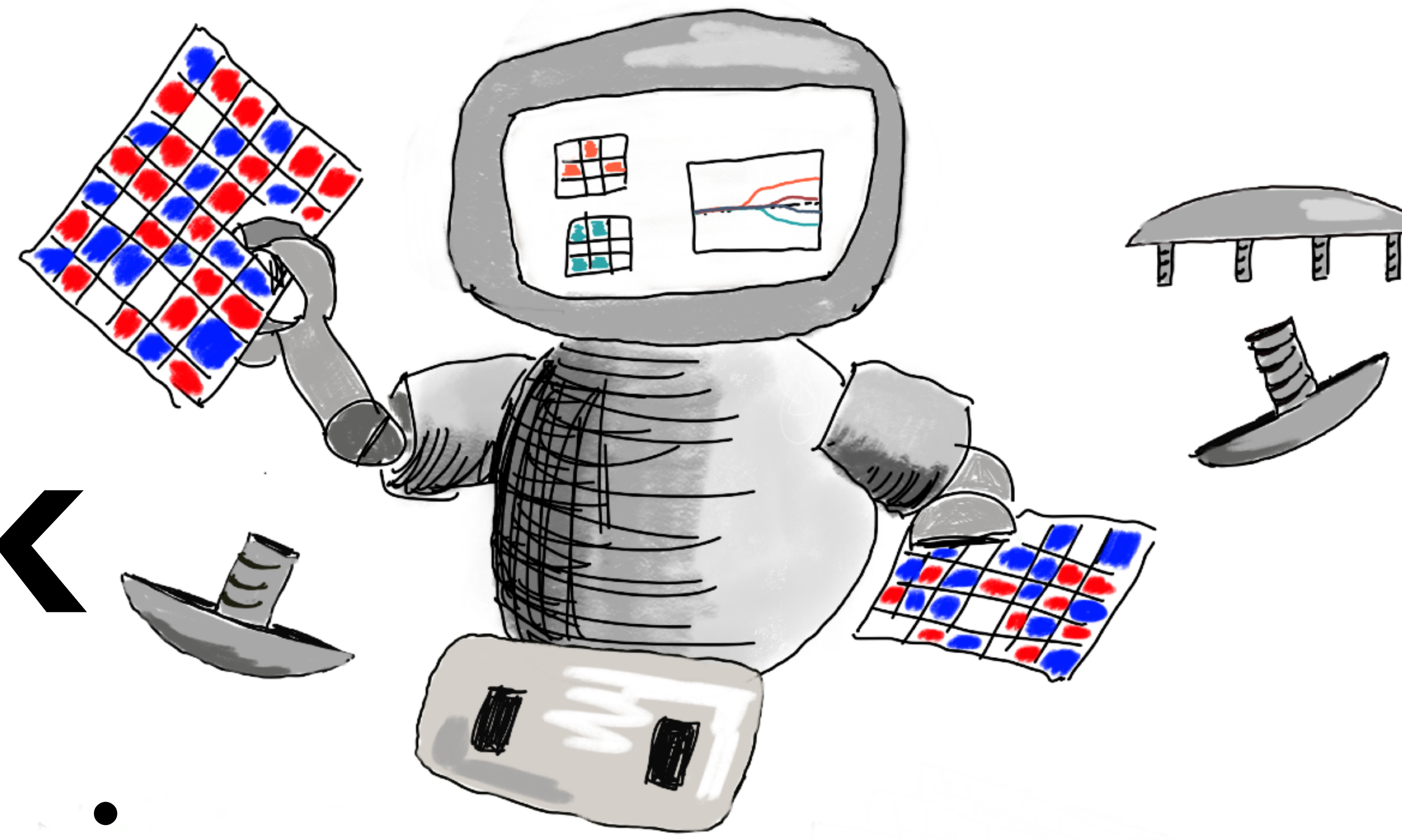
Annabelle Bohrdt



Understanding your network

towards interpretability in machine learning applications in many-body physics

ASC summer school 2022

Overview: Machine learning for quantum many-body physics

Annabelle Bohrdt

Recent review: Dawid et al., arXiv:2204.04198

1) Phase classification: supervised/unsupervised (+ many different physical systems) & characterizing phases Original work: Carrasquilla & Melko, Nature Physics 13 (2017)
2) Interpretability (analysis of bottlenecks, (symbolic) regression, CCNN architecture, Hessian based approach,…) See e.g. Dawid et al., Mach. Learn.: Sci. Technol. 3 (2021)
3) Neural network quantum states Review: Torlai & Melko, Annu. Rev. Condens. Matter Phys. (2020)
   - Representation — learning the state
   - Finding ground states/dynamics Original work: Carleo & Troyer, Science 355 (2017)
4) Reinforcement learning
   - Quantum error correction/state control
   - Experiment design See e.g. Melnikov et al., PNAS 115 (2017)
   - …
5) Quantum machine learning See e.g. Biamonte et al., Nature 549 (2017)

Annabelle Bohrdt

What problems are we trying to solve

Some (interesting) model with some tuning parameter

- (quantum) magnetism
- Topological transition
- Non-equilibrium dynamics
- Characterize a quantum system
- …

Some Input

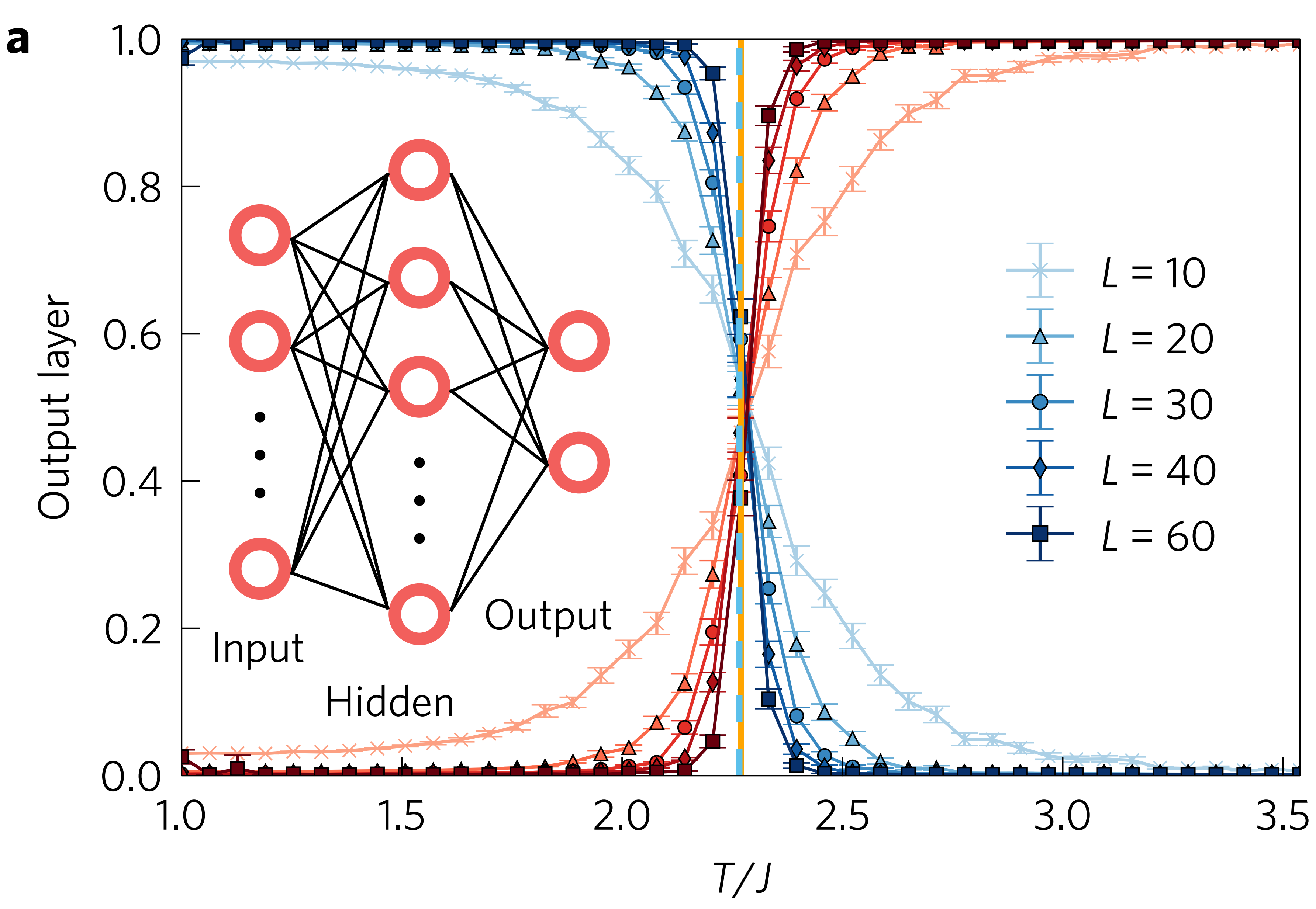
- Spin configurations
- Expectation values of observables
- Quantum states
- Entanglement spectra
- …

Some (machine learning) method

- Supervised and unsupervised methods
- Different neural network architectures
- Support vector machines
- …

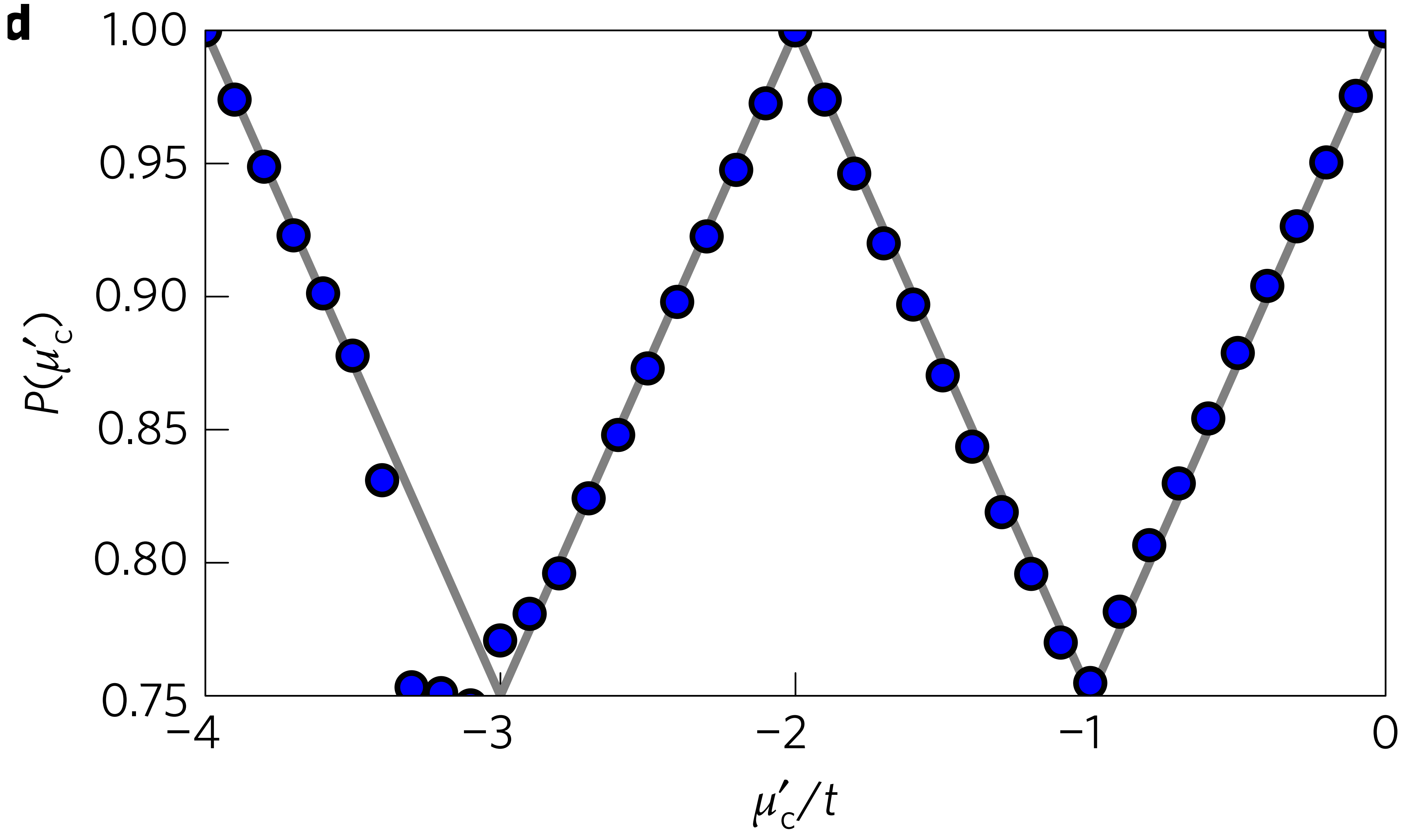
Machine learning phases of matter

2D Ising model at finite temperature
Input: spin configurations
Method: fully connected neural network



Carrasquilla & Melko, Nature Physics 13 (2017)

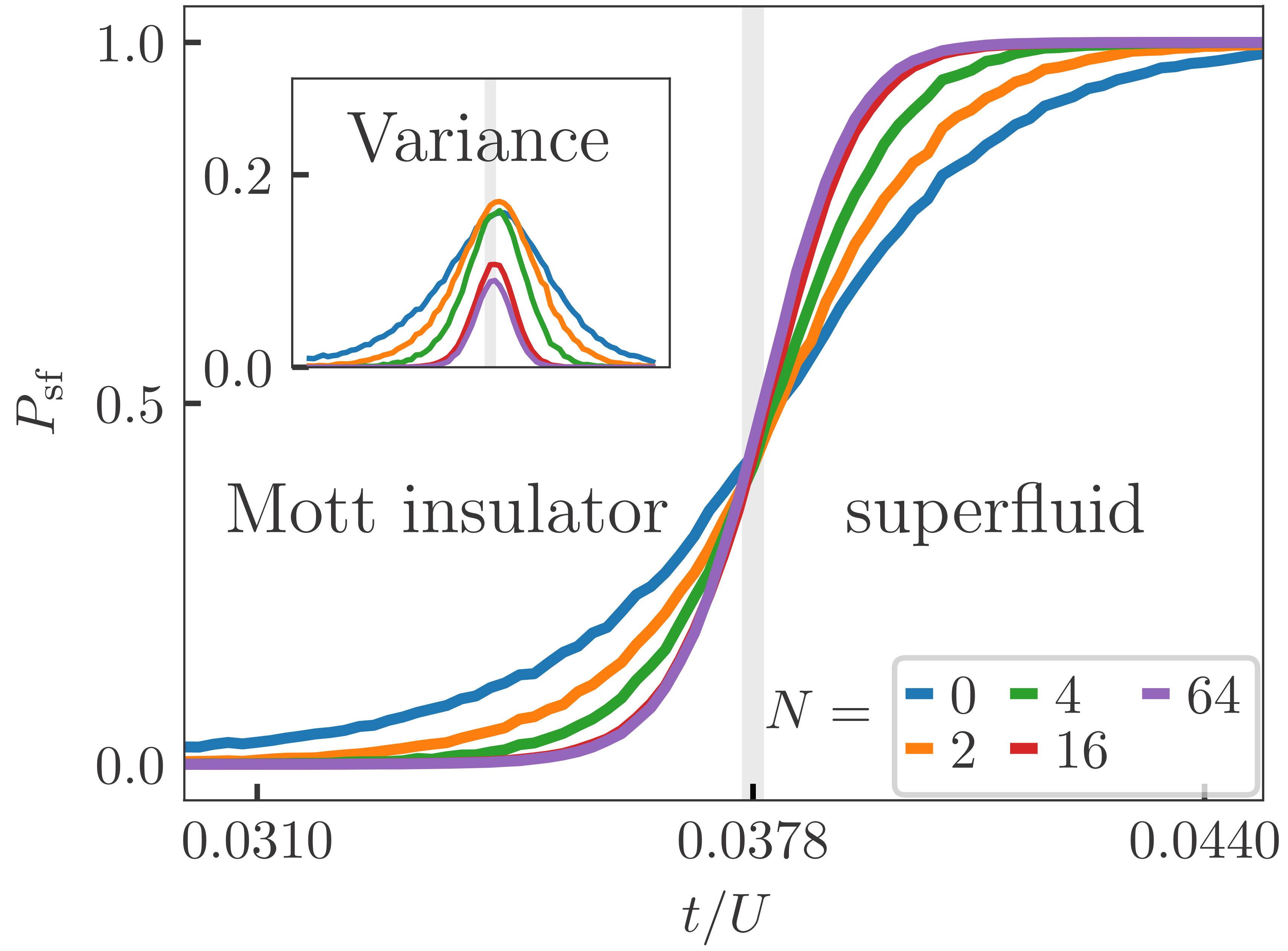
1D Kitaev chain
Input: entanglement spectra
Method: learning by confusion (with neural net)



Nieuwenburg et al., Nature Physics 13 (2017)

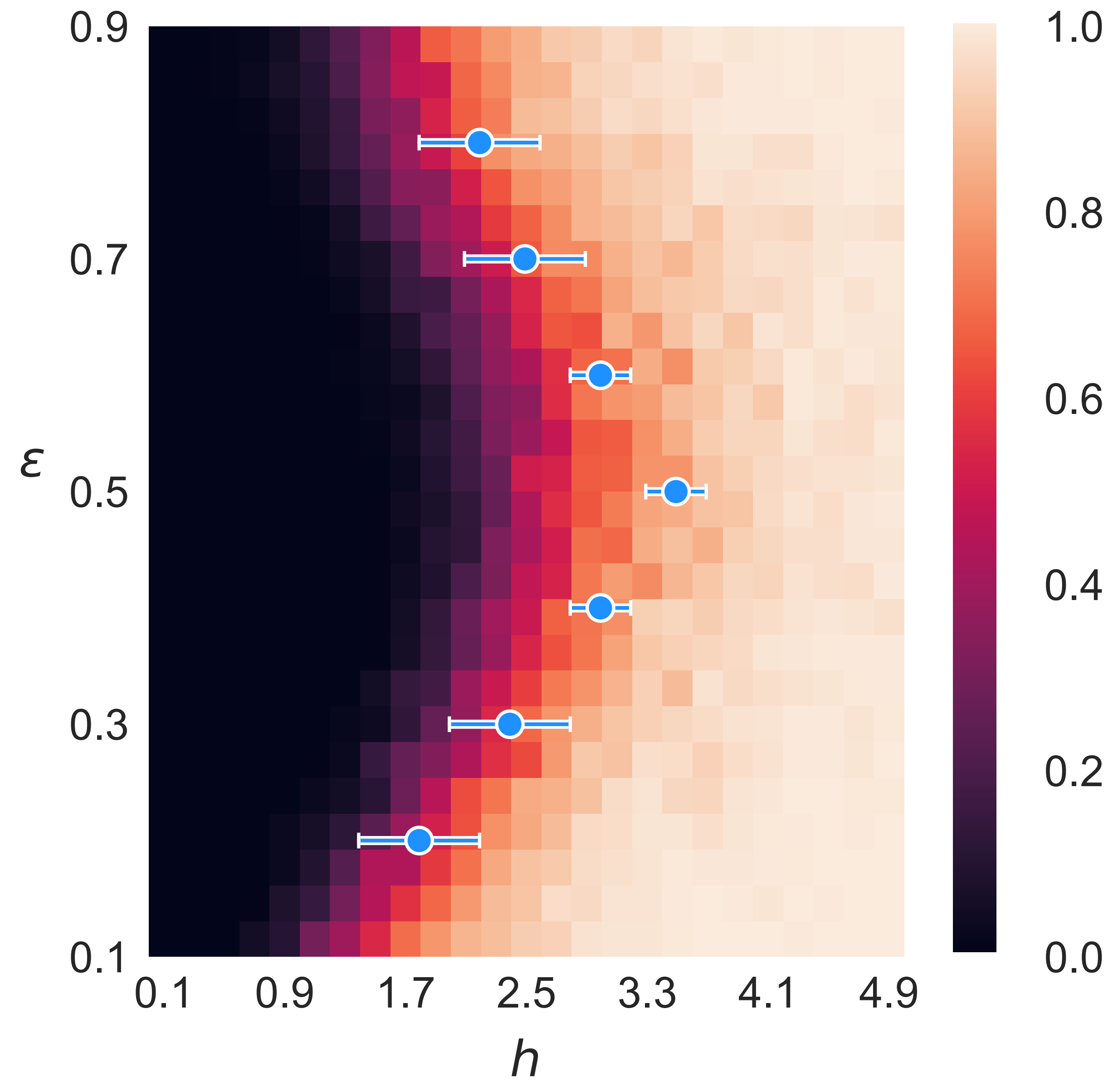
Machine learning phases of matter

Mott insulator to superfluid transition
Input: QMC configurations from SSE
Method: convolutional neural network



Dong et al., PRB 99 (2019)

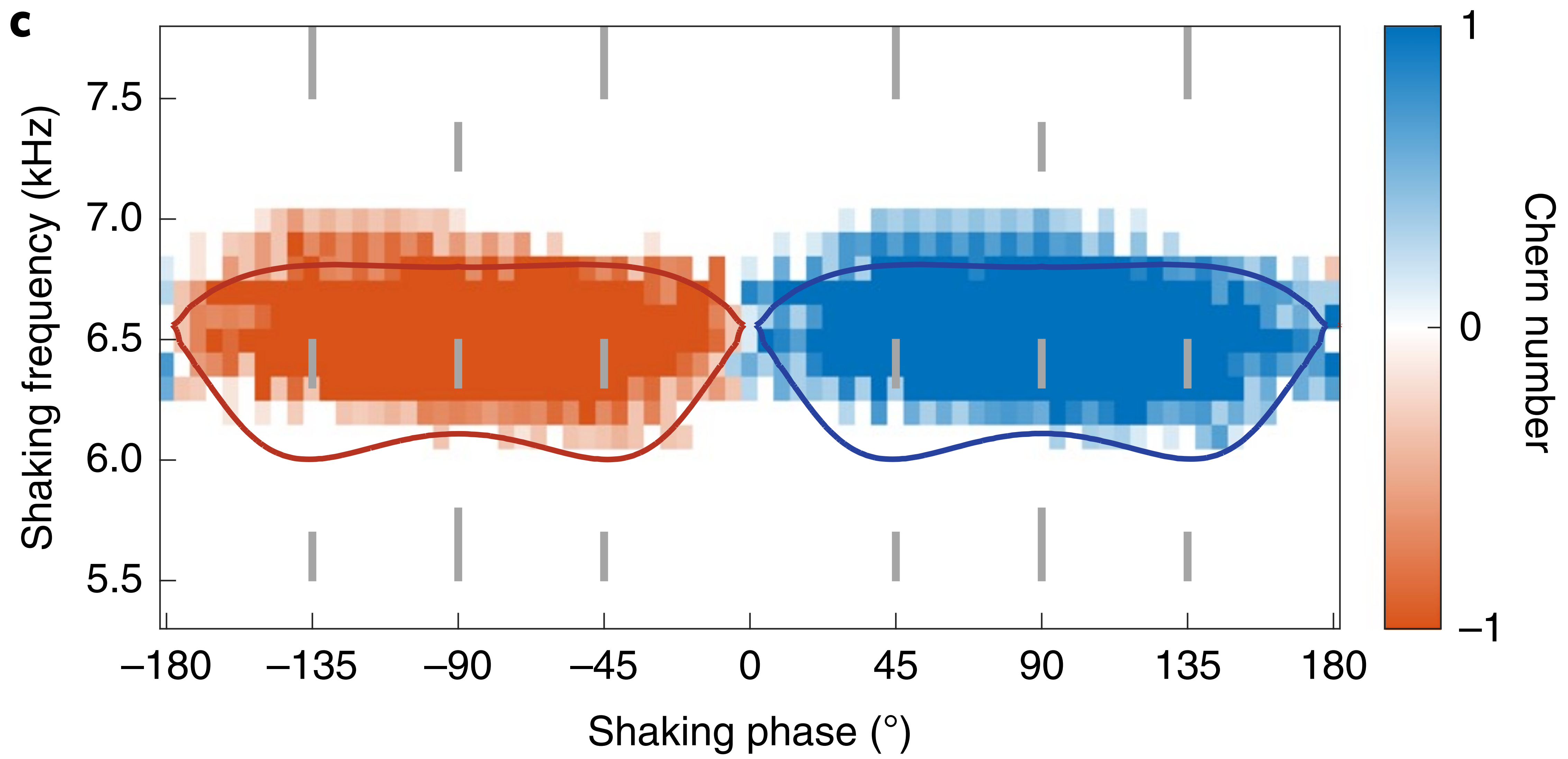
Many-body localization
Input: eigenstates of the Hamiltonian
Method: Adversarial training (two neural nets)



Huembeli et al., PRB 99 (2019)

Machine learning phases of matter

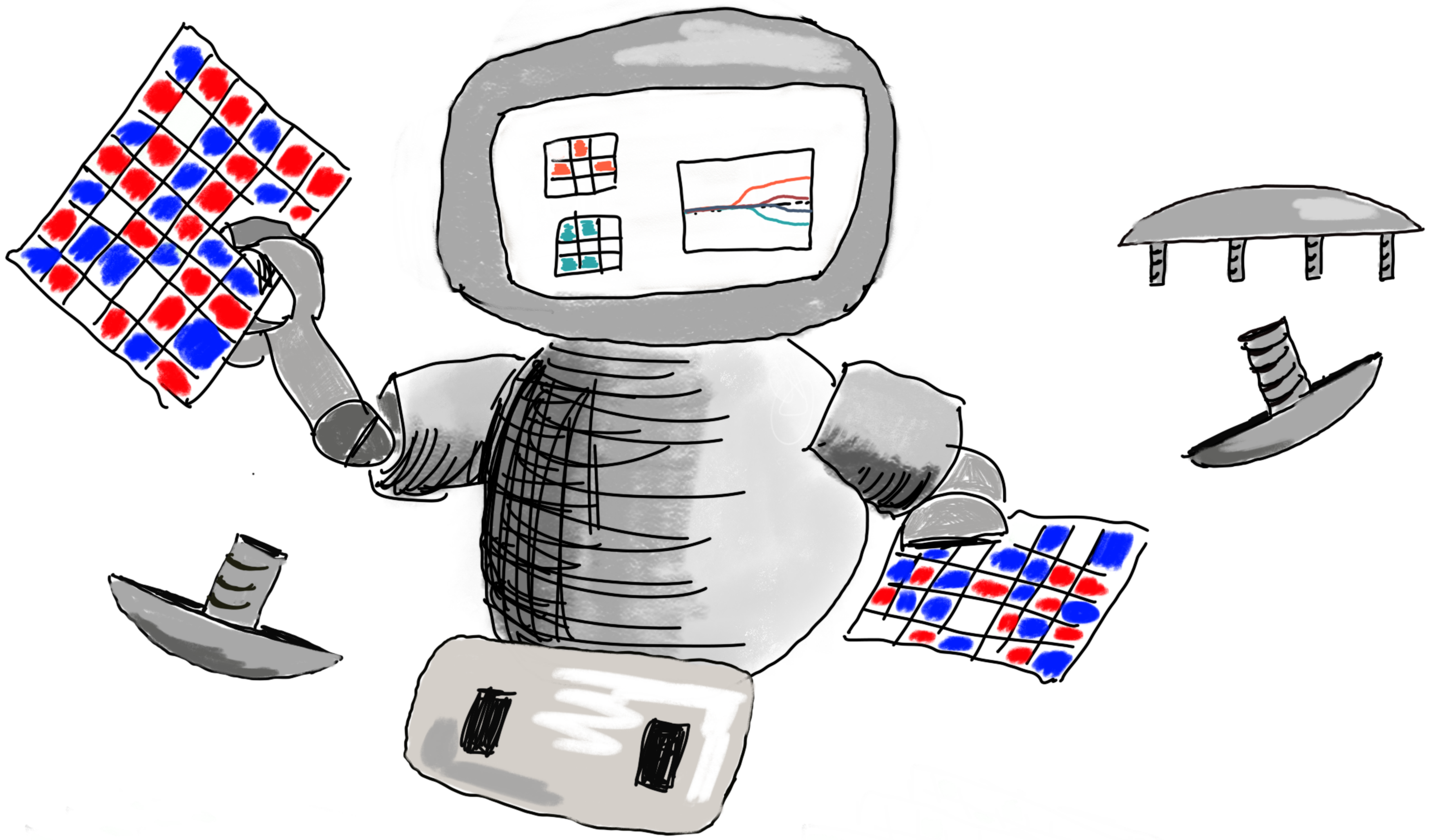
Topological phase diagram of Haldane model
Input: experimental time of flight images
Method: convolutional neural network



Rem et al., Nature Physics 15 (2019)

Annabelle Bohrdt

Interpretability

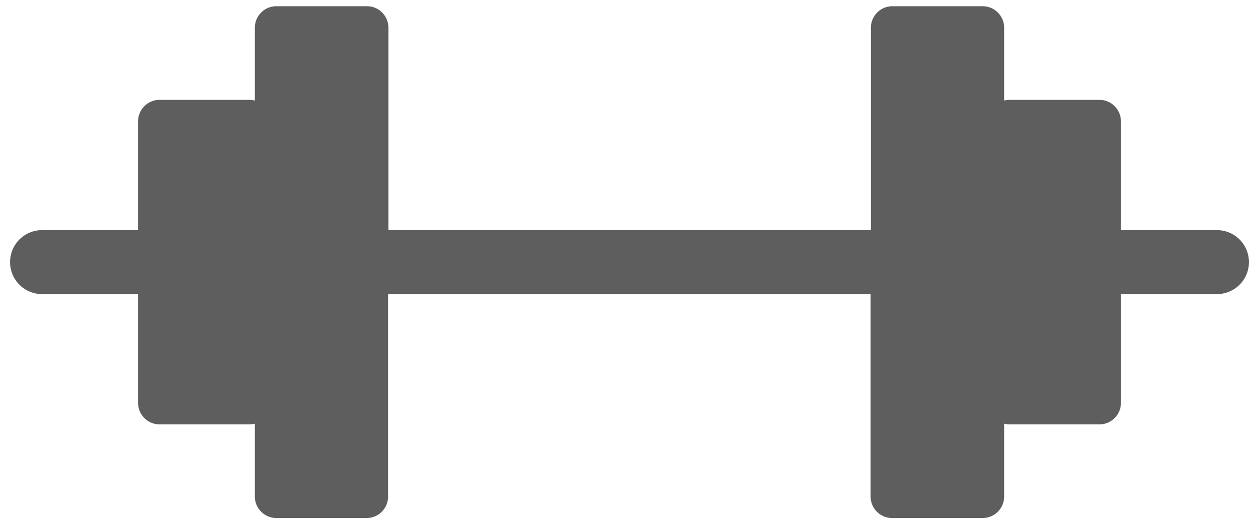


Different routes to interpretability

An incomplete and biased overview

- Directly looking at weights
- Bottleneck based approach
- Support vector machines
- Hessian based approach
- Taylor expansion type approach
  - Correlator convolutional neural network

Looking at weights



Annabelle Bohrdt

Looking at weights

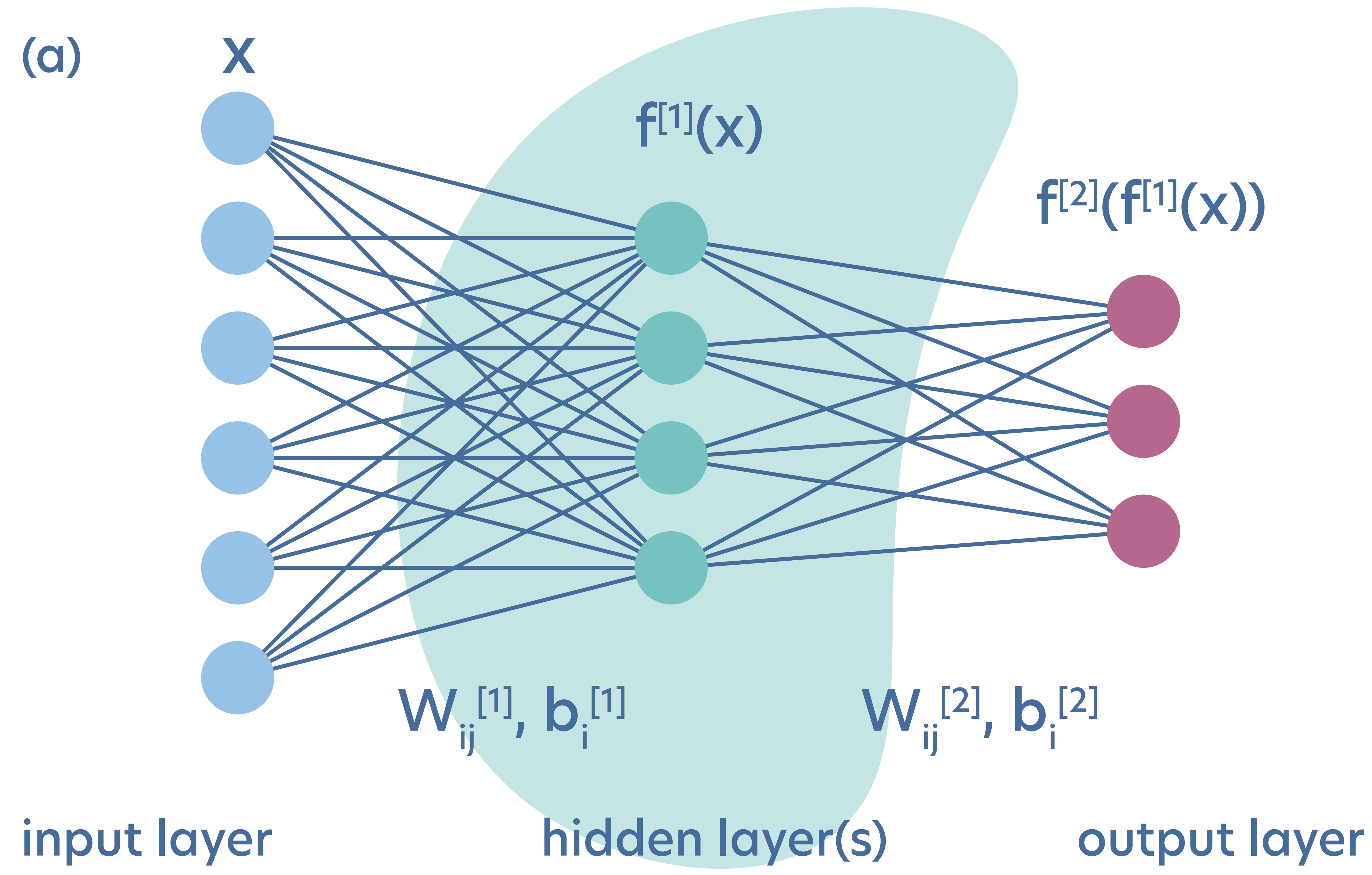
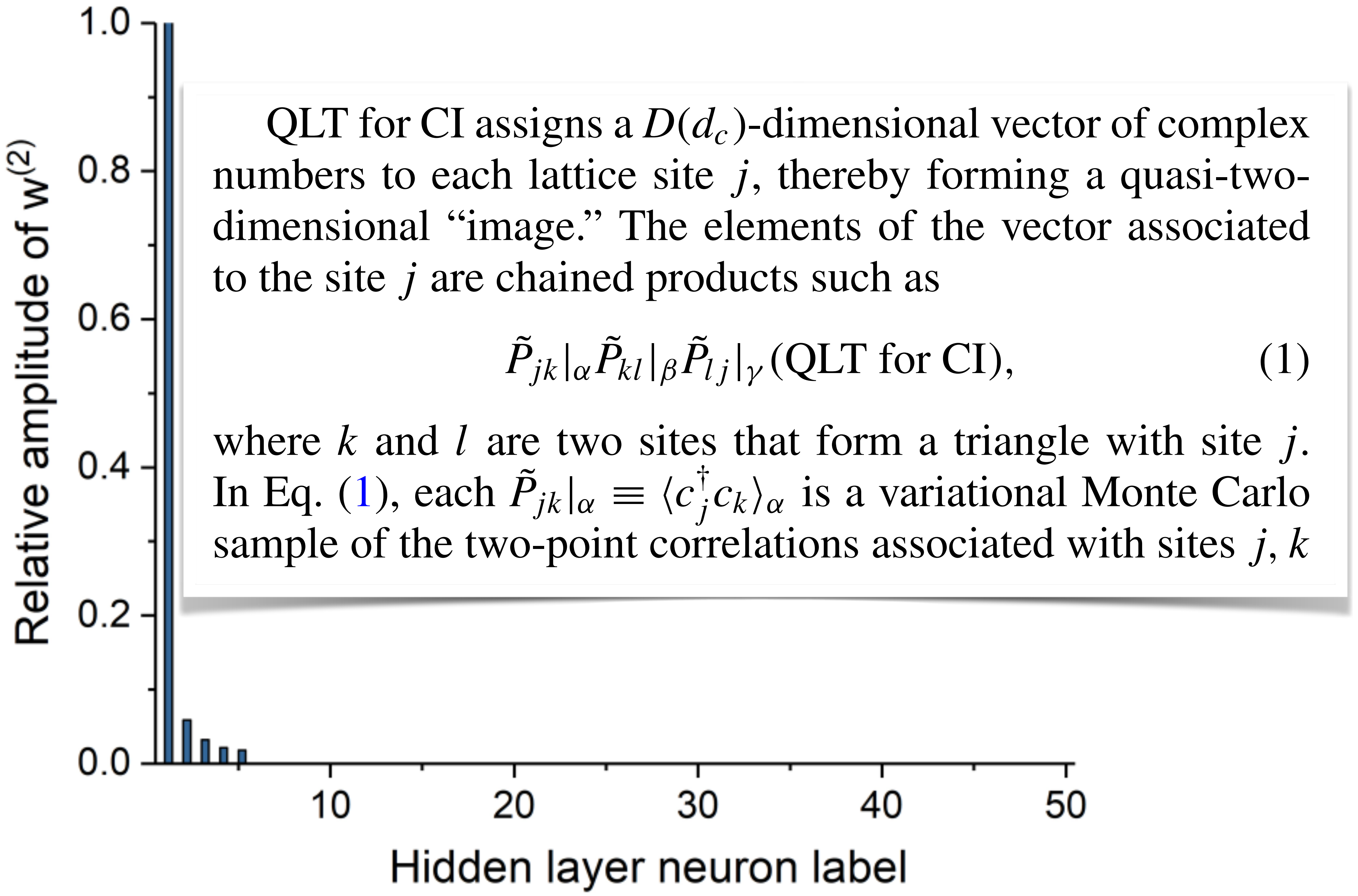


Image: Dawid et al., arXiv:2204.04198

Annabelle Bohrdt

Analyzing network weights

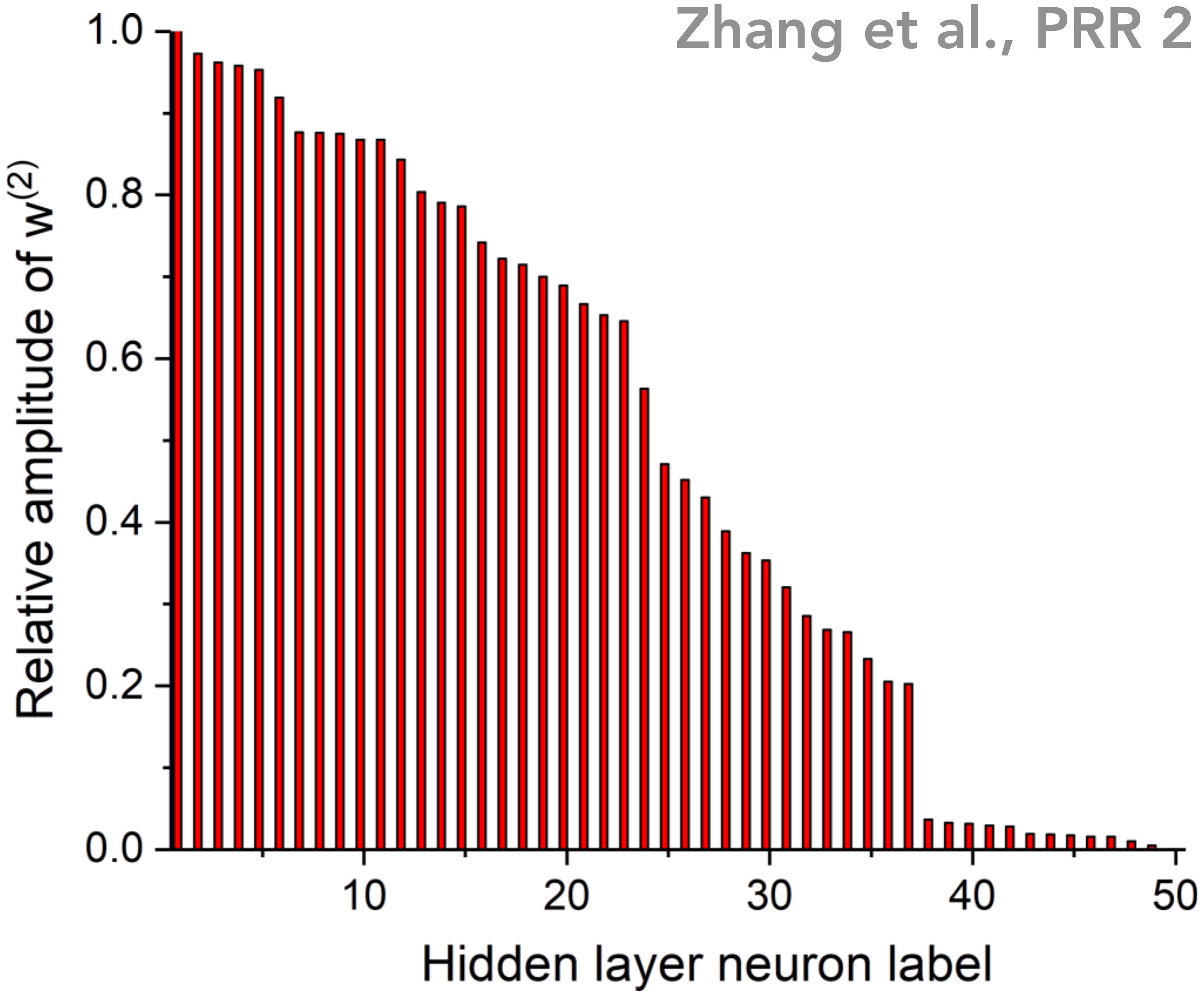
Zhang et al., PRR 2 (2020)



QLT for CI assigns a $D(d_c)$-dimensional vector of complex numbers to each lattice site j, thereby forming a quasi-two-dimensional "image." The elements of the vector associated to the site j are chained products such as

$$\tilde{P}_{jk}|_\alpha \tilde{P}_{kl}|_\beta \tilde{P}_{lj}|_\gamma \,(\text{QLT for CI}), \qquad (1)$$

where k and l are two sites that form a triangle with site j. In Eq. (1), each $\tilde{P}_{jk}|_\alpha \equiv \langle c_j^\dagger c_k \rangle_\alpha$ is a variational Monte Carlo sample of the two-point correlations associated with sites j, k



Linear dependence on input

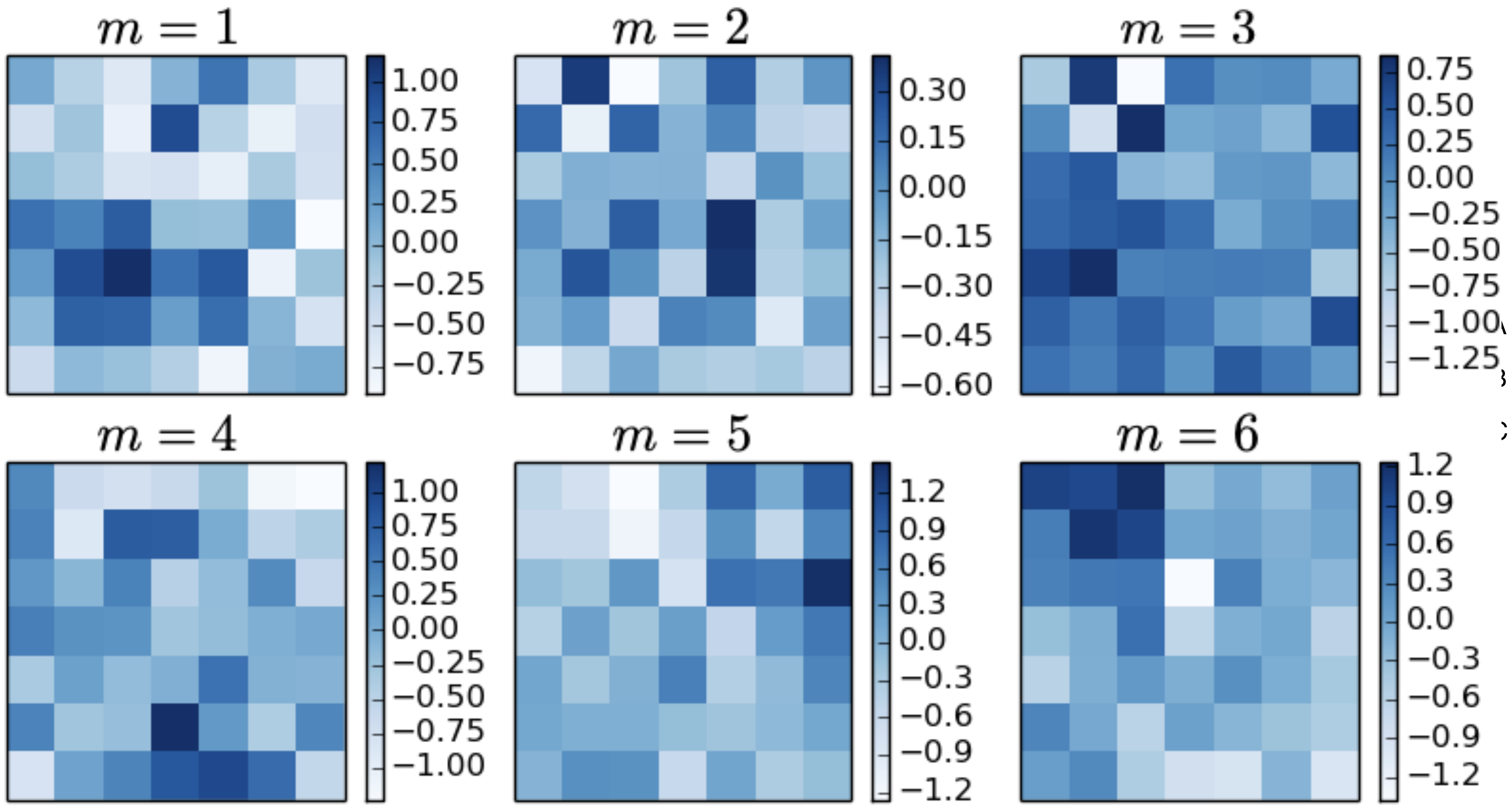
Non-linear dependence on input

=> Introduce non-linearities in input to regain linear dependence

Problem: pre-processing of input data, need a priori physical insights for feature selection

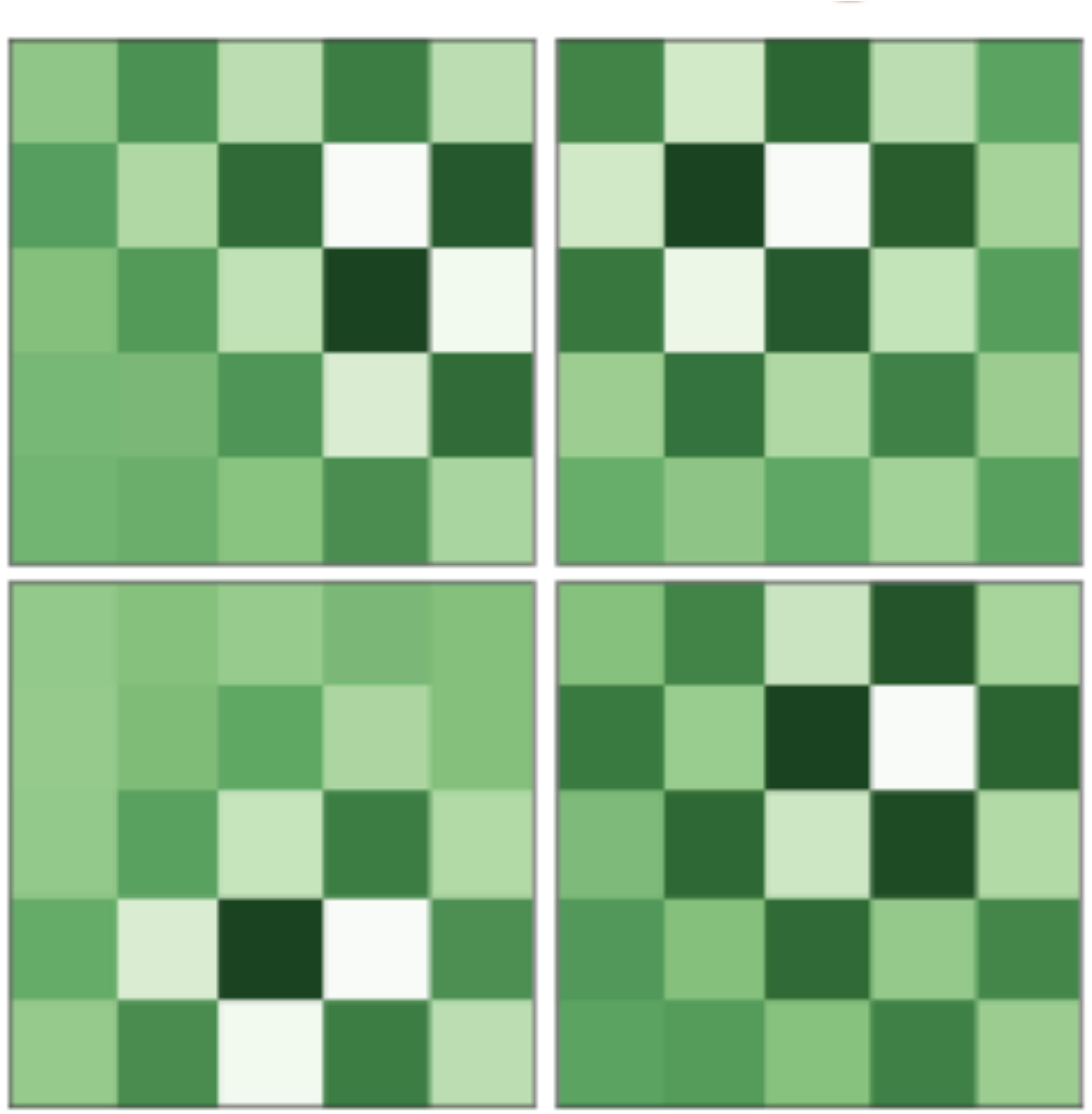
Convolutional neural networks: looking at filters

2D Fermi-Hubbard @ finite doping: charge configurations



Bohrdt et al., Nature Physics 15 (2019)

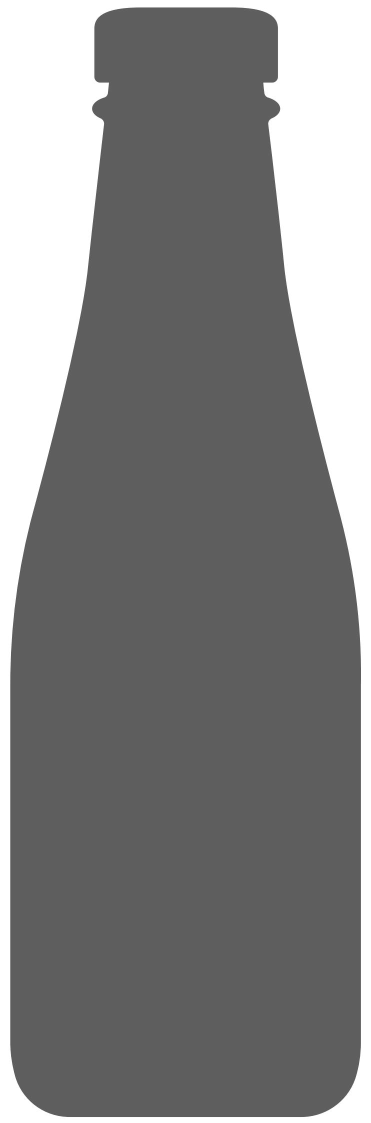
2D Fermi-Hubbard @ half filling: spin configurations at high vs low temperature



Khatami et al., PRA 102 (2020)

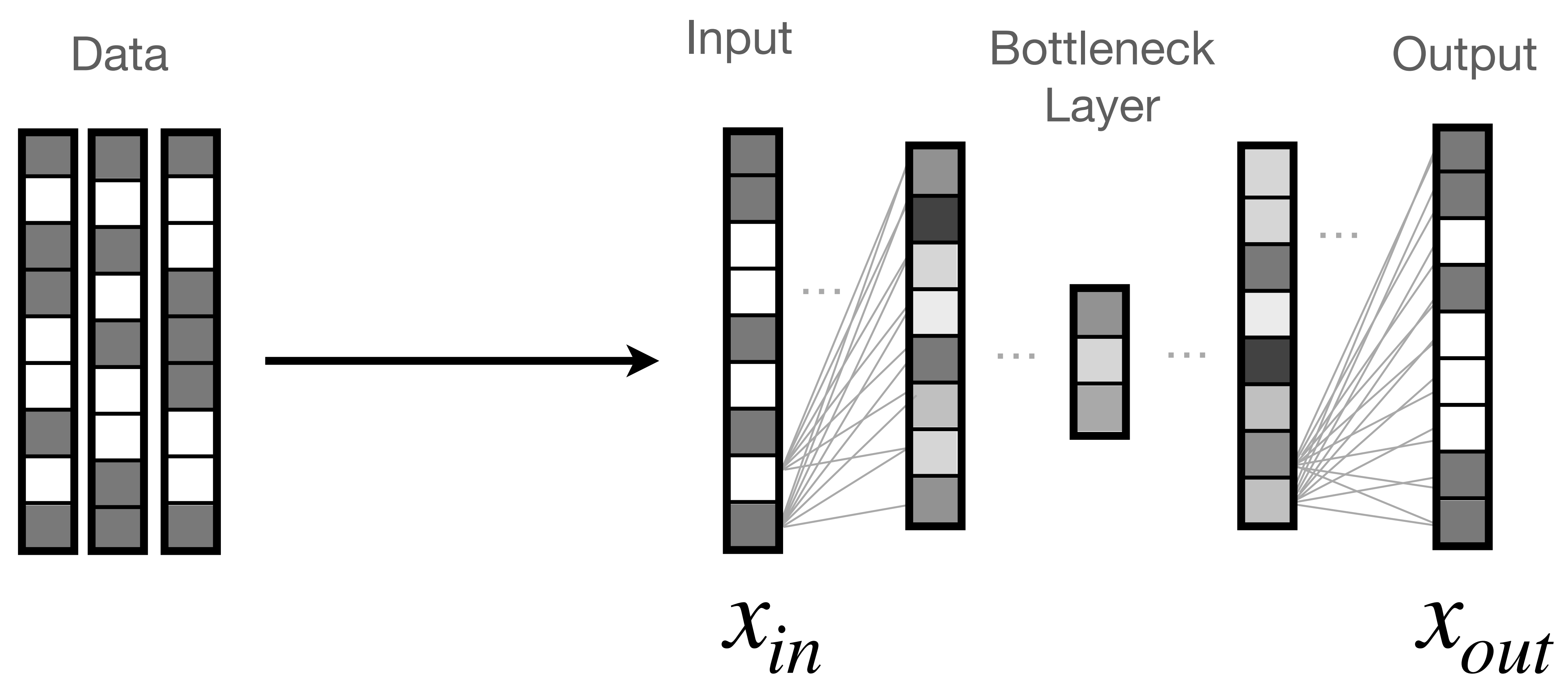
Problem: non-linearities introduce correlations which cannot be visualized this way

Bottleneck based approach



Annabelle Bohrdt

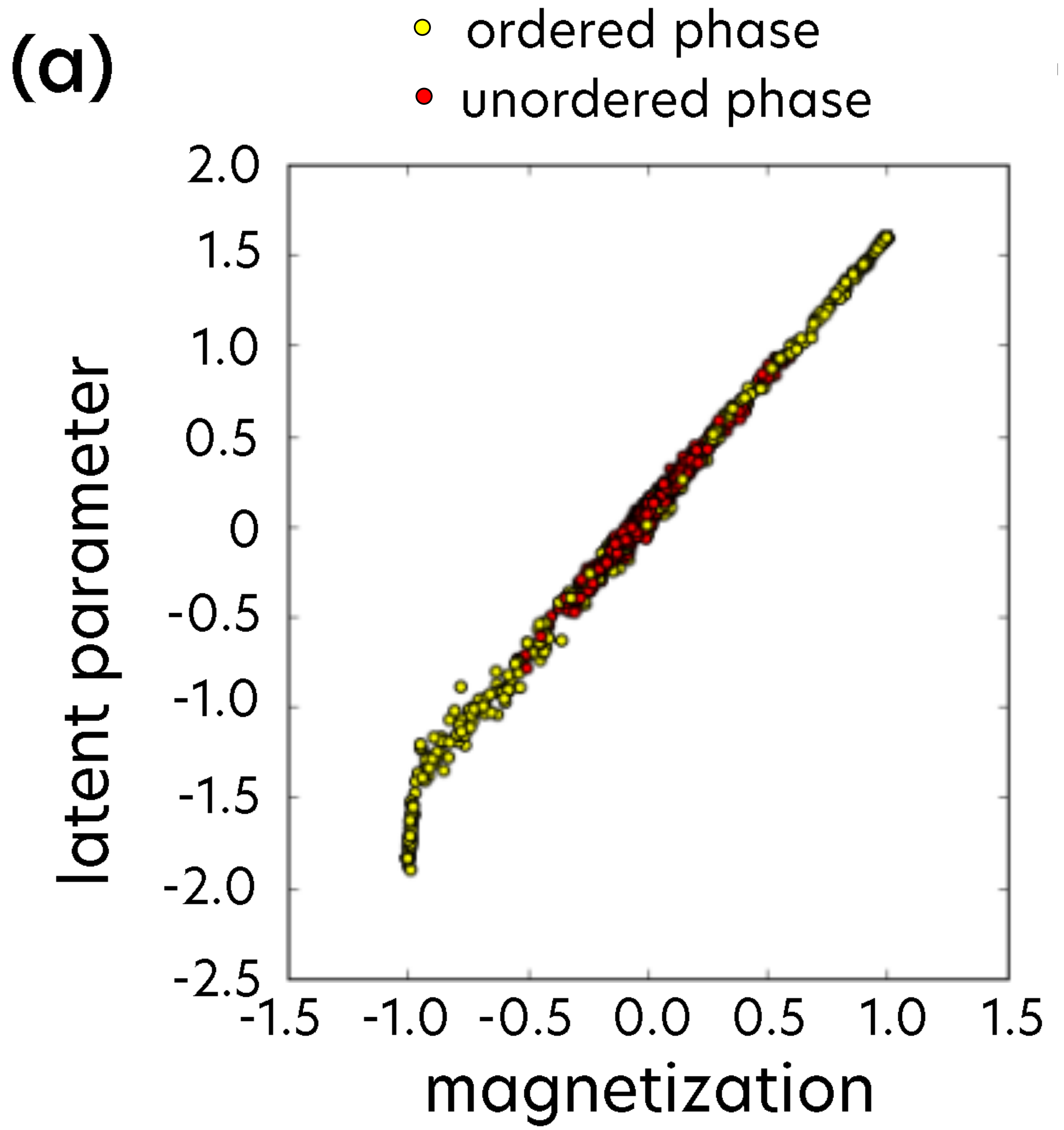
Bottleneck-based approach



$$L(x_{in}, x_{out}) = |x_{in} - x_{out}|^2$$

problem: assumes that we already know which dependence to look for —> we can only compare it against the quantities or features we suspect to be important.

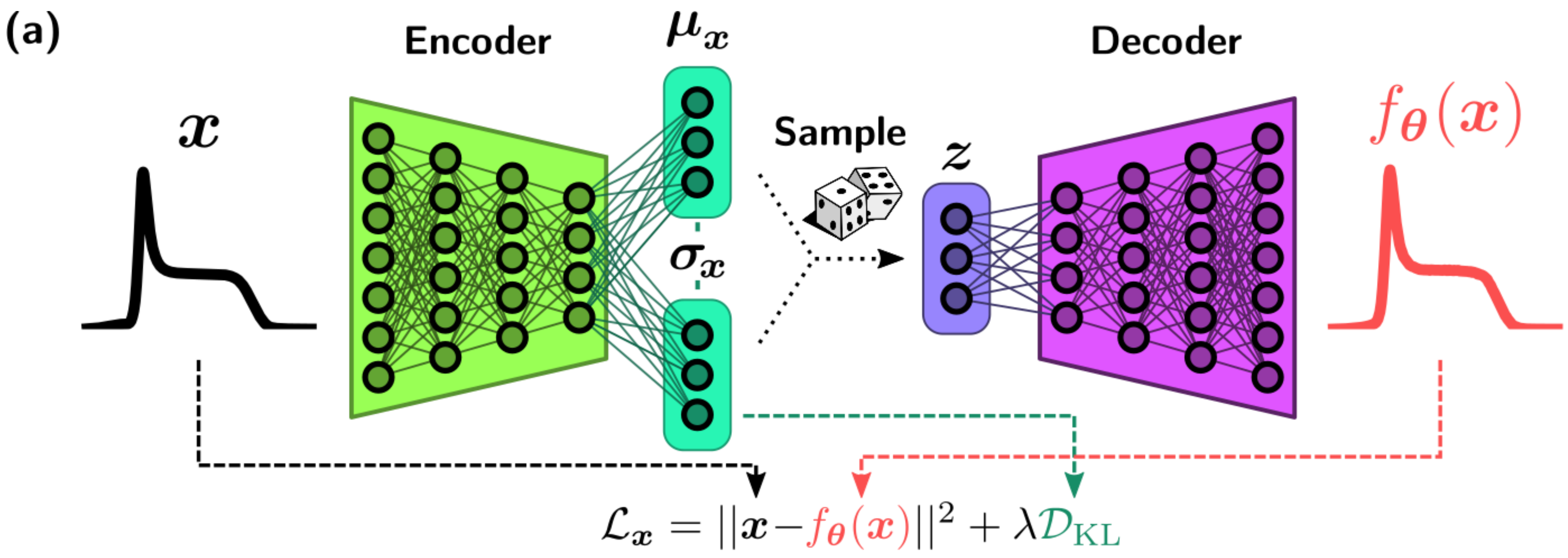
spin configurations of 2D Ising model



Wetzel, PRE 96 (2017)

Bottleneck-based approach

One possibility: symbolic regression on the latent space



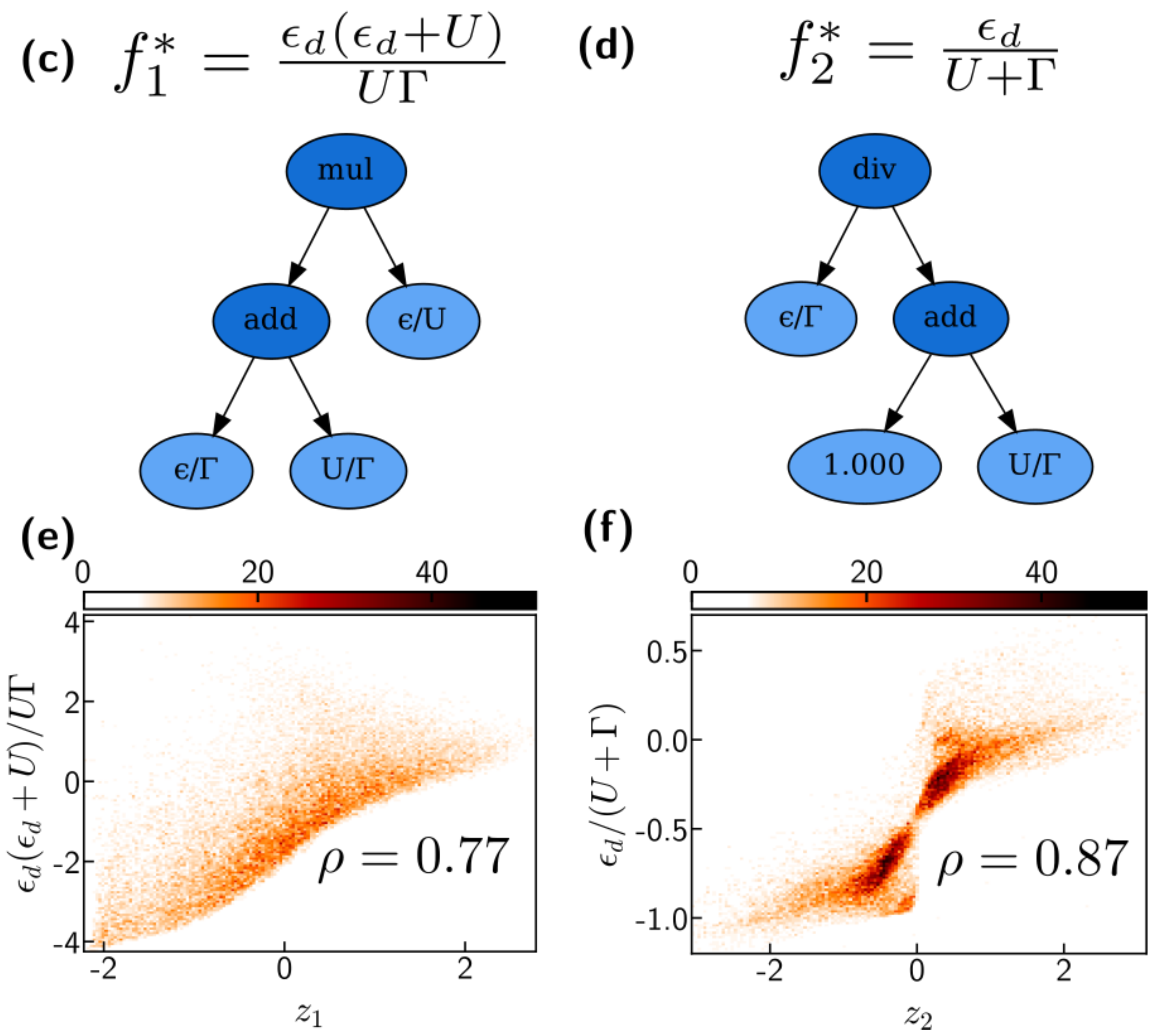
Goal: Functional form to relate physical input parameters to latent vector z

Miles et al., PRB 104 (2021)

Symbolic regression

One possibility: symbolic regression on the latent space

- Here: one-particle Anderson impurity model spectral functions
- Use genetic algorithm to obtain functional form

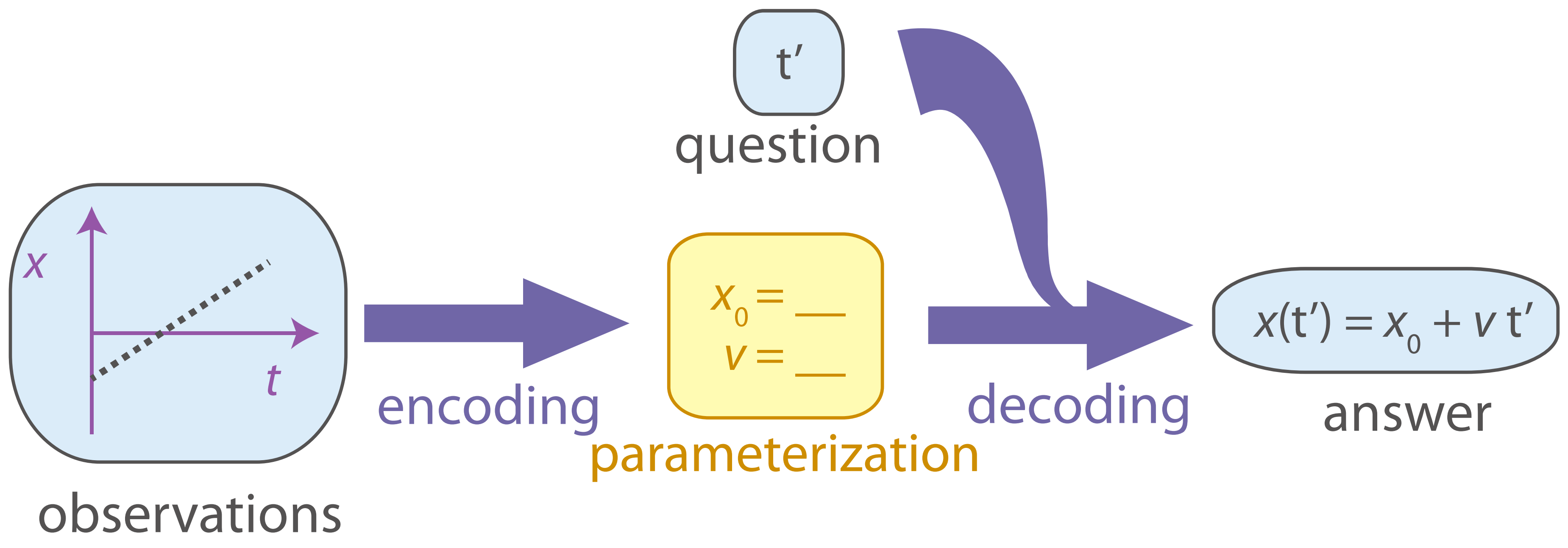


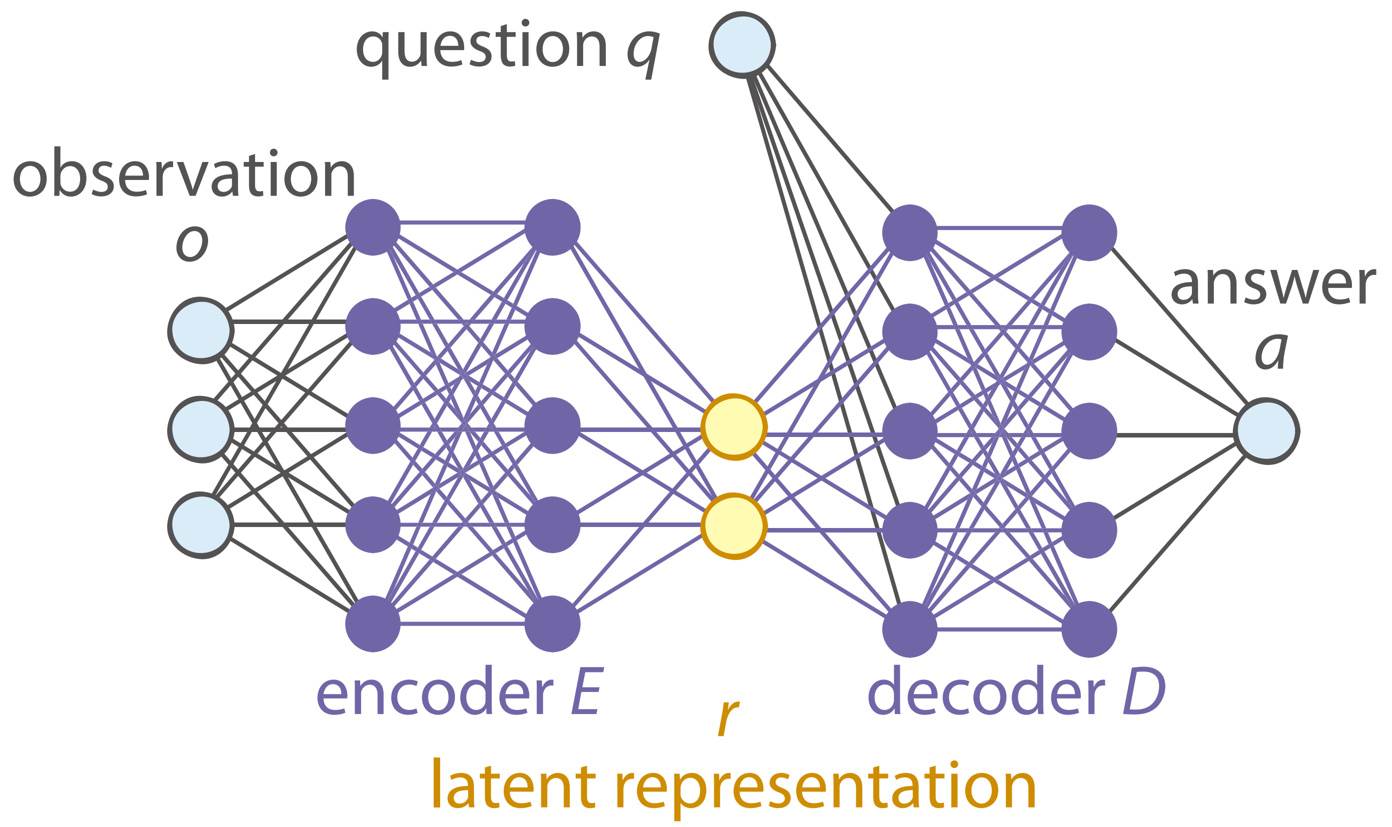
Miles et al., PRB 104 (2021)

More on symbolic regression (using neural nets):
Feynman AI: Udrescu & Tegmark, Sci. Adv. 6 (2020)

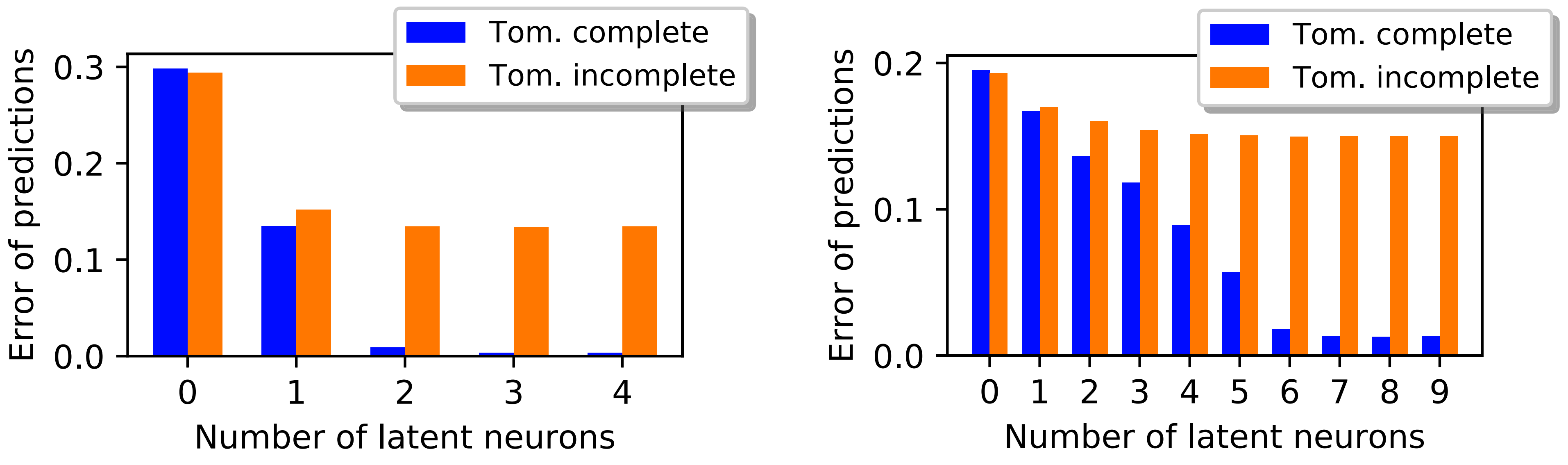
Bottleneck-based approach

Training using question neuron: Input: observations, question, correct answer





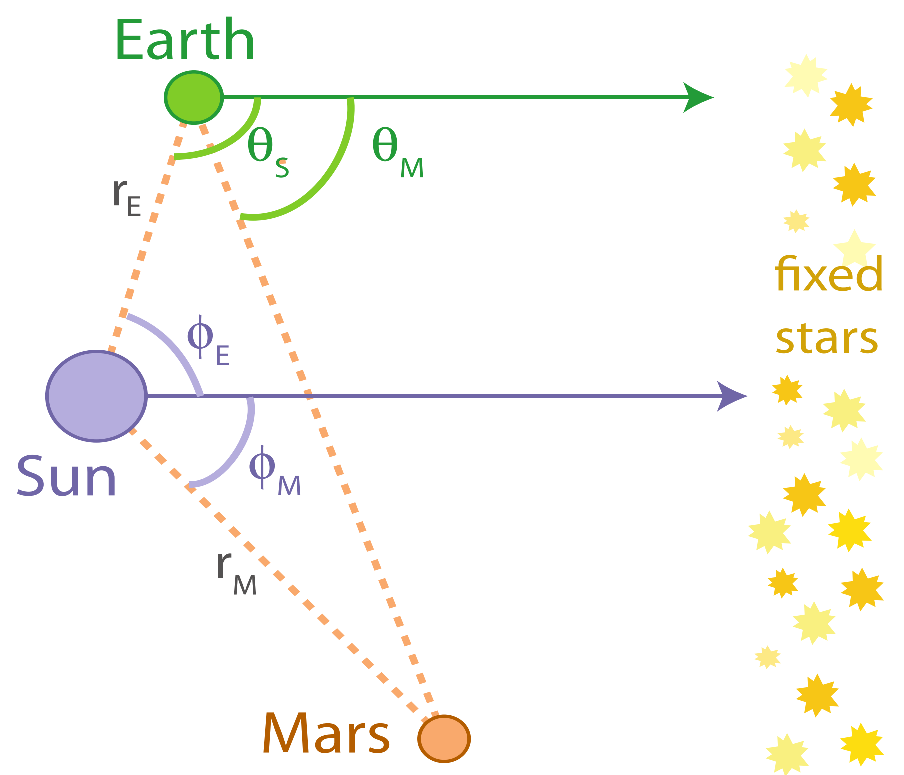
Quantum state tomography



(a) One qubit

(b) Two qubits

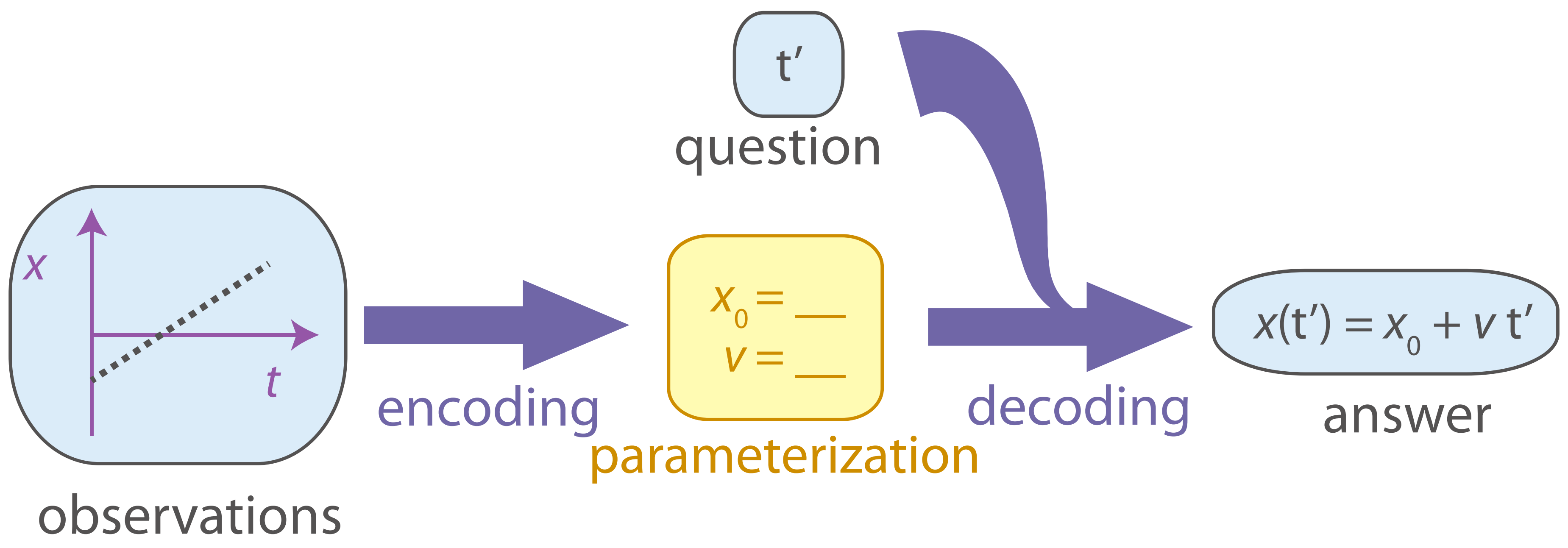
Heliocentric solar system

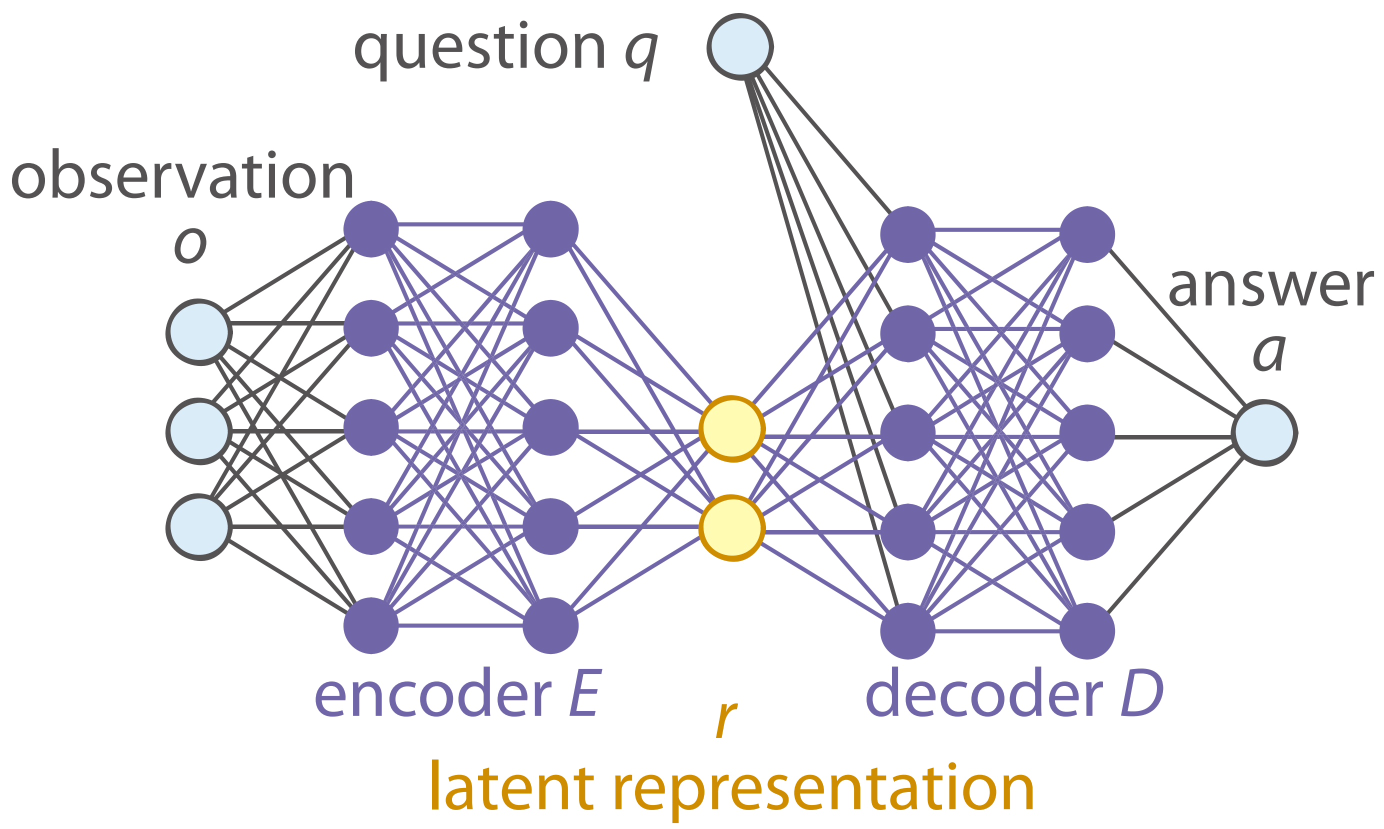


Iten et al., PRL 124 (2020)

Bottleneck-based approach

Training using question neuron: Input: observations, question, correct answer



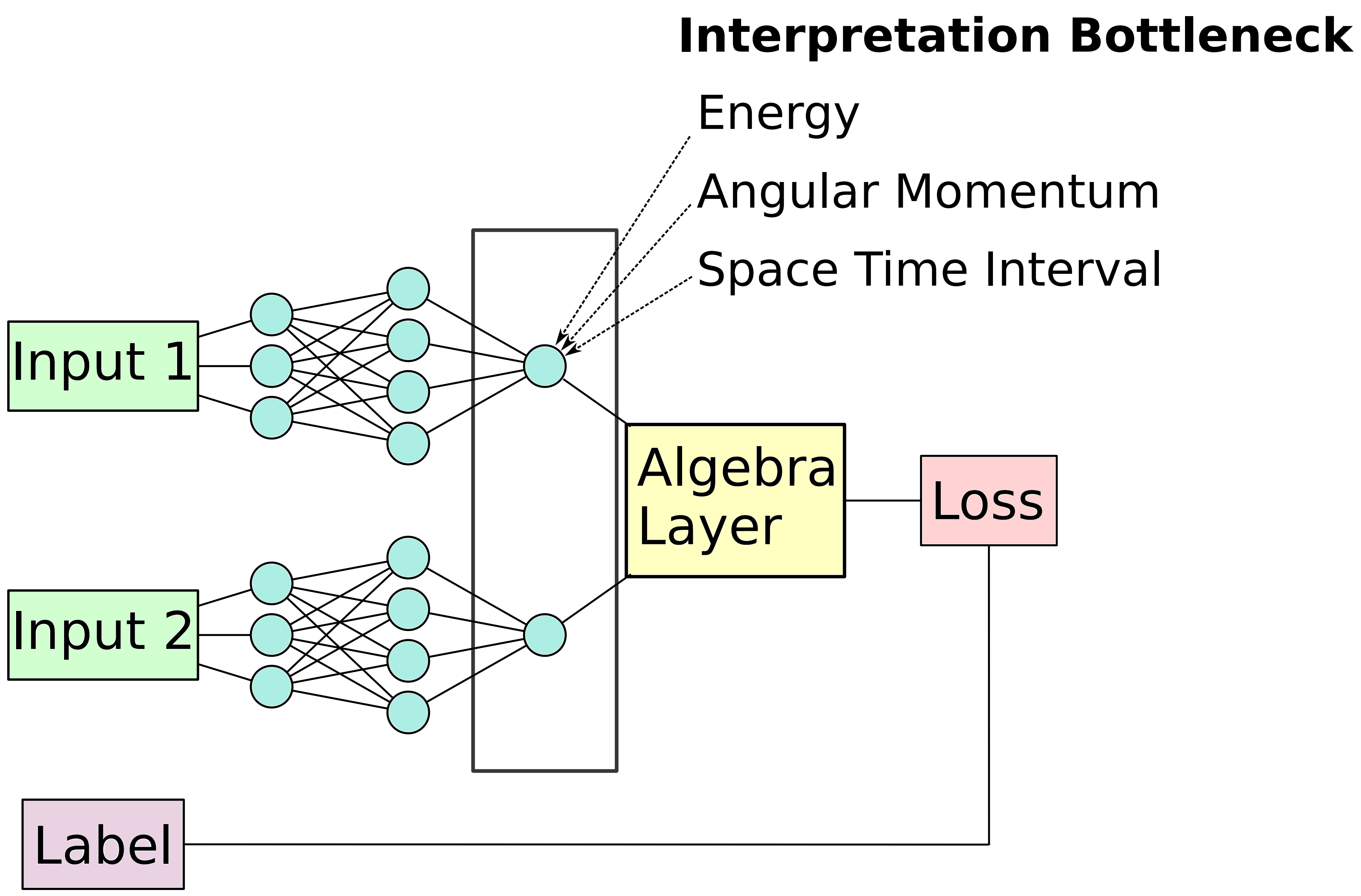


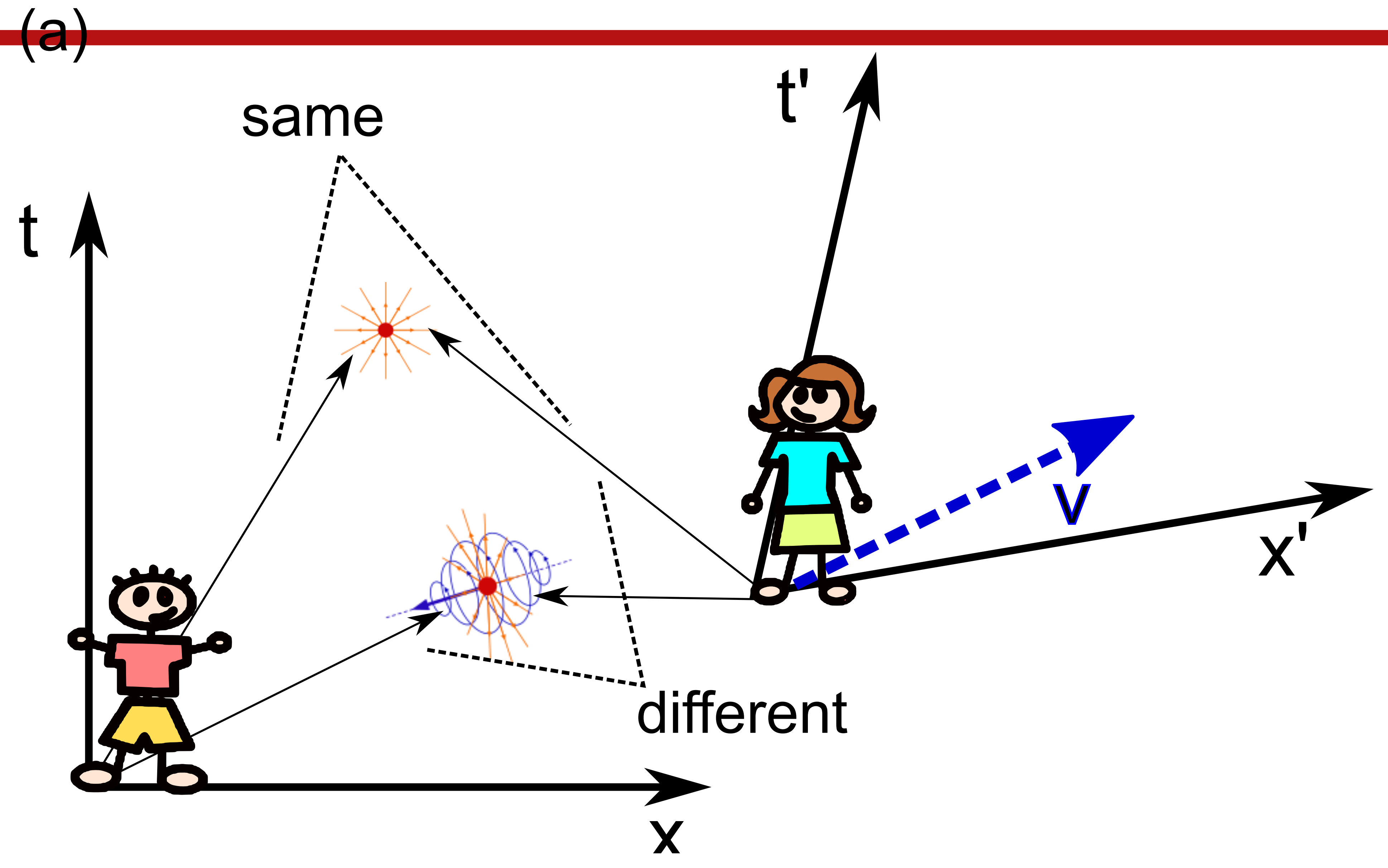
- all useful data stored in representation in compact form
- Comparison to hypothesis possible (if hypothesis exists)
- insights possible from the number of required parameters
- insights from change in the representation when manually changing the input, and the change in output when manually changing the representation

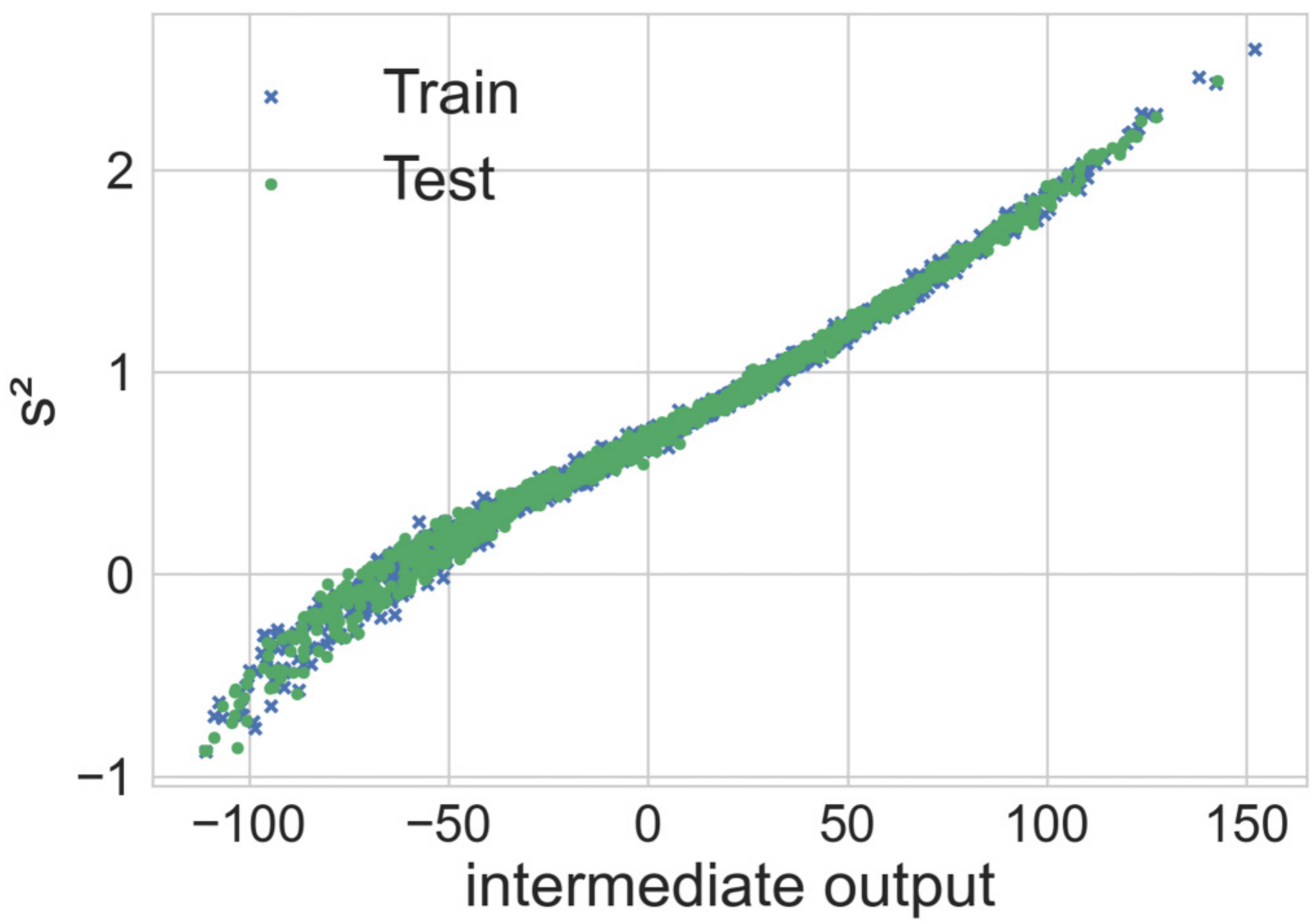
Iten et al., PRL 124 (2020);
review on AI physicists: Wu & Tegmark, PRE 100 (2019)

Bottleneck-based approach

Siamese neural network: classify same vs different







Polynomial regression on bottleneck layer

Support vector machines

Support vector machines

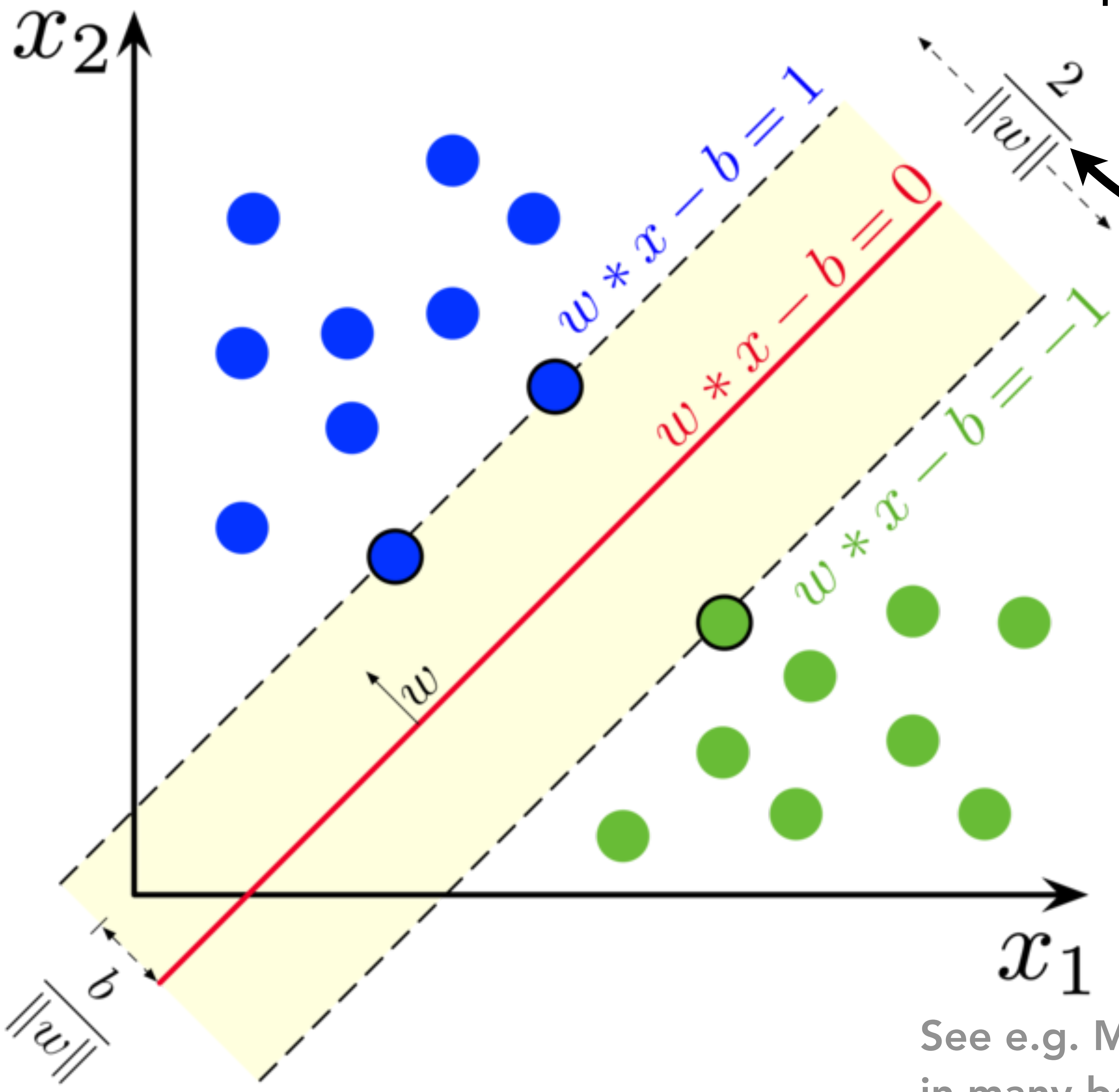


Image: wikipedia

Want to minimize (optimization with constraints: maximize margin while classifying correctly):

$$L = \frac{1}{2}|\boldsymbol{\theta}|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i + \theta_0 \right) - 1 \right]$$

Inverse margin (want to maximize margin)

Lagrange multipliers ($\alpha_i \neq 0$ for support vectors)

$$\alpha_i \left[y_i \left(\boldsymbol{\theta}^\intercal \boldsymbol{x}_i + \theta_0 \right) - 1 \right] = 0 \quad \forall\, i = 1, \ldots, n\,.$$

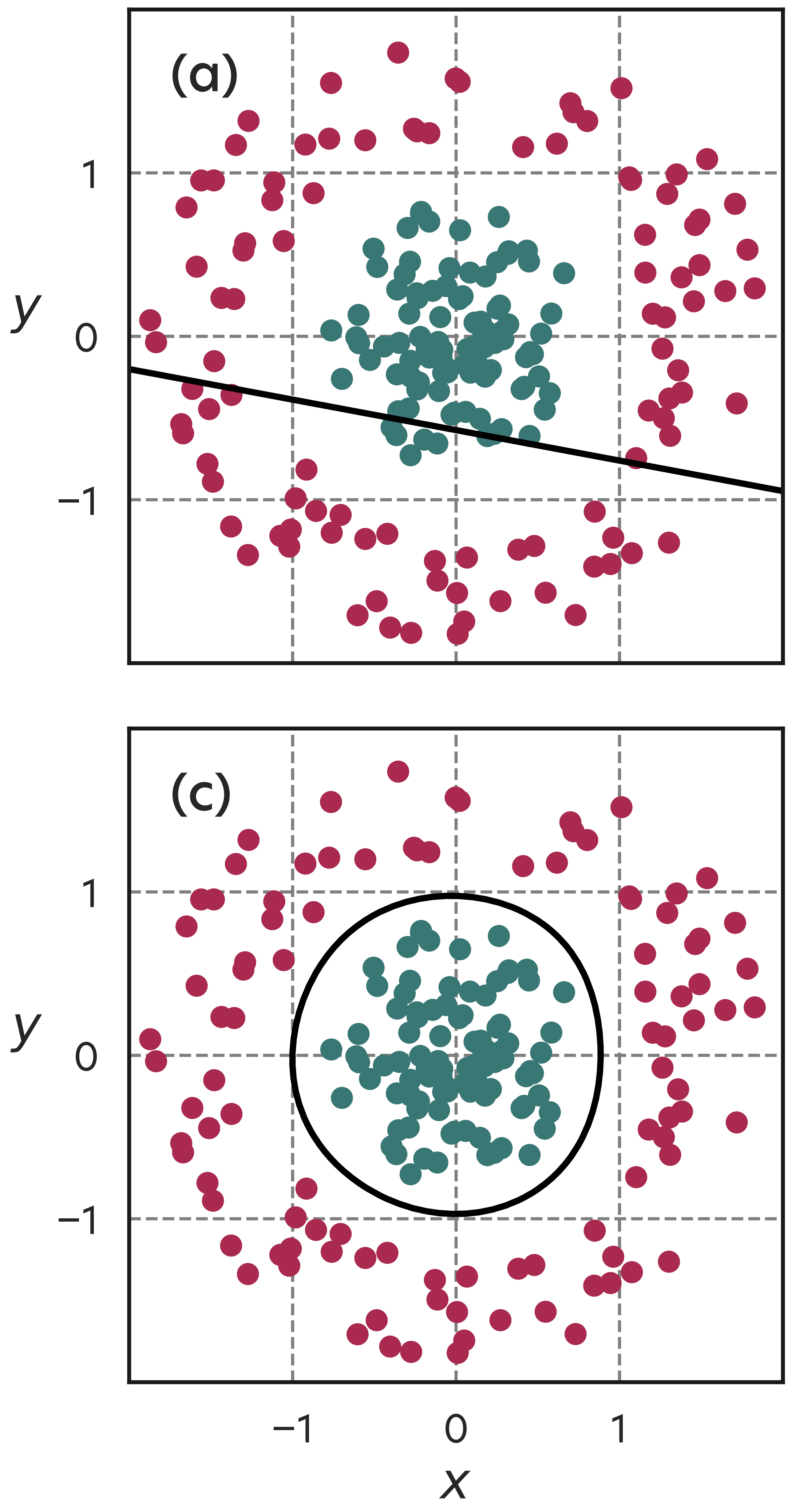
Dual formulation:

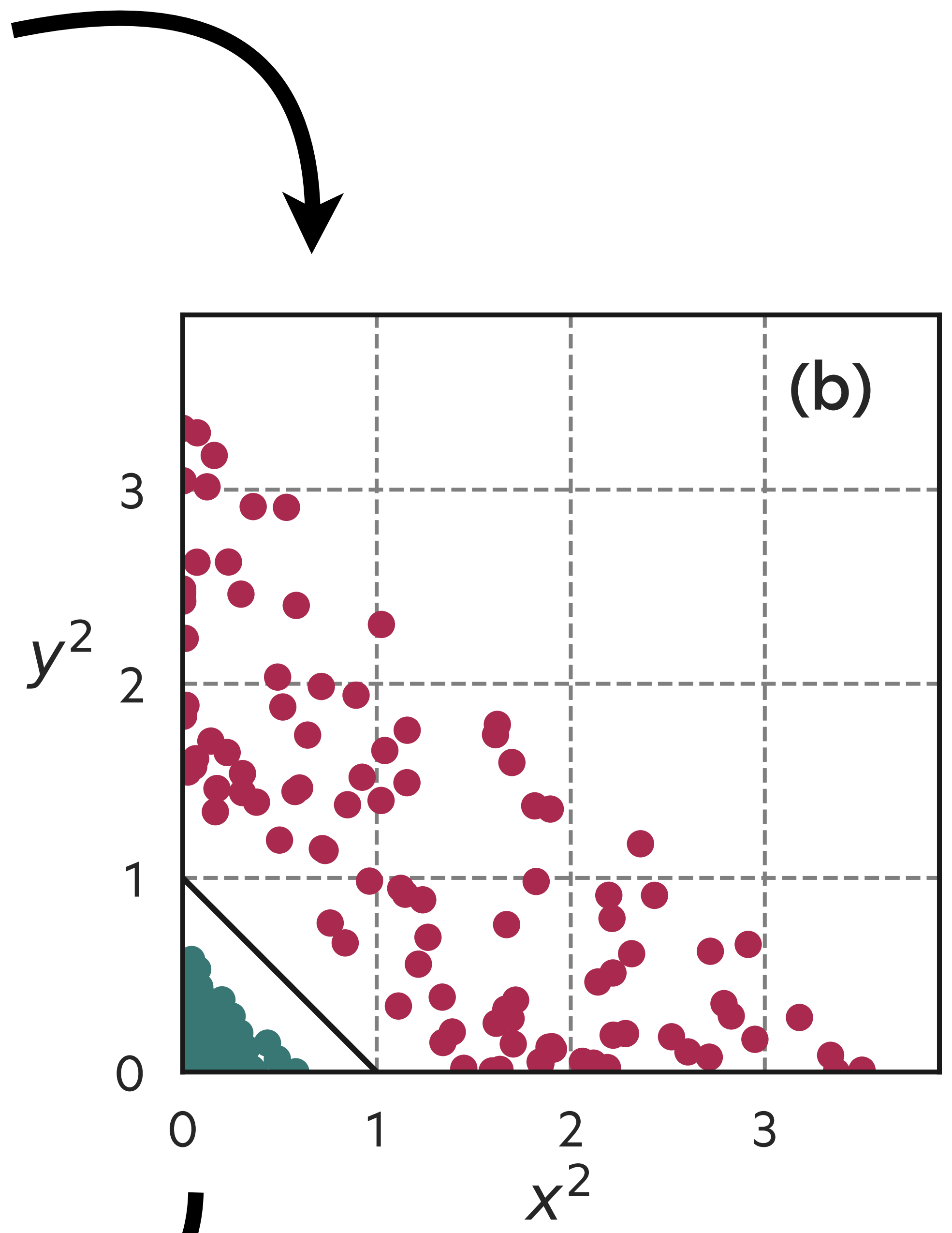
$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\intercal \boldsymbol{x}_j \quad \text{subject to } \alpha_i \geq 0\,.$$

See e.g. Murphy, Machine Learning: A Probabilistic Perspective (MIT Press, Cambridge, 2012)
in many-body physics: Ponte & Melko, PRB 96 (2017); review: Dawid et al., arXiv:2204.04198

Annabelle Bohrdt

Support vector machines





Want to optimize:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_j \quad \text{subject to } \alpha_i \geq 0\,.$$

Introducing a feature map (Kernel trick):

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left\langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \right\rangle$$

$$\phi(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$$

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} + ||\mathbf{x}||^2 ||\mathbf{y}||^2$$

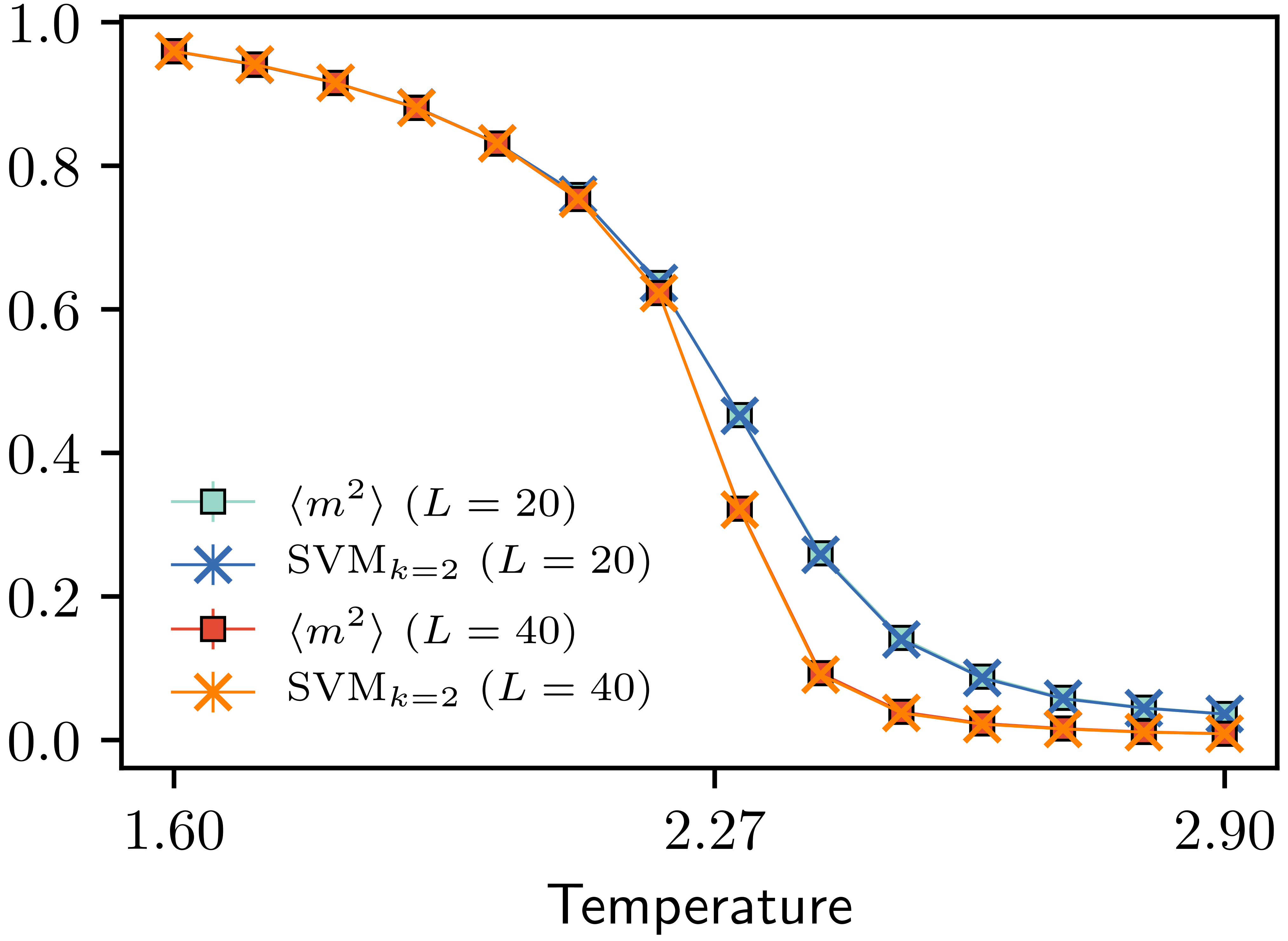
More applications: Greitemann et al., PRB 99 (2019); Liu et al., PRR 3 (2021), …

See e.g. Murphy, Machine Learning: A Probabilistic Perspective (MIT Press, Cambridge, 2012)
in many-body physics: Ponte & Melko, PRB 96 (2017); review: Dawid et al., arXiv:2204.04198

Image: wikipedia

SVM: example

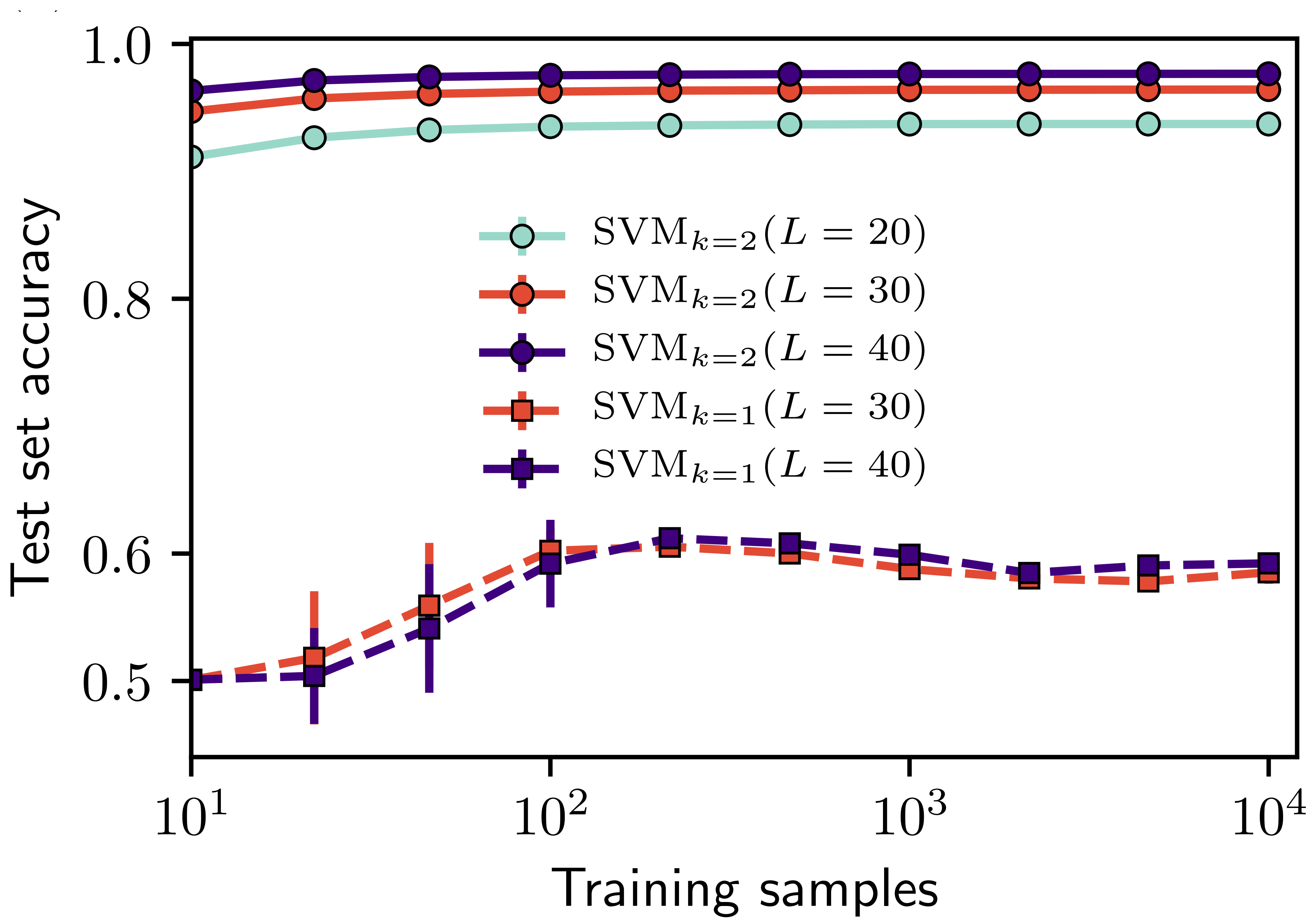
SVM decision function vs squared magnetization



Ponte & Melko, PRB 96 (2017)

Spin configurations 2D Ising model

Polynomial kernel $K(\sigma, \sigma') = (\sigma \cdot \sigma')^k$



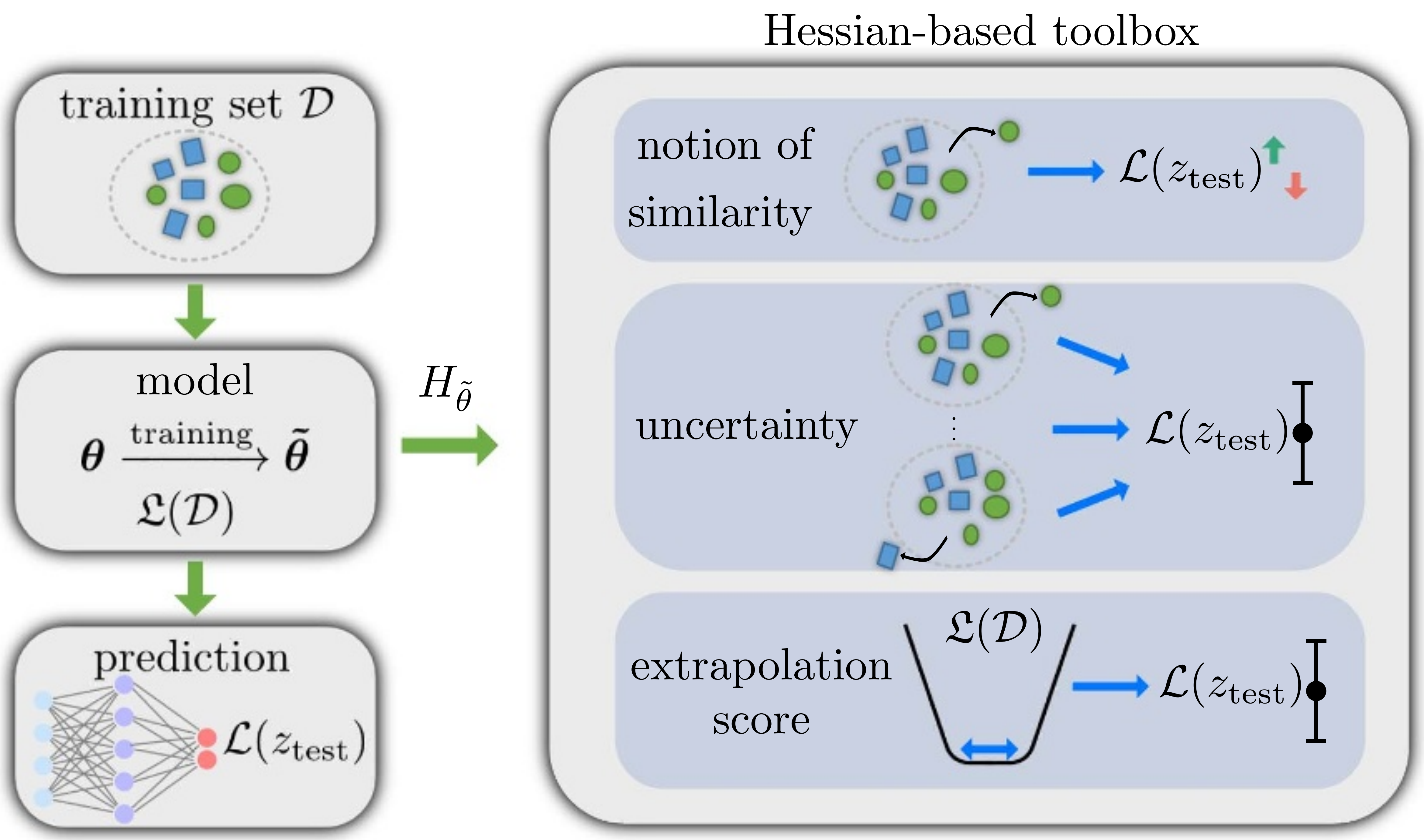
Linear decision boundary (k=1) not enough to distinguish FM (m=+-1) and paramagnet (m=0)

Hessian-based approach

Hessian-based approach

- Independent of ML architecture
- Look at curvature around minimum of the training loss $(H_{\tilde{\boldsymbol{\theta}}})_{ij} = \partial^2_{\theta_i \theta_j} \mathfrak{L}(\mathcal{D}, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}$

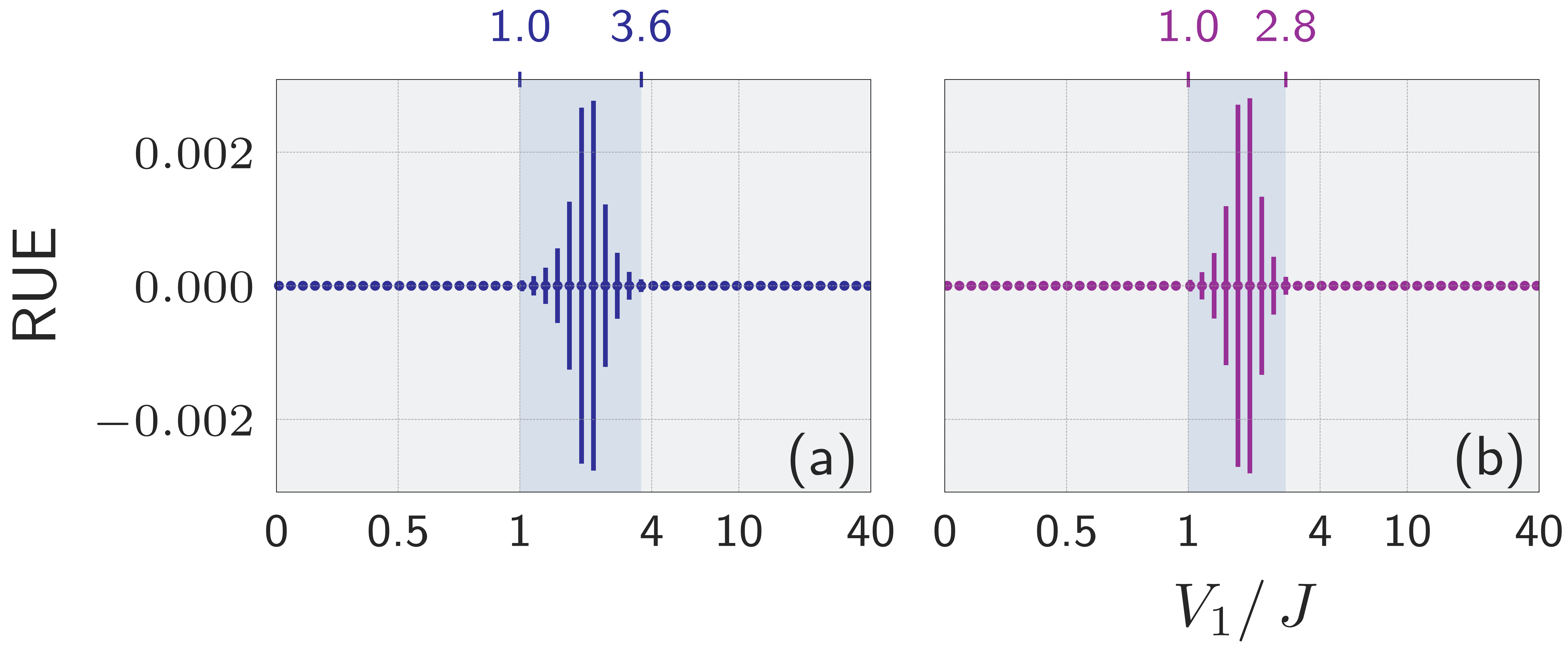
- Influence function: indicates which training points are influential for a chosen (test) prediction. Analysis of the most influential examples can reveal the characteristics which impacts the machine learning predictions.
- Resampling uncertainty estimation: check whether there are training samples similar to test sample & how big/small errors are on this training data
- Extrapolation: explore flat basin around minimum, make predictions for the same test point, calculate variance of the test loss



Dawid et al., Mach. Learn.: Sci. Technol 3 (2022)

Hessian-based approach

Phase transition less/more sharp for L=12/14



Dawid et al., Mach. Learn.: Sci. Technol 3 (2022)

Correlator convolutional neural network

Setting the stage: quantum simulation

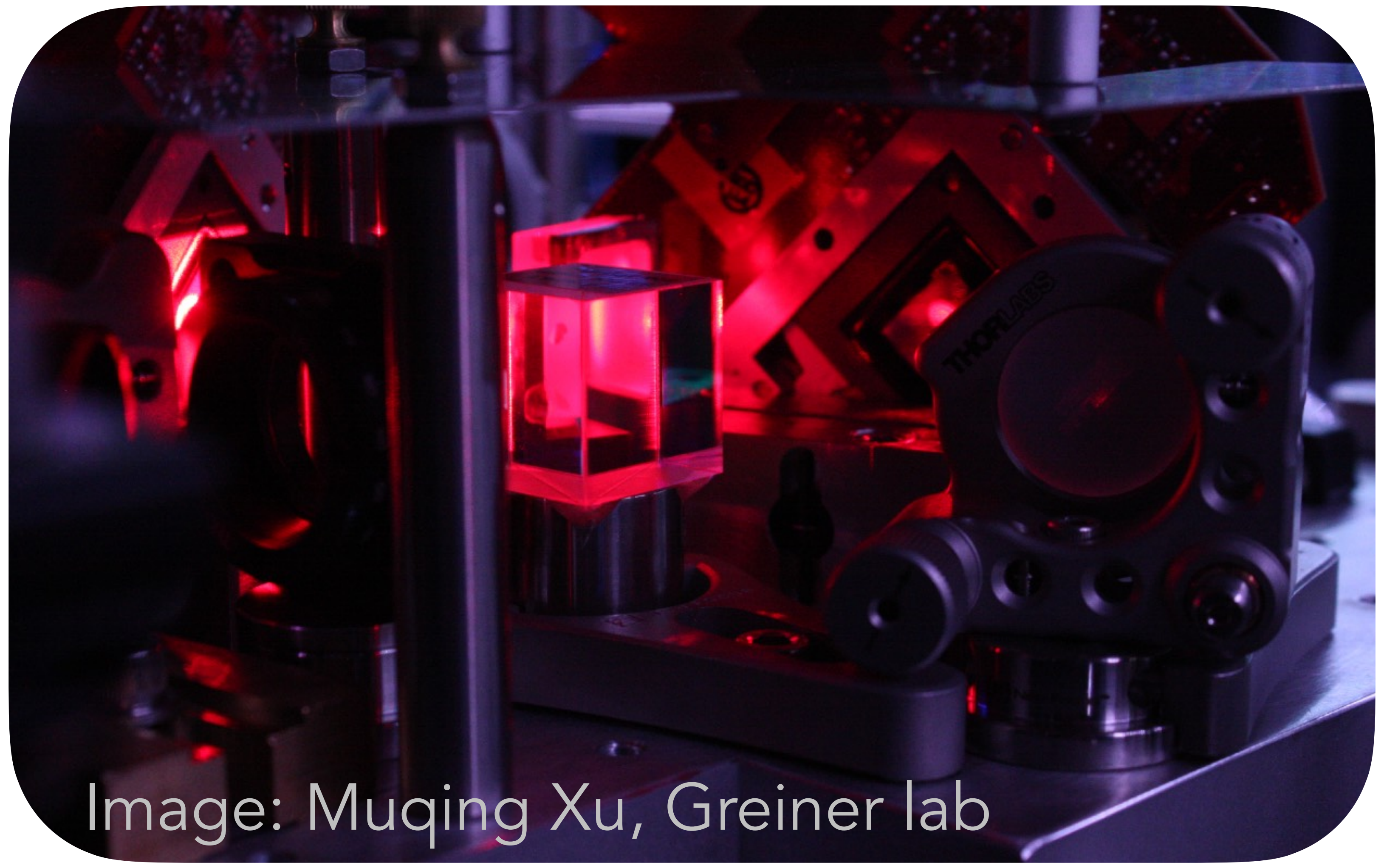


Image: Muqing Xu, Greiner lab

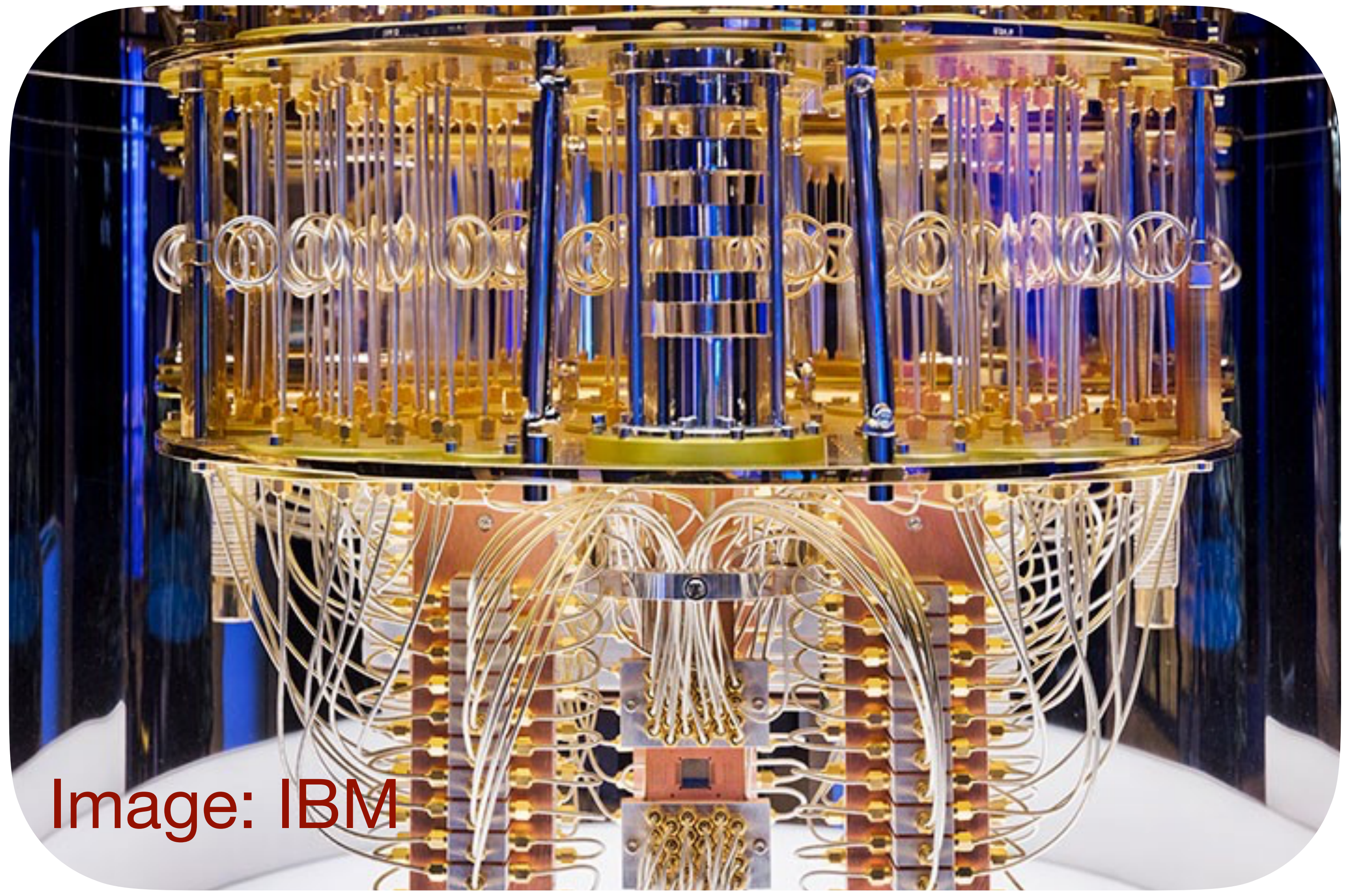


Image: IBM

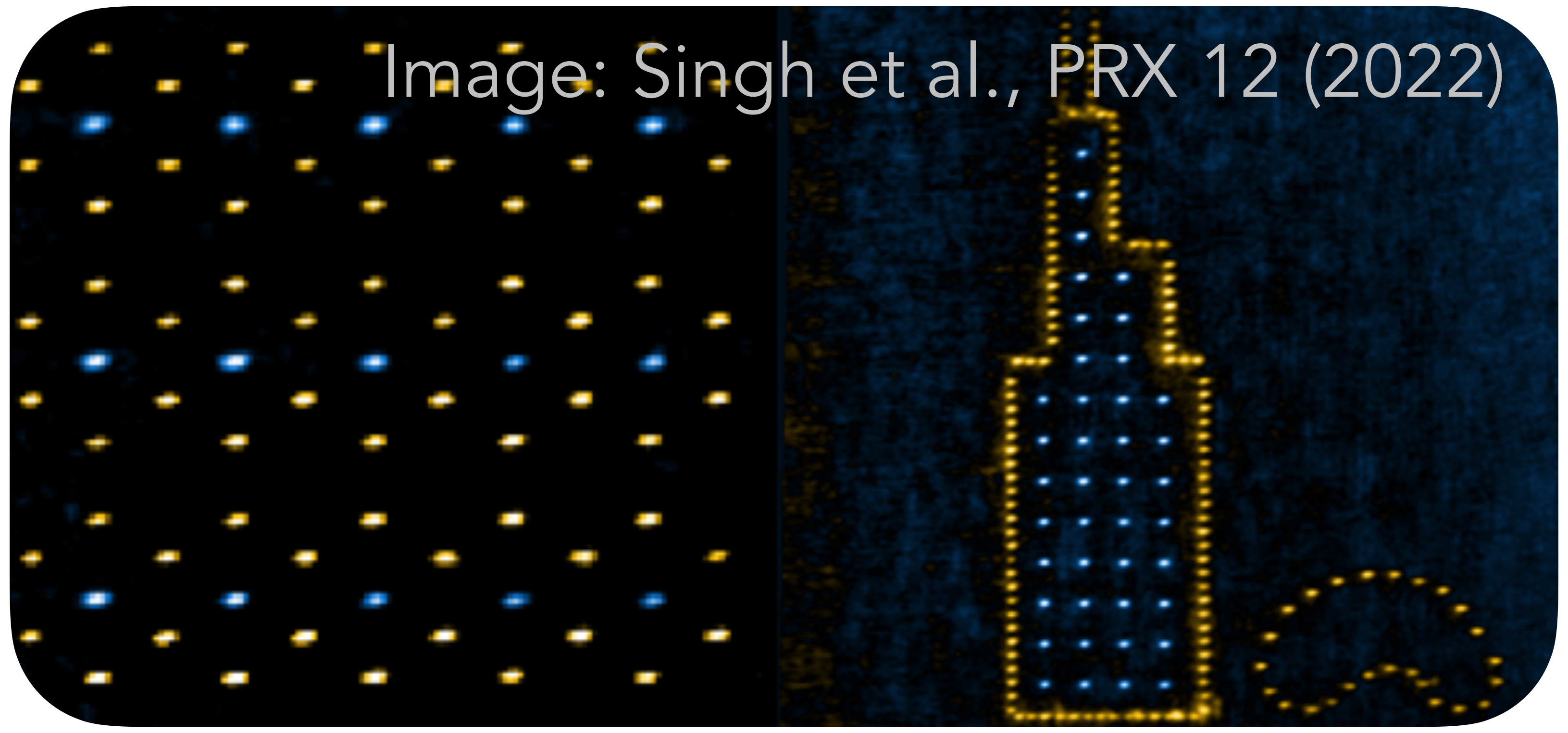


Image: Singh et al., PRX 12 (2022)

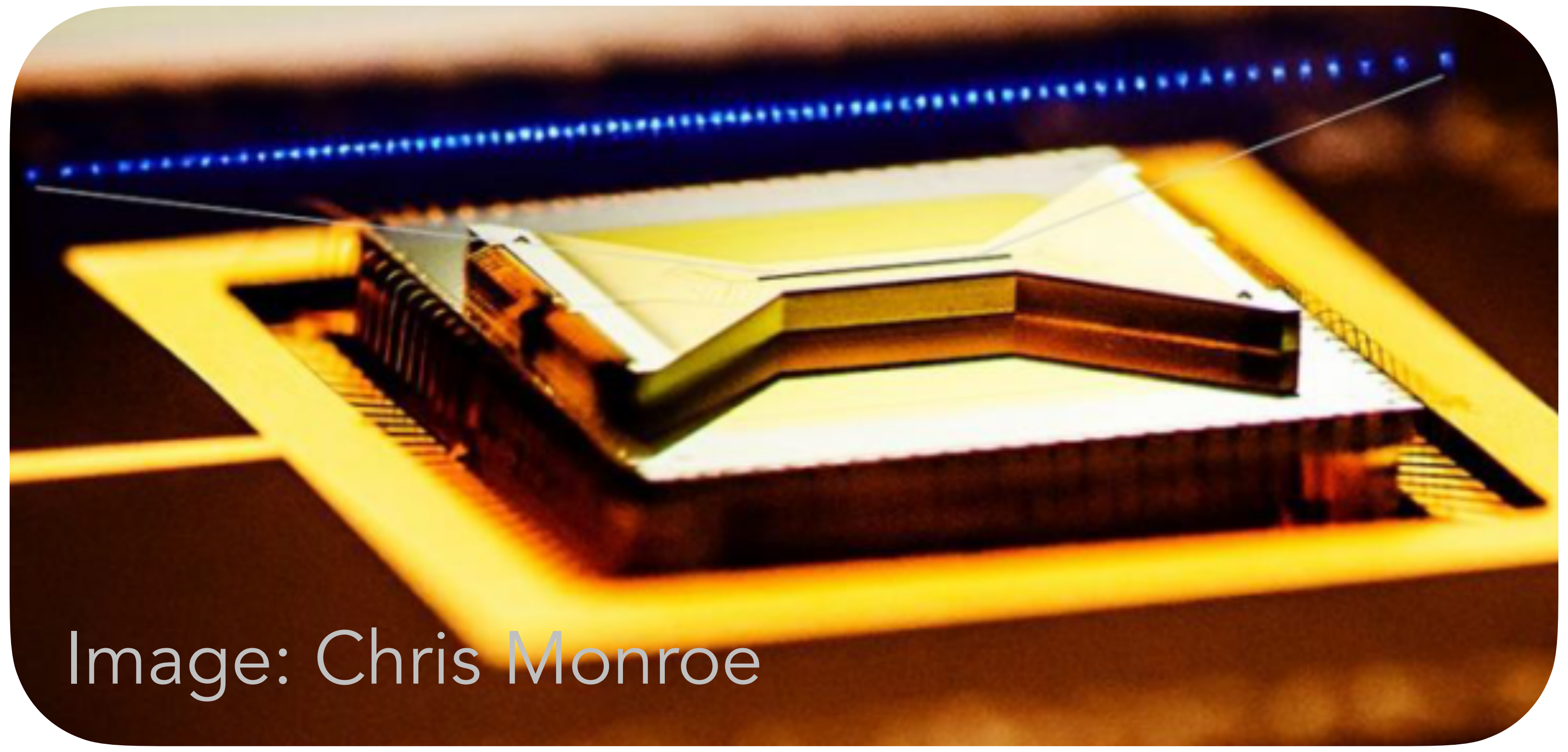
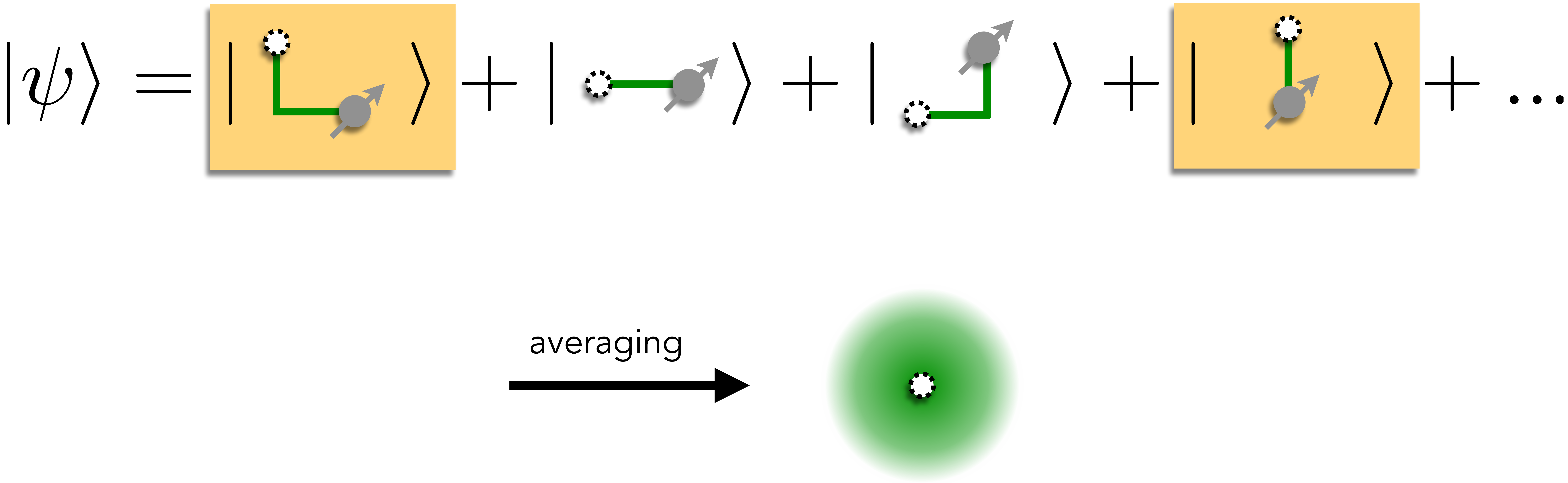


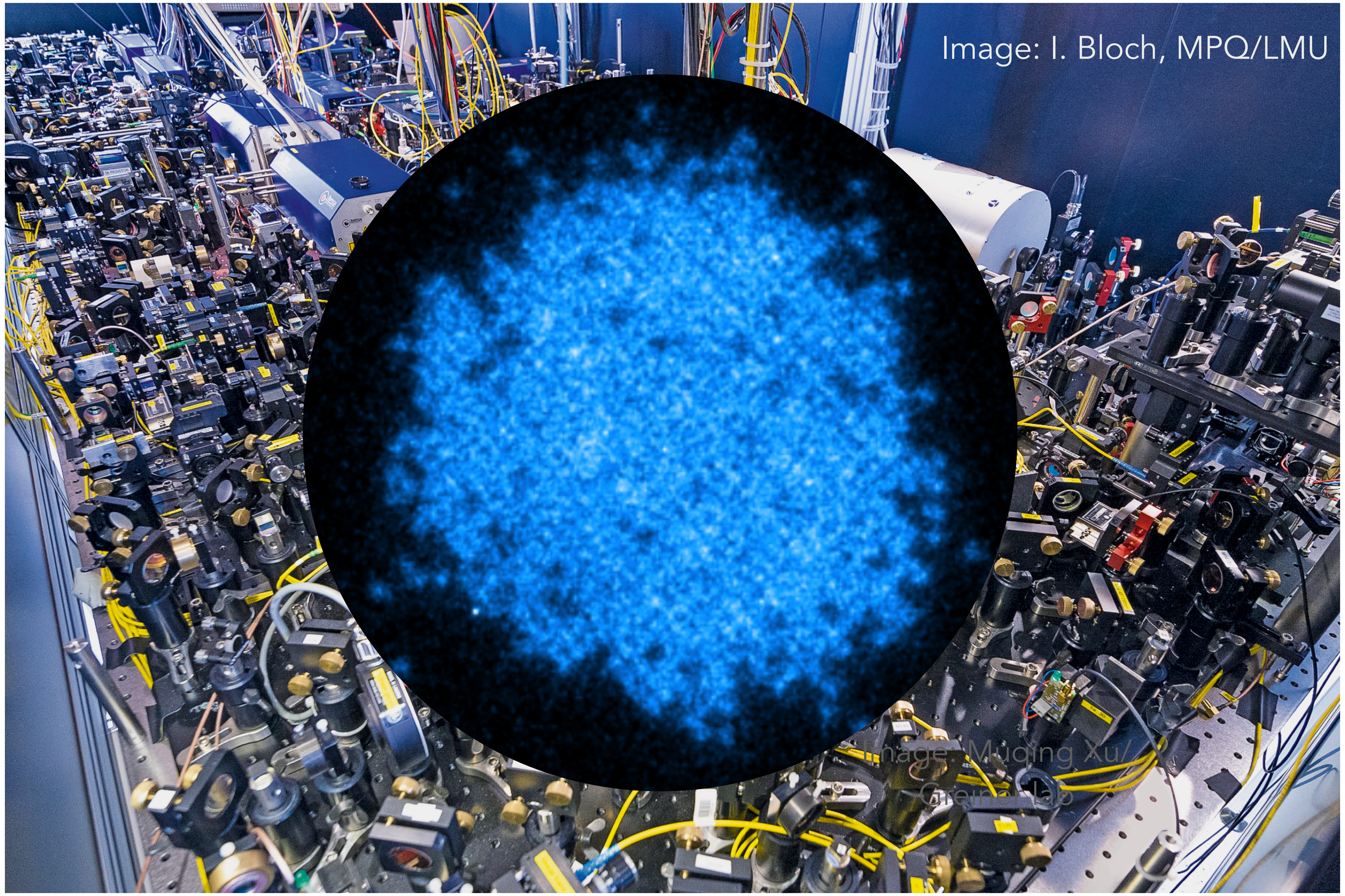
Image: Chris Monroe

Annabelle Bohrdt

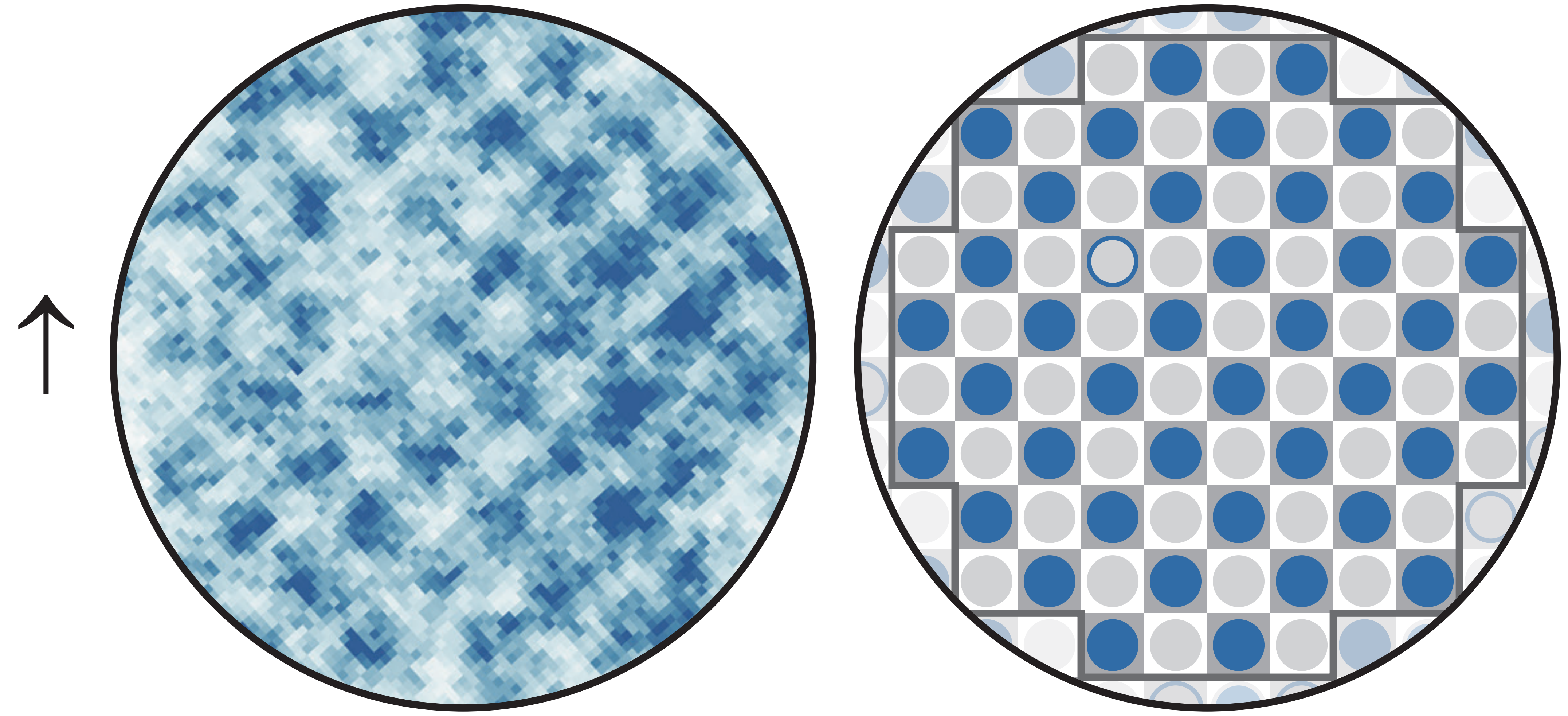
Setting the stage: Projective measurements



Setting the stage: Projective measurements

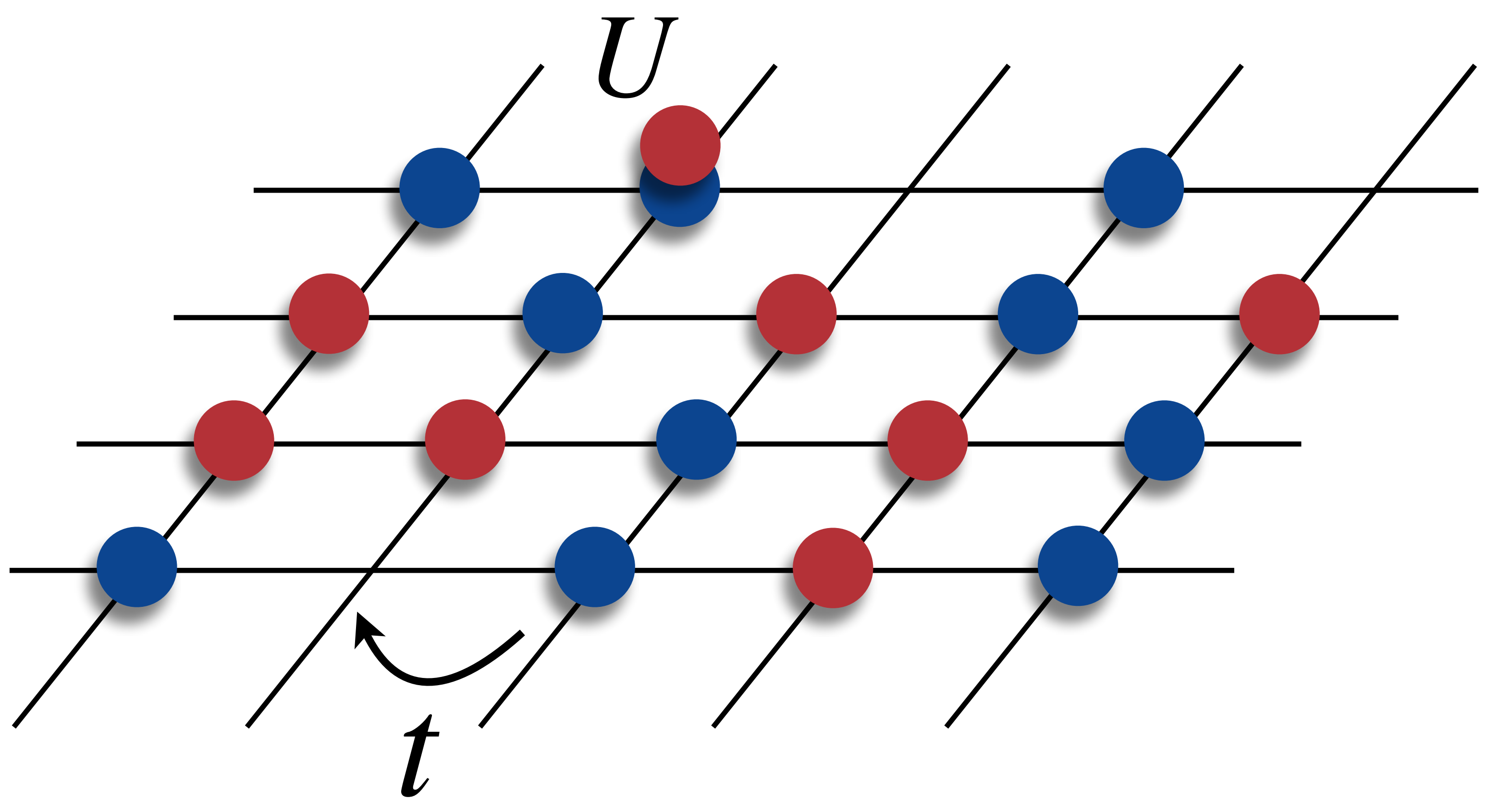


Setting the stage: Projective measurements



Mazurenko et al., Nature 545 (2017)

Setting the stage: Fermi-Hubbard model



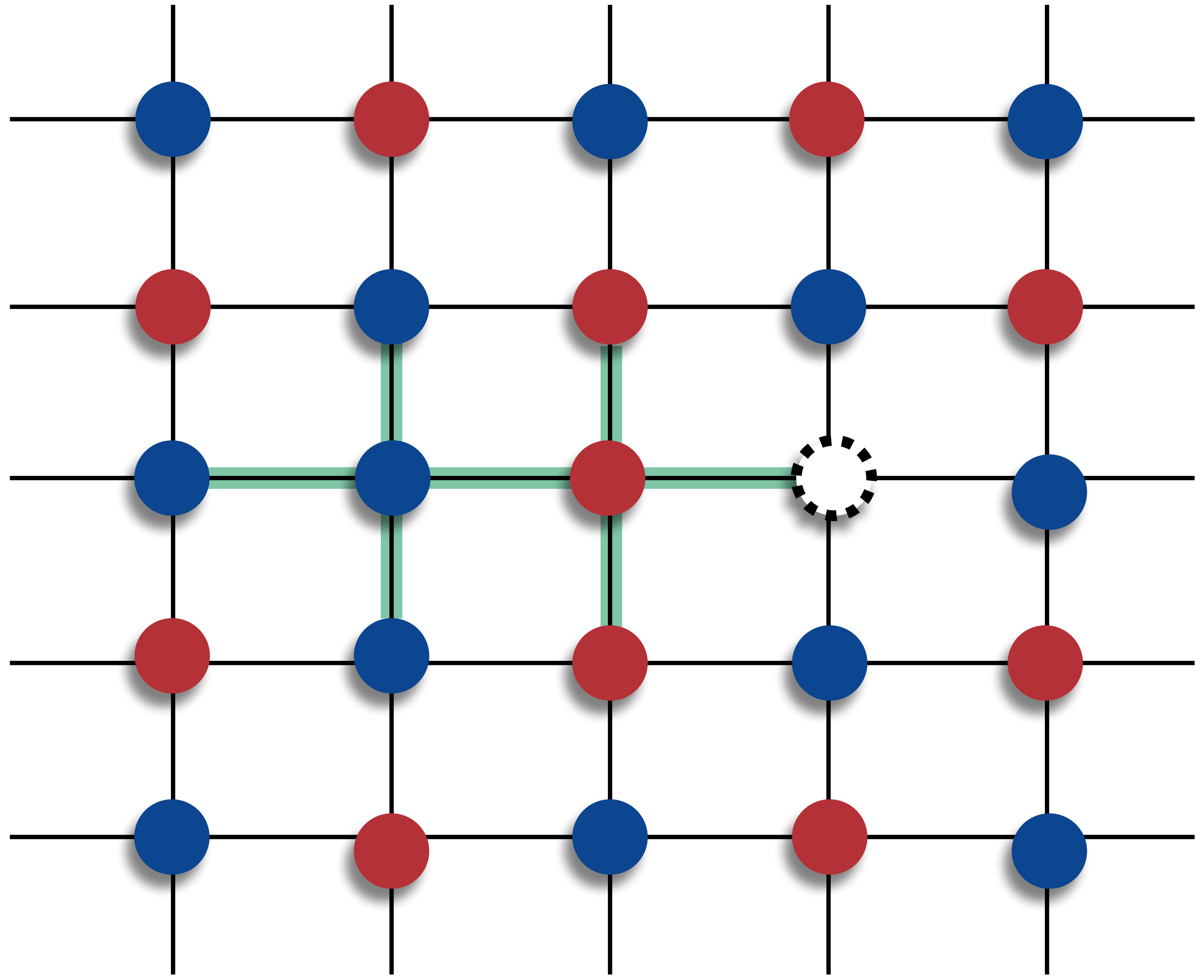
$$\hat{H} = -t \sum_{\langle i,j \rangle, \sigma} \left(\hat{c}^{\dagger}_{i,\sigma} \hat{c}_{j,\sigma} + h.c. \right) + U \sum_i \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow}$$

tunneling

interaction

Setting the stage: Candidate theories

Geometric string theory

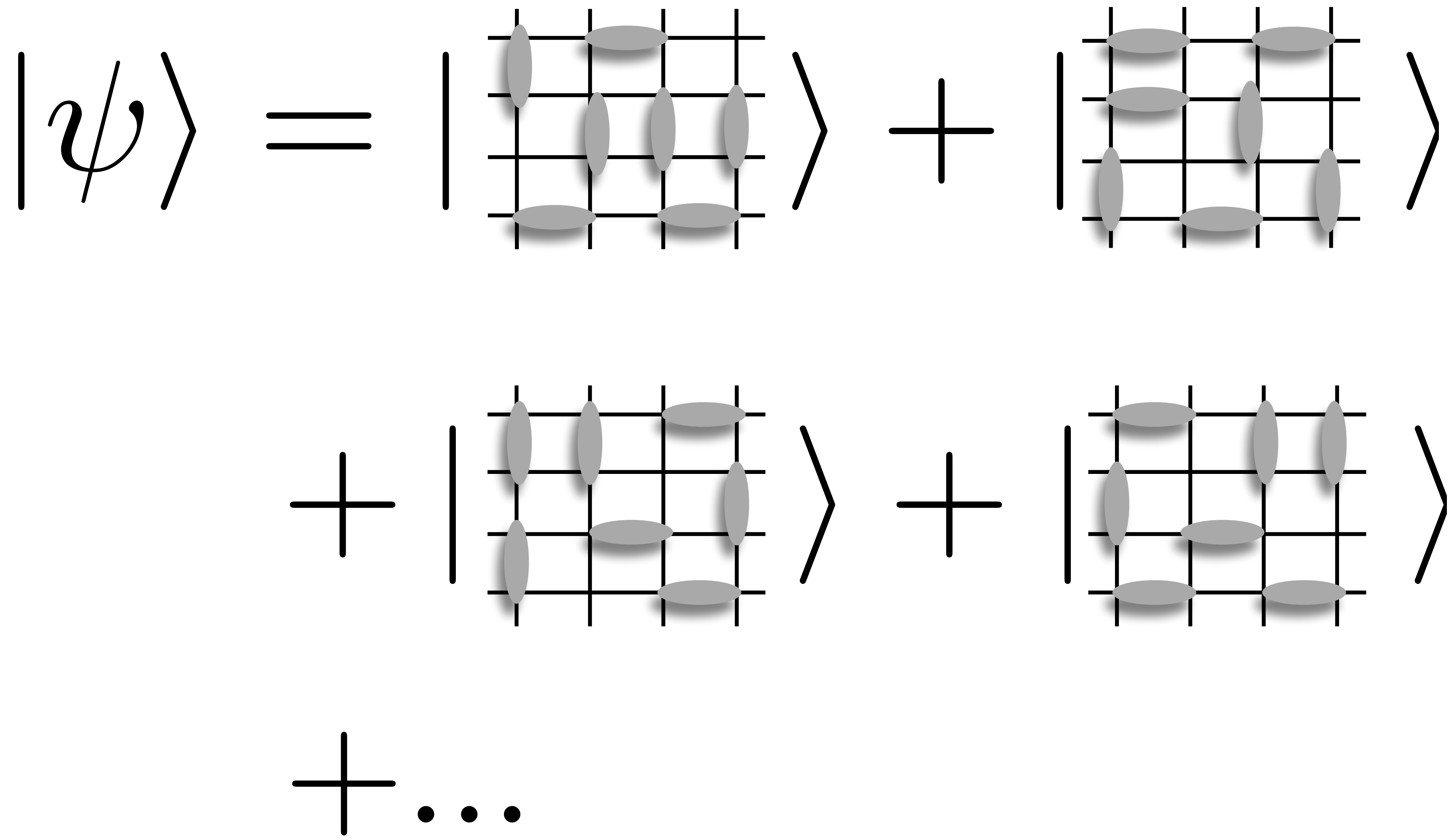


Grusdt et al., PRX 8 (2018), Grusdt et al., PRB 99 (2019), Bohrdt et al., NJP 22 (2020)

Early work: Bulaevskii et al., JETP 27 (1968), Trugman, PRB 37 (1988), Manousakis, PRB 75 (2007)

Resonating valence bond theory

(π-flux theory)

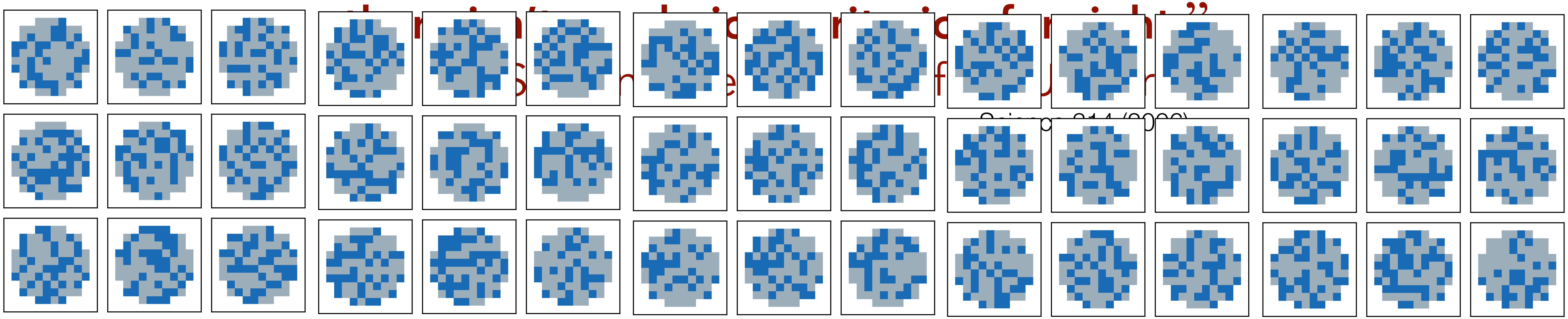


P. W. Anderson, Science 235, 1196 (1987)

Annabelle Bohrdt

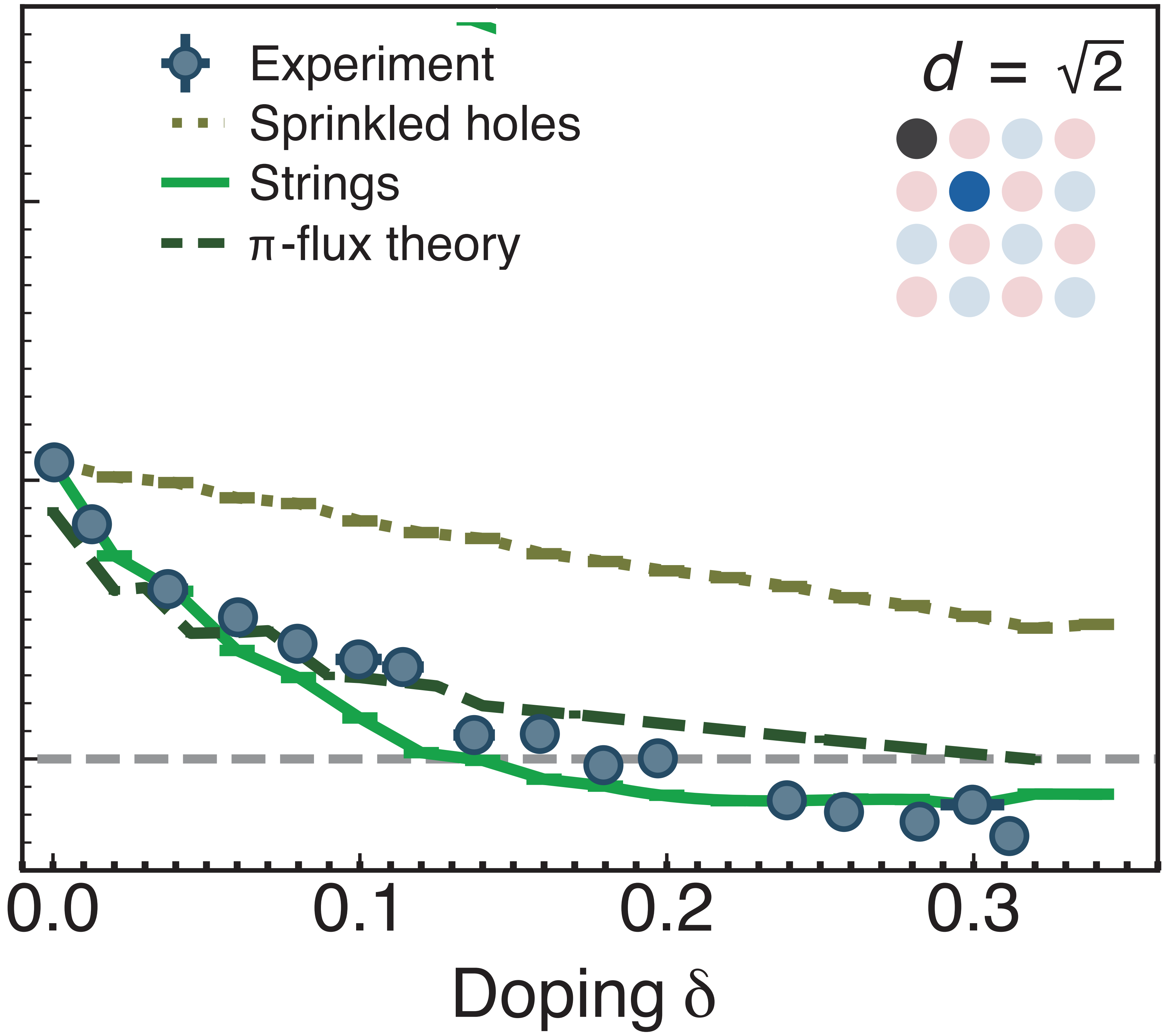
Setting the stage: snapshots

“The theoretical problem is so hard that

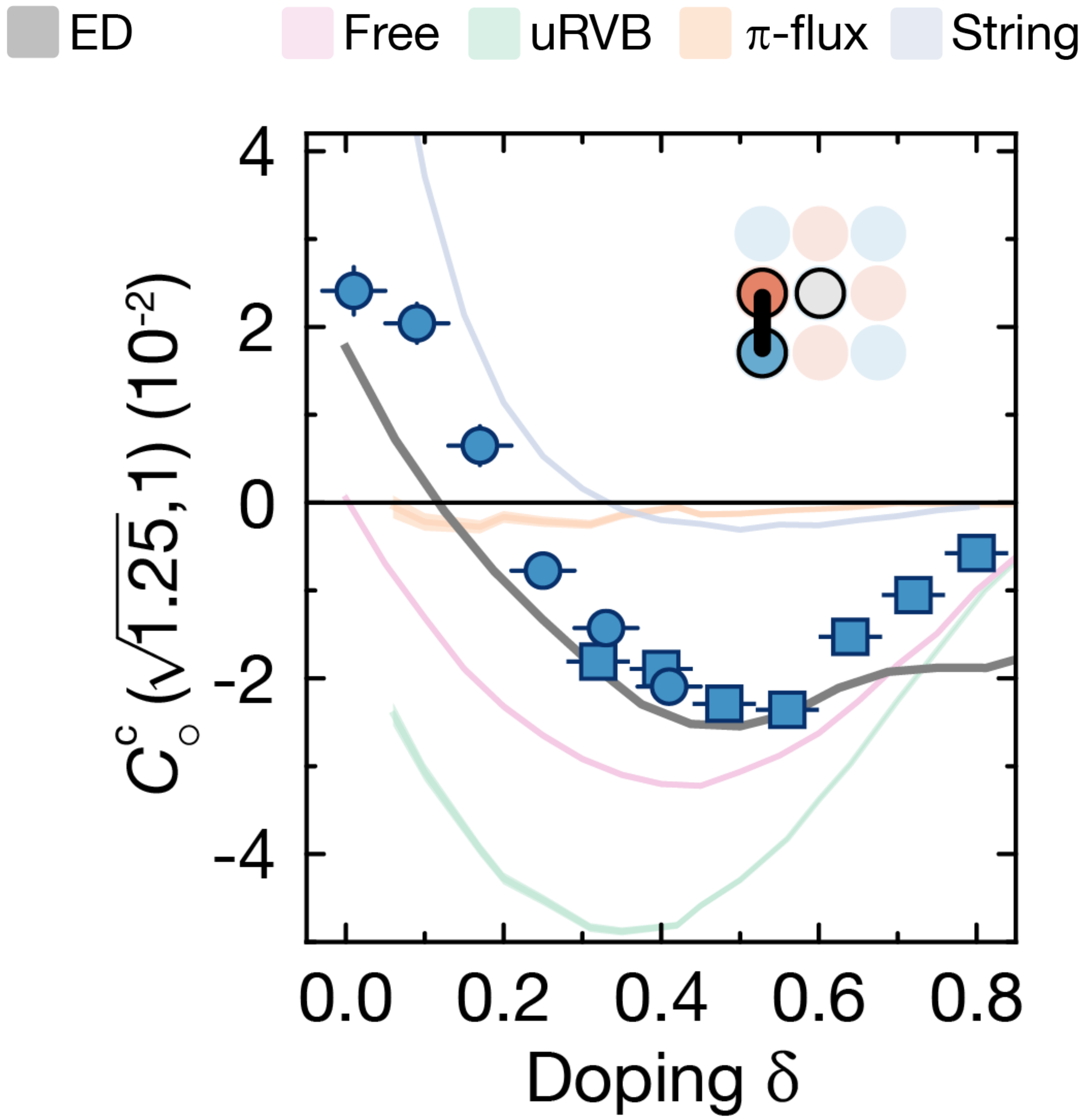


A lot of information about the quantum state is contained in these snapshots!

Setting the stage: correlations



Chiu et al., Science 365, 6450 (2019)



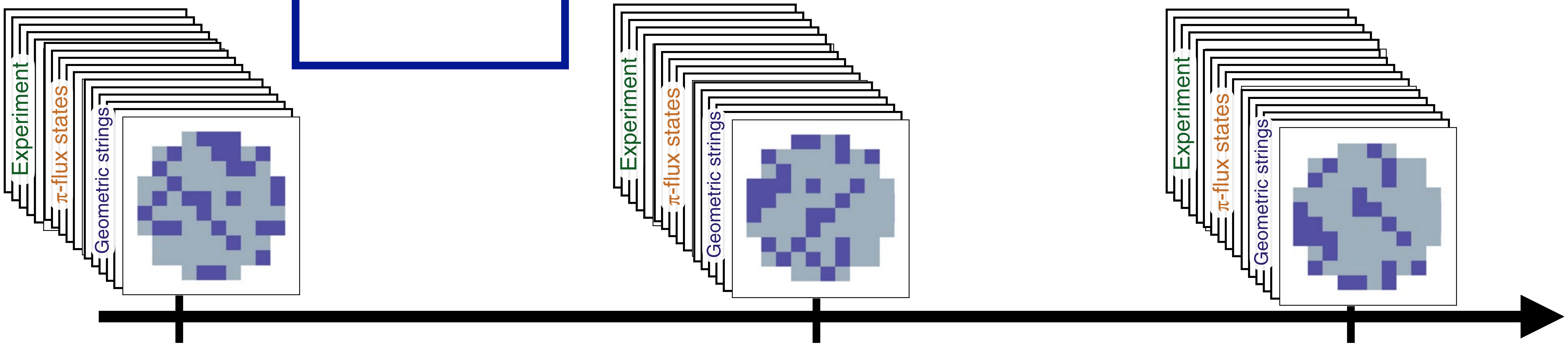
Koepsell et al., Science 374 (2021)

Training a neural network

Annabelle Bohrdt

Training
Classify

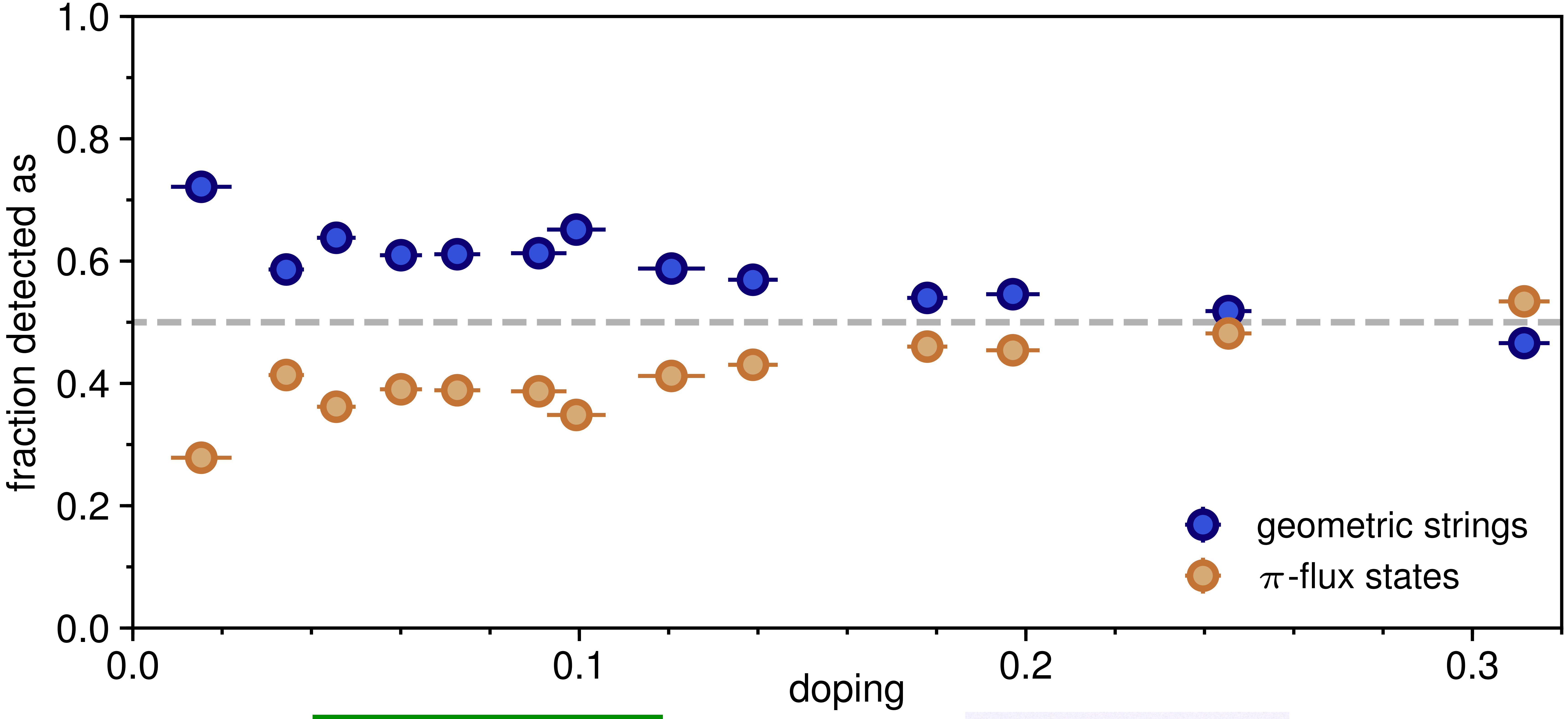
Output



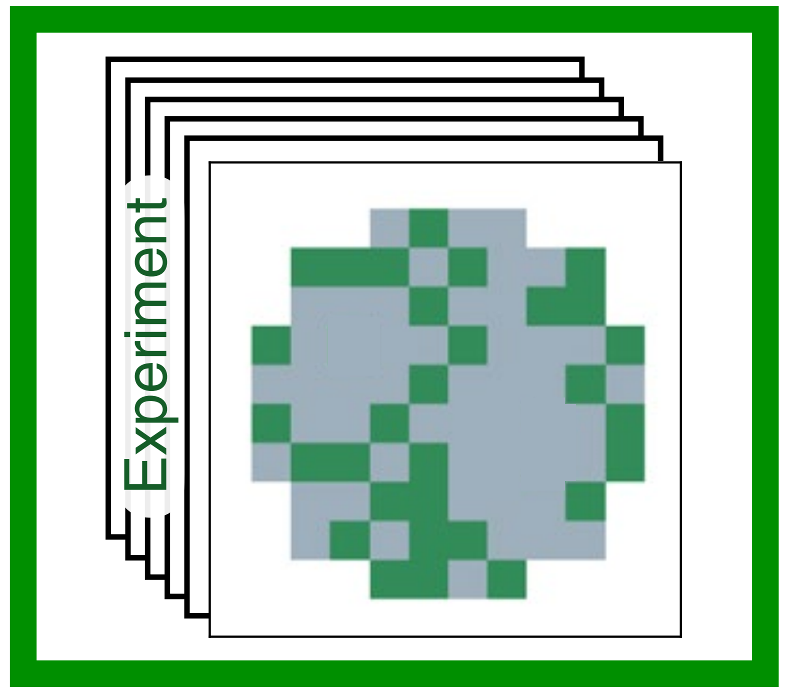
doping

Annabelle Bohrdt

Evaluating experimental data



Classify



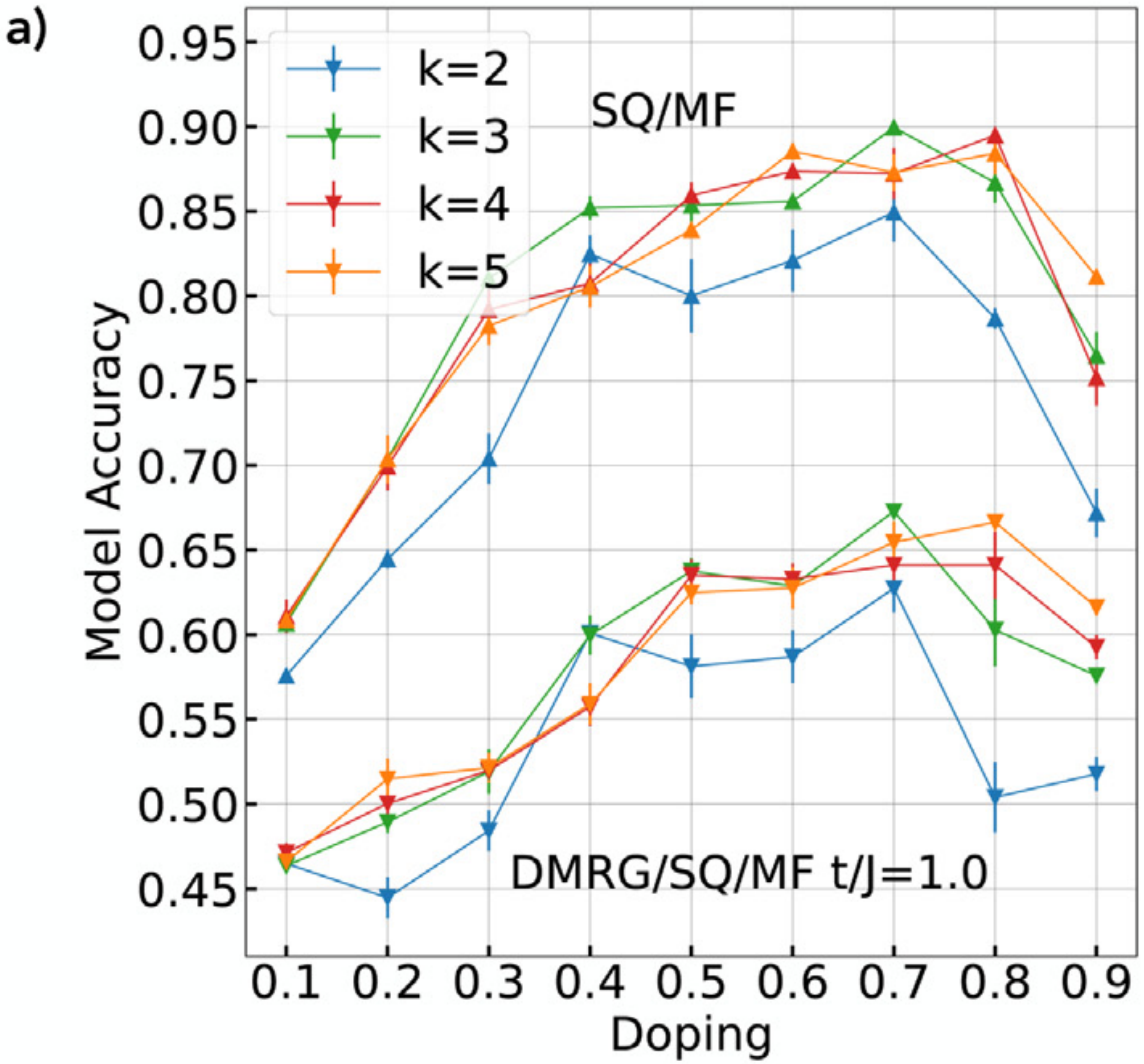
Output

Bohrdt et al., Nature Physics 15 (2019)

Looking at filters

- Descriptions for the 1D t-J model
- Convolutional neural network: varying the filter size

Problem: many correlation functions possible in given filter (in 2D!)

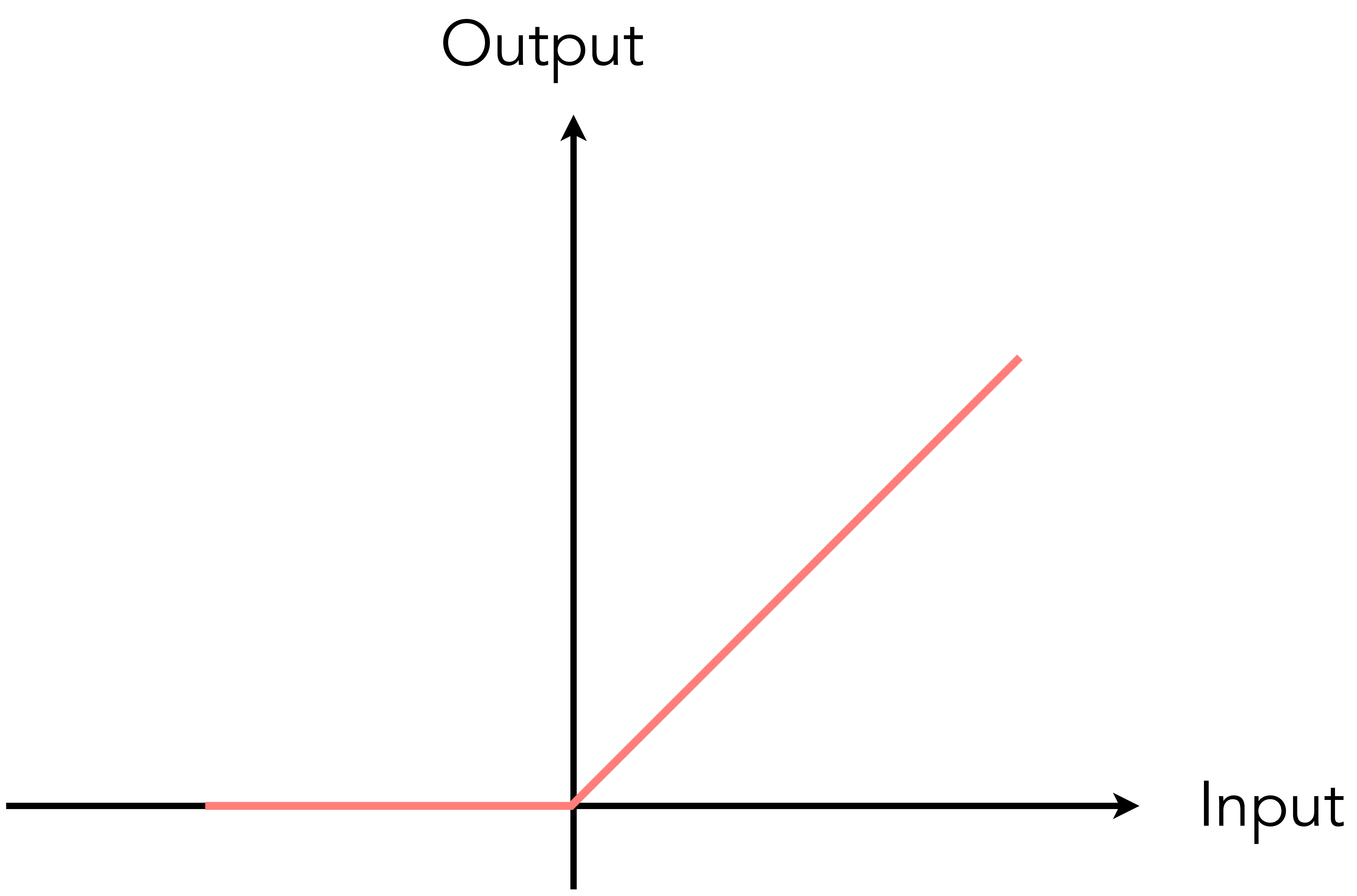


Bohrdt et al., Annals of Physics (2021)

Interpretability?

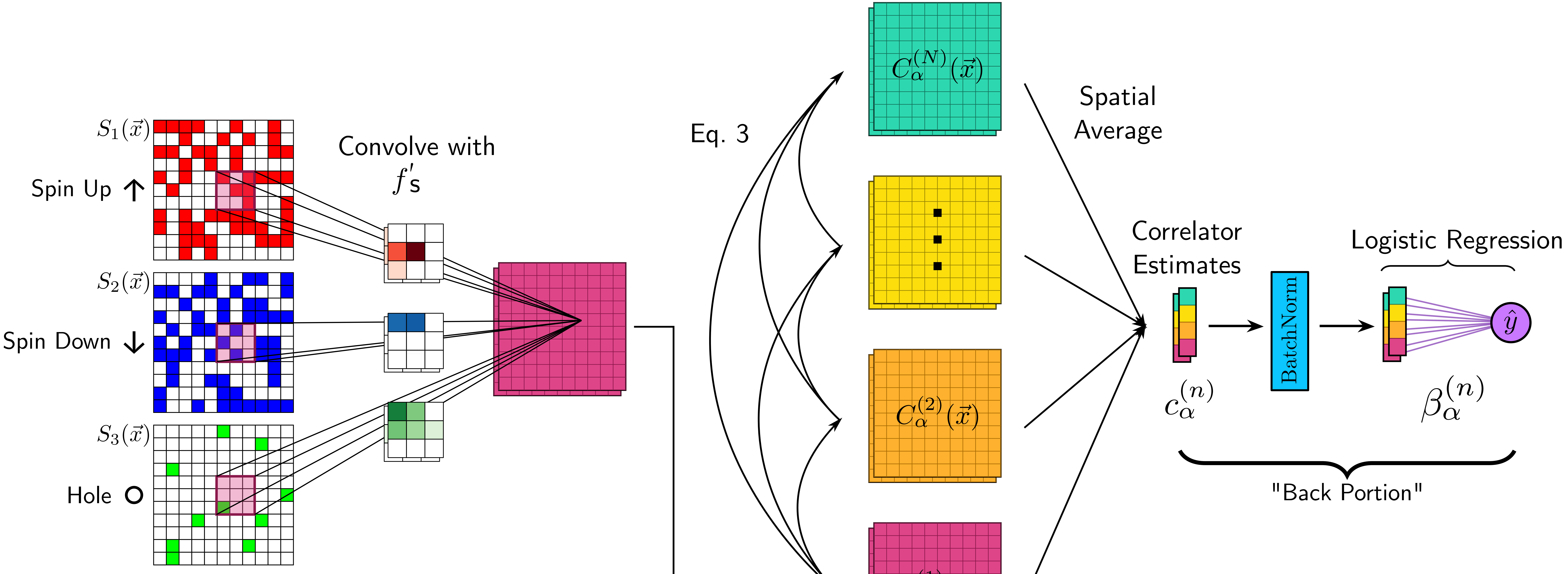
- Looking at filters difficult: non-linearity mixes all orders of correlations
- Non-linearity necessary:
  otherwise, network corresponds to function that multiplies the input by a weight matrix and adds to it an additional bias vector.

Typical activation function: rectified linear unit



Annabelle Bohrdt

Correlator convolutional neural network



$$C_\alpha^{(1)}(\vec{x}) = \sum_{\vec{a},k} f_{\alpha,k}(\vec{a}) S_k(\vec{x}+\vec{a})$$

$$C_\alpha^{(2)}(\vec{x}) = \sum_{(\vec{a},k)\neq(\vec{b},k')} f_{\alpha,k}(\vec{a}) f_{\alpha,k'}(\vec{b}) S_k(\vec{x}+\vec{a}) S_{k'}(\vec{x}+\vec{b})$$

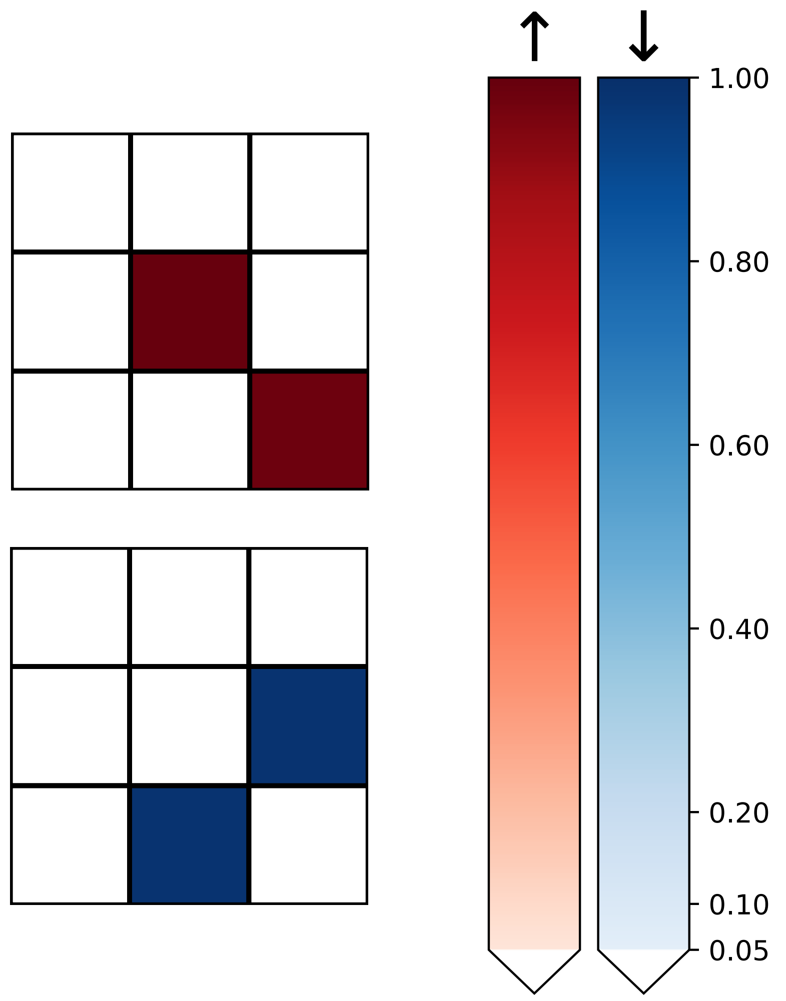
Miles et al., Nature Comm. 12 (2021)

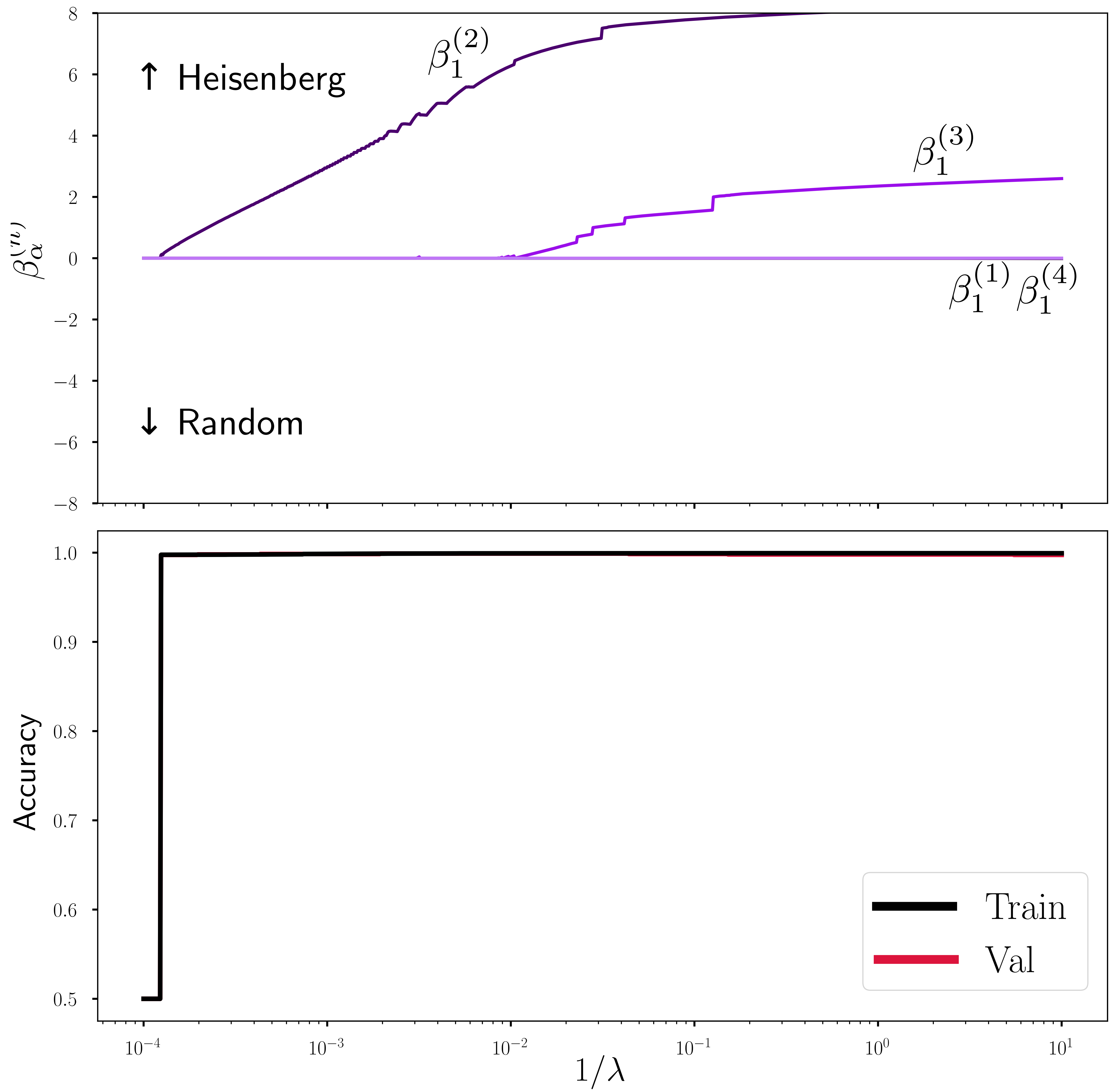
See also: Wetzel & Scherzer, PRB 96 (2017)

Correlator convolutional neural network

With fixed filters, re-train back portion of the network with loss:

$$L_{\text{path}}(y,\hat{y}) \equiv -y\log\hat{y} - (1-y)\log(1-\hat{y}) + \lambda\sum_{\alpha,n}|\beta_\alpha^{(n)}|$$



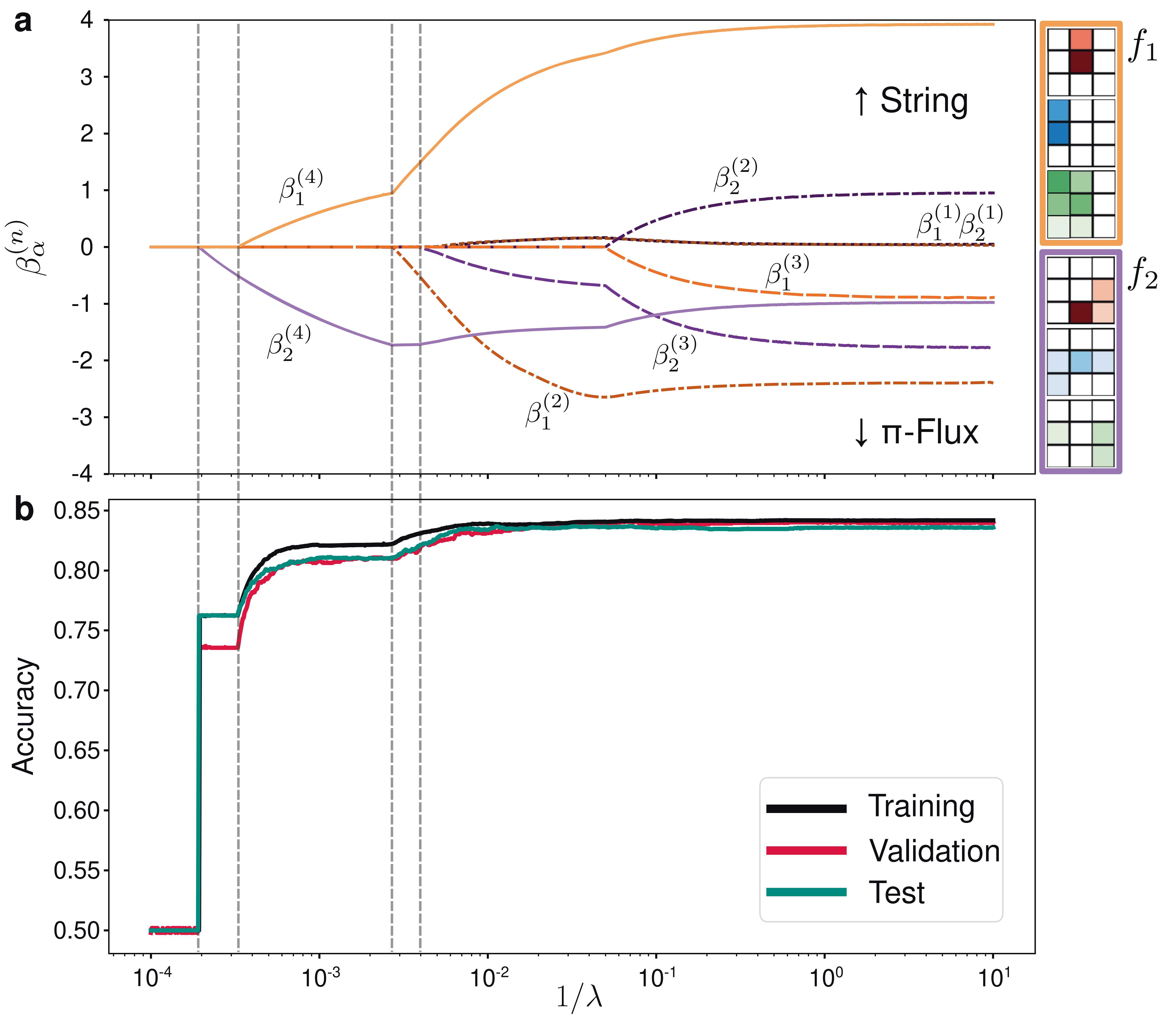


Miles et al., Nature Comm. 12 (2021)

Correlator convolutional neural network

With fixed filters, re-train back portion of the network with loss:

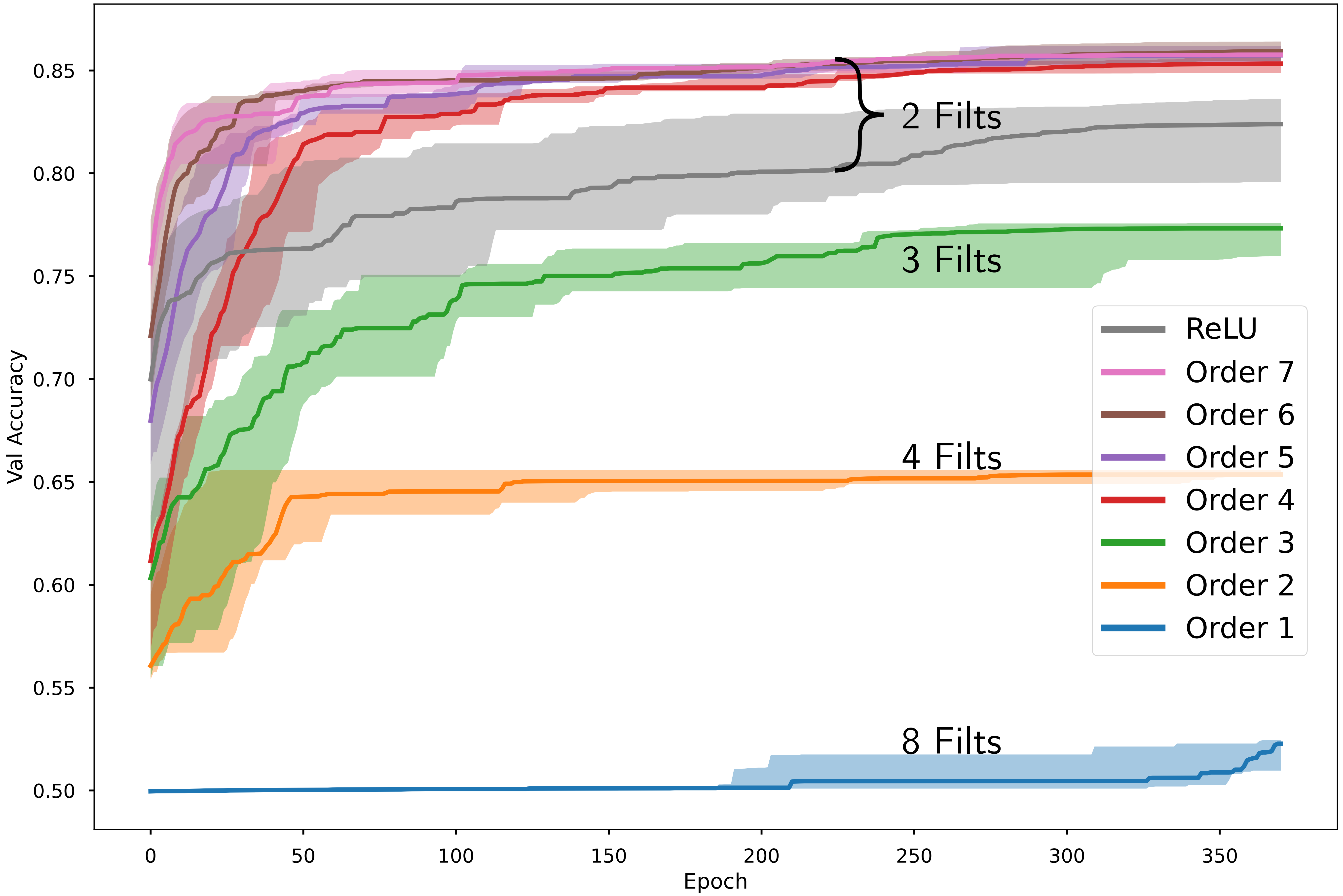
$$L_{\text{path}}(y,\hat{y}) \equiv -y\log\hat{y} - (1-y)\log(1-\hat{y}) + \lambda\sum_{\alpha,n}|\beta_\alpha^{(n)}|$$



Miles et al., Nature Comm. 12 (2021)

Annabelle Bohrdt

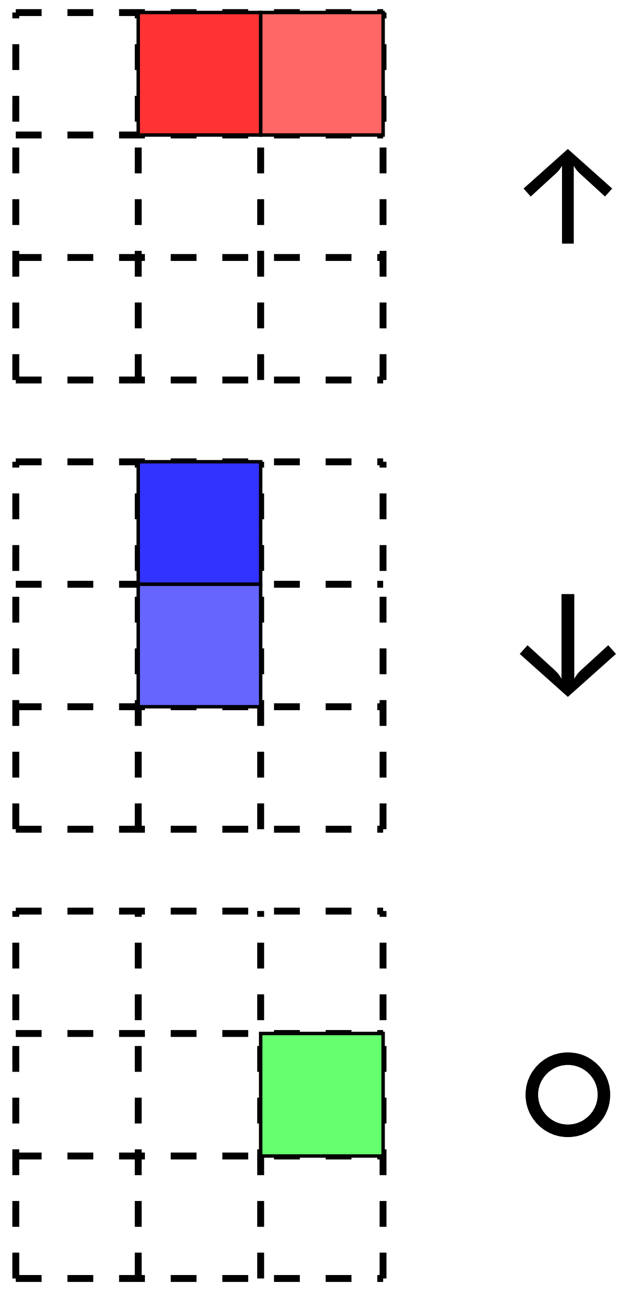
Higher-order correlations

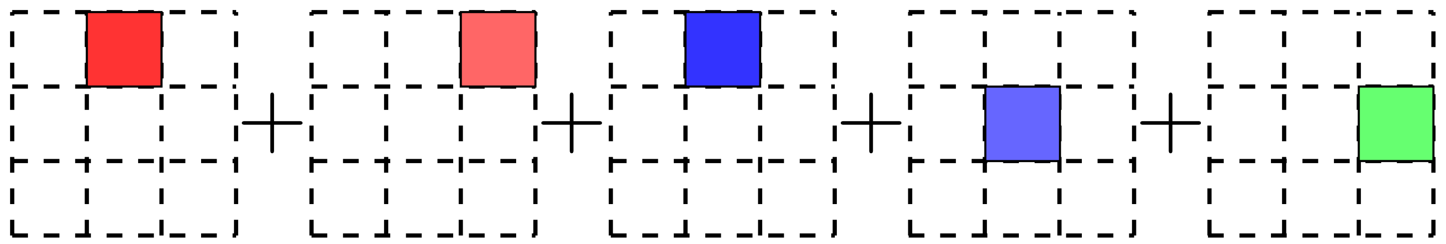


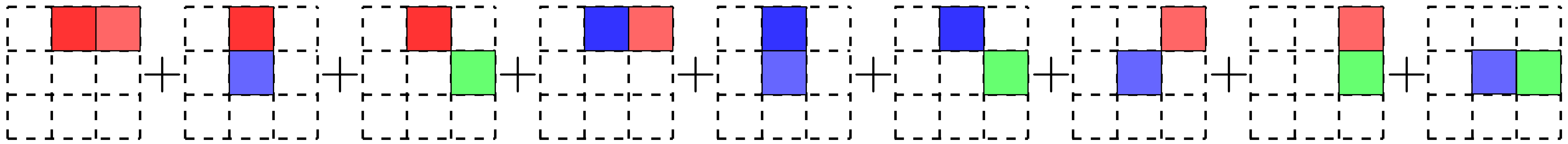
Miles et al., Nature Comm. 12 (2021)

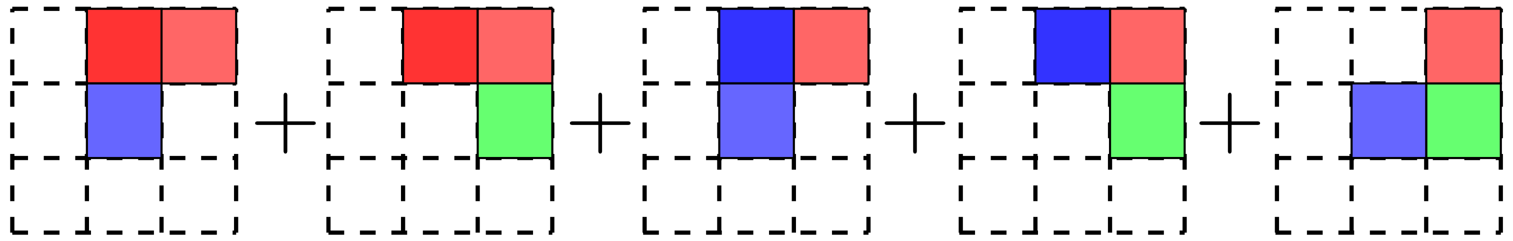
From filter to correlation

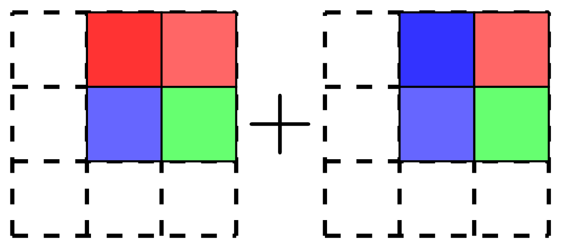
Example Filter



$c^{(1)} =$ 

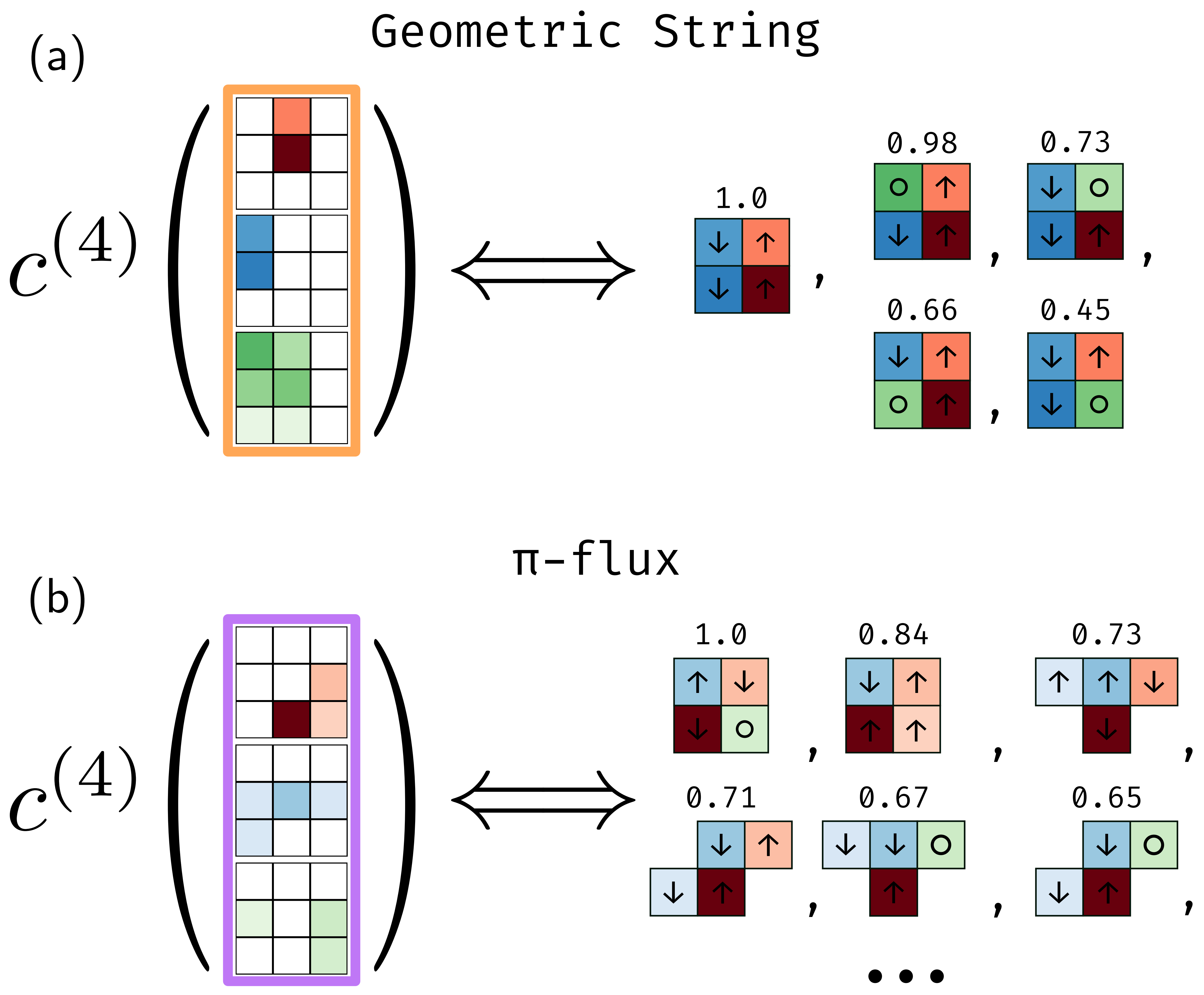
$c^{(2)} =$ 

$c^{(3)} =$ 

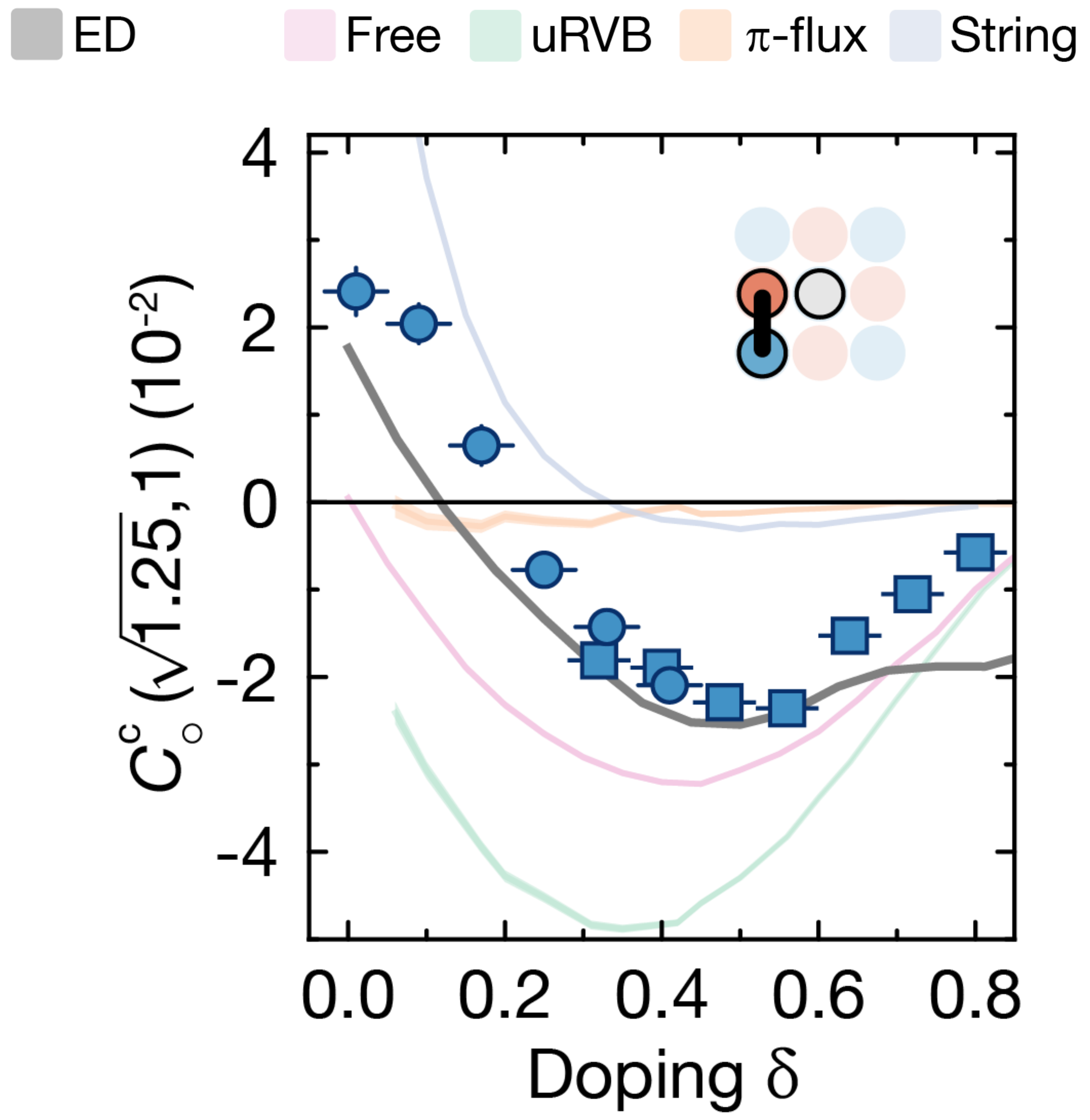
$c^{(4)} =$ 

Miles et al., Nature Comm. 12 (2021)

Higher-order correlations

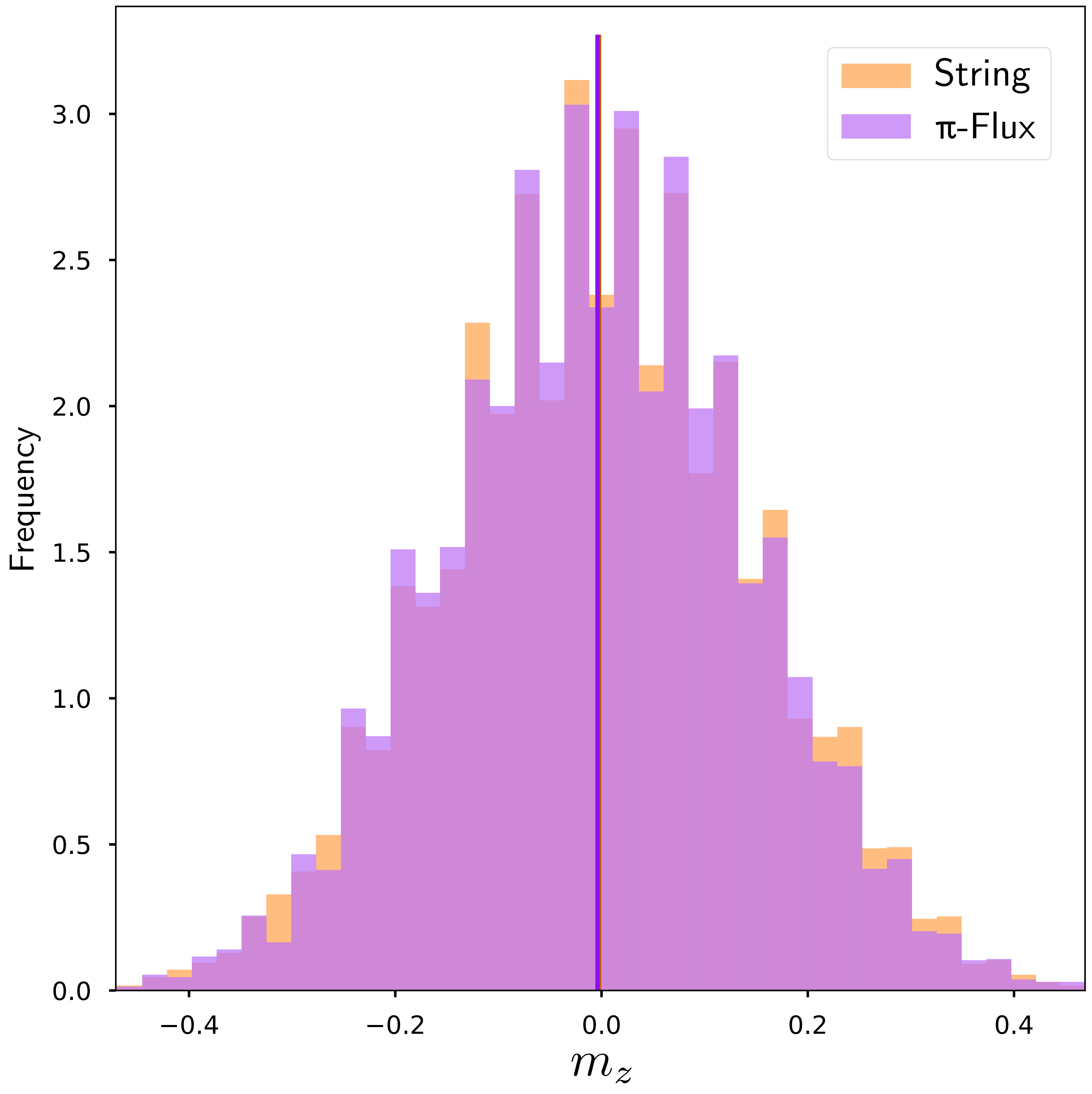


Miles et al., Nature Comm. 12 (2021)

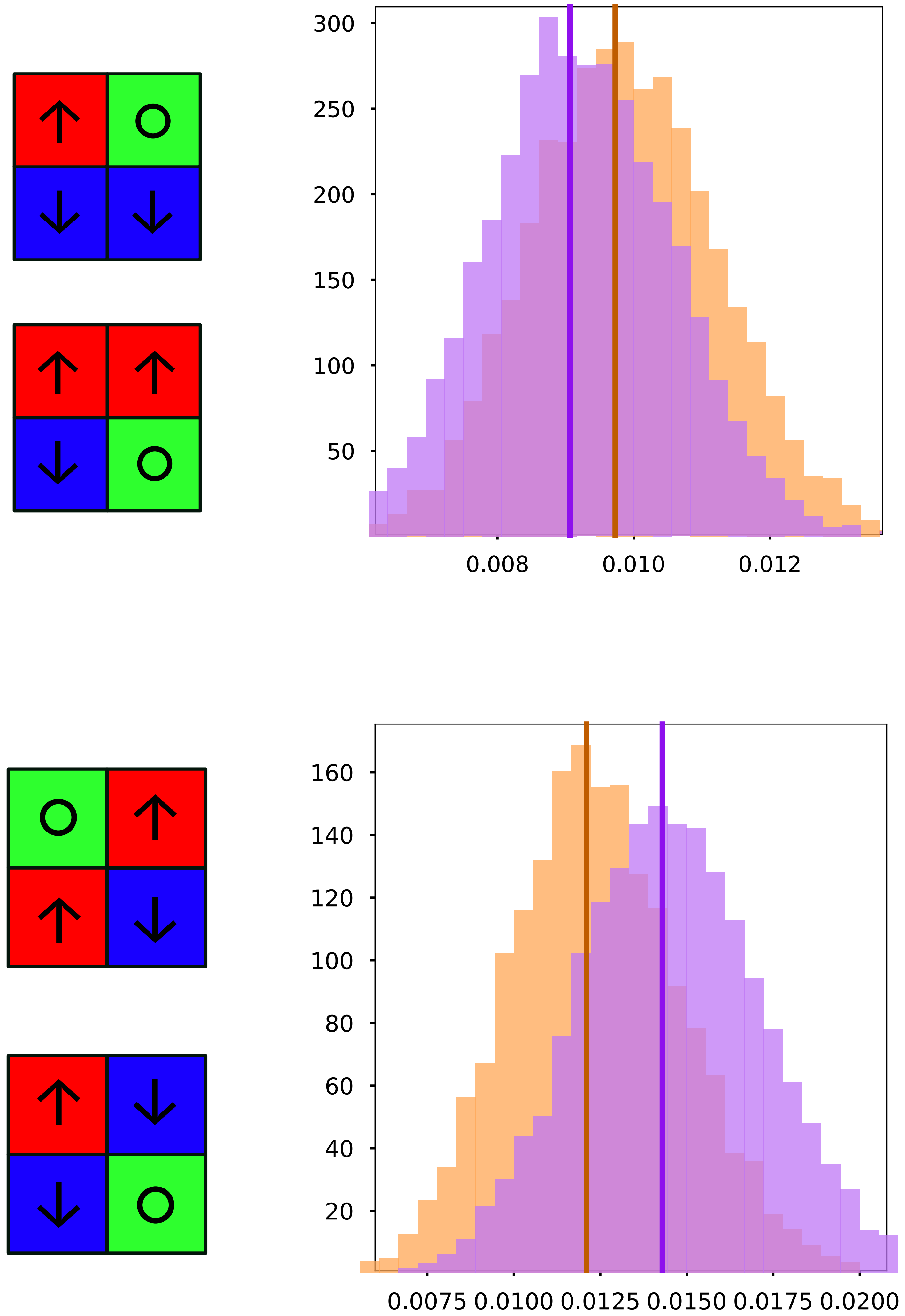


Koepsell et al., Science 374 (2021)

Histograms of pattern occurrence

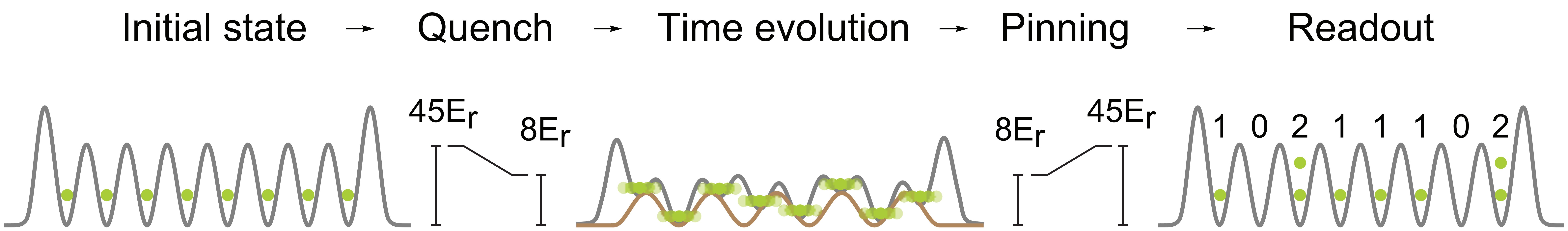


(a) Staggered magnetization.



CCNN: Application 2

Non-equilibrium dynamics



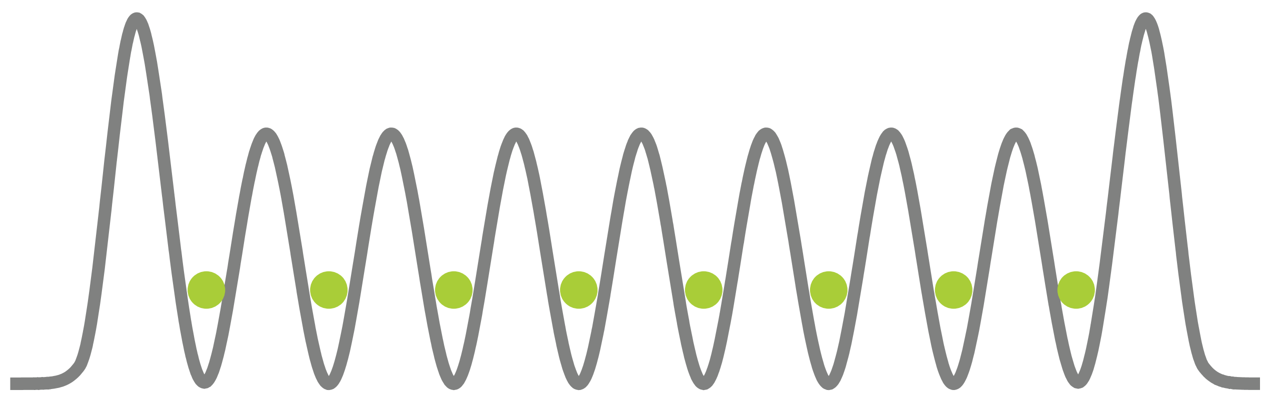
Lukin et al., Science 364 (2019)

Annabelle Bohrdt

Non-equilibrium dynamics

Initial state:

$$E_i \;=\; \langle \psi_0 | \hat{H} | \psi_0 \rangle$$



Thermal density matrix:

$$\hat{\rho}_\beta \;=\; \frac{1}{Z} \exp(-\beta_{\text{eff}} \hat{H})$$

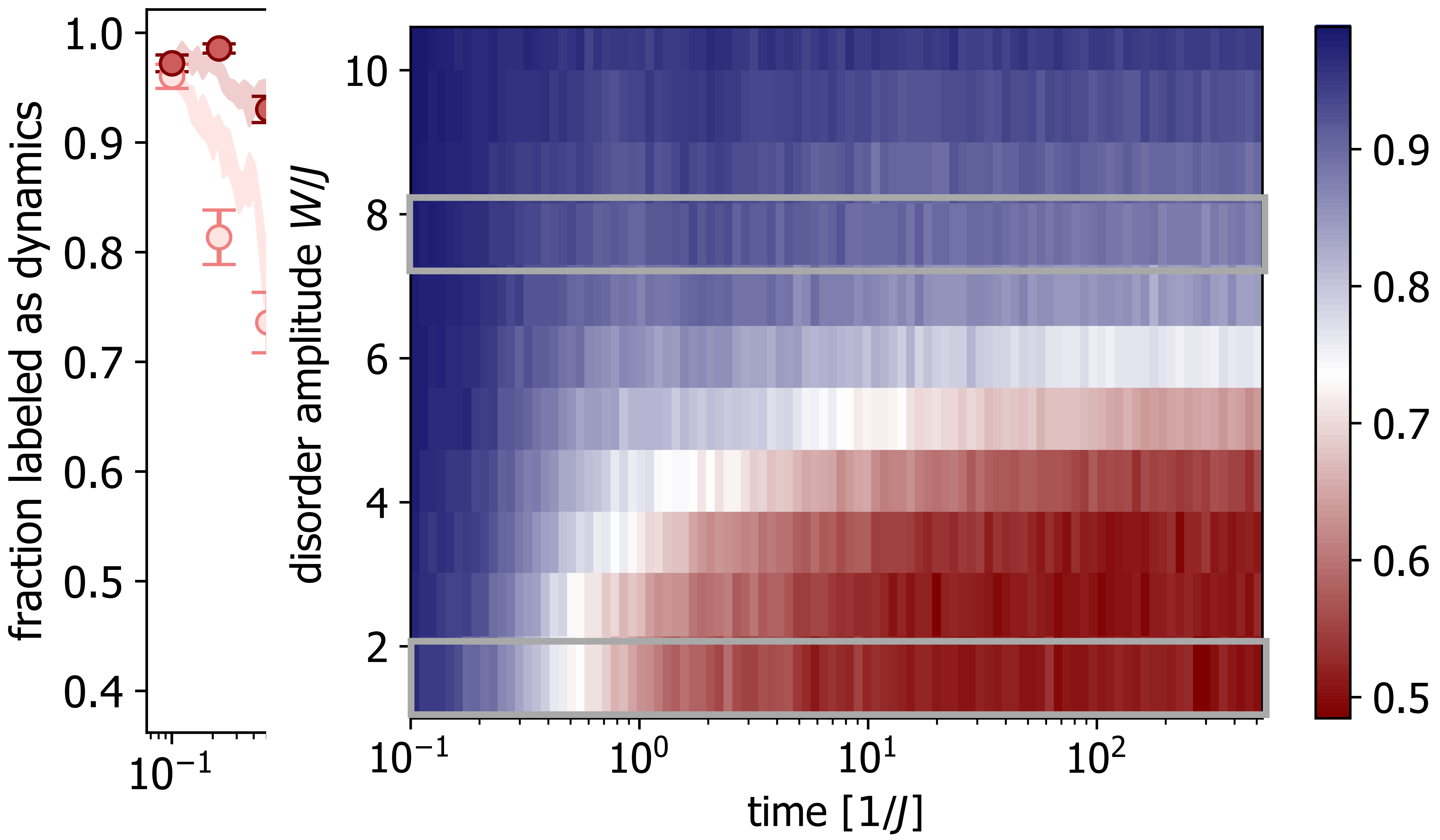
$$\beta_{\text{eff}} = 1/T_{\text{eff}}$$

$$E_i = \operatorname{tr}\left(\hat{H}\hat{\rho}_\beta\right)$$

Annabelle Bohrdt

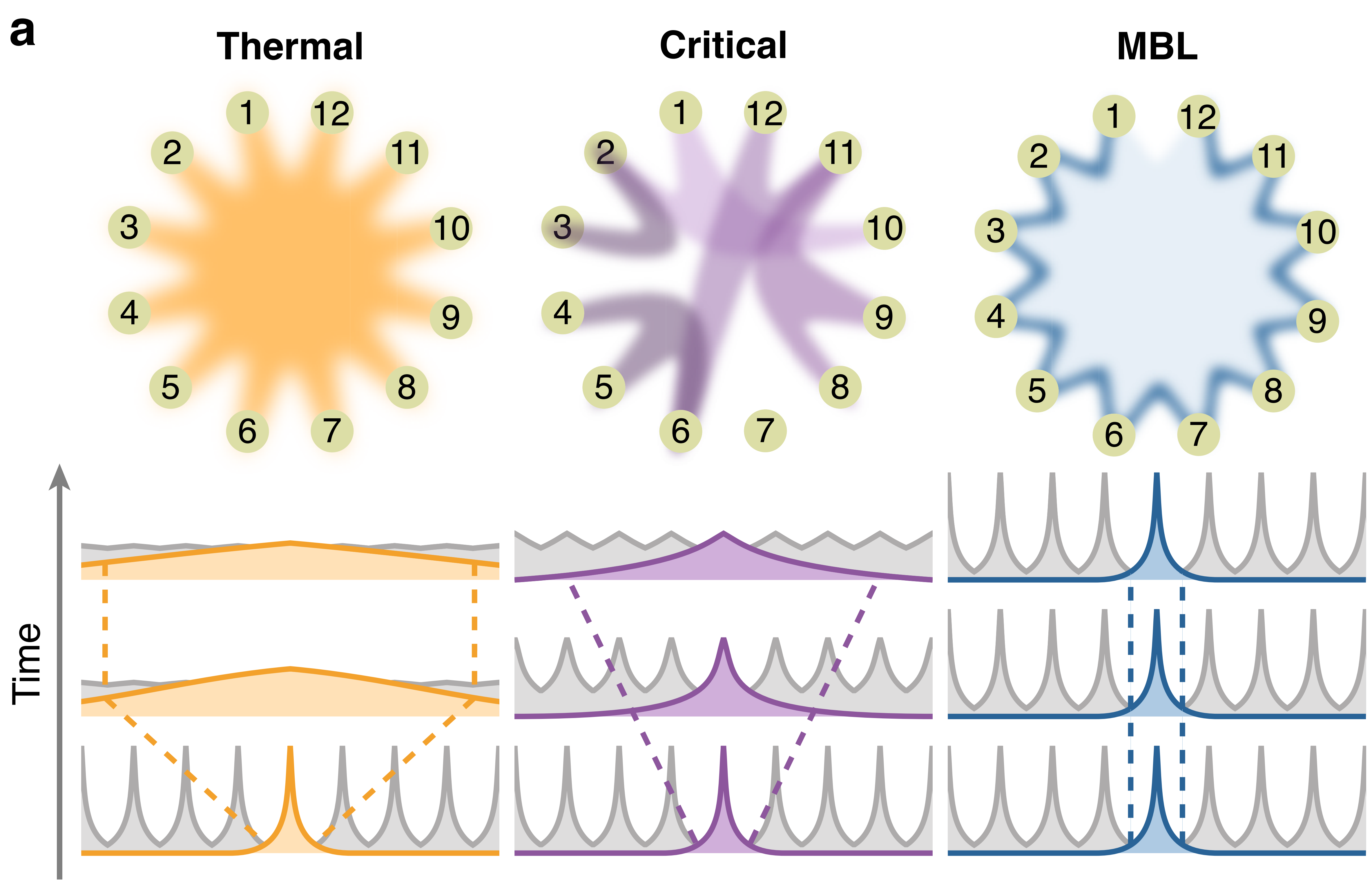
Non-equilibrium dynamics

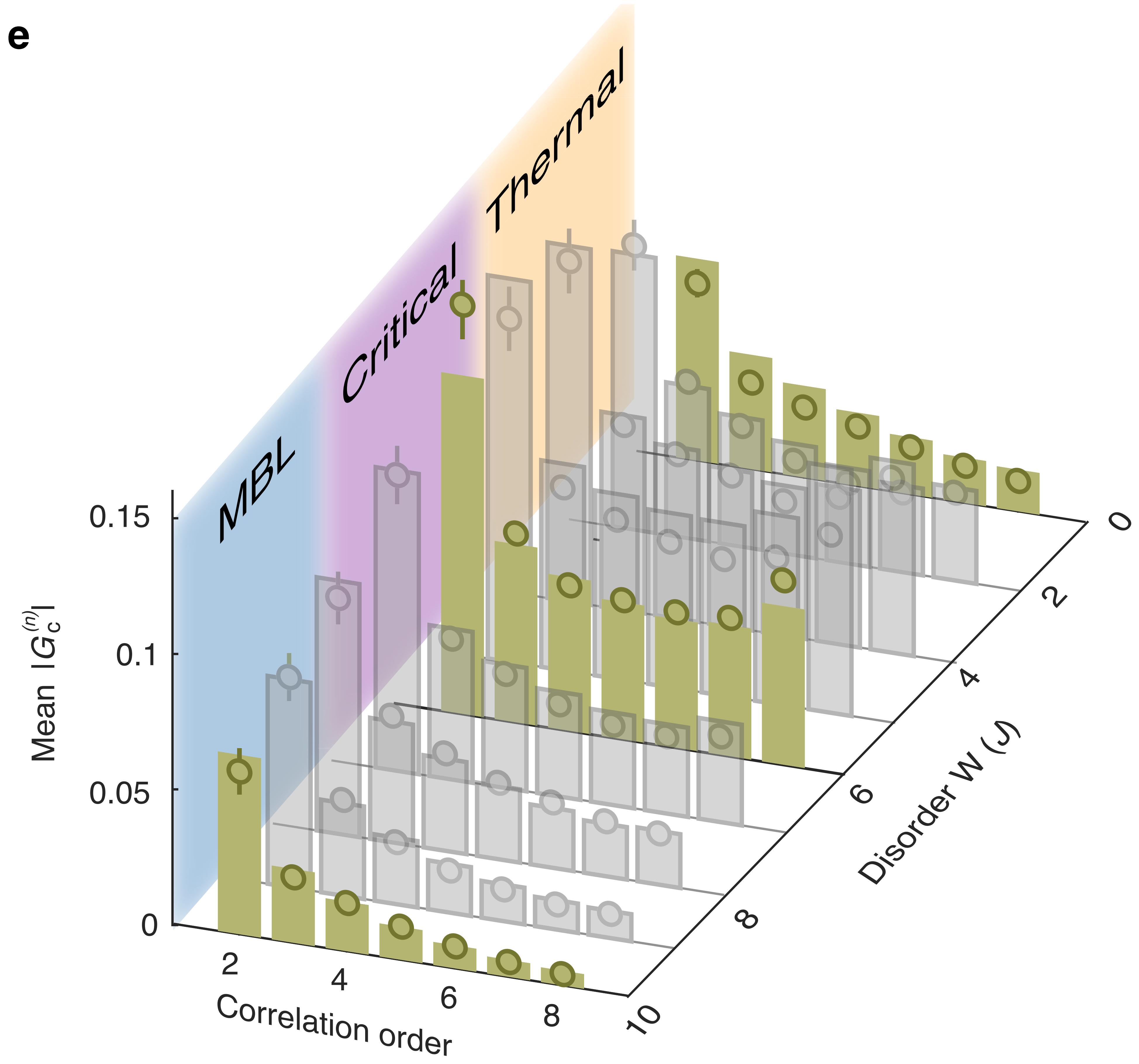
Non-equilibrium dynamics



Bohrdt et al., PRL 127 (2021)

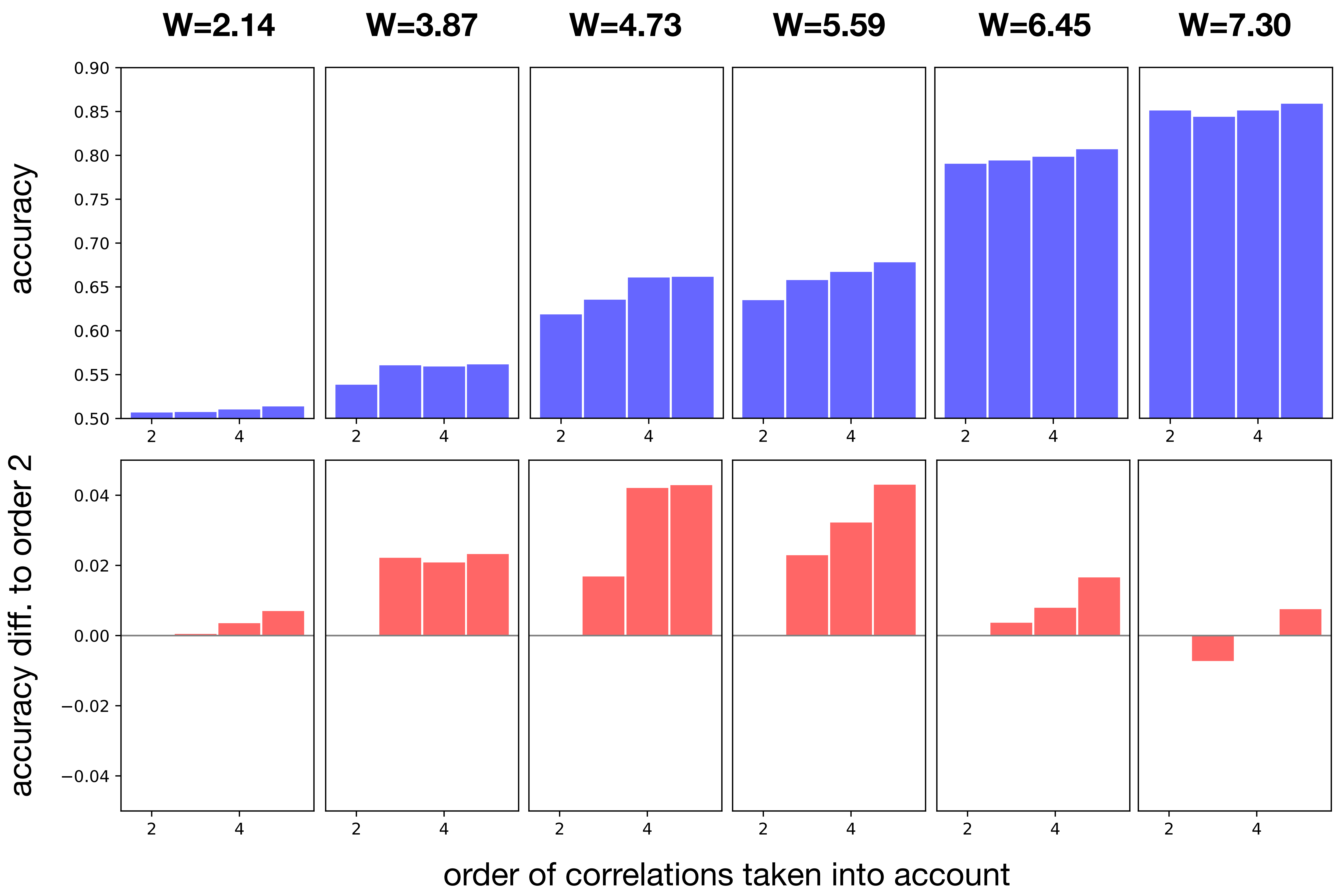
Non-equilibrium dynamics





Rispoli et al., Nature 573 (2019)

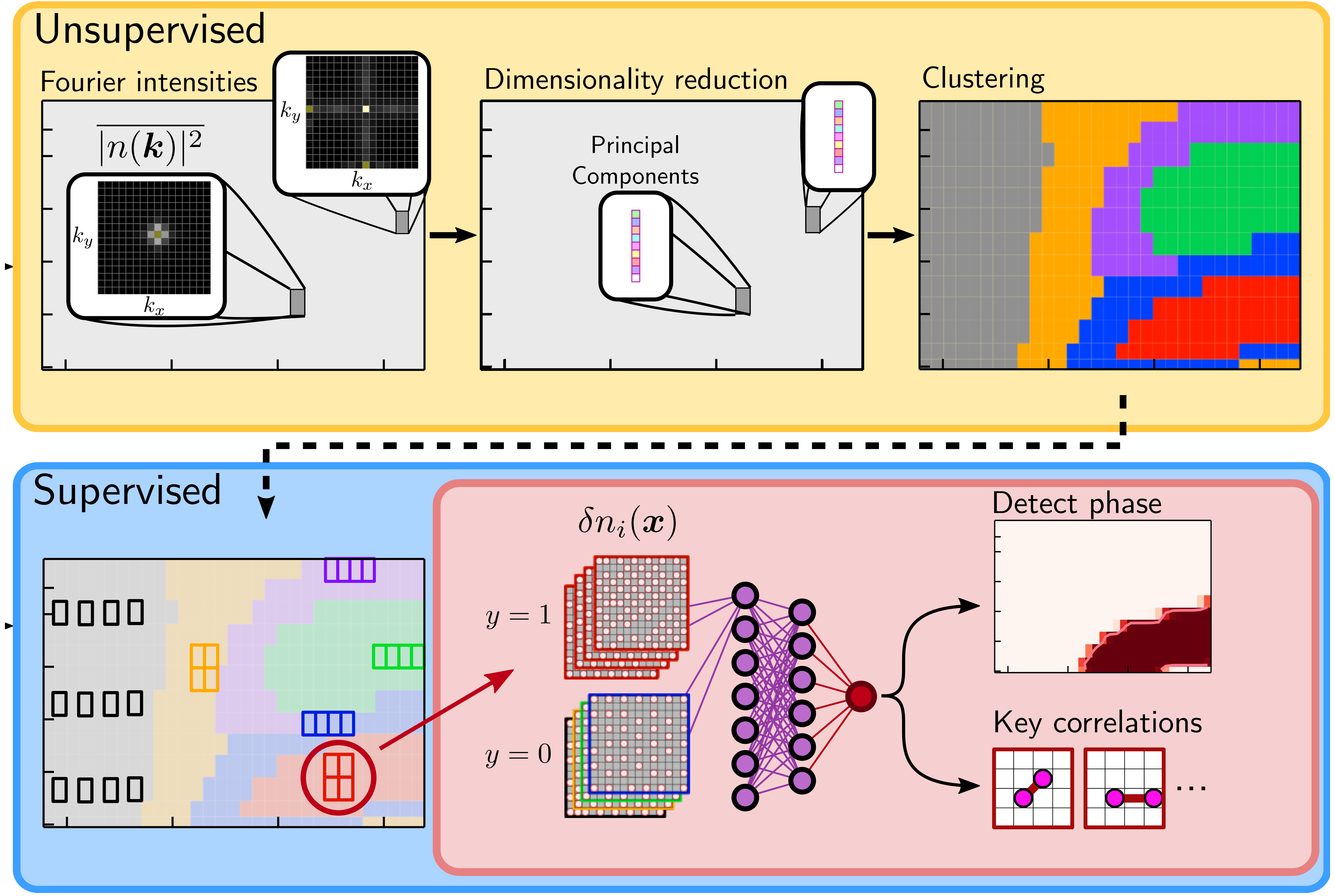
Non-equilibrium dynamics

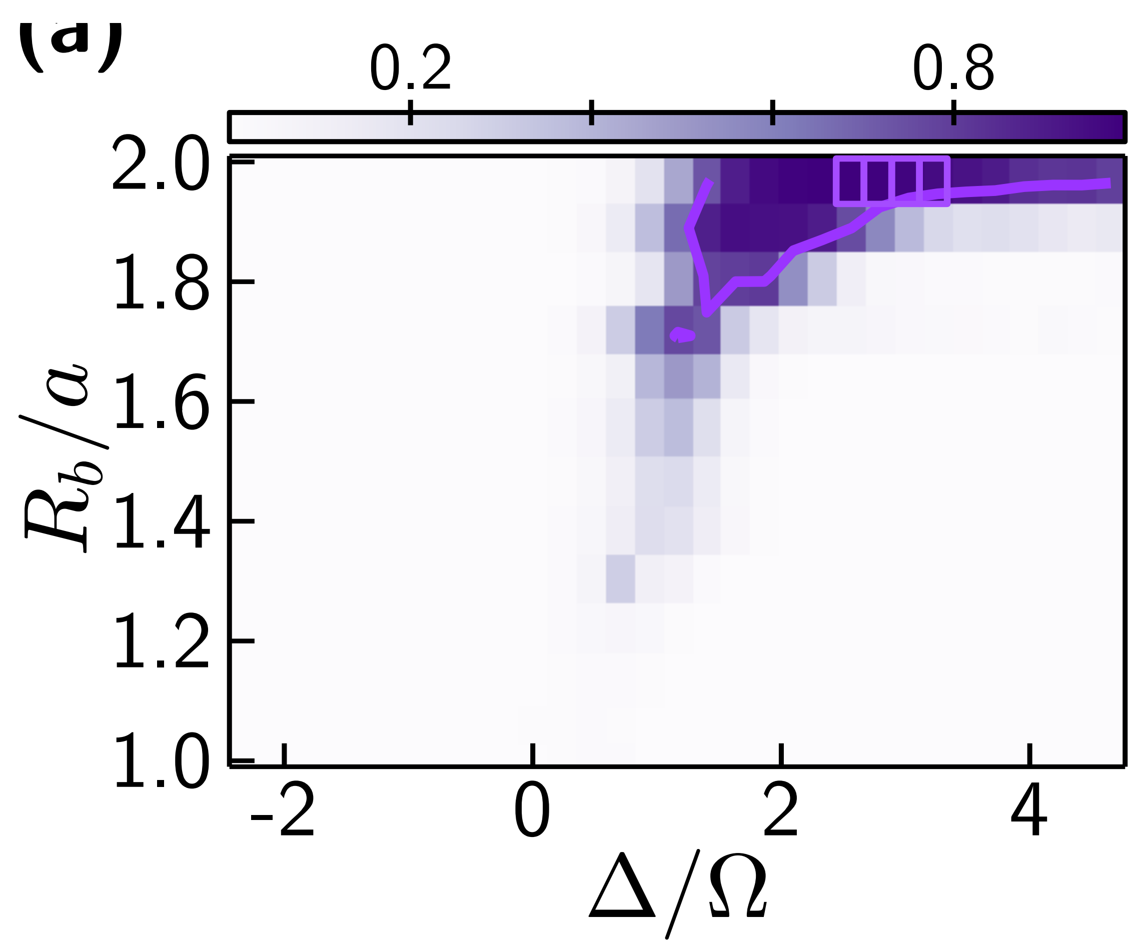


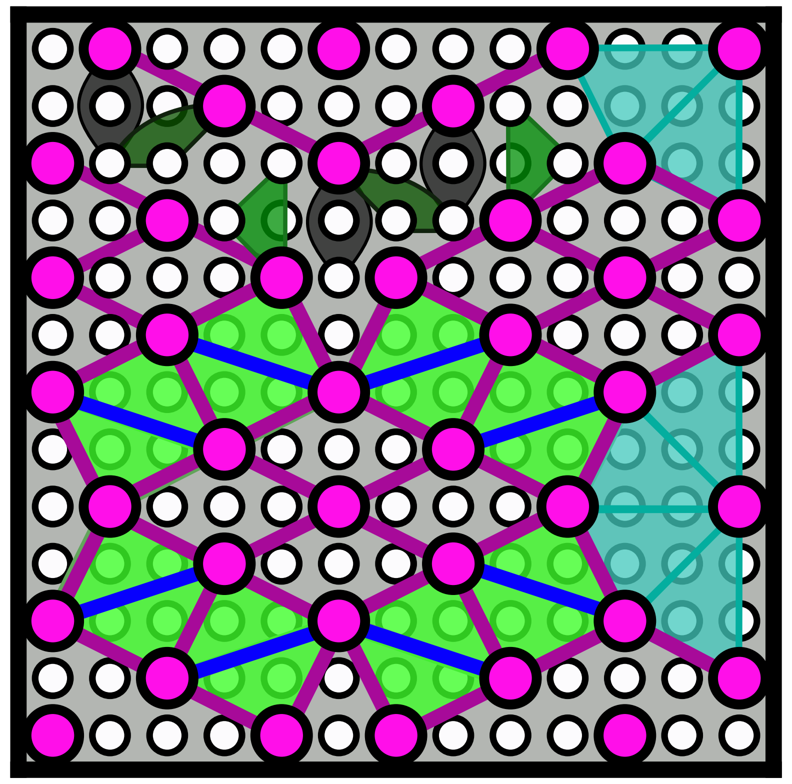
Bohrdt et al., PRL 127 (2021)

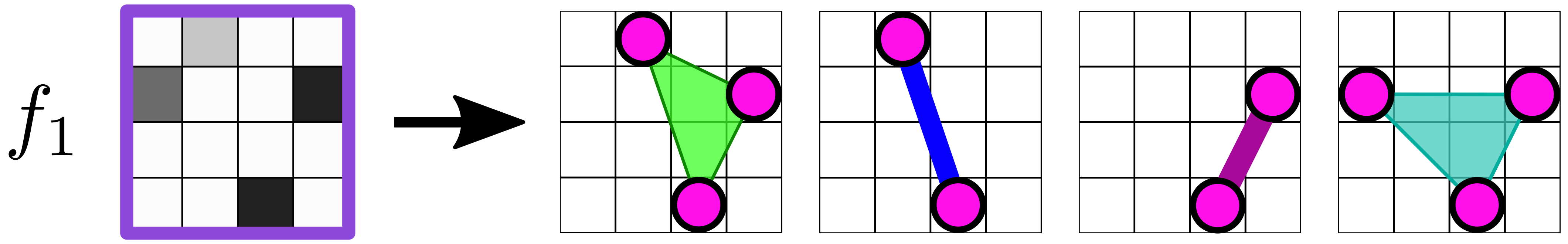
CCNN: Application 3

Rydberg system phase diagram



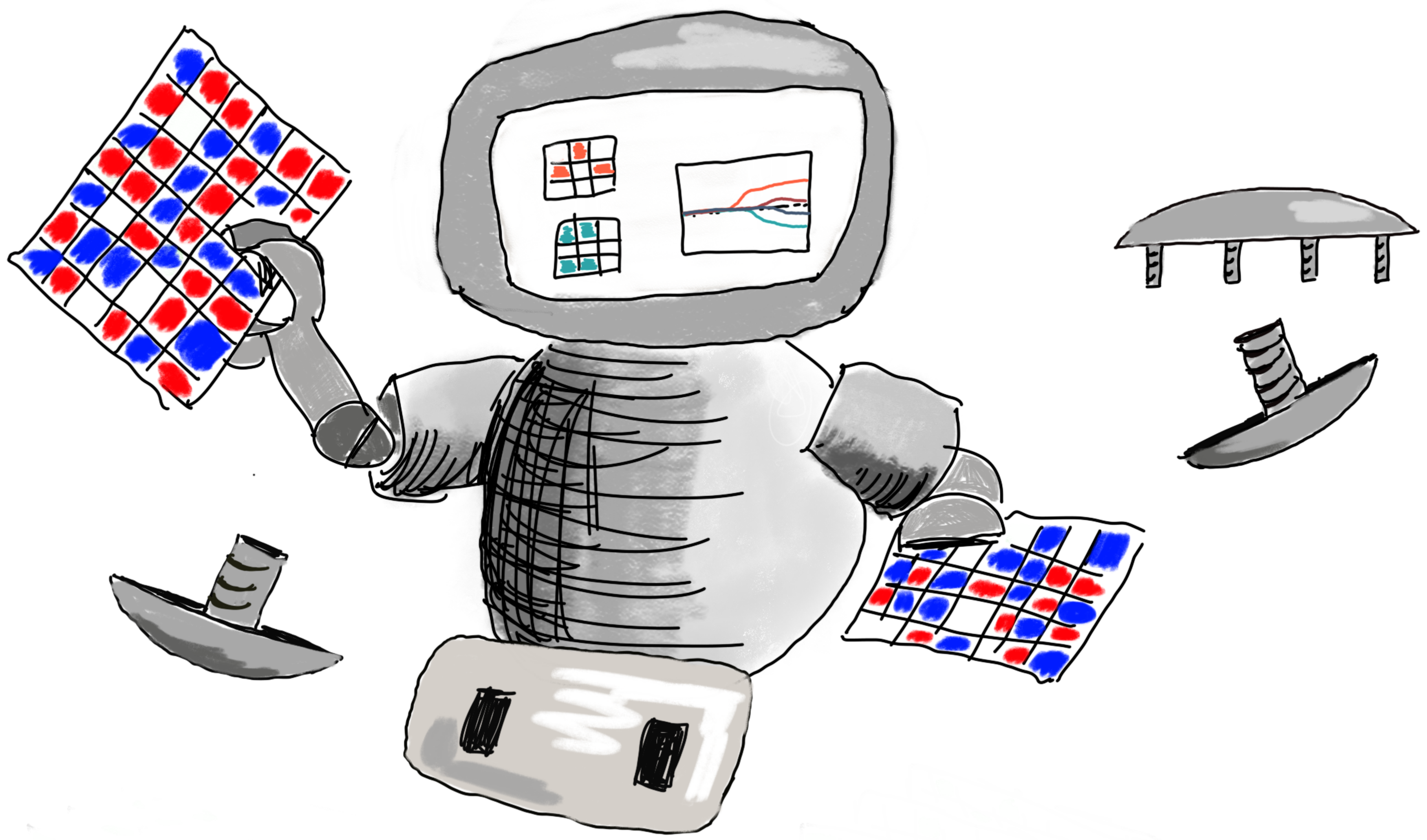






Miles et al., arXiv2112.10789

Thanks for your attention!



Ulrich Schollwöck
Lode Pollet
Fabian Grusdt
Hannah Lange
Matjaz Kebric
Felix A. Palm
Maximilian Buser
Lukas Homeier
Henning Schlömer
Christian Reinmoser

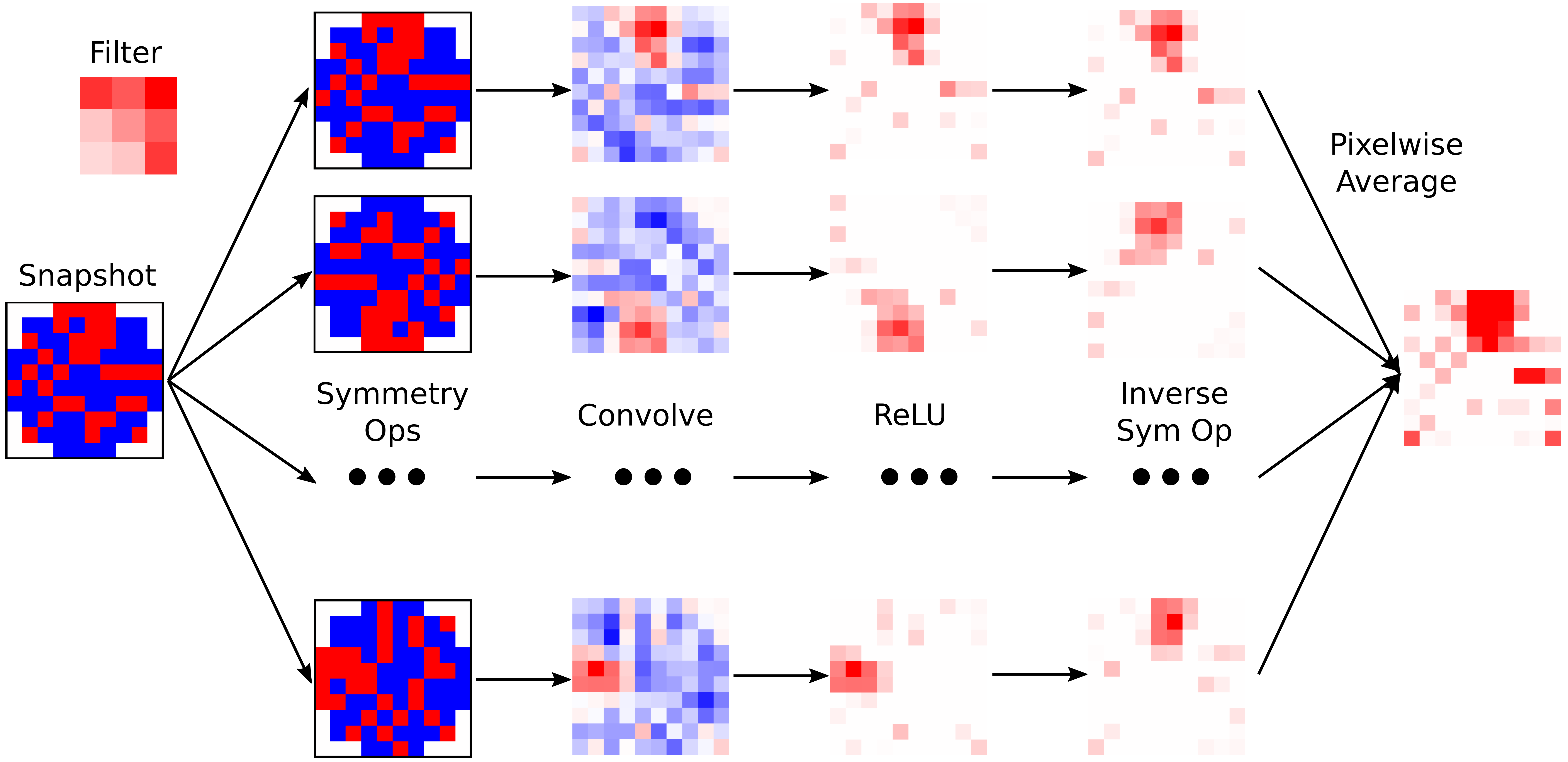
Eugene Demler
Yao Wang
Cole Miles
Eun-Ah Kim
Michael Knap
Frank Pollmann
…

Immanuel Bloch
Christian Gross
Joannis Koepsell
Jayadev Vijayan
Pimonpan Sompet
Guillaume Salomon
Dominik Bourgund
Petar Bojovic
Timon Hilker
Thomas Chalopin
Sarah Hirthe

Markus Greiner
Daniel Greif
Christie Chiu
Geoffrey Ji
Muqing Xu
Martin Lebrat
Lev Kendrick
Justus Brüggenjürgen
Anant Kale
Julian Léonard
Sooshin Kim
Matthew Rispoli
Alex Lukin
Robert Schittko

Bonus slides

CCNN: symmetric convolutions



Miles et al., Nature Comm. 12 (2021)