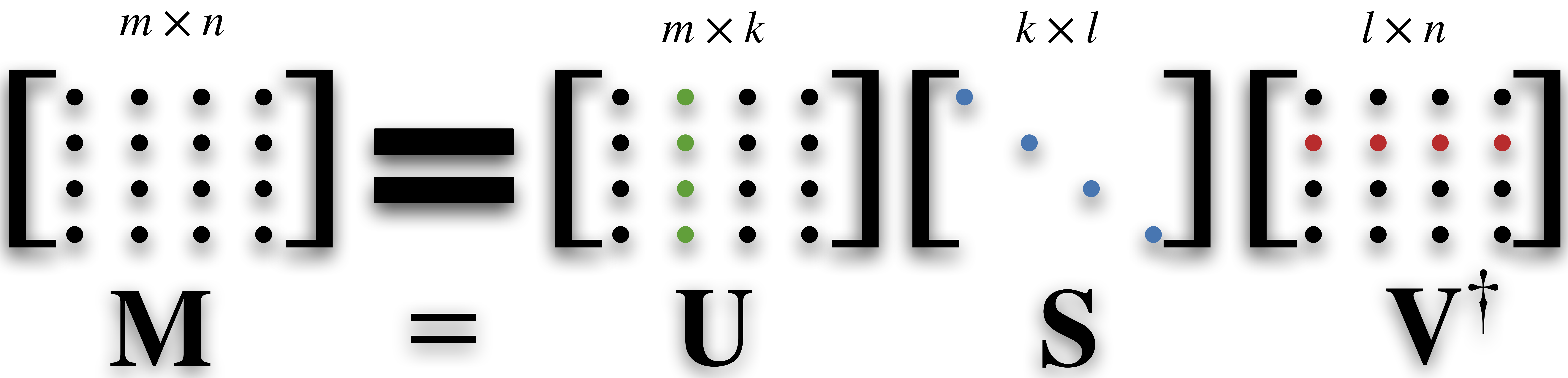
λ_i also known as Schmidt values

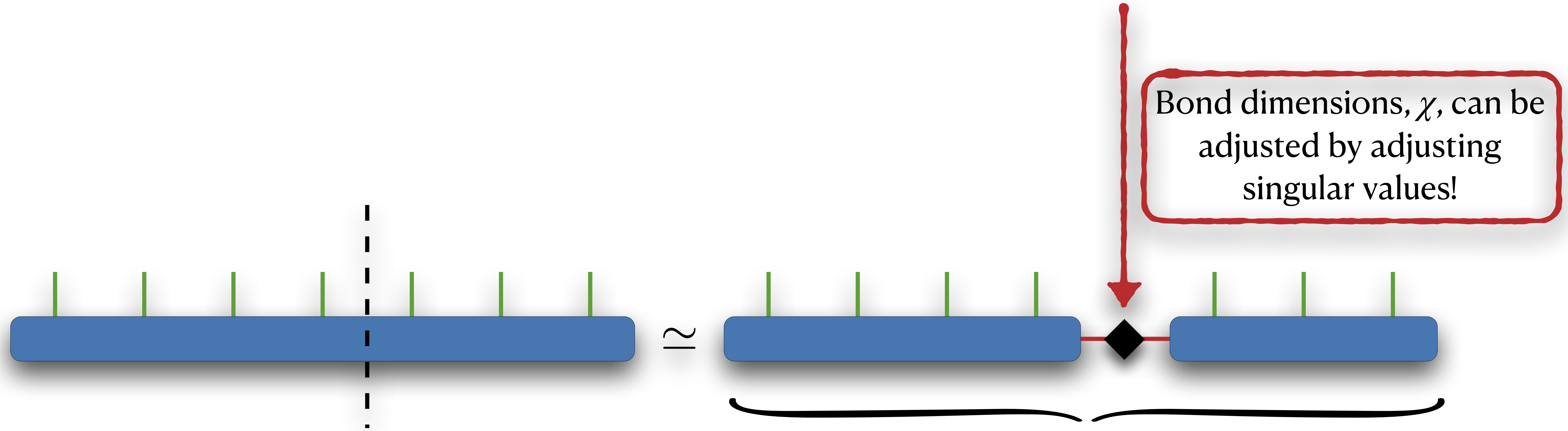
Singular Value Decomposition

$$\overset{m\times n}{\mathbf{M}} = \overset{m\times k}{\mathbf{U}}\;\overset{k\times l}{\mathbf{S}}\;\overset{l\times n}{\mathbf{V}^{\dagger}}$$



Singular Value Decomposition





Computational cost is $\mathcal{O}(d^{N-1}\chi^2)$!!!

Matrix Product States for Classification

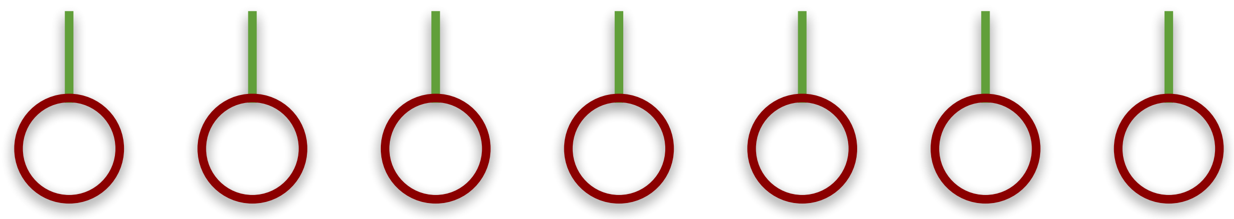
Sub-Outline

- How to embed the data?
- How to form a network?
- How to train the network?

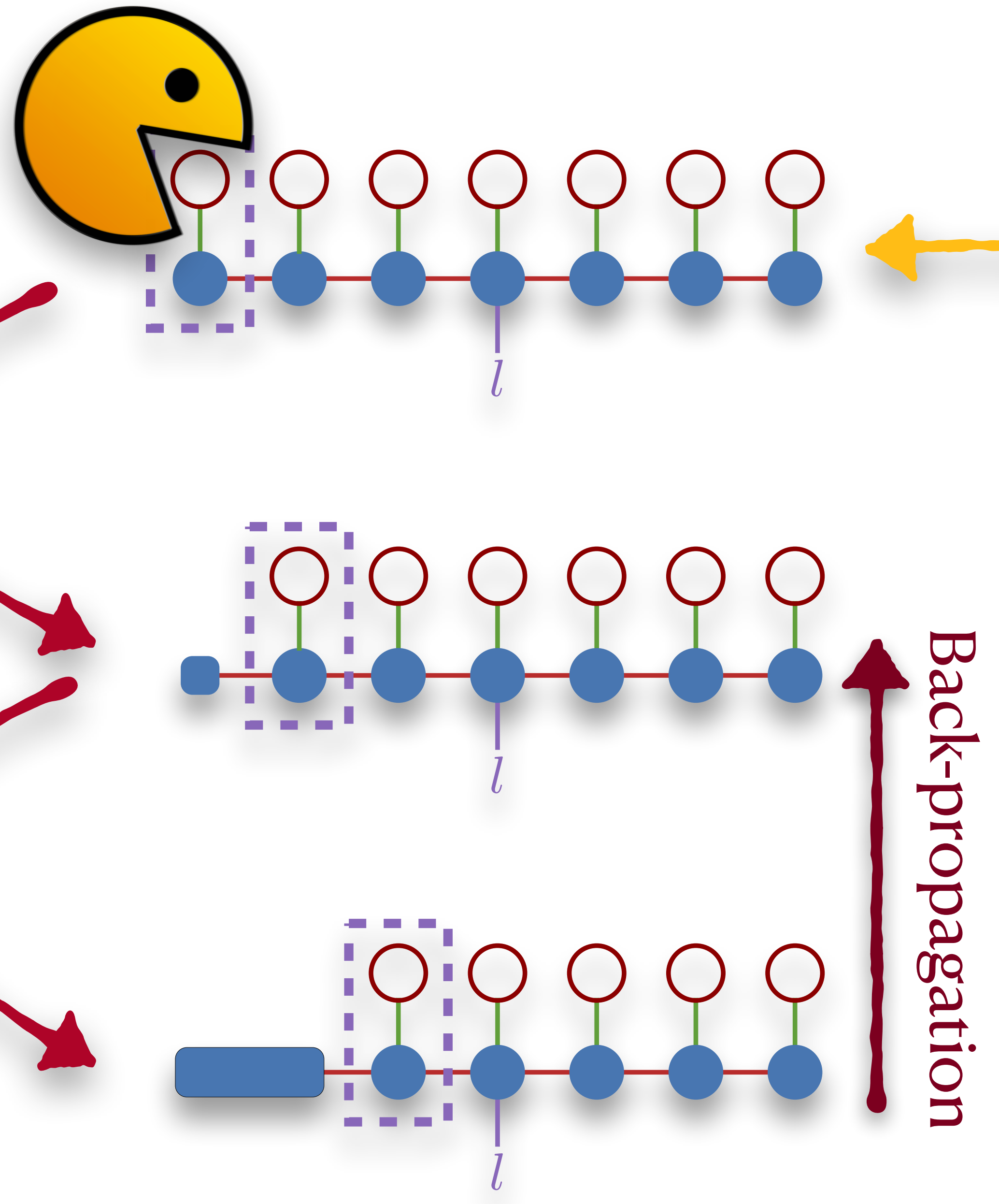
Data Embedding

$$\Phi^{p_1\cdots p_n}(\mathbf{x}) = \phi^{p_1}(x_1)\otimes\phi^{p_2}(x_2)\otimes\cdots\otimes\phi^{p_n}(x_n)$$

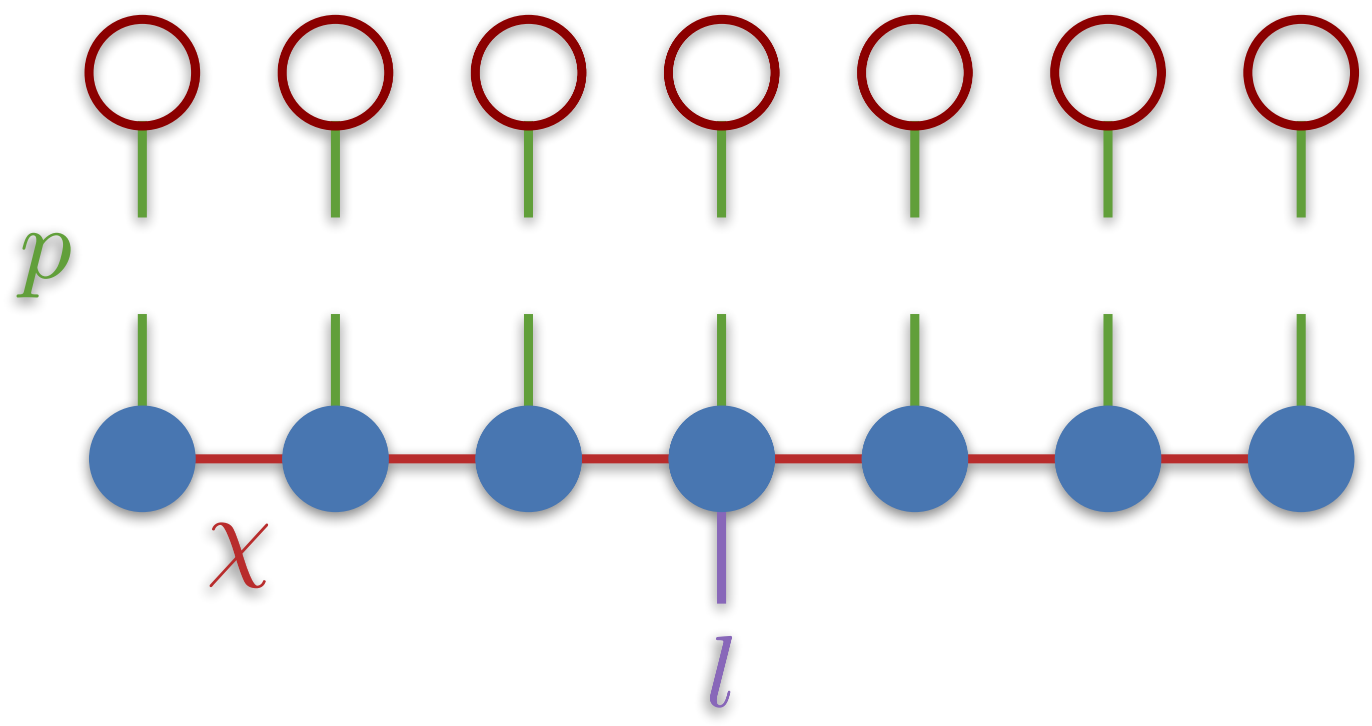
$$\phi^{p_i}(x_i) = \begin{bmatrix}\cos(x_i\ \pi/2)\\ \sin(x_i\ \pi/2)\end{bmatrix} \quad \text{or} \quad \phi^{p_i}(x_i) = \begin{bmatrix}1\\ x_i\\ x_i^2\end{bmatrix} \quad \text{or} \quad \cdots$$



$$= \Phi^{p_1\cdots p_n}(\mathbf{x})$$

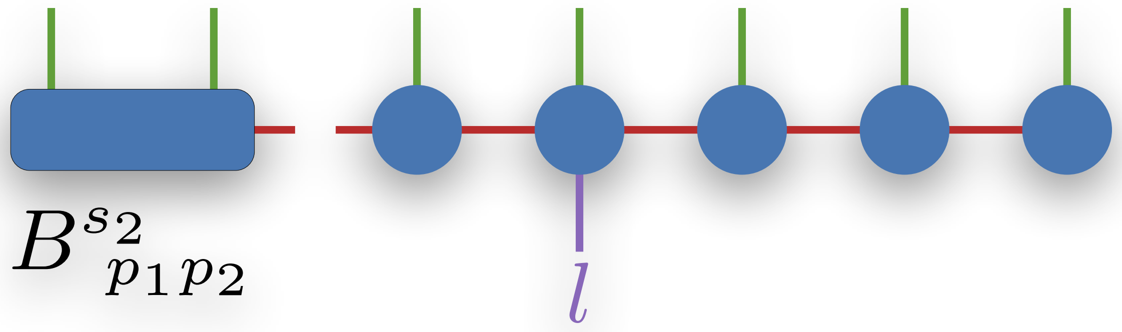


An efficient contraction algorithm is essential for training!

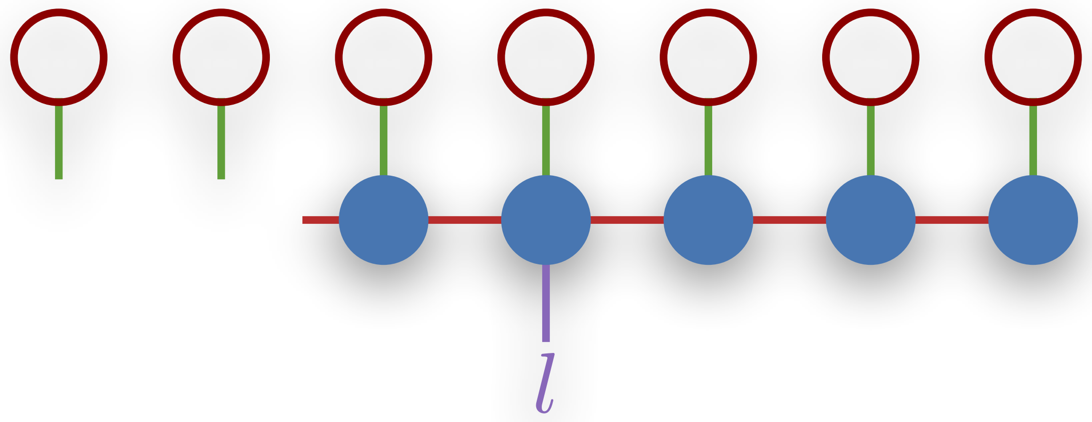


Density Matrix Renormalization Group Algorithm

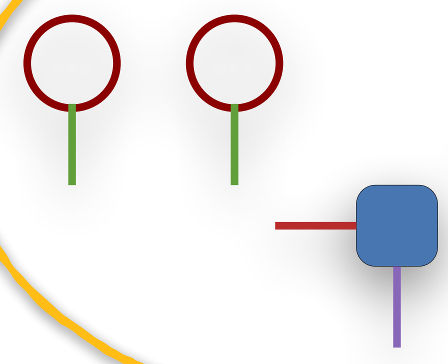
$I.$



$II.$



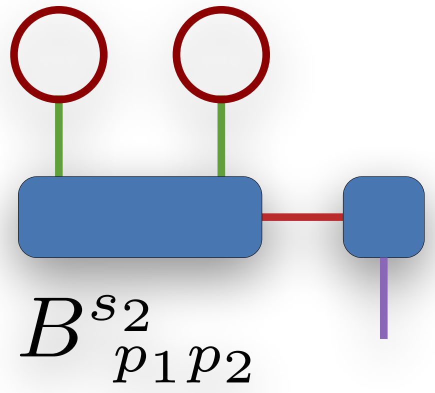
$=$



$= \Gamma^{l p_1 p_2}{}_{s_2}$

$$\frac{\partial f^l(\mathbf{x}^{(i)})}{\partial B^{s_2}{}_{p_1 p_2}} = \Gamma^{l p_1 p_2}{}_{s_2}$$

$III.$



$$= \boxed{B^{s_2}{}_{p_1 p_2} \Gamma^{l p_1 p_2}{}_{s_2} = f^l\left(\mathbf{x}^{(\mathbf{n})}\right)} \to p(l, \mathbf{x}) = \left| f^l\left(\mathbf{x}^{(\mathbf{n})}\right) \right|^2$$

$$\underline{\mathcal{L} = L\left(p(l, \mathbf{x}),\ l^{truth}\right)}$$

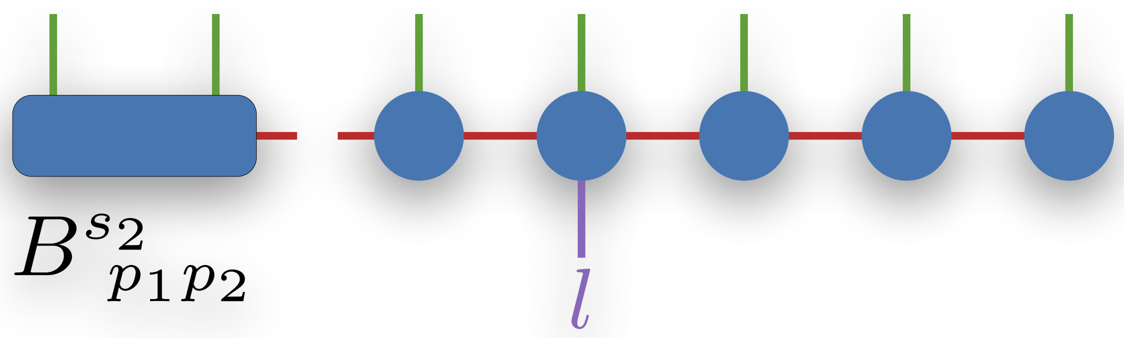
$$L = \frac{1}{N} \sum_{x \in \mathbf{x}^N} l^{truth}\ \log\left(p(l, x)\right)$$

$IV.$

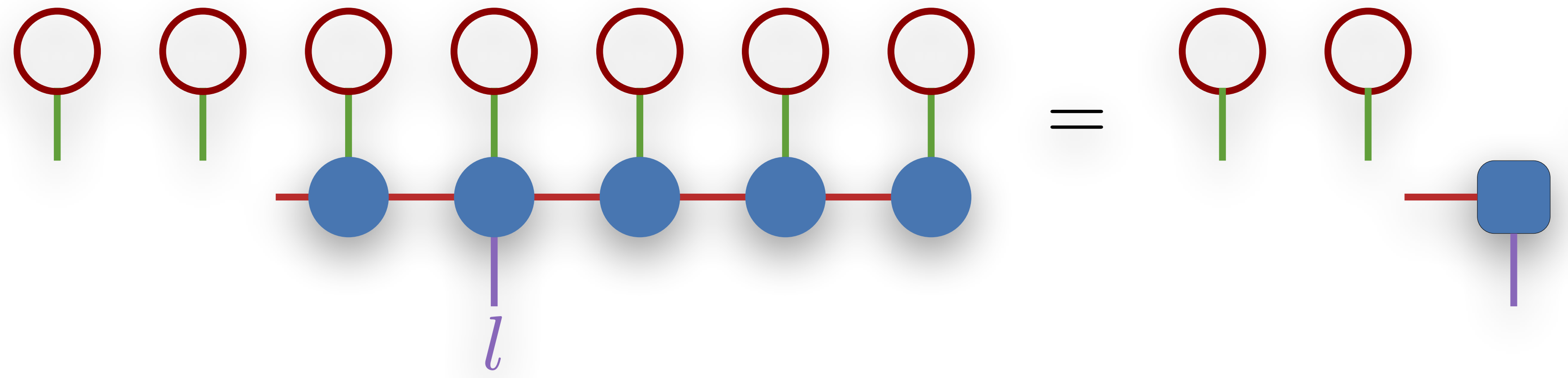
$$\tilde{B}^{s_2}{}_{p_1 p_2} = B^{s_2}{}_{p_1 p_2} - \eta\ \frac{\partial \mathcal{L}}{\partial B^{s_2}{}_{p_1 p_2}}$$

Density Matrix Renormalization Group Algorithm

$I.$

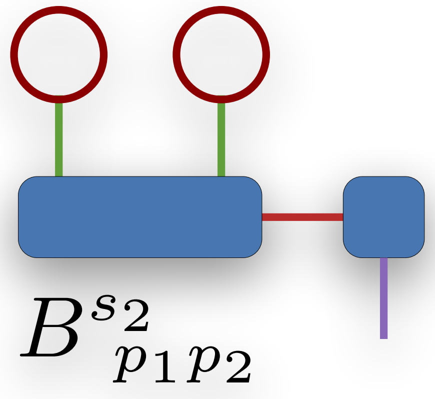


$II.$



$= \Gamma^{l p_1 p_2}{}_{s_2}$

$III.$



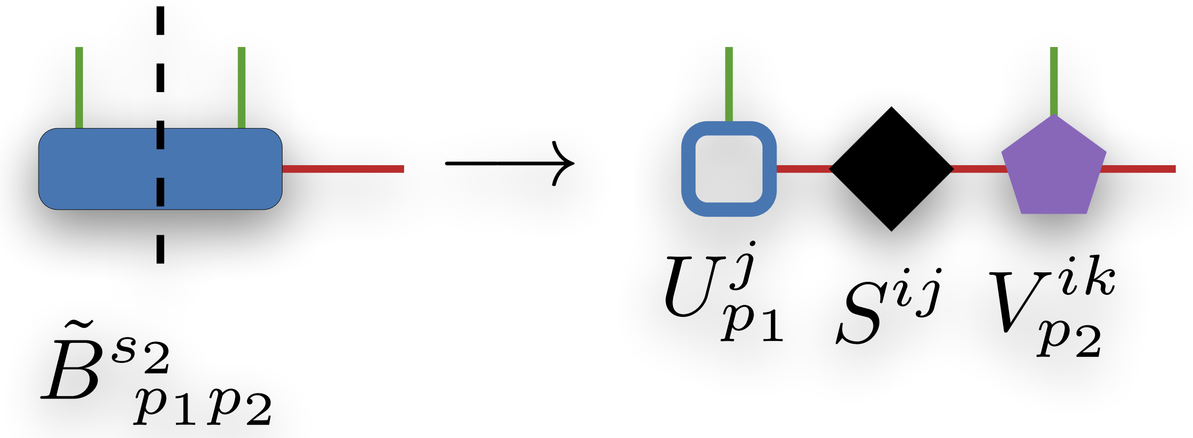
$$= B^{s_2}{}_{p_1 p_2} \Gamma^{l p_1 p_2}{}_{s_2} = f^l(\mathbf{x}^{(\mathbf{n})}) \to p(l, \mathbf{x}) = \left| f^l(\mathbf{x}^{(\mathbf{n})}) \right|^2$$

$$\mathcal{L} = L\left(p(l, \mathbf{x}),\ l^{truth}\right)$$

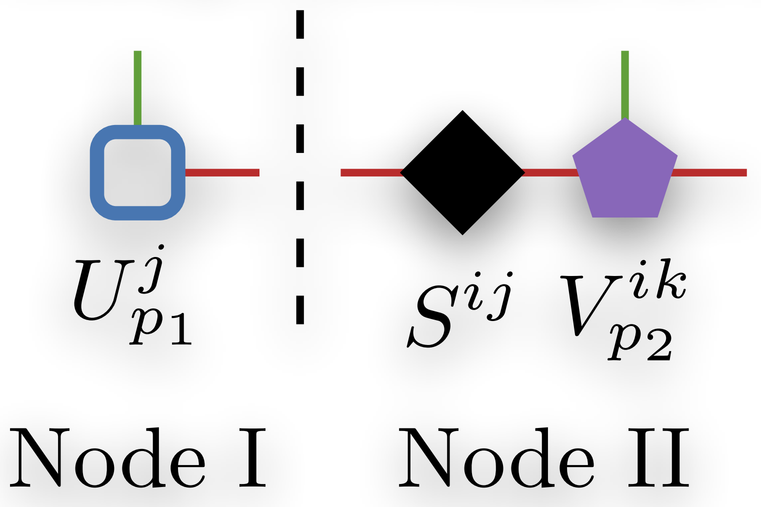
$IV.$

$$\tilde{B}^{s_2}{}_{p_1 p_2} = B^{s_2}{}_{p_1 p_2} - \eta \frac{\partial \mathcal{L}}{\partial B^{s_2}{}_{p_1 p_2}}$$

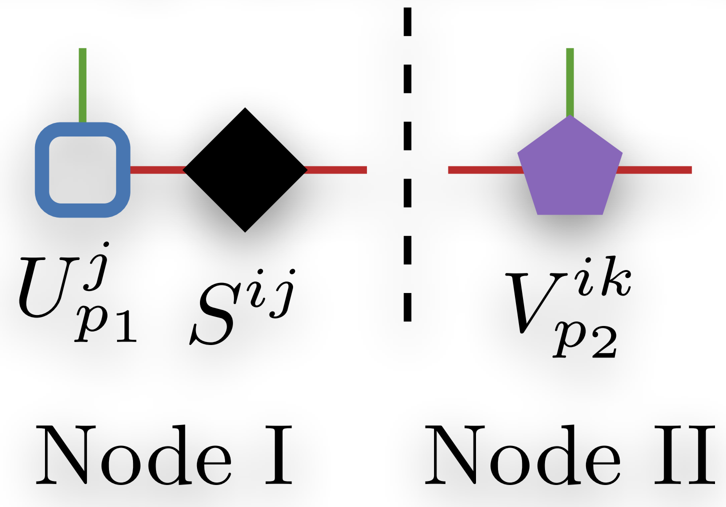
$V.$



$VI.$ Sweep direction to the right:

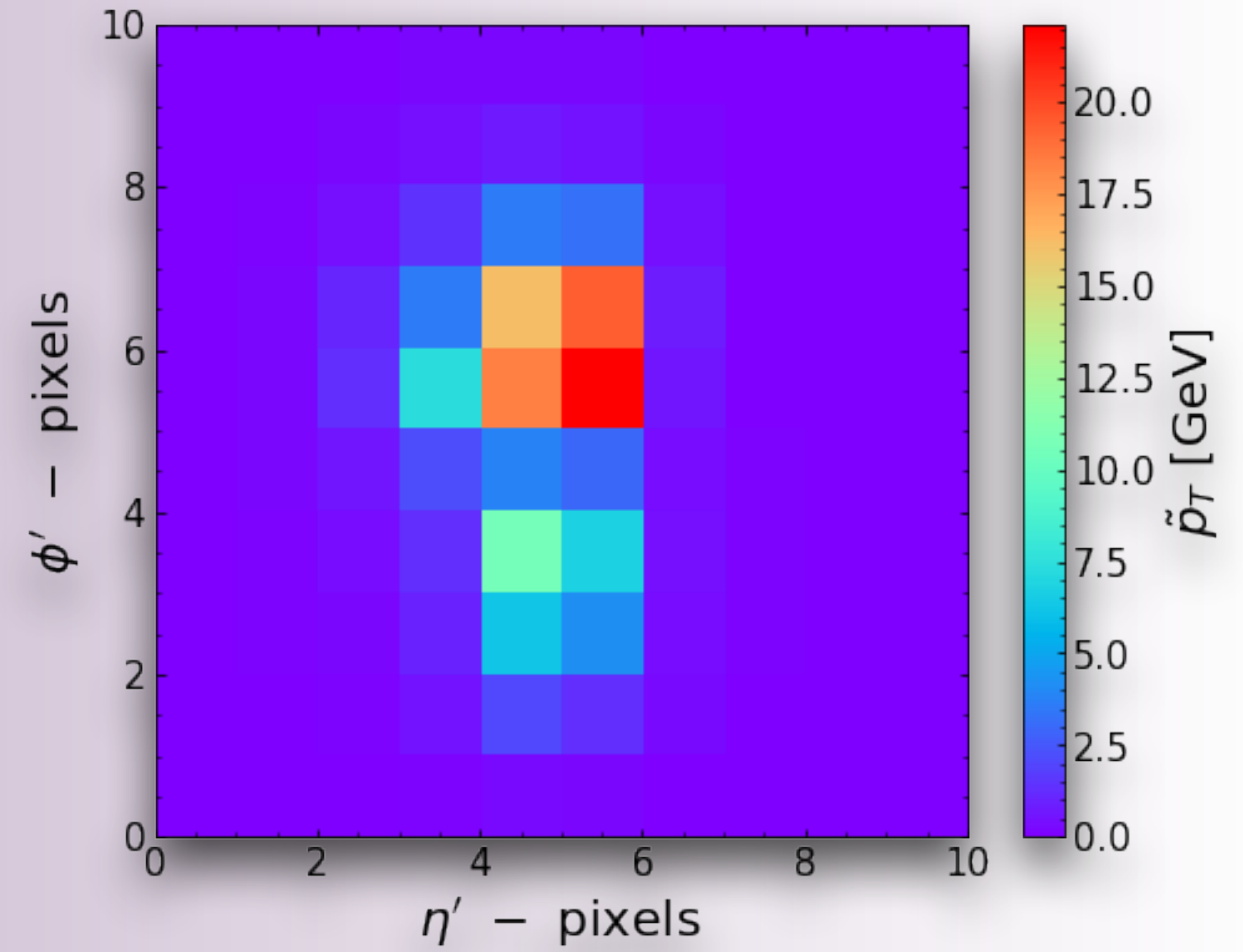


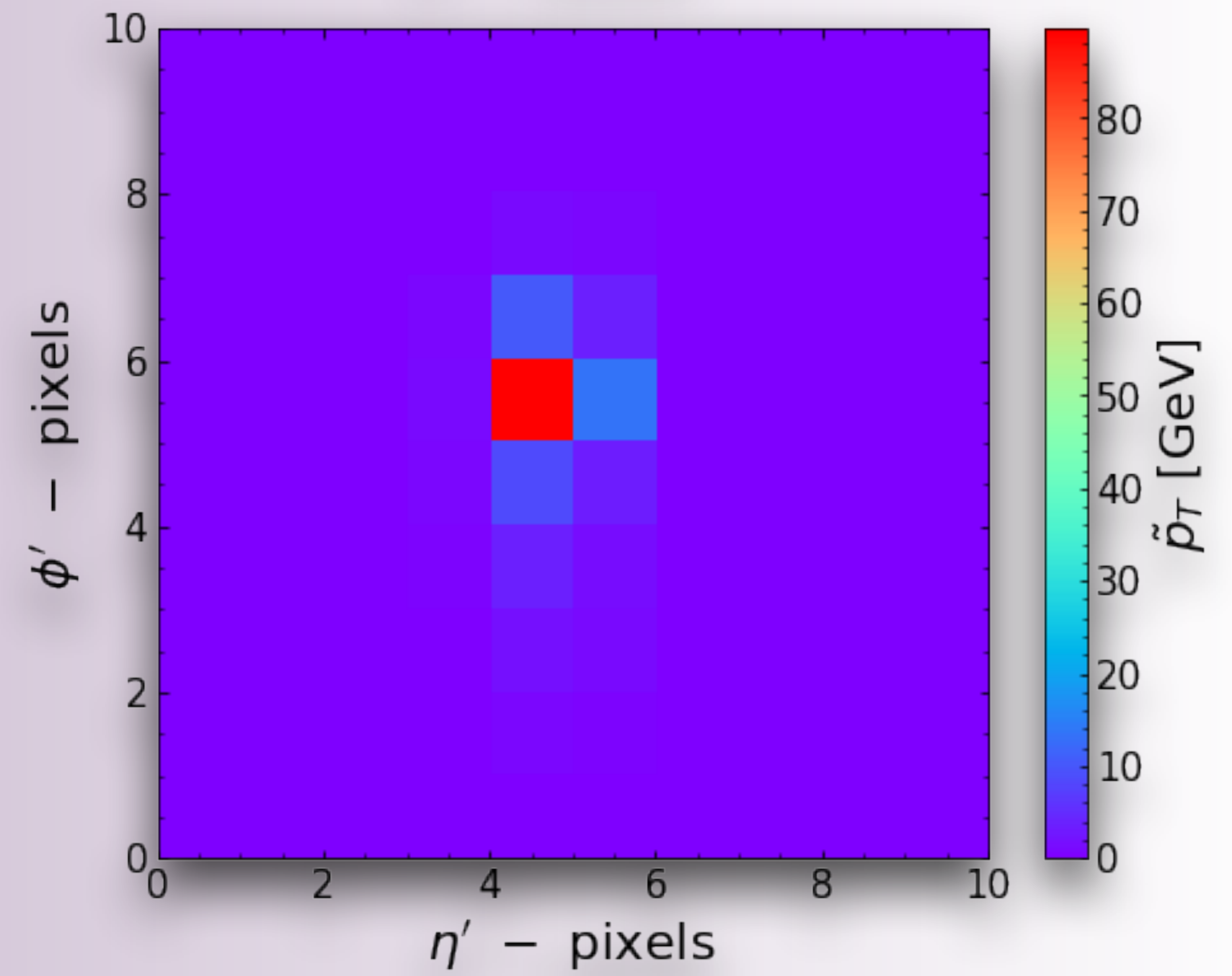
Sweep direction to the left:



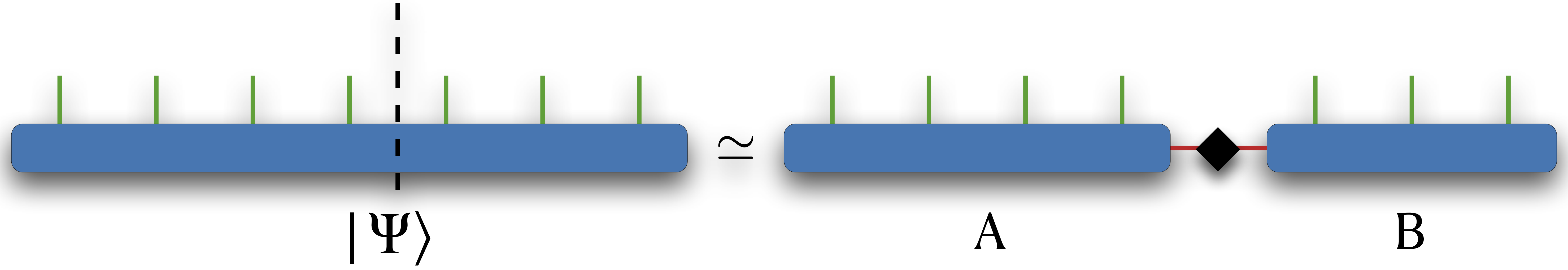
Adjust the bond dimension via SVD!

Top Tagging through MPS





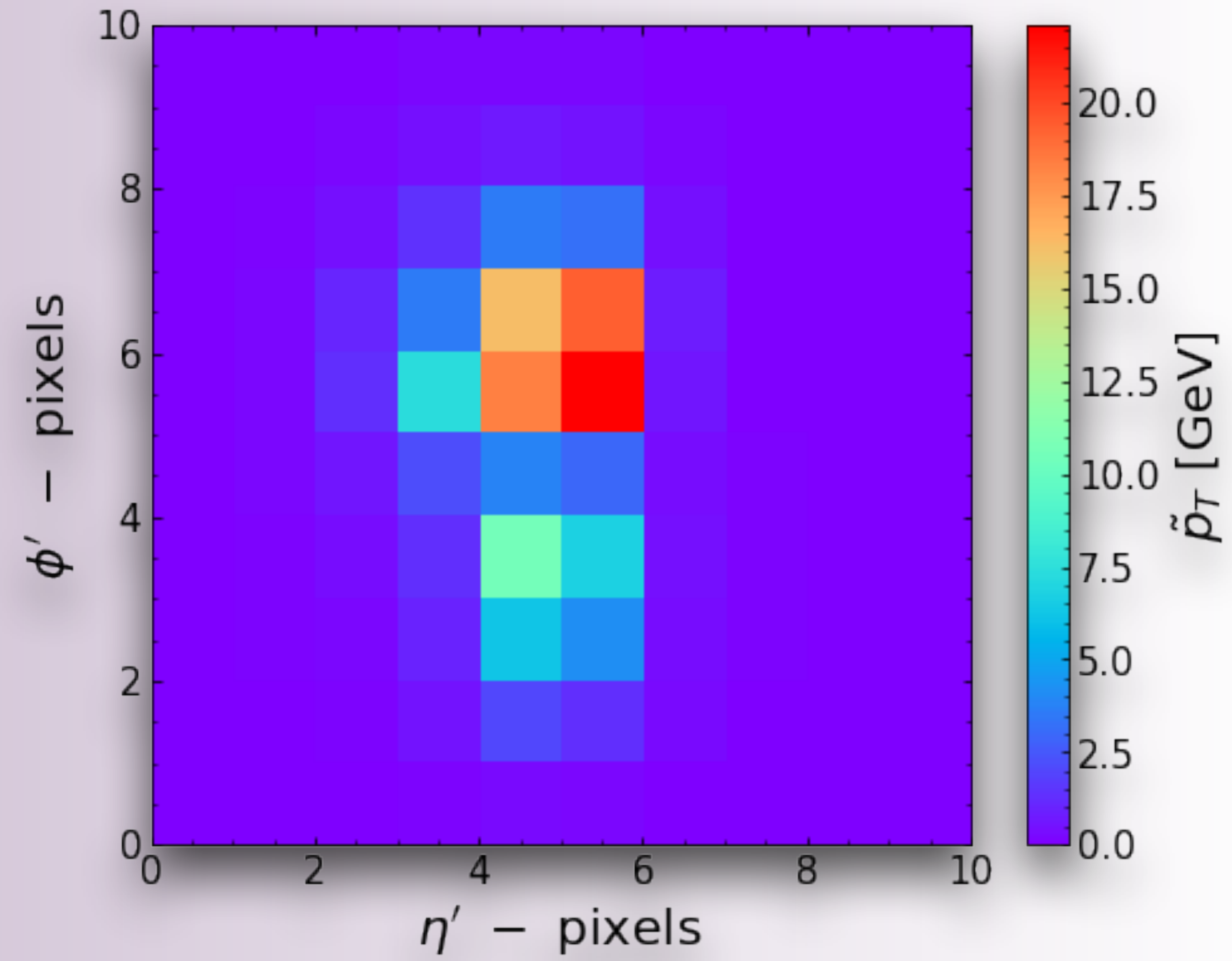
Entanglement Entropy & Schmidt Decompostion

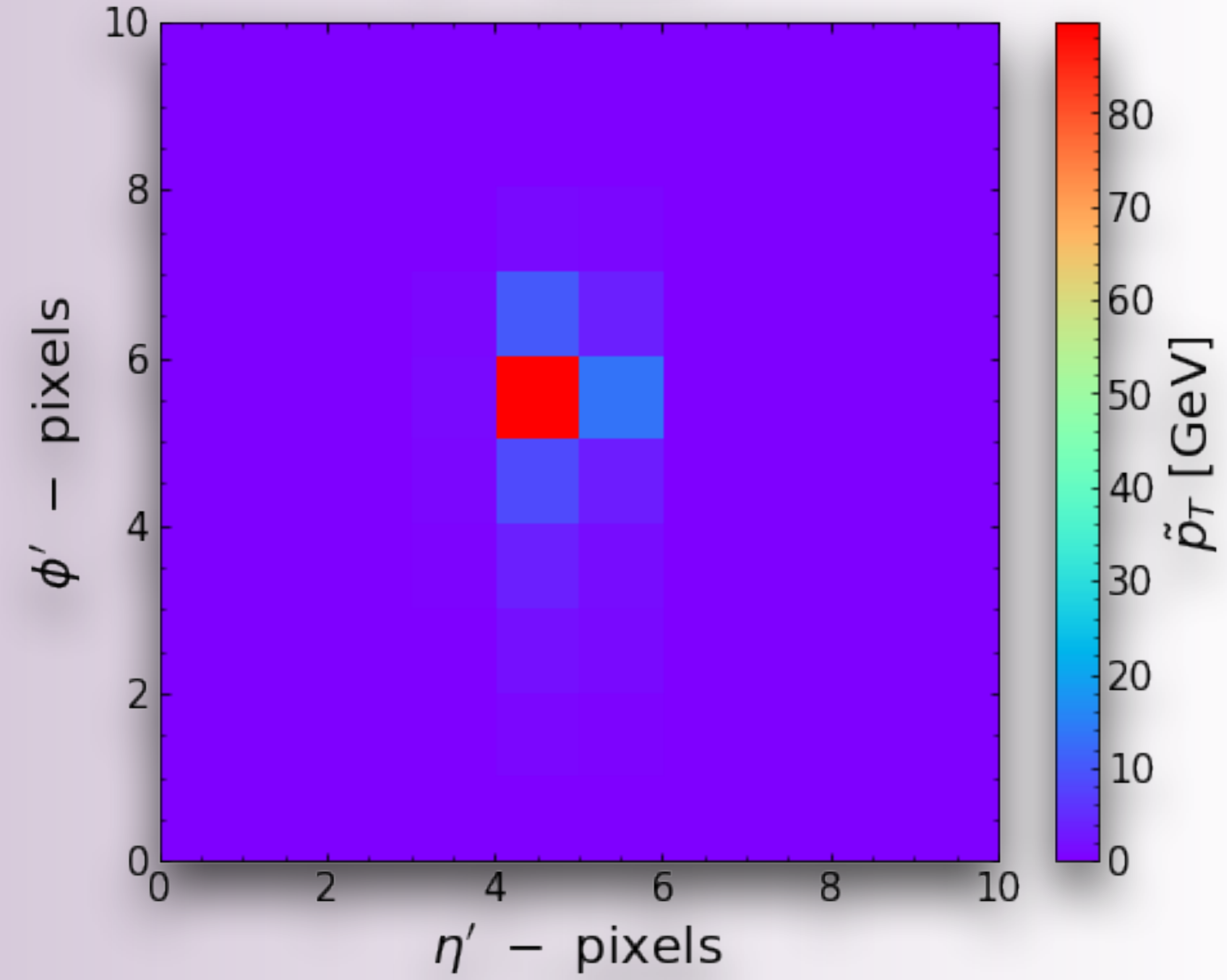


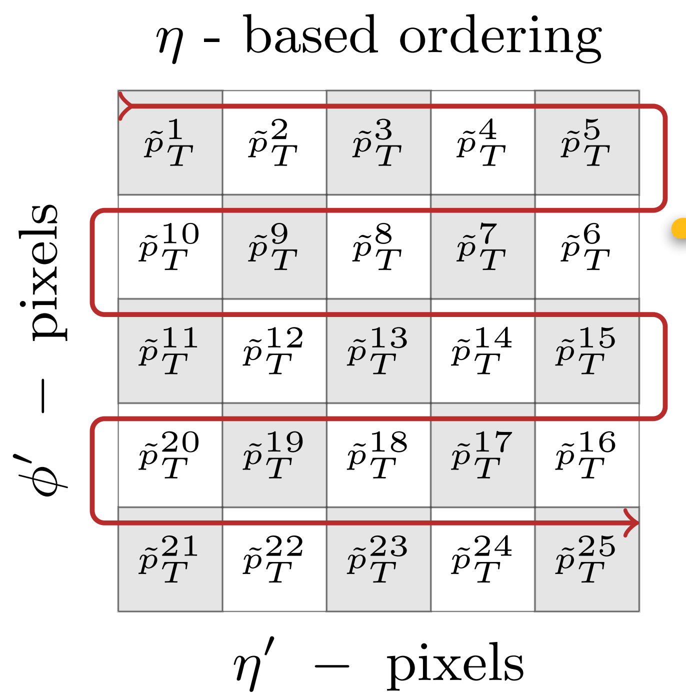
$$|\Psi\rangle_{AB} = \sum_{i}^{\chi} \lambda_\alpha |\alpha\rangle_A |\alpha\rangle_B \quad \rightarrow \quad \lambda_\alpha := \text{Schmidt values}$$

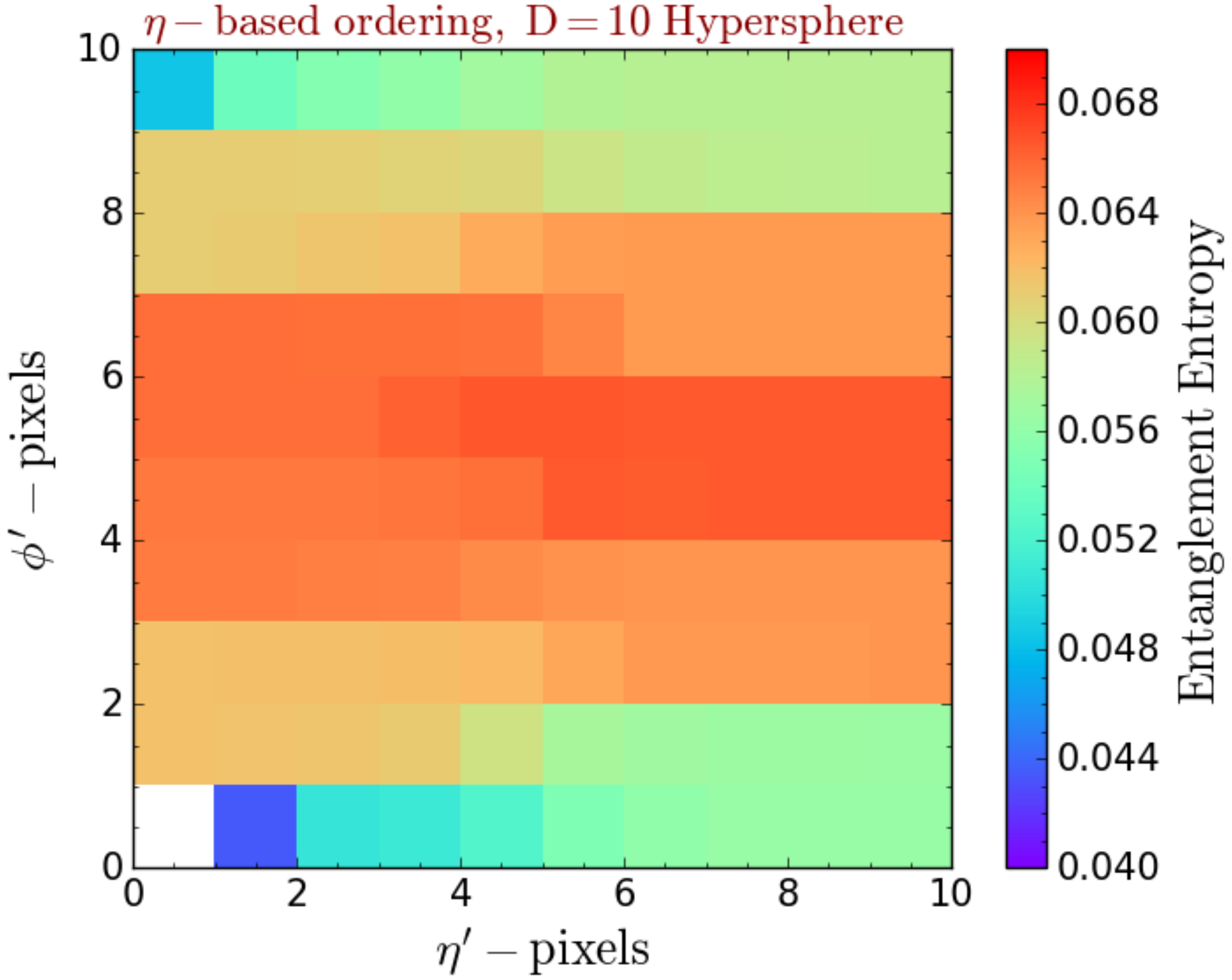
$$\boxed{\mathcal{S} = -\sum_{i}^{\chi} \lambda_i^2 \log \lambda_i^2} \quad \rightarrow \quad \text{von Neumann entropy}$$

Top Tagging through MPS

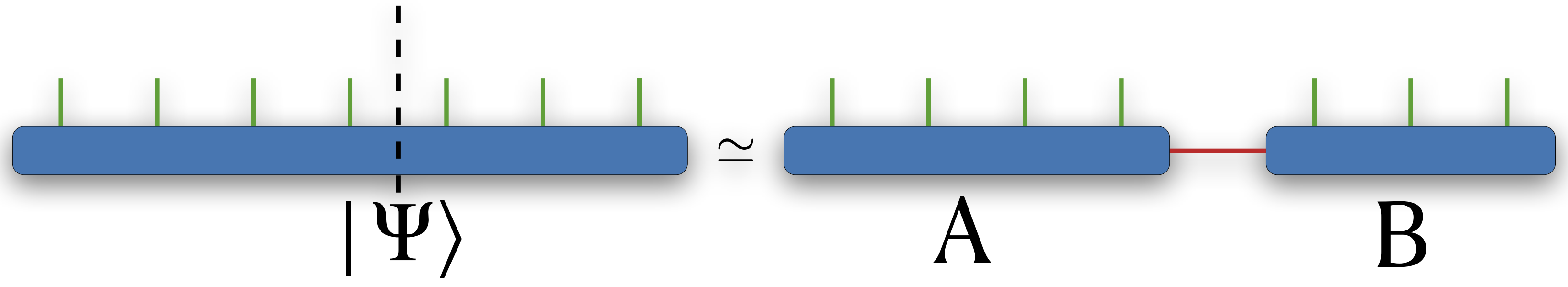






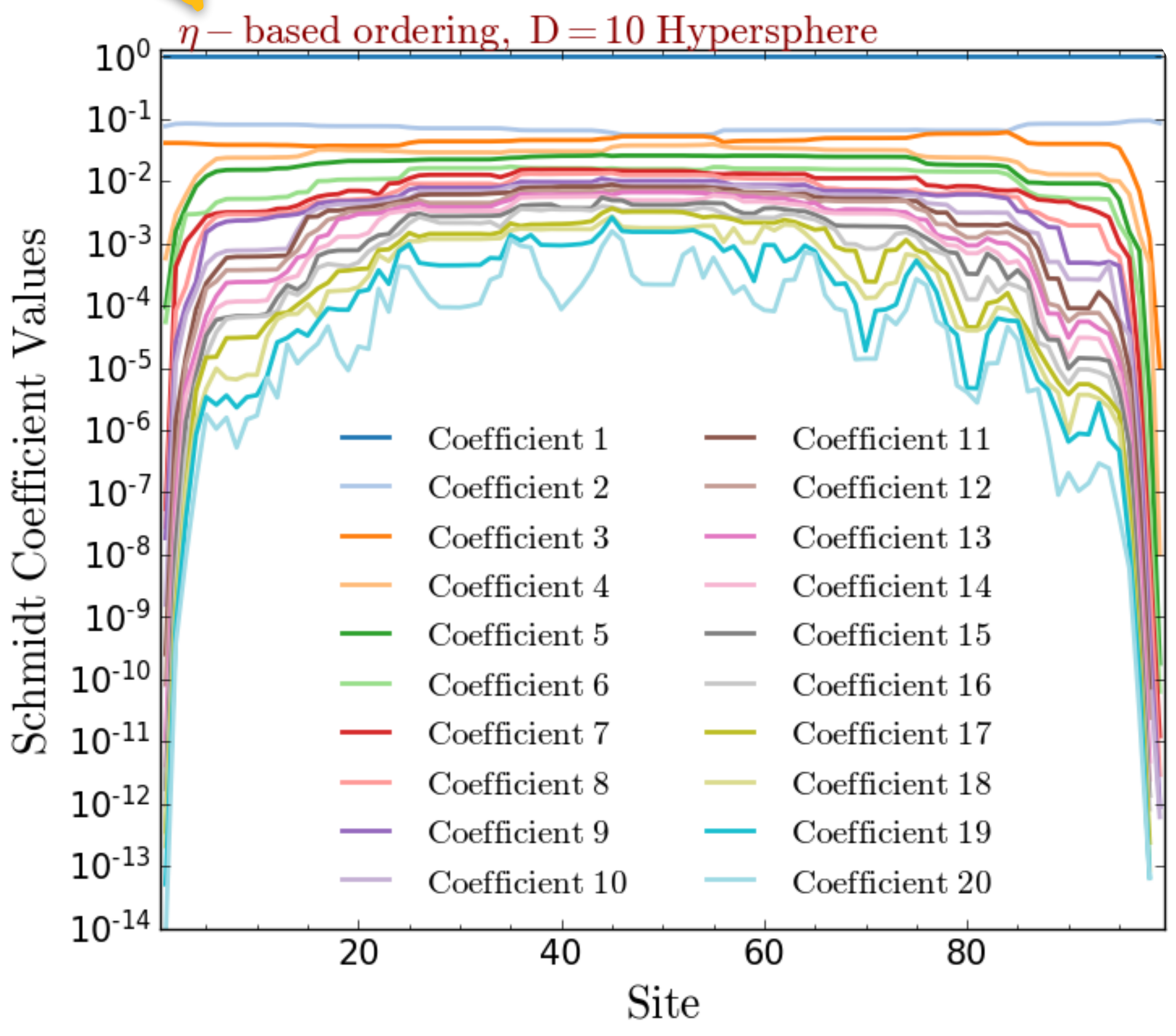


Entanglement Entropy & Schmidt Decompostion



$$|\Psi\rangle_{AB} = \sum_i^{\chi} \lambda_\alpha |\alpha\rangle_A |\alpha\rangle_B$$

$$\boxed{\mathcal{S} = -\sum_i^{\chi} \lambda_i^2 \log \lambda_i^2}$$



Fisher Information & Effective Dimensions