Quantum HyperNetworks

Training binary neural networks in quantum superposition

https://arxiv.org/abs/2301.08292

Juan Felipe Carrasquilla Álvarez

Vector Institute —> ETH zürich

VECTOR INSTITUTE | INSTITUT VECTEUR
UNIVERSITY OF WATERLOO
UNIVERSITY OF TORONTO
CIFAR
NSERC CRSNG
compute canada | calcul canada

Deep learning

- Computer vision, natural language processing, machine translation, self driving cars, game playing, physics, chemistry, finance, healthcare, demographics, entertainment, music, art, robotics.

- Availability of datasets, specialized hardware, and outstanding algorithmic developments have ushered a new generation of large models displaying unprecedented accuracy across a wide array of technologically and scientifically relevant tasks.

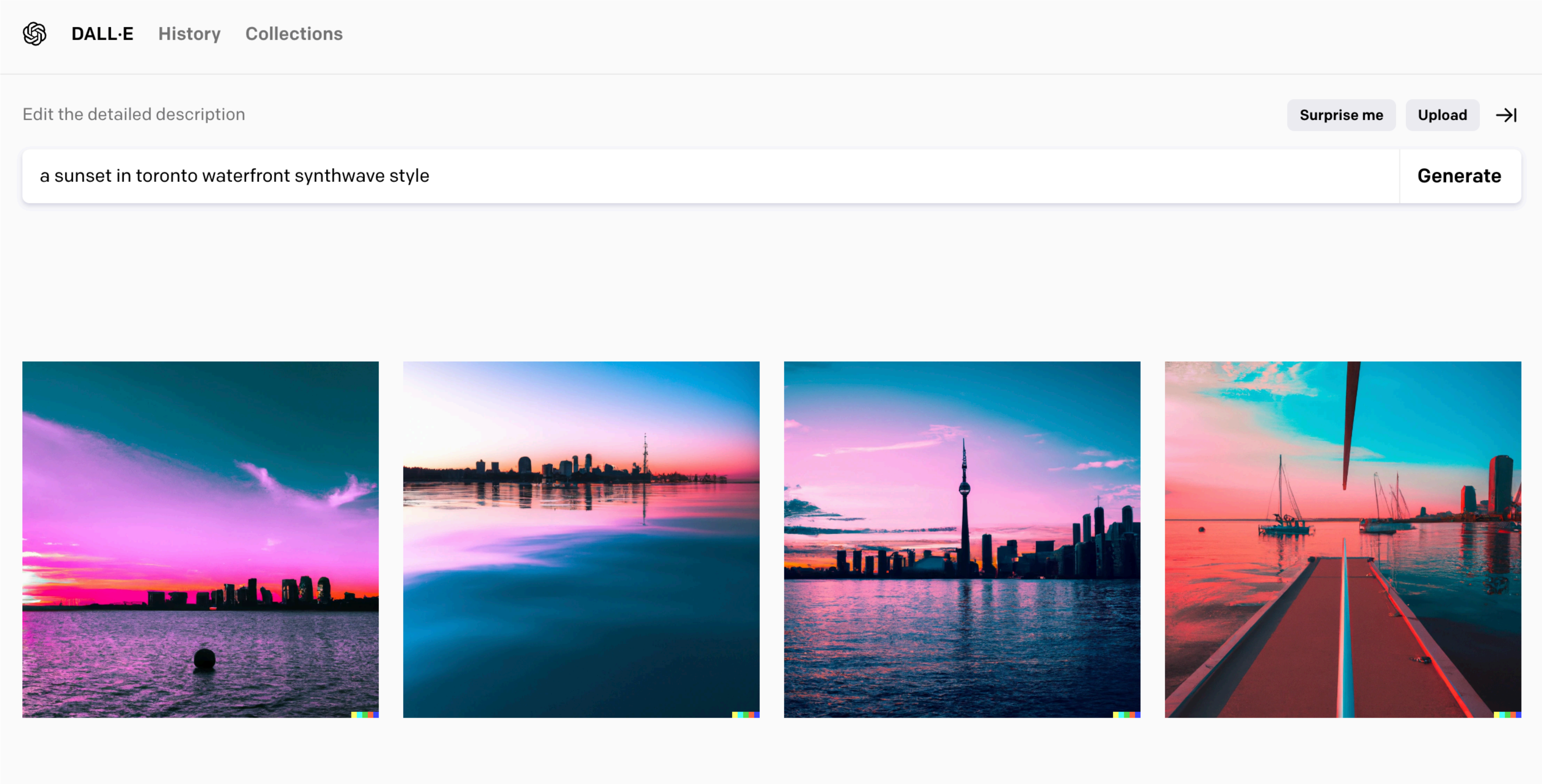
- Example: Diffusion models

- Impressive results for image generation that suggest that art will change

- Dall-E-2: Prompt the model with natural language and it draws artistic pictures

https://openai.com/dall-e-2/

Deep learning: Dall-E



https://openai.com/dall-e-2/

Natural language processing

- Branch of computer science, linguistics, and machine learning concerned with giving computers the ability to process text and spoken words in a similar way humans do it.
- Machine translation
- Speech recognition
- Sentiment analysis
- Automatic summarization of text
- Text to image/video generation

Deep learning

- Example: “Galactica: A Large Language Model for Science” https://arxiv.org/abs/2211.09085
- Prompt a scientific topic and the language model writes a manuscript for you. Surprising results.
- Meta shuts down public test of Galactica, its ‘AI for Science’ because it produced pseudoscientific papers

Deep learning

× not verified

Recurrent Neural Network wavefunctions

We can use a Recurrent Neural Network (RNN) to model the quantum wavefunction of a many-body system.

In this notebook we show how to implement this model and train it on the ground state of the Ising model.

For this notebook we will use the TensorFlow Keras API.

The model

The RNN wavefunction has the following structure:

Each site is represented by a spin variable s_i which can take two values, ± 1.

The wavefunction is parametrized by an RNN, where each spin s_i is treated as a time step in the RNN. The hidden state h_i at each time step i is updated according to the following rule:

$$h_i = f(s_i W^{(1)} + h_{i-1} W^{(2)})$$

where f is an activation function, $W^{(1)}$ and $W^{(2)}$ are matrices which parametrize the RNN. The output at each time step is given by:

$$\psi(s_1, \ldots, s_N) = f(h_N W^{(3)})$$

The matrices $W^{(1)}$, $W^{(2)}$ and $W^{(3)}$ are trained using Variational Monte Carlo (VMC).

Data

We train the RNN on the ground state of the one-dimensional transverse field Ising model:

ChatGPT

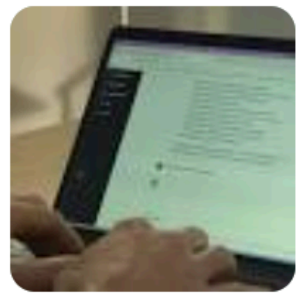
- ChatGPT is a machine learning model which interacts in a conversational way
- Dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

CBC

Can the new AI tool ChatGPT replace human work? Judge for yourself

A new artificial intelligence tool using natural language processing has captured the public's imagination, amassing more than a million...

1 day ago



Global News

ChatGPT: Everything to know about the viral, 'groundbreaking' AI bot - National | Globalnews.ca

Users can ask the AI to write essays, poems or scripts, or even translate or summarize text. ChatGPT can also answer questions on a wide...

21 hours ago

CNET

The 5 Best Uses (So Far) for ChatGPT's AI Chatbot

The new AI tool ChatGPT has inspired excitement and worry with its ability to instantly answer complex questions. In the days after its...

1 day ago

The Toronto Star

I wrote a story about ChatGPT's AI. Then I dared it to write a better one

Like a million or so other artificial intelligence dumb-dumbs who discovered ChatGPT this week, I couldn't wait to show off the capabilities...

https://openai.com/blog/chatgpt/

ChatGPT

- People now believe these models will likely have deep technological, educational, and societal implications.

Daily Mail

ChatGPT can pass the United States Medical Licensing Exam and the Bar Exam

The rise of the machines? ChatGPT CAN pass US Medical Licensing Exam and the Bar, experts warn - after the AI chatbot received B grade on...

4 hours ago

Firstpost

OpenAI's ChatGPT passes Wharton's MBA Exam, while also qualifying for a medical licence in the US

Medical researchers are actually considering using ChatGPT in actual, clinical cases to assist real doctors. OpenAI's chatbot ChatGPT has been...

5 hours ago

02-21-23 | 9:59 AM

A science fiction magazine closed submissions after being bombarded with stories written by ChatGPT

In a case of life (or something) imitating art, an award-winning publisher of science fiction says it's being overrun with AI-generated work.

https://openai.com/blog/chatgpt/

ChatGPT

I'll do my best to create an ASCII art representation of the Mona Lisa, however, please keep in mind that it may not be a very detailed or realistic representation. Here is my attempt:

Copy code

```
     ._""""""_.
   .'          '.
  /   O      O   \
 :                :
 |                |
 :  ',        ,'  :
  \  '-......-'  /
   '.          .'
     '-......-'
```

Please note that this is a very simple and minimal representation of Monalisa, as ASCII art is not my primary function. It may not be the best representation of Monalisa.

https://openai.com/blog/chatgpt/

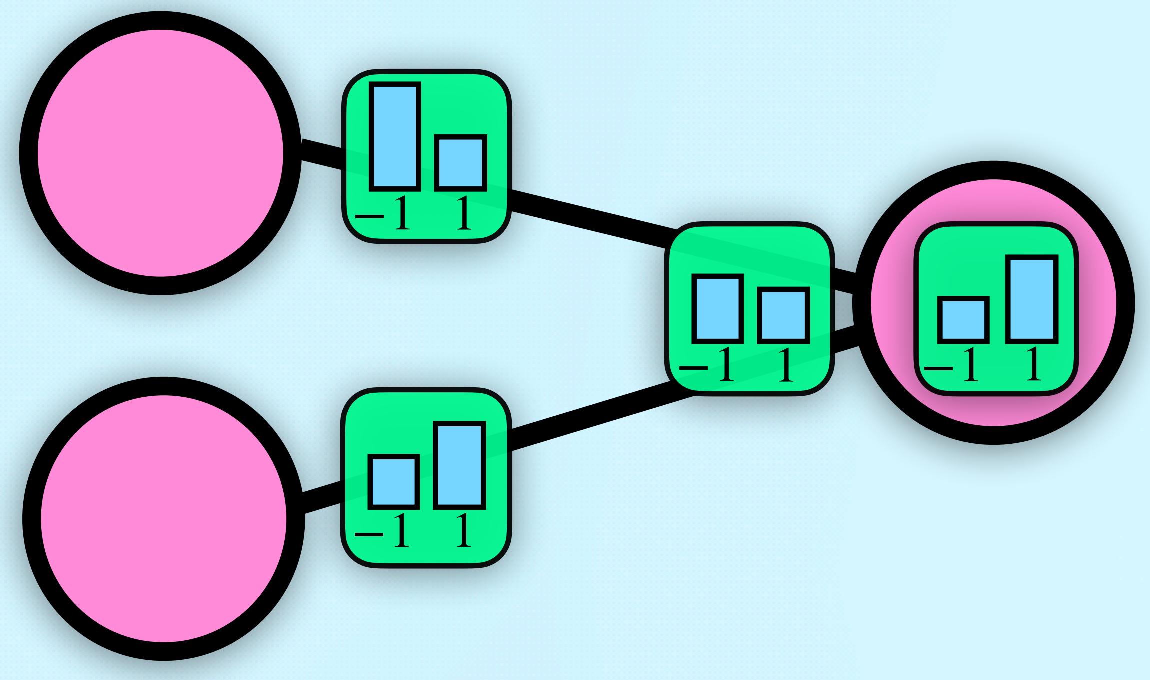
Training these models this is expensive

- These advances crucially depend on the availability of specialized computational resources such as graphics and tensor processing units, which demand a high electricity consumption.
- In particular, a set of key but computationally expensive elements in the modern machine learning (ML) workflow include hyperparameter optimization and neural architecture search.
- GPT-3's training costs up to $27.6 million
- GPT-4

The GPT-3 economy: https://bdtechtalks.com/2020/09/21/gpt-3-economy-business-model/

Possible solution is quantization of weights and biases

Binary Neural networks



- Neural networks with binary weights and activations (BiNNs) partially alleviate these issues as they are computationally efficient, hardware-friendly, and energy efficient.
- 32-fold reduction in memory.
- Robust to adversarial attacks.
- Specialized hardware implementations that simultaneously increase computational speed and improve their energy efficiency.
- Parameter, hyperparameter, and architectural searches remains computationally expensive— multiple nested combinatorial optimization problems

Binary Neural Networks remain expensive to train

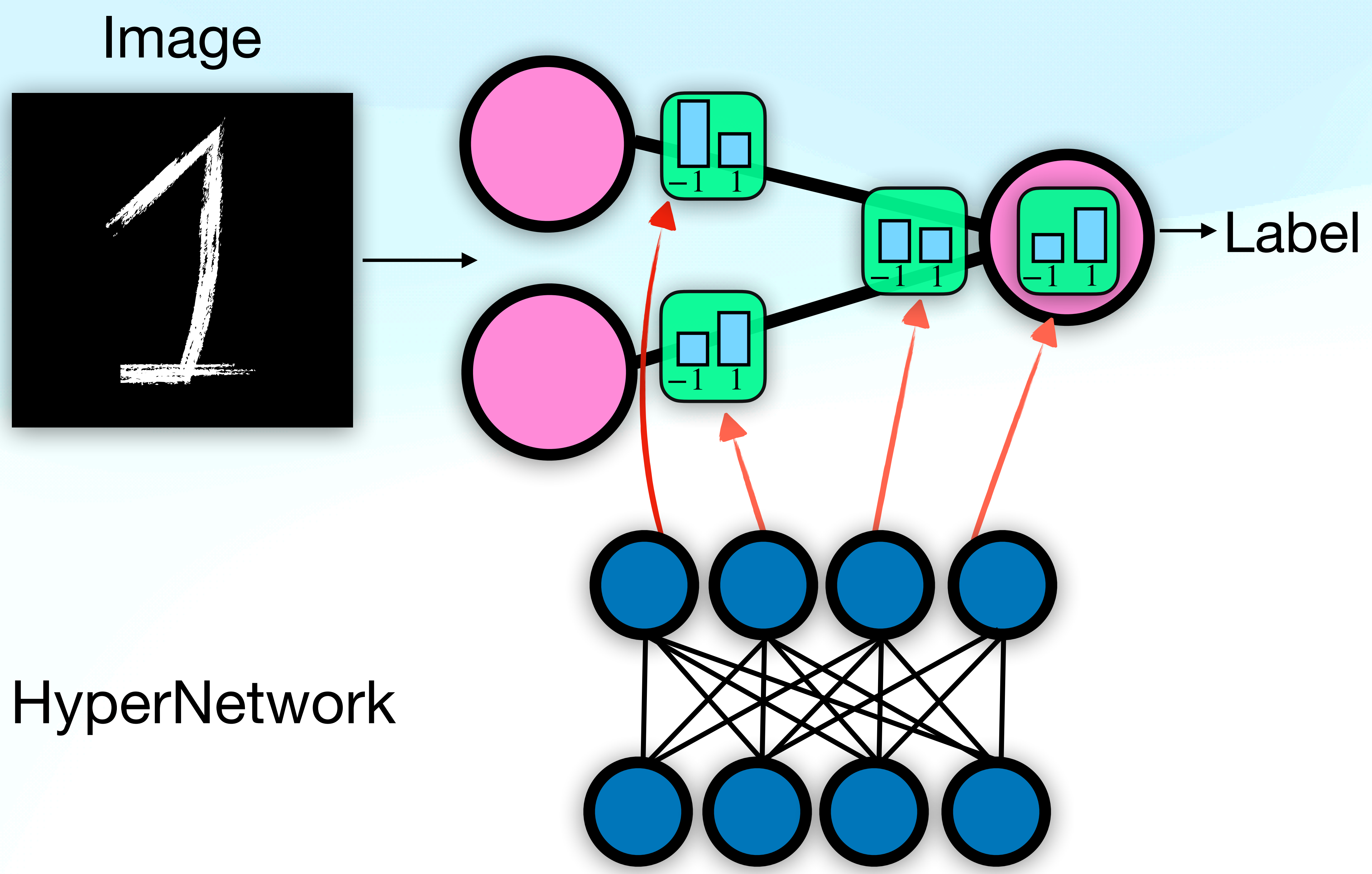
- Traditionally, there are two nested loops:
- Outer optimization—loop through the hyperparameter and architectural state spaces on a validation set
- Inner optimization—adjusts the weights of the neural network on a training set.
- Such a nested optimization process remains the most computationally demanding task in the modern ML workflow and entails an unsustainable carbon footprint
- Call for computationally efficient hardware and algorithms to train and search for neural architectures

Binary Neural Networks are expensive to train

- Traditionally, there are two nested loops:
- Outer optimization—loop through the hyperparameter and architectural state spaces on a validation set
- Inner optimization—adjusts the weights of the neural network on a training set.
- Objective: $C(\boldsymbol{w}) = \dfrac{1}{N_t} \sum_{i=1}^{N_t} \mathscr{L}\left(\mathrm{NN}(\boldsymbol{x}_i; \{\boldsymbol{w}\}), \boldsymbol{y}_i\right).$
- Real boolean function of the binary (hyper-) parameters of the neural network.

HyperNetworks

- HyperNetworks: an approach of using a one network, also known as a hypernetwork, to generate the weights for another network.



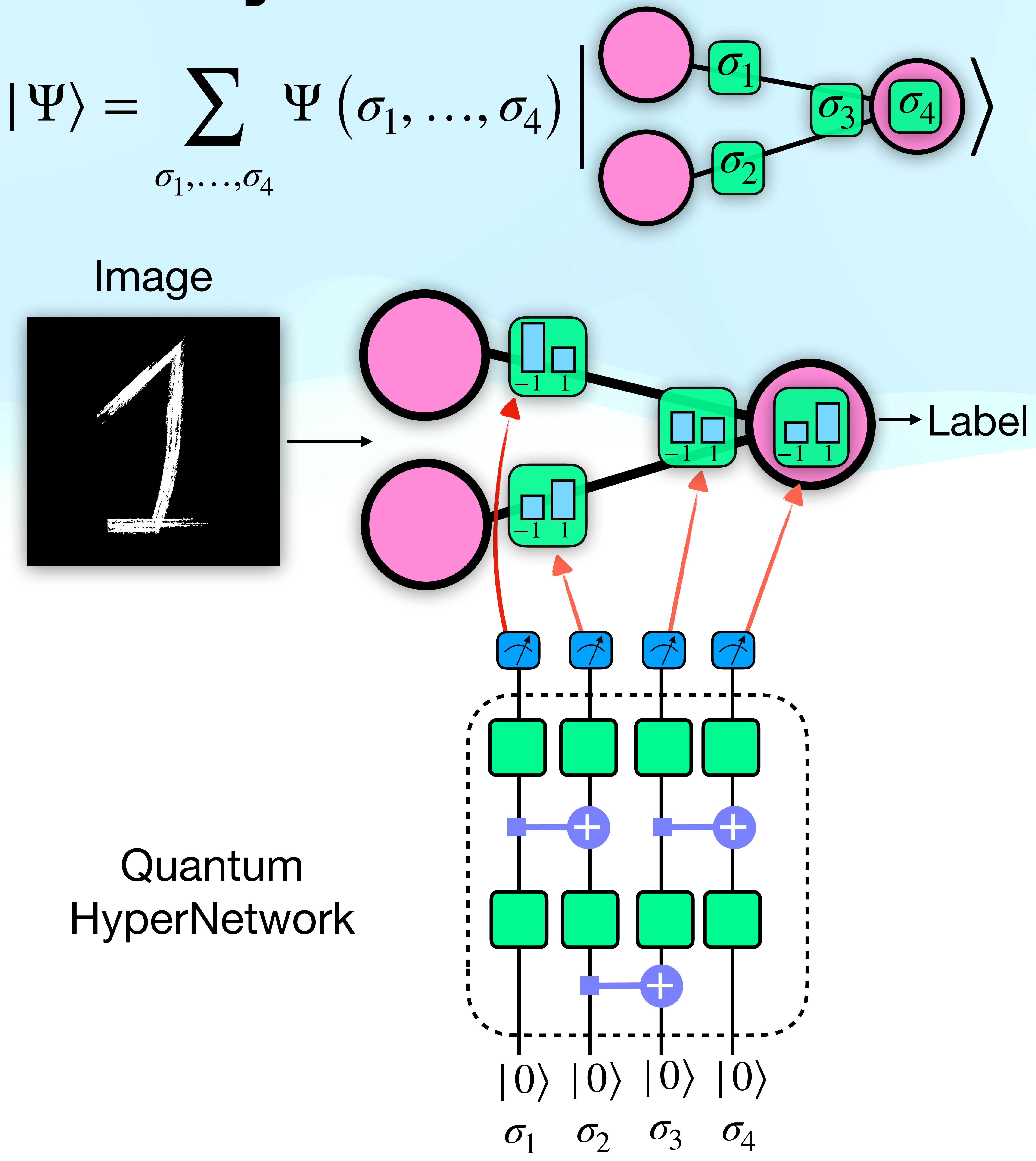
- Used in natural language processing, computer vision, hyperparmeter tuning, neural architectural search, meta-learning.

- HyperNetworks. https://arxiv.org/abs/1609.09106

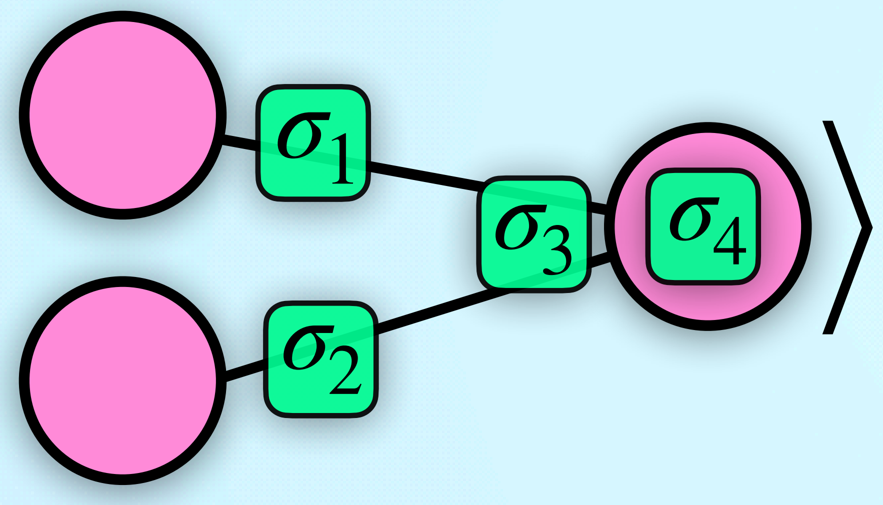
Quantum HyperNetworks

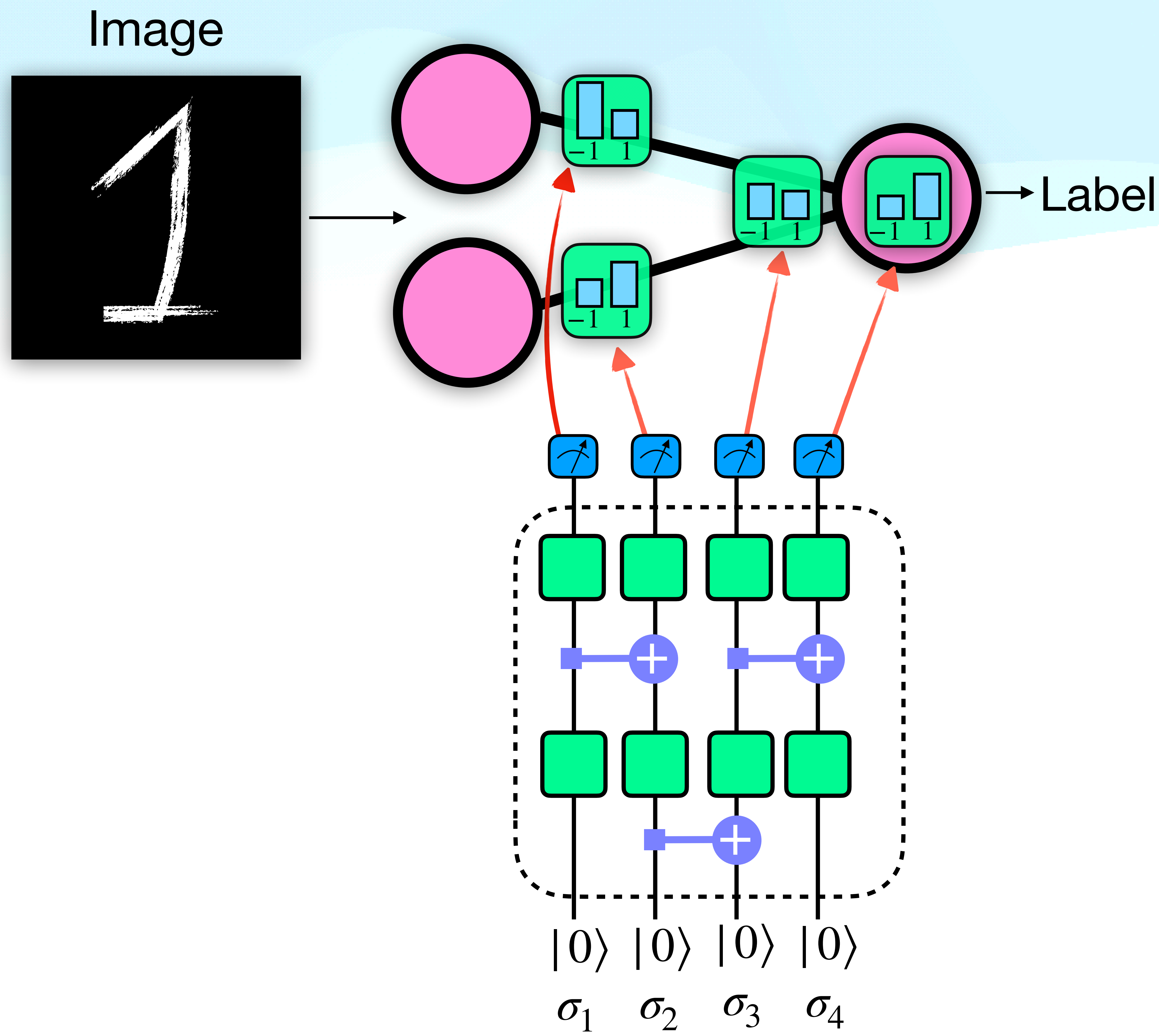
Quantum HyperNetworks to train binary neural networks

- We define Quantum HyperNetworks and use them to unify parameter, hyperparameter, and architectural search for binary neural networks
- Can be understood as training binary neural networks in quantum superposition
- Superpositions contain exponentially many binary neural networks with different parameters, architectural choices, and hyperparameters



Encoding BiNNs in a quantum state

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi\left(\sigma_1,\ldots,\sigma_4\right) \Big|$$  $$\Big\rangle$$



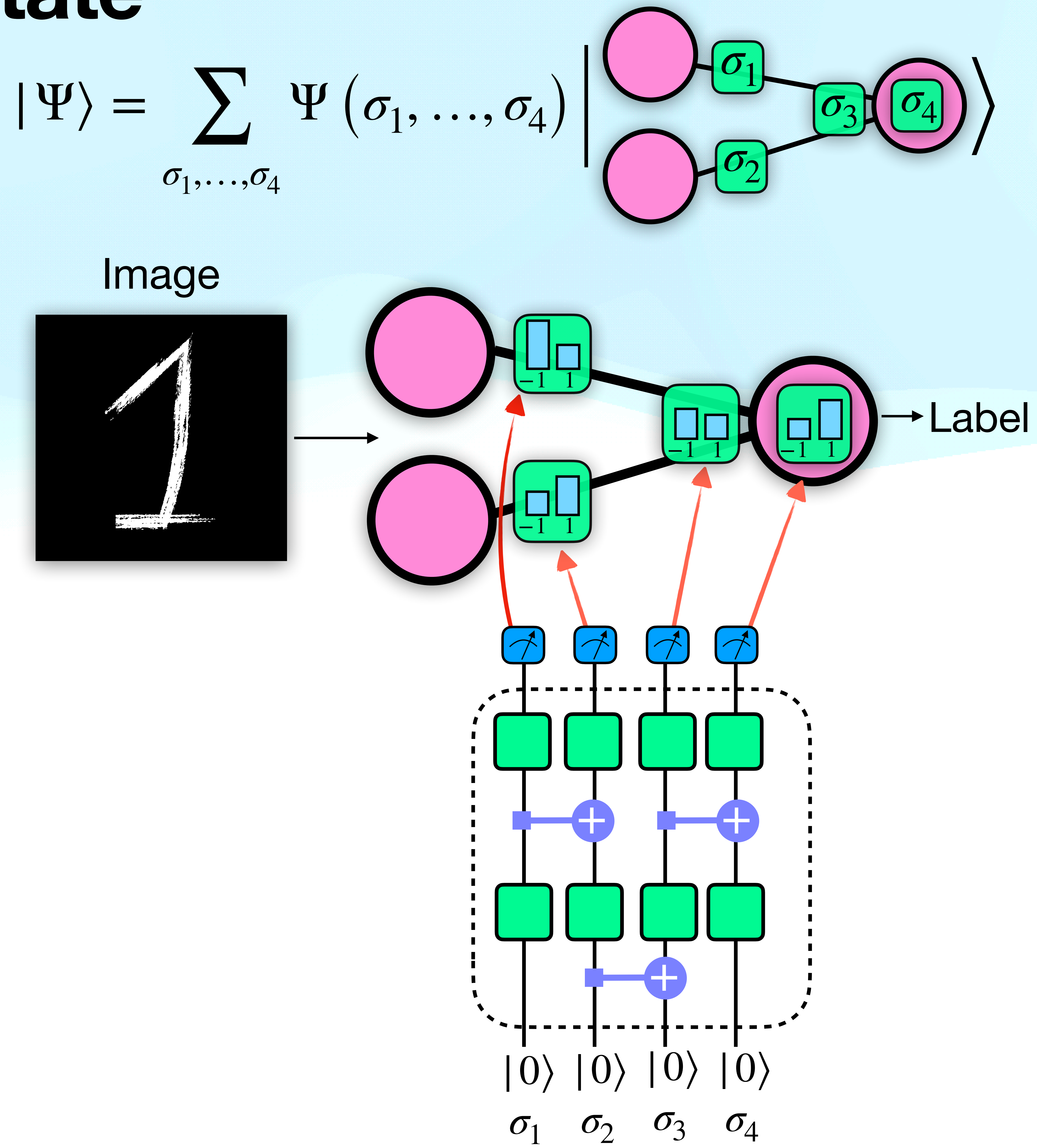
- Consider a quantum state
- $$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_N} \Psi(\sigma_1,\ldots,\sigma_N)\,|\sigma_1,\ldots,\sigma_N\rangle$$
- To each basis element $|\boldsymbol{\sigma}\rangle = |\sigma_1,\ldots,\sigma_N\rangle$ −> augmented model comprising the parameters, hyperparameters, and any desired architectural choices.
- The selection of activation function from two possibilities f_1 or f_2, we make the activation function qubit dependent (qubit σ_4).
  $f(x) \rightarrow f(x, \sigma_4)$

Encoding BiNNs in a quantum state

- $f(\boldsymbol{x};\sigma) = \begin{cases} f_1(\boldsymbol{x}) & \text{if } \sigma = 0 \\ f_2(\boldsymbol{x}) & \text{if } \sigma = 1. \end{cases}$

- Other architectural choices (skip connections, dimension of the hidden layer, # of layers, etc)—add more qubits.

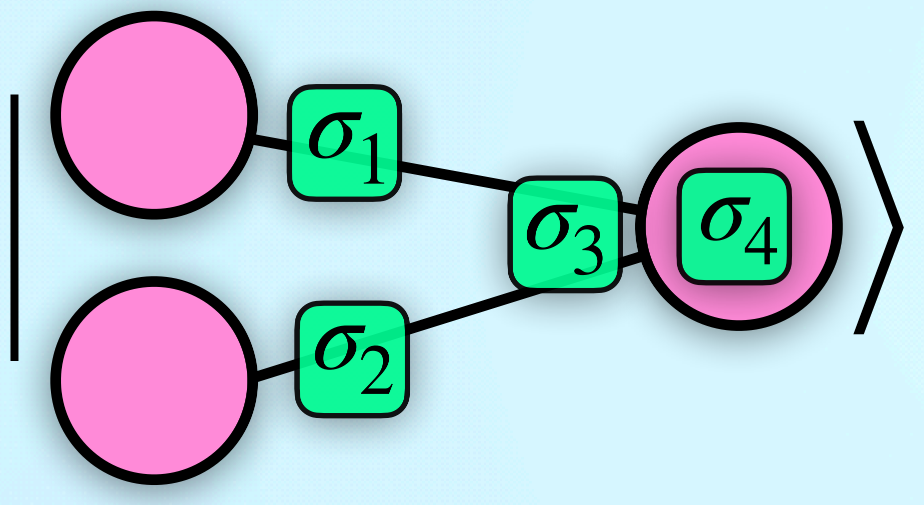
- We “nudge” the state so that when we measure it in an experiment, it returns neural networks with good architectural choices, parameters, and hyperparameters



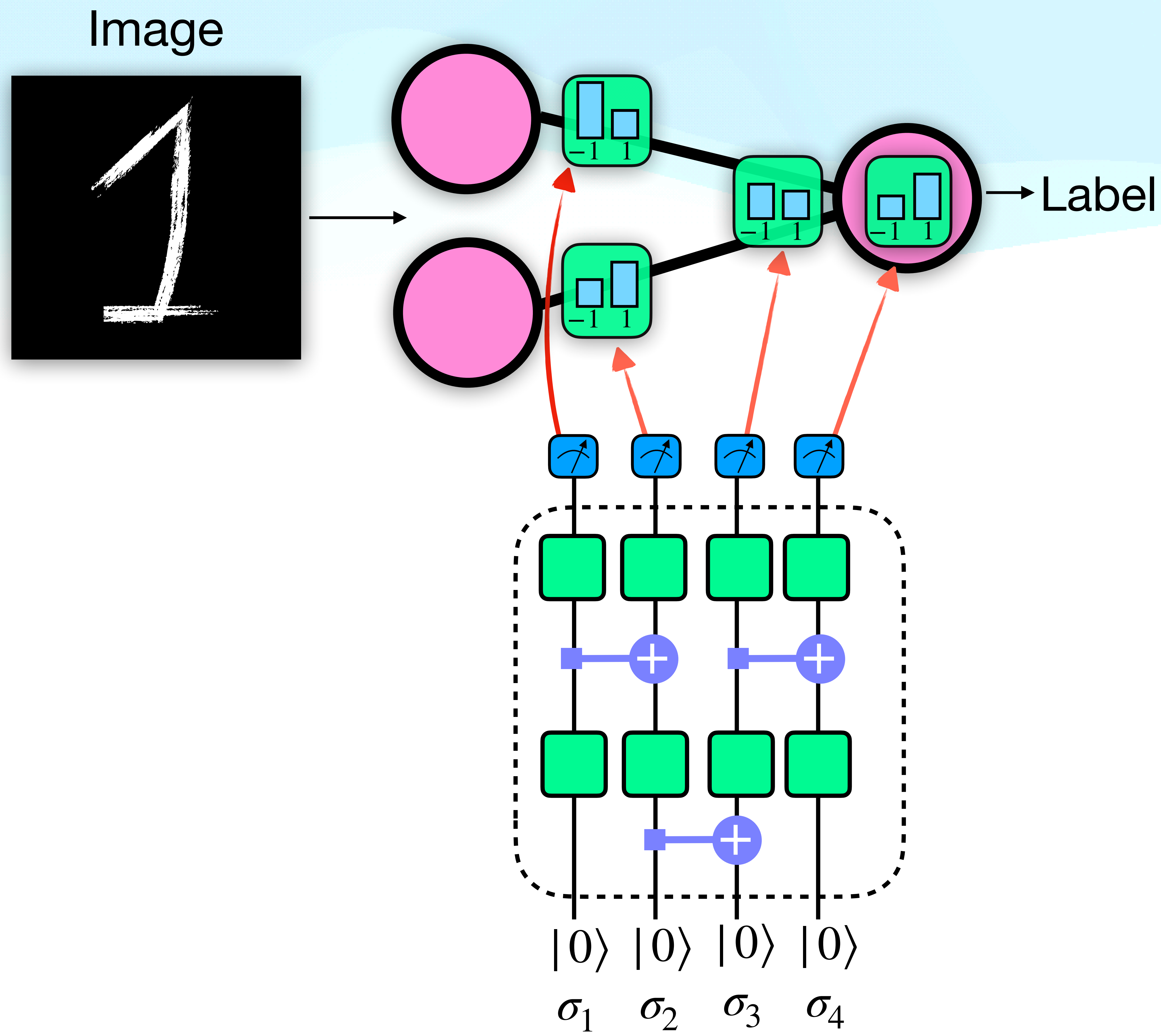
Variational quantum algorithm

Encode the problem in a form suitable to optimization by a variational quantum algorithm

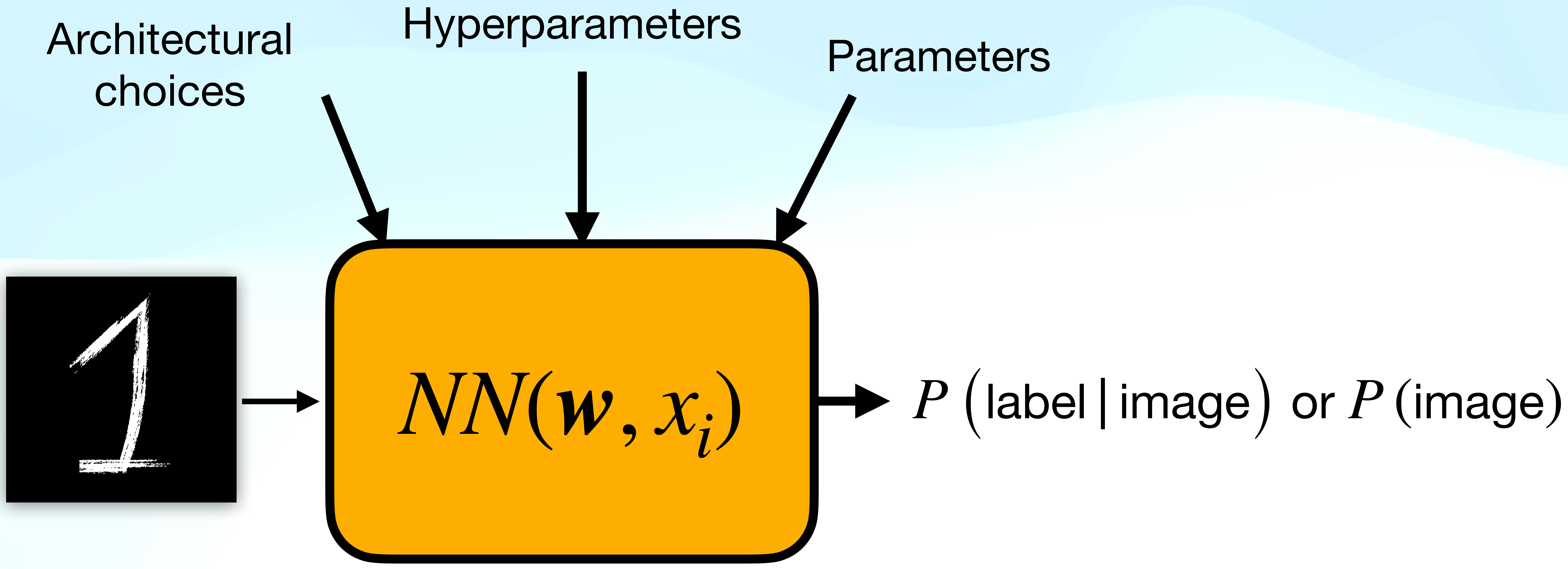
$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi\left(\sigma_1,\ldots,\sigma_4\right)$$



- One idea: a variational quantum algorithm (VQA).
- A VQA employs a classical optimizer acting on a parameterized quantum circuit, with the purpose of finding solutions to a problem encoded in an objective function.
- Objective: $C(\boldsymbol{w}) = \dfrac{1}{N_t}\displaystyle\sum_{i=1}^{N_t} \mathscr{L}\left(\mathrm{NN}(\boldsymbol{x}_i; \{\boldsymbol{w}\}), \boldsymbol{y}_i\right).$
- We define an augmented model with parameters $\boldsymbol{w} = \{w_1, \ldots, w_N\}$ to include the neural network weights, biases, hyperparameters, and architectural choices.



Augmented model



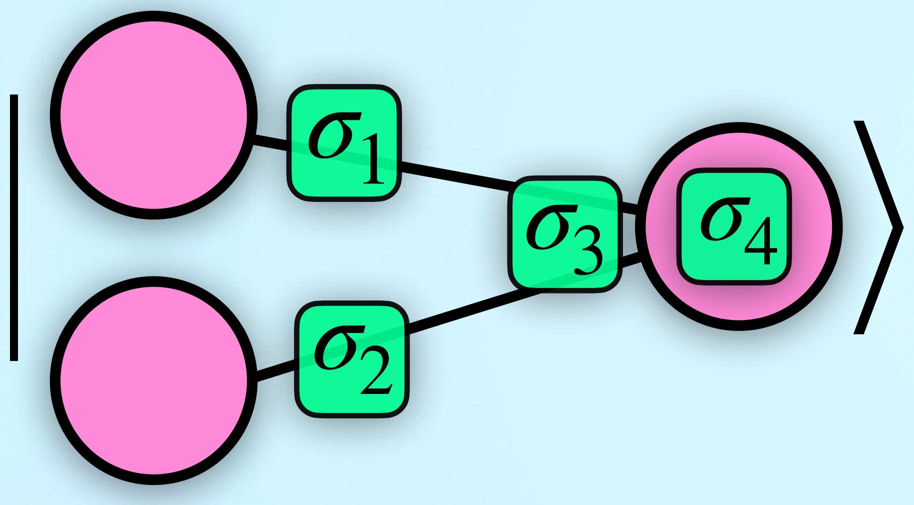
Variational quantum algorithm

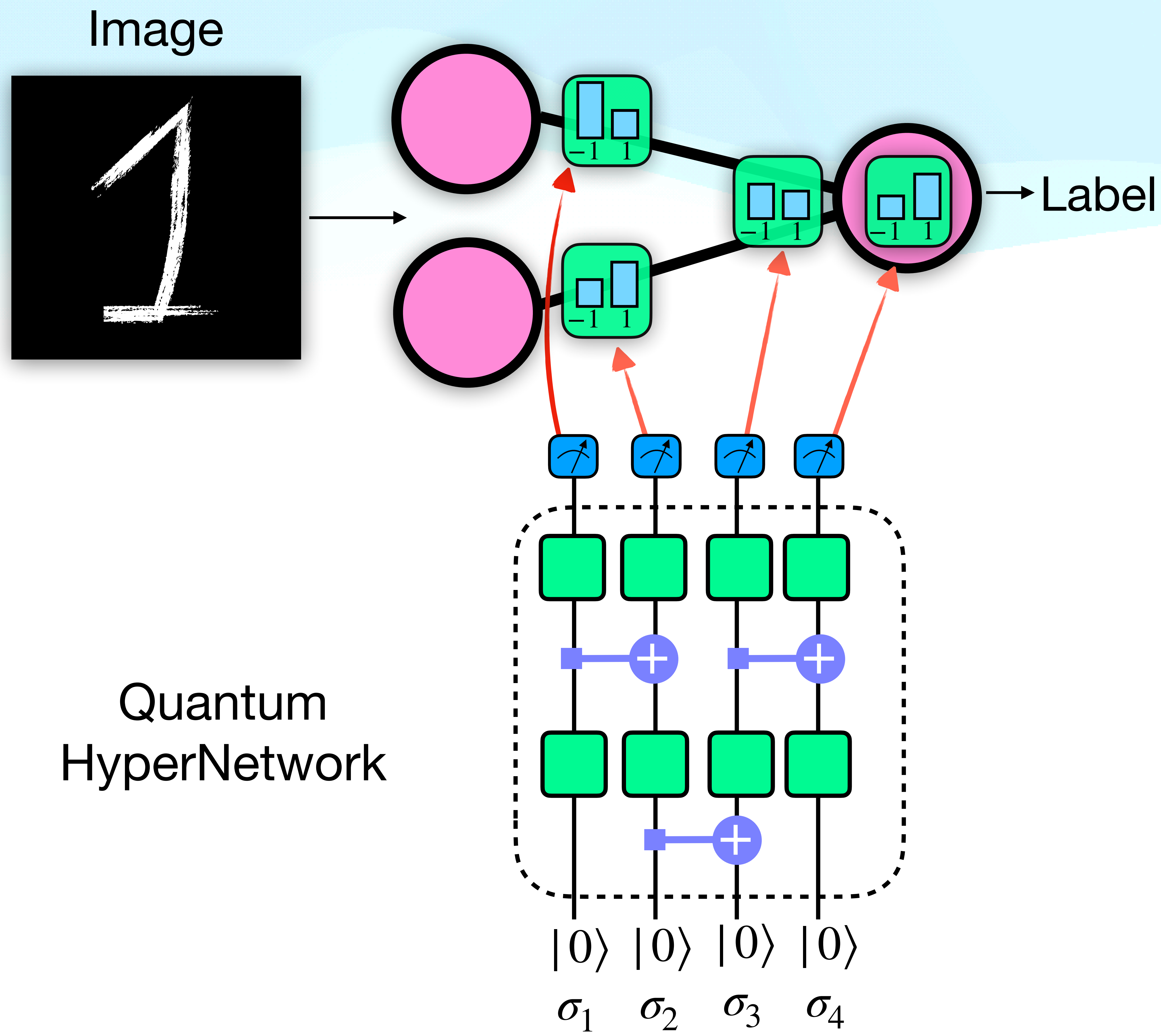
Encode the problem in a form suitable to optimization by a variational quantum algorithm

- Making the objective function quantum
- Promote the parameters of the BiNN to a set of Pauli matrices
  $\boldsymbol{w} \rightarrow \hat{\boldsymbol{\sigma}}_z = \left(\hat{\sigma}_1^z, \hat{\sigma}_2^z, \ldots, \hat{\sigma}_N^z\right),$
- $C(\boldsymbol{w}) \rightarrow \hat{C}$ (i.e. go from a Boolean function to a $2^N \times 2^N$diagonal matrix).
- This encoding is flexible — off-diagonal operators, multi-basis encoding, tensor product of Pauli operators, etc.

Variational Quantum Optimization with Multi-Basis Encodings. Taylor L. Patti, Jean Kossaifi, Anima Anandkumar, Susanne F. Yelin. https://arxiv.org/abs/2106.13304

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi\left(\sigma_1,\ldots,\sigma_4\right)$$

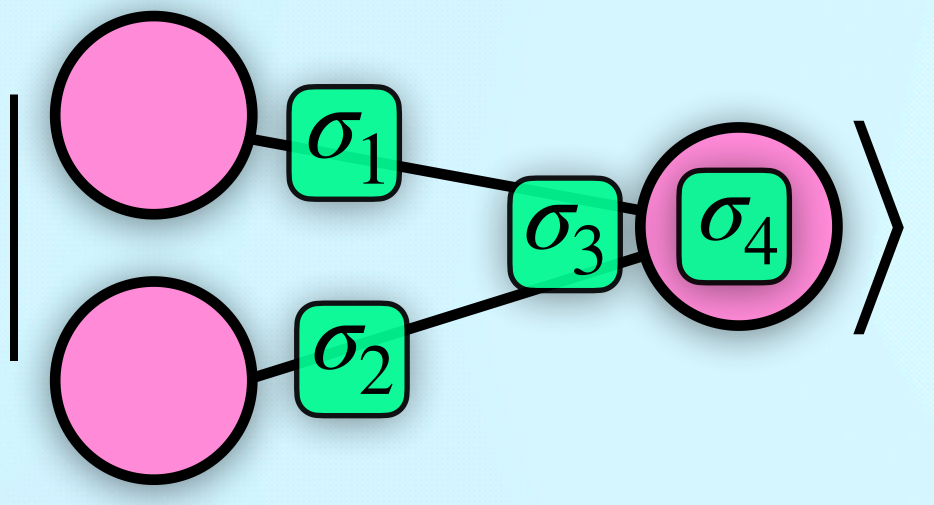




Variational quantum algorithm

Encode the problem in a form suitable to optimization by a variational quantum algorithm

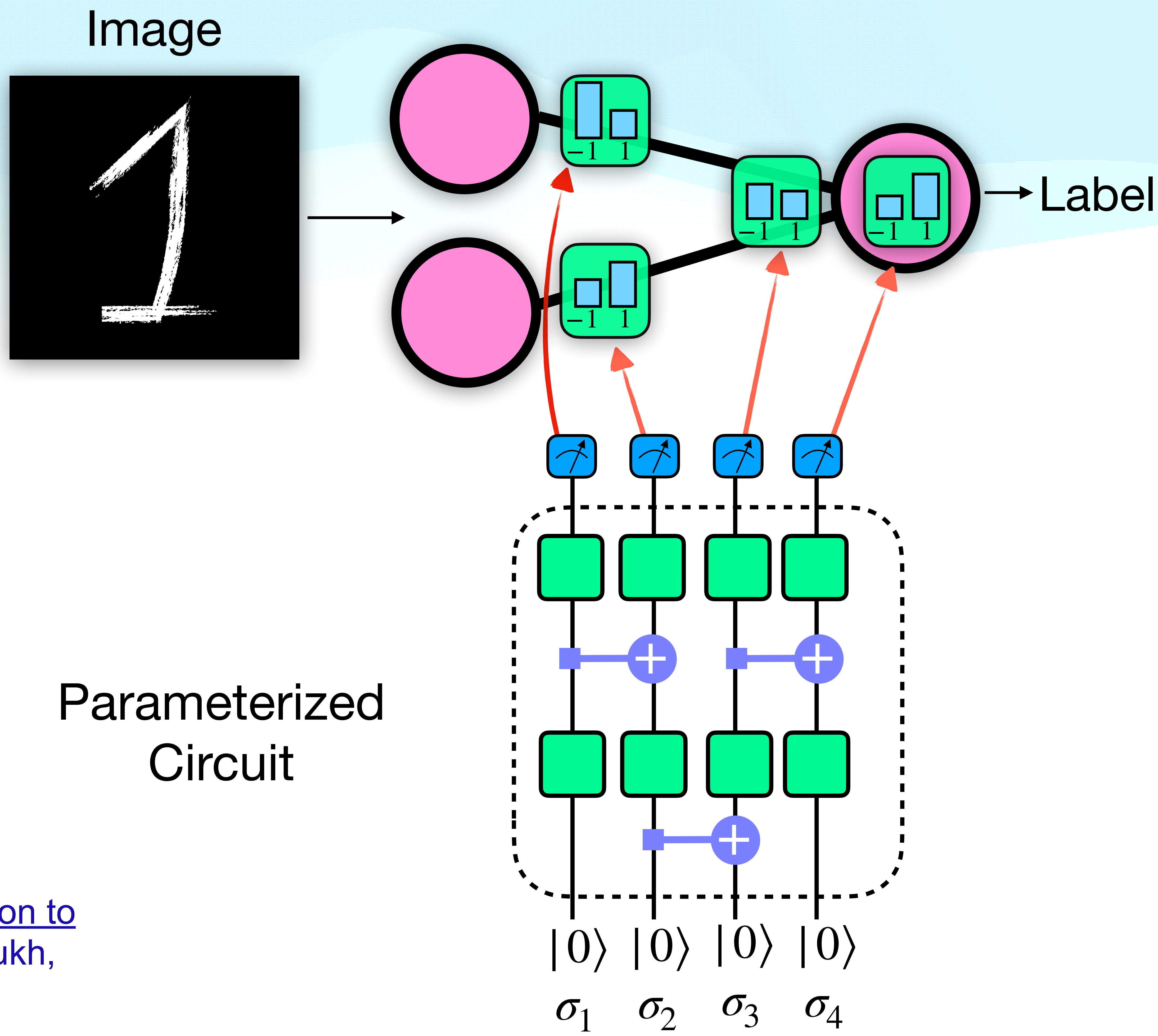
$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi\left(\sigma_1, \ldots, \sigma_4\right)$$



- We construct a quantum state $|\Psi\rangle$ through a parameterized quantum circuit $U(\boldsymbol{\theta})$ with continuous parameters $\boldsymbol{\theta}$ such that $|\Psi\rangle \to |\Psi_{\boldsymbol{\theta}}\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}$
- We aim at finding solutions to the training of the BiNN solving for
- $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} E\left(\boldsymbol{\theta}\right),$

$$\begin{aligned} E\left(\boldsymbol{\theta}\right) &= \langle \Psi_{\boldsymbol{\theta}} | \hat{C} | \Psi_{\boldsymbol{\theta}} \rangle \qquad (6) \\ &= \sum_{\sigma_1,\sigma_2,\ldots,\sigma_N} |\Psi_{\boldsymbol{\theta}}(\sigma_1,\sigma_2,\ldots,\sigma_N)|^2 C(\sigma_1,\sigma_2,\ldots,\sigma_N) \\ &= \mathbb{E}_{\boldsymbol{\sigma}\sim|\Psi_{\boldsymbol{\theta}}|^2}\left[C(\boldsymbol{\sigma})\right] \approx \frac{1}{N_s}\sum_{i=1}^{N_s} C(\boldsymbol{\sigma}_i), \end{aligned}$$

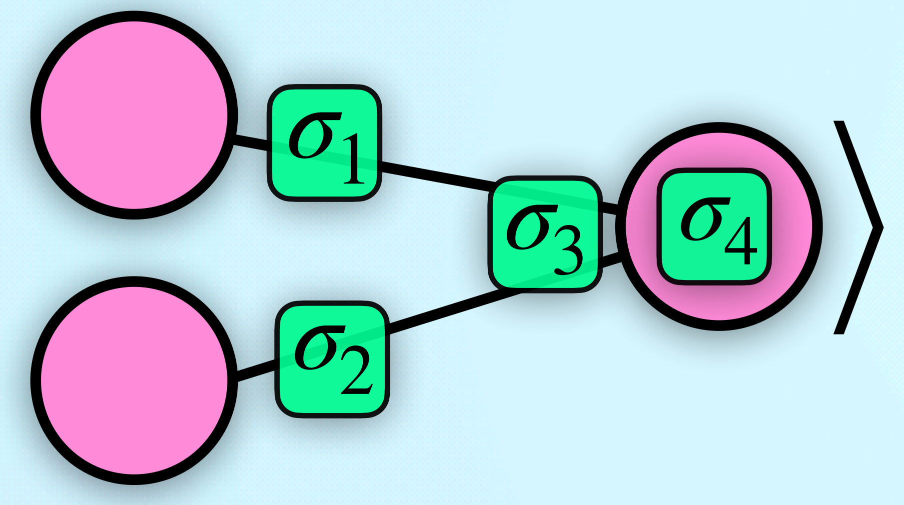
- [Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection](). C Zoufal, RV Mishmash, N Sharma, N Kumar, A Sheshadri, A Deshmukh, Noelle Ibrahim, Julien Gacon, and Stefan Woerner. Quantum 7, 909 (2023)



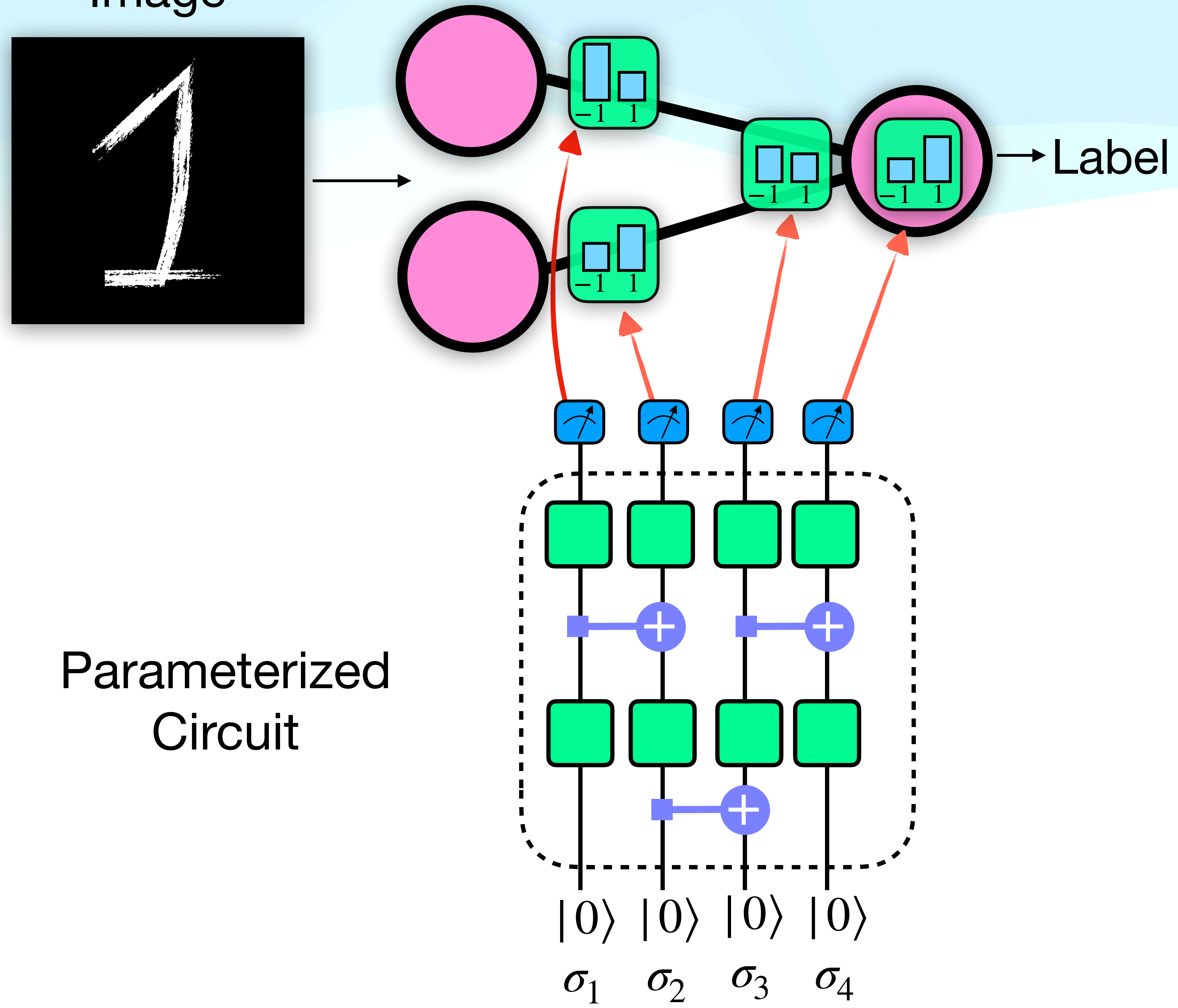
Variational quantum algorithm

Encode the problem in a form suitable to optimization by a variational quantum algorithm

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi\left(\sigma_1,\ldots,\sigma_4\right) \Bigg|$$



$$\begin{aligned} E\left(\boldsymbol{\theta}\right) &= \langle \Psi_{\boldsymbol{\theta}} | \hat{C} | \Psi_{\boldsymbol{\theta}} \rangle \qquad (6) \\ &= \sum_{\sigma_1,\sigma_2,\ldots,\sigma_N} |\Psi_{\boldsymbol{\theta}}(\sigma_1,\sigma_2,\ldots,\sigma_N)|^2 C(\sigma_1,\sigma_2,\ldots,\sigma_N) \\ &= \mathbb{E}_{\boldsymbol{\sigma}\sim|\Psi_{\boldsymbol{\theta}}|^2}\left[C(\boldsymbol{\sigma})\right] \approx \frac{1}{N_s}\sum_{i=1}^{N_s} C(\boldsymbol{\sigma}_i), \end{aligned}$$

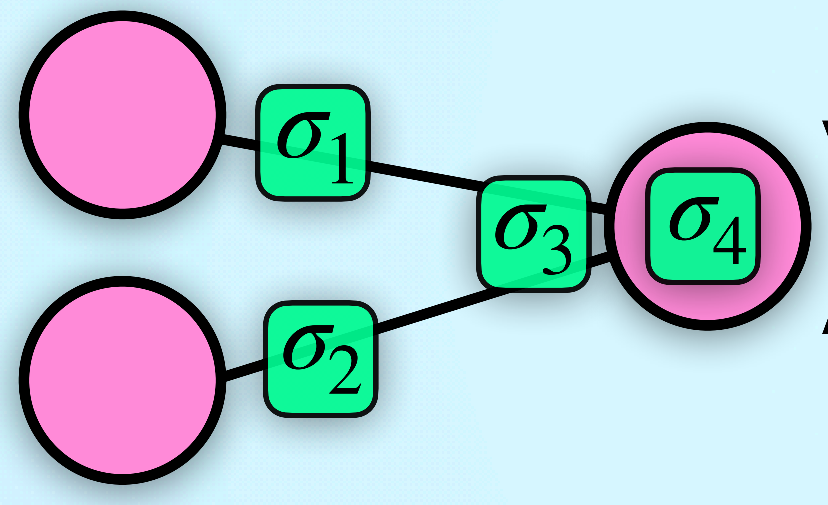


- From an ML perspective—stochastic relaxation of the discrete optimization problem. This is close to a Bayesian BiNN with a “quantum” approximating posterior.
- Instead of optimizing binary variables, optimize continuous parameters $\boldsymbol{\theta}$.

Variational quantum algorithm

Encode the problem in a form suitable to optimization by a variational quantum algorithm

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi\left(\sigma_1,\ldots,\sigma_4\right) \Big| \cdots \Big\rangle$$

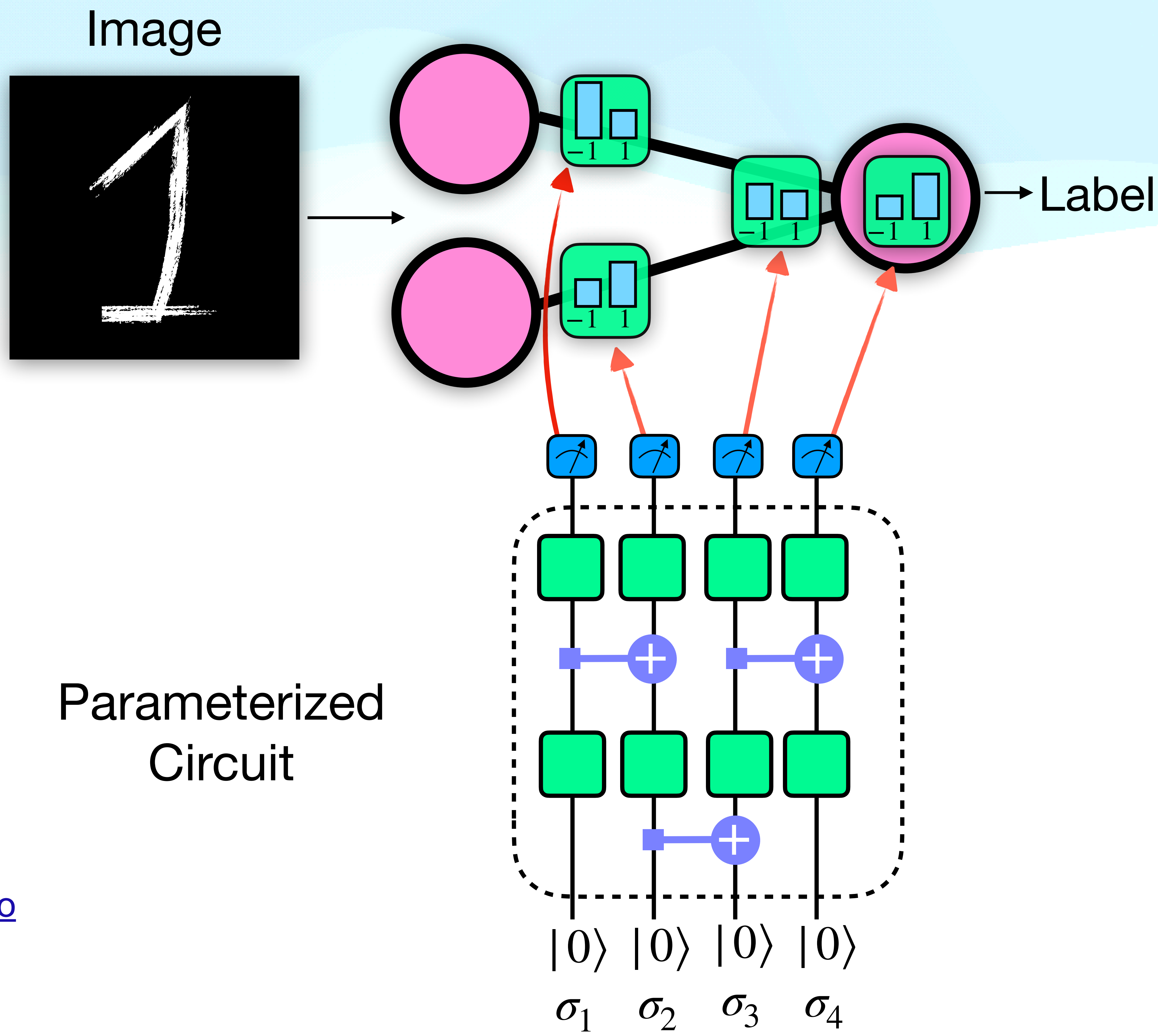


- Design of the circuit is important. Depth, connectivity of the gates etc.

- We don’t know a whole lot about C— How to choose a good ansatz?

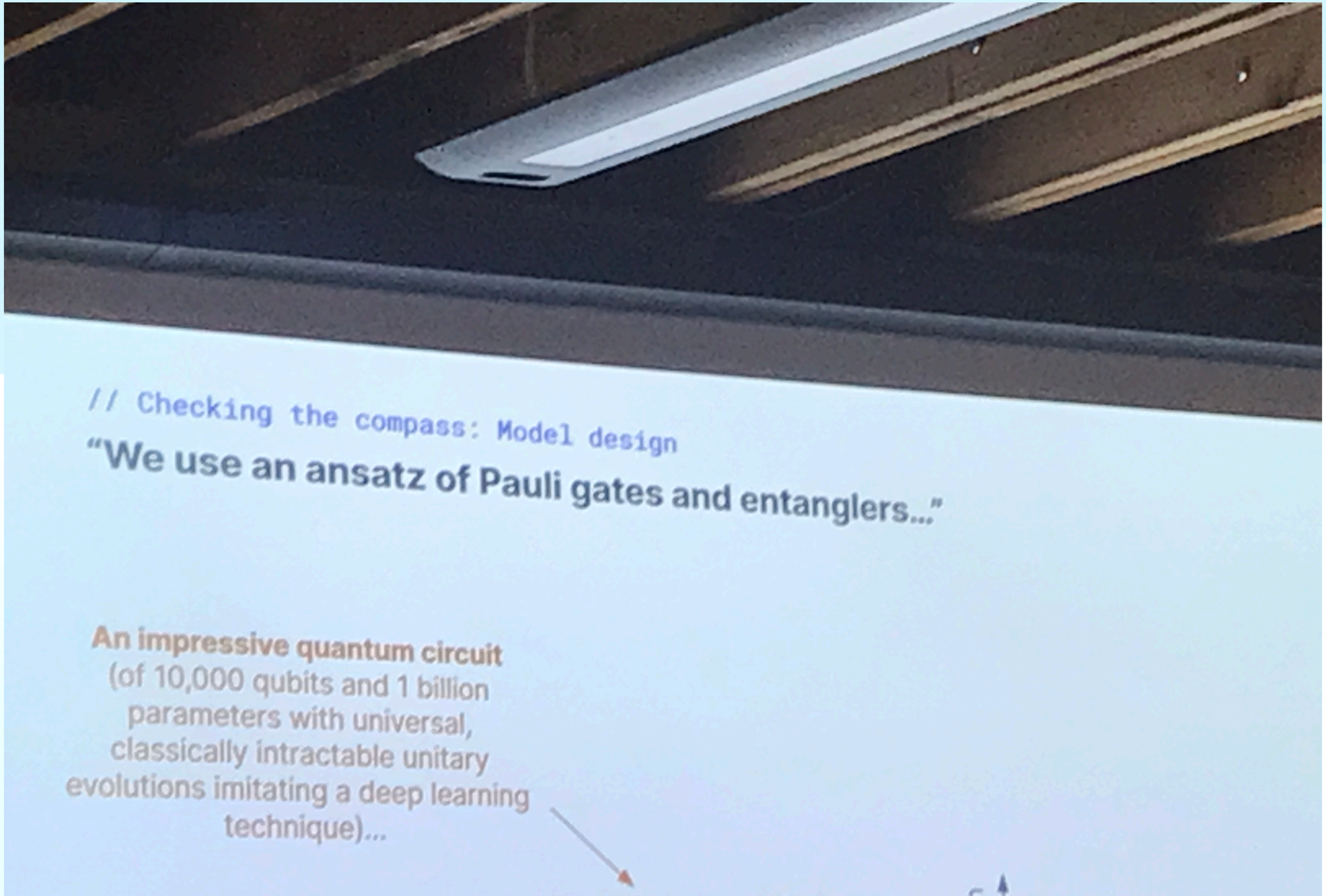
- People use QAOA $e^{i\alpha\hat{C}} e^{\gamma\sum_i \sigma_i^X}$ — hard to use as need to compile $e^{i\alpha\hat{C}}$

- We choose a circuit with linear connectivity and vary its depth.



Variational quantum algorithm for unconstrained black box binary optimization: Application to feature selection. C Zoufal, RV Mishmash, N Sharma, N Kumar, A Sheshadri, A Deshmukh, Noelle Ibrahim, Julien Gacon, and Stefan Woerner. Quantum 7, 909 (2023)

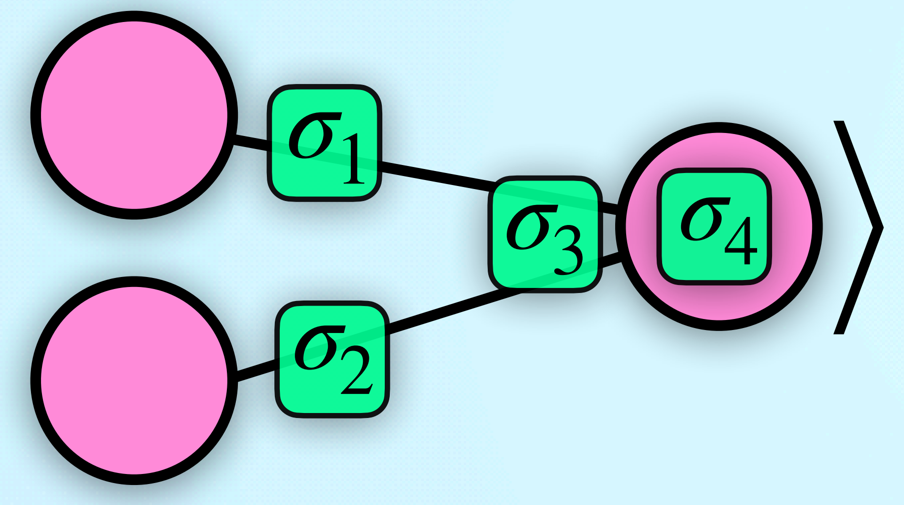
How do we choose ansatz?



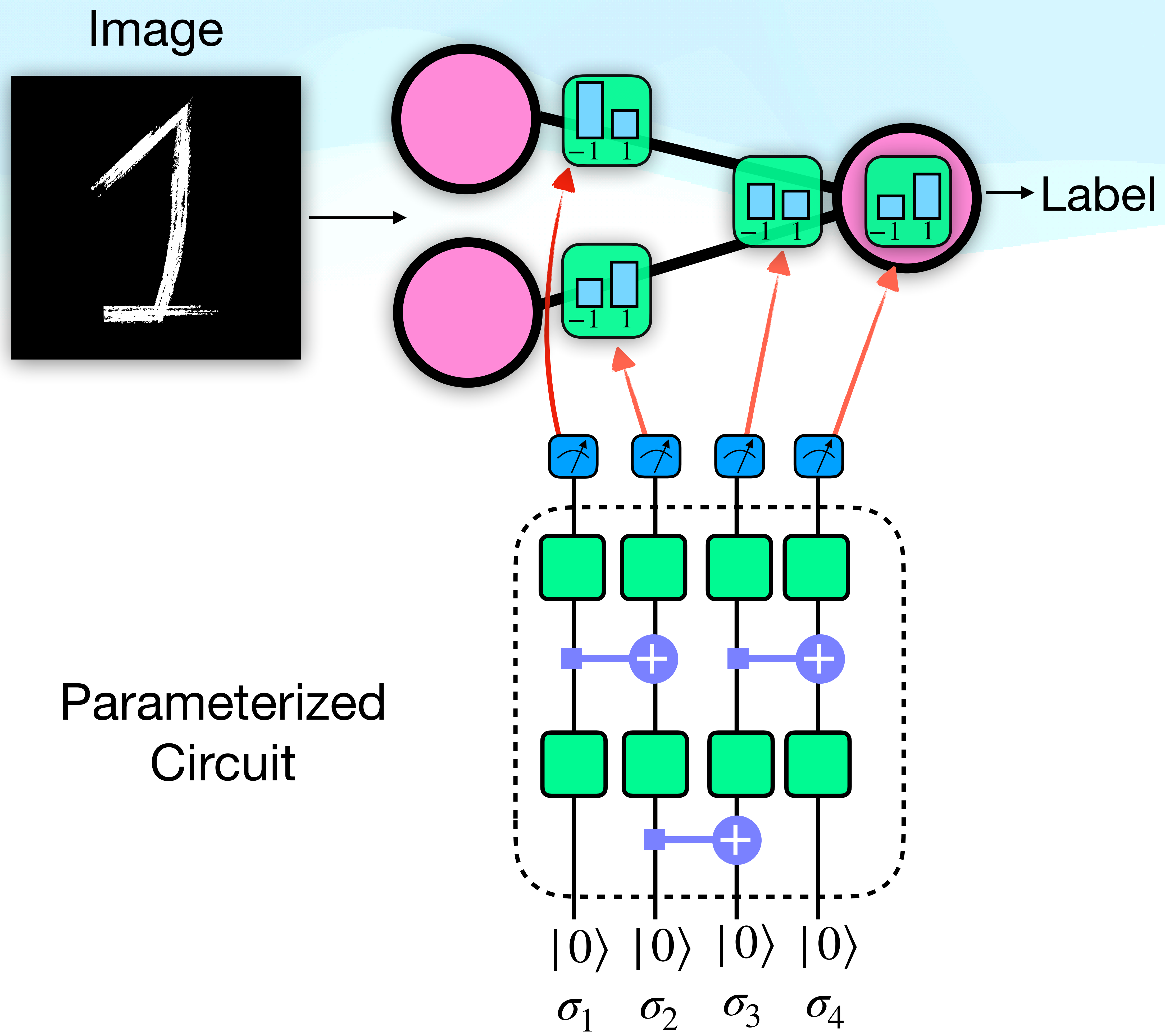
Optimization

Encode the problem in a form suitable to optimization by a variational quantum algorithm

$$|\Psi\rangle = \sum_{\sigma_1,\ldots,\sigma_4} \Psi(\sigma_1,\ldots,\sigma_4)\Big|$$

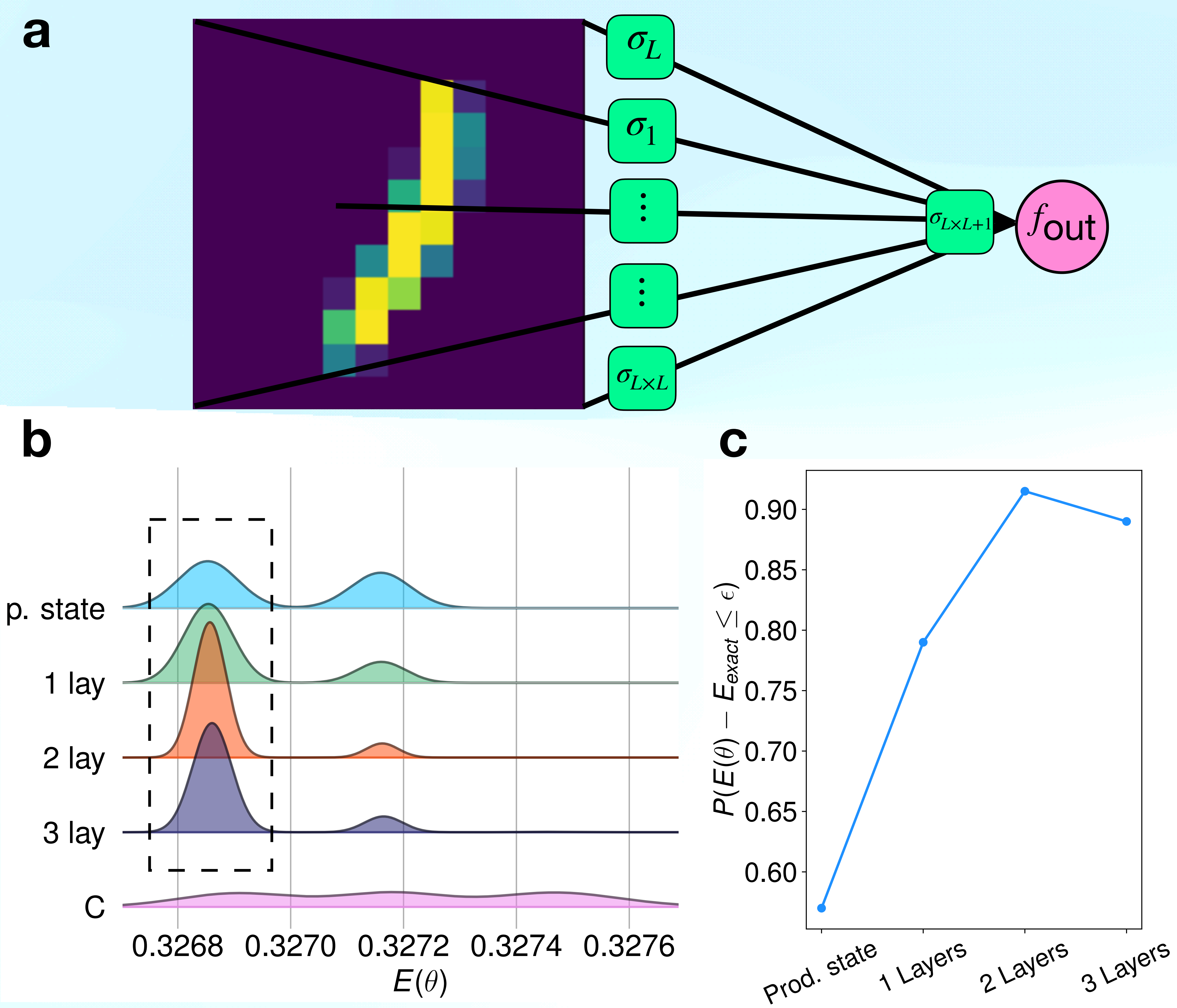


- Use gradient descent to optimize $E(\boldsymbol{\theta})$
- $$\frac{\partial E(\boldsymbol{\theta})}{\partial \theta_{\alpha,j,k}} = \frac{1}{2}\left[E(\boldsymbol{\theta}^{+}_{\alpha,j,k}) - E(\boldsymbol{\theta}^{-}_{\alpha,j,k})\right],$$
- The shifted parameter vector $\boldsymbol{\theta}^{\pm}_{\alpha jk}$ is such that $\theta^{\pm}_{\beta,i,l} = \theta_{\beta,i,l} \pm \frac{\pi}{2}\delta_{\alpha,\beta}\delta_{i,j}\delta_{k,l}$
- Thus, the calculation of the gradient corresponds to the evaluation of a shifted version of the objective function $E(\boldsymbol{\theta})$.
- However, we used a tensor network simulator and automatic differentiation.



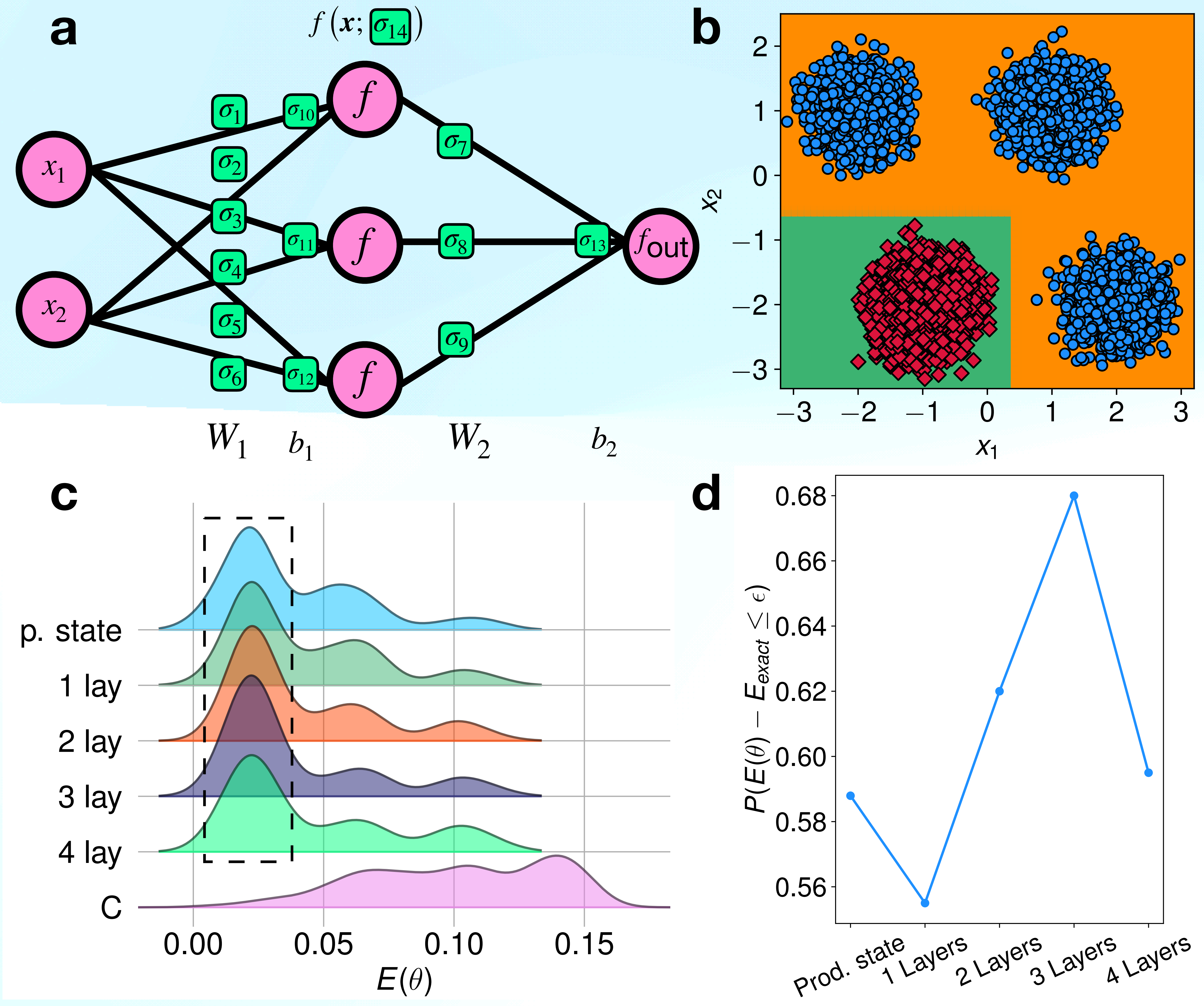
Some results

Results: MNIST, binary logistic regression

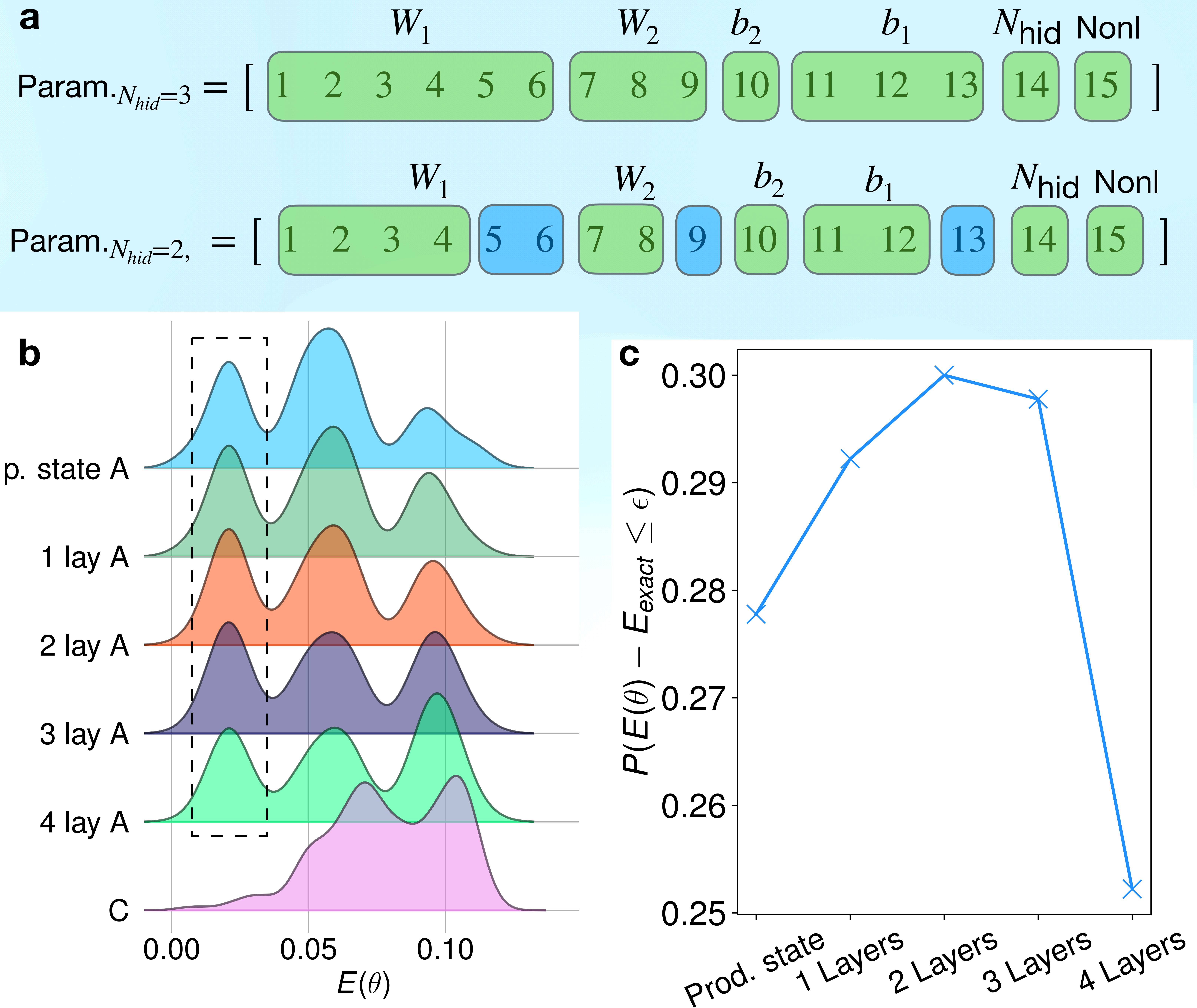


- Train weights and bias.
- Run optimization at least 200 times and evaluate the probabilities of finding an objective function with value $E(\boldsymbol{\theta})$
- Compute Probability that $E(\boldsymbol{\theta})$ is less than ϵ.
- Optimization is successful frequently
- Optimal circuit depth suggests an optimal use of entanglement

Results



- Train weights + architectural choice of non-linearity.
- Run optimization at least 200 times and evaluate the probabilities of finding an objective function with value $E(\boldsymbol{\theta})$
- Compute Probability that $E(\boldsymbol{\theta})$ is less than
- Optimization is successful frequently
- Optimal circuit depth— optimal use of entanglement

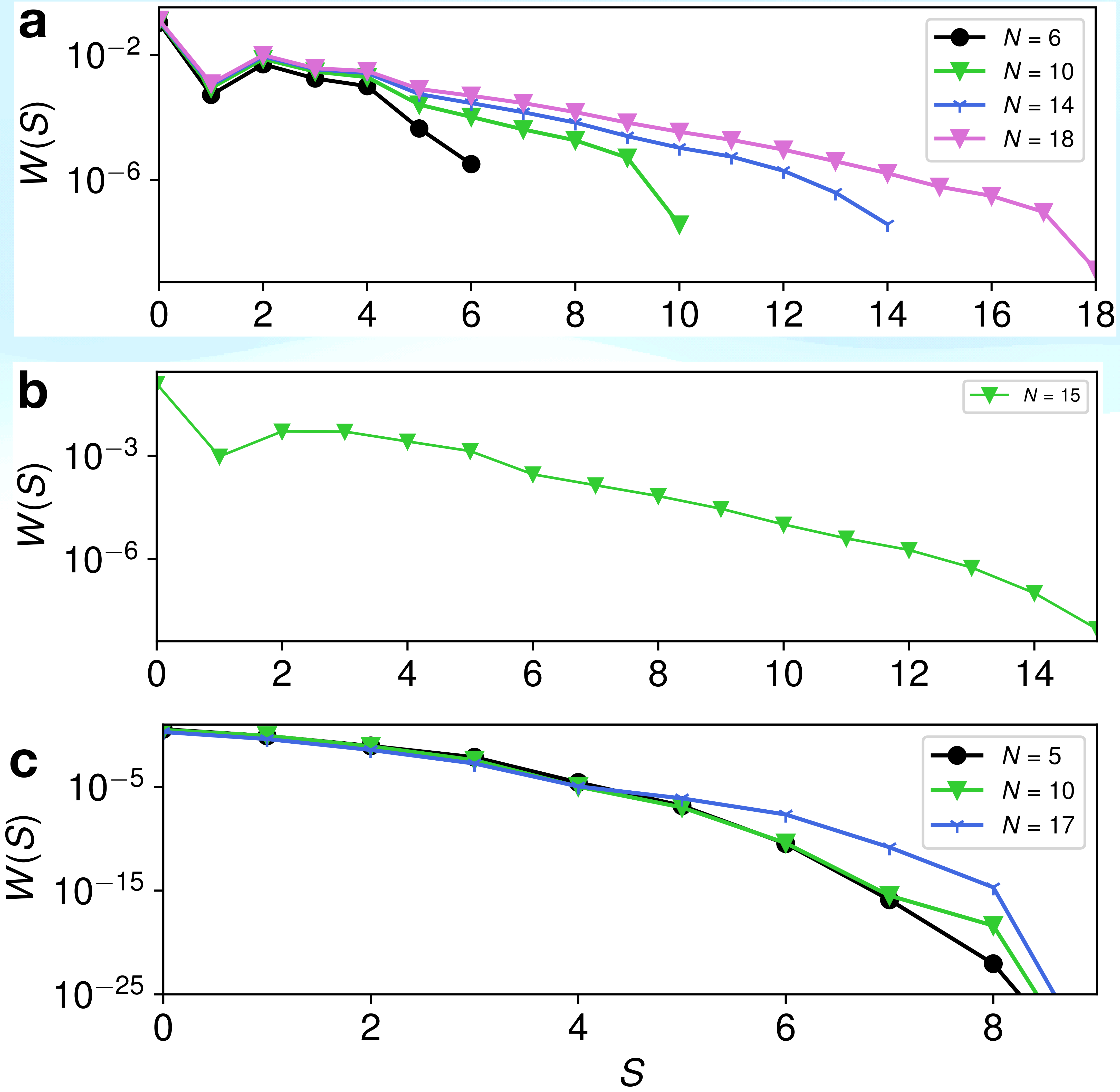


- Train weights + architectural choice of non-linearity + hidden dimension (2 or 3, binary choice)
- Optimal circuit depth
- Success probability a bit smaller
- But overall successful optimization

Any idea about the structure of C

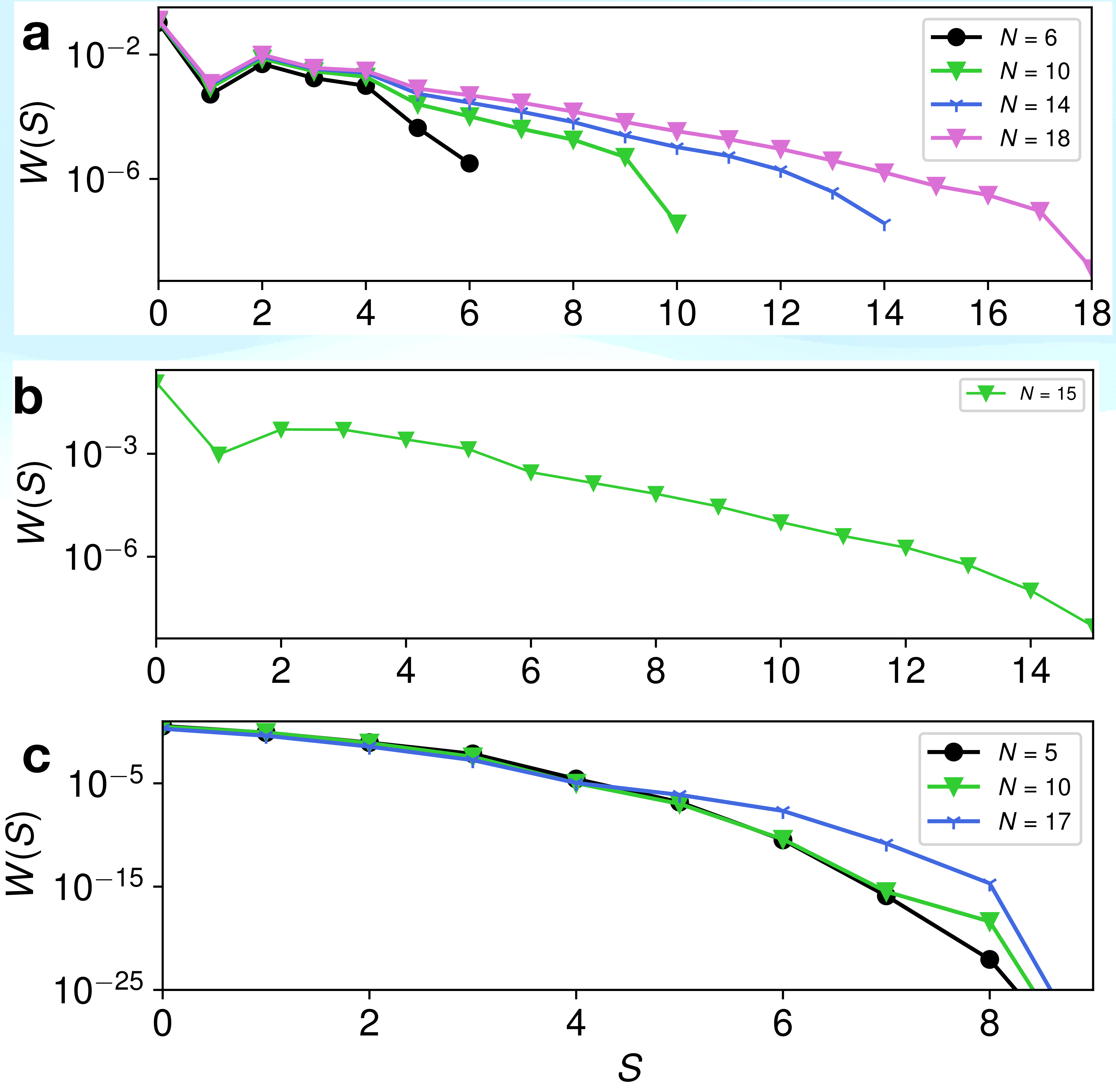
Fourier Analysis

- $$\hat{C} = \sum_{\hat{\sigma}_1,\ldots\hat{\sigma}_N} f(\hat{\sigma}_1,\ldots\hat{\sigma}_N) \bigotimes_{i=1}^{N} \hat{\sigma}_i$$
- Effective Ising model with multi-variable all-to-all interactions
- Fourier coefficients are given by $f(\hat{\sigma}_1,\ldots\hat{\sigma}_N) = \frac{1}{2^N}\mathrm{Tr}\left[\hat{C} \otimes_{i=1}^{N} \hat{\sigma}_i\right] \in \mathbb{R}$
- $$W(S) = \sum_{\hat{\sigma}_1,\ldots\hat{\sigma}_N} |f(\hat{\sigma}_1,\ldots\hat{\sigma}_N)|^2 \delta_{S,S(\hat{\sigma}_1,\ldots\hat{\sigma}_N)}\,.$$



Fourier Analysis

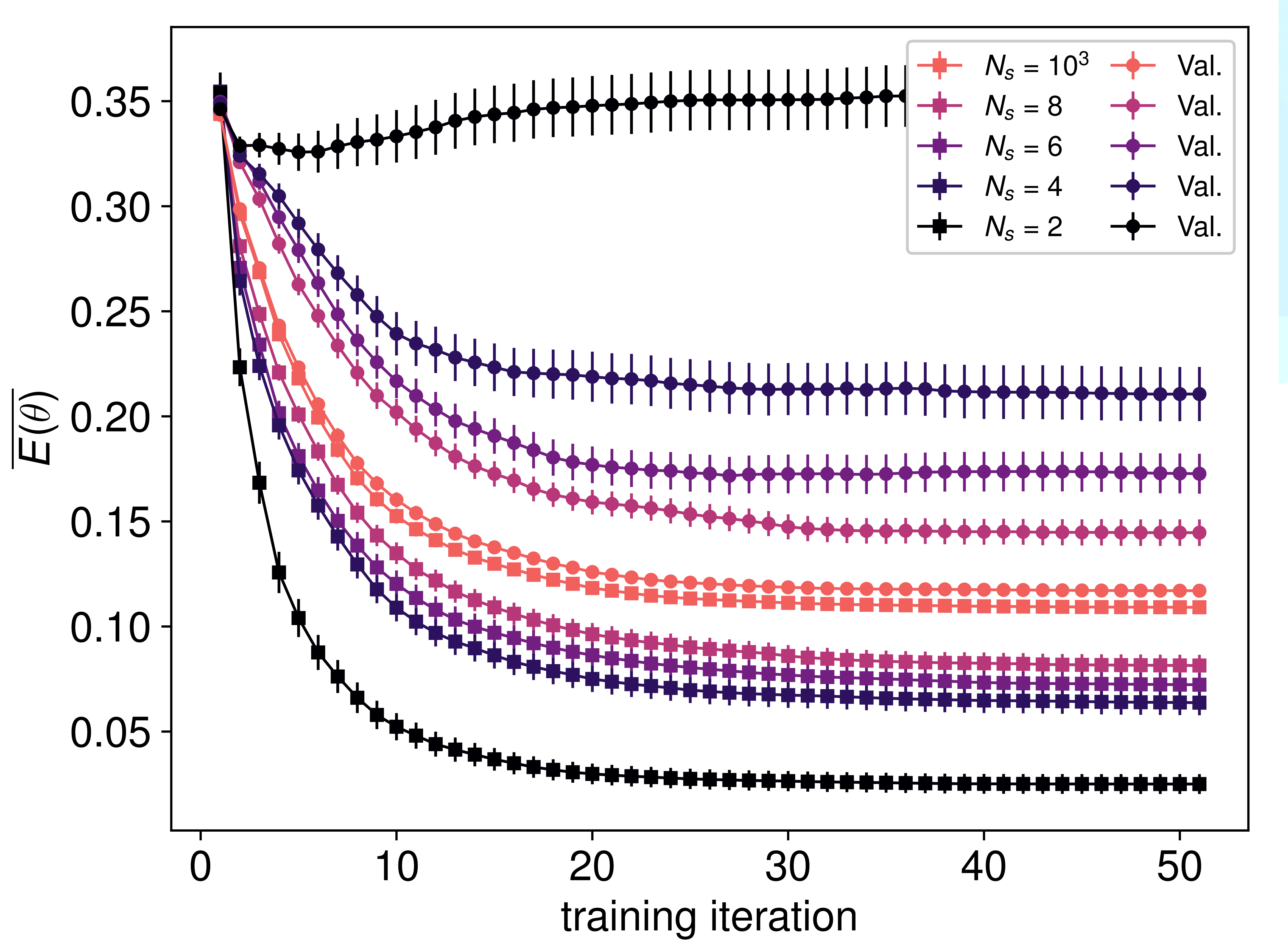
- Data suggest that the objective functions C are predominantly local
- MNIST C is nearly single-body, which is easier to optimize even by a product state
- No tractability issues arising from vanishing gradients induced by C



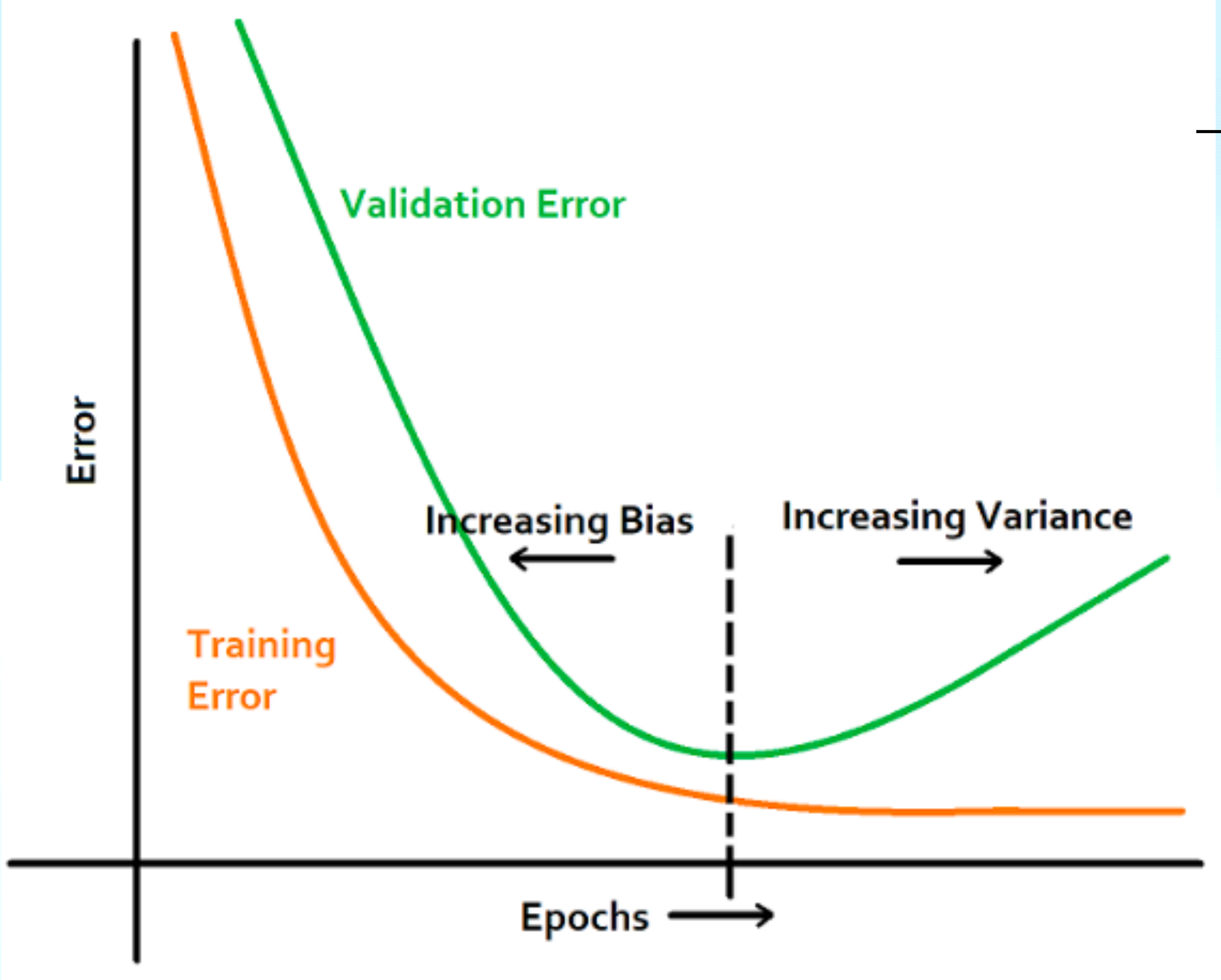
“Augmented” model selection

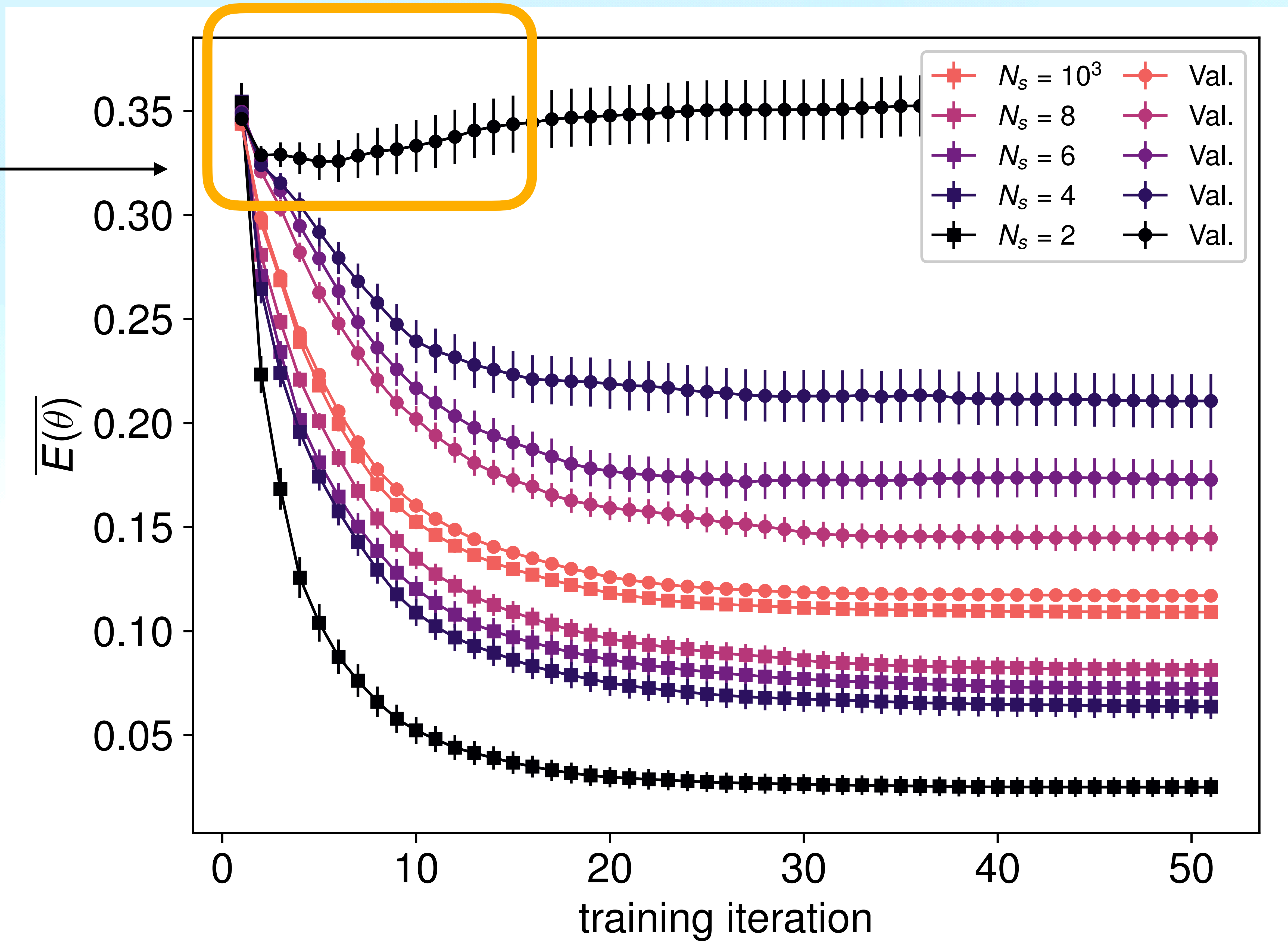
- Augmented model encapsulating the parameters, hyperparameters and architecture of a neural network which we jointly optimize on a training dataset. How to choose model using a validation set?

- The data suggests that these augmented models behave like traditional statistical models which follow the usual bias-variance decomposition.



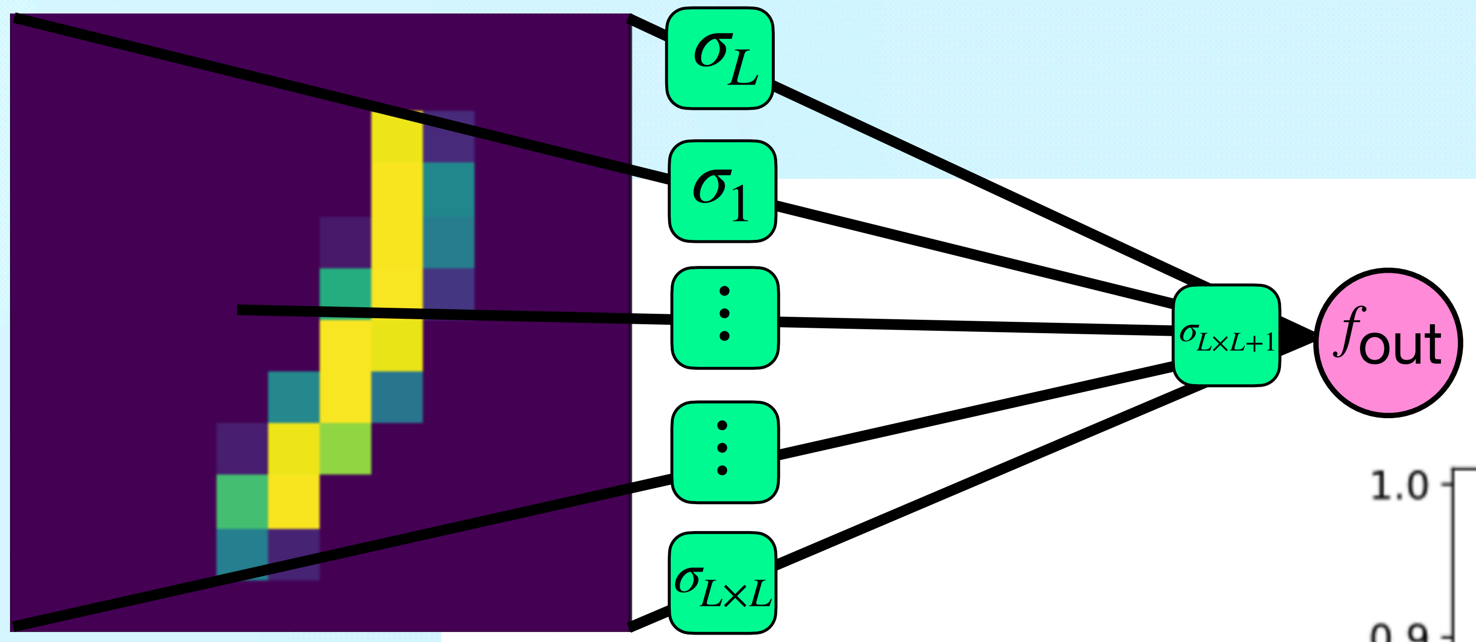
“Augmented” model selection





https://medium.com/@rahuljain13101999/why-early-stopping-works-as-regularization-b9f0a6c2772

Simulations based on Matrix product states



8X8 +1 + 1 = 66 qubits

Weights, bias, activation function choice

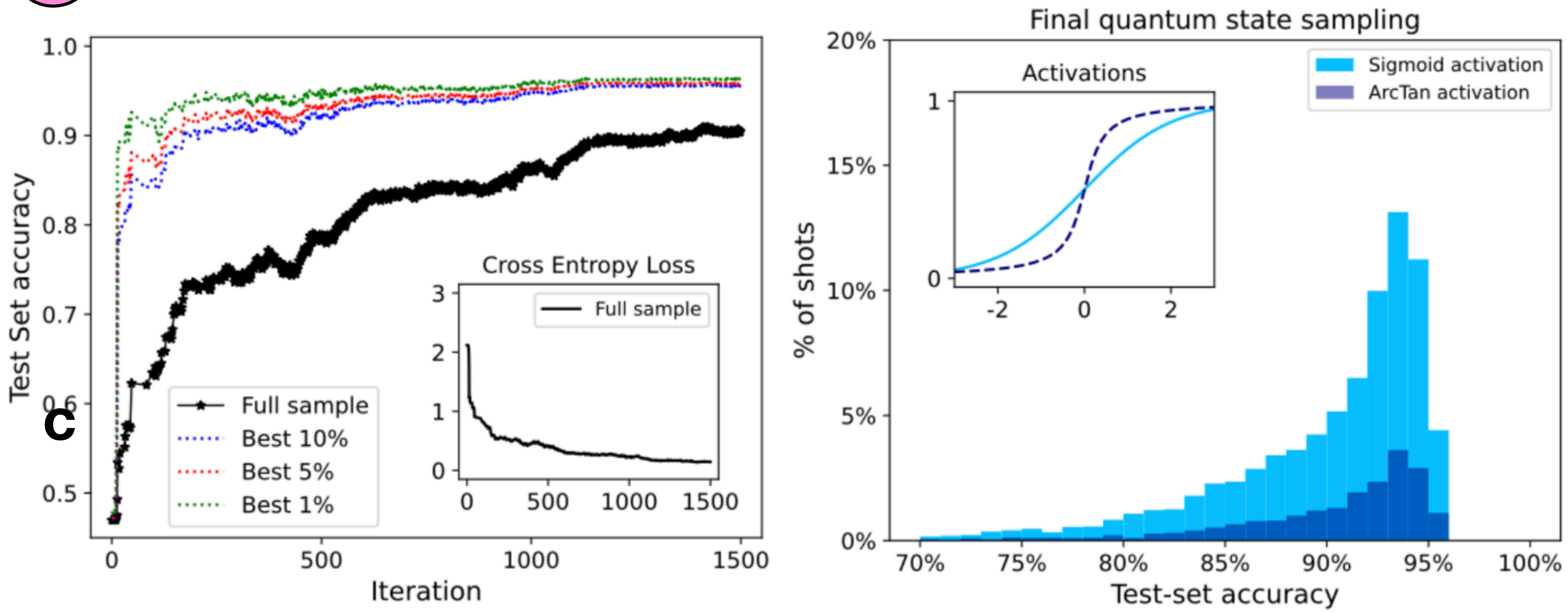


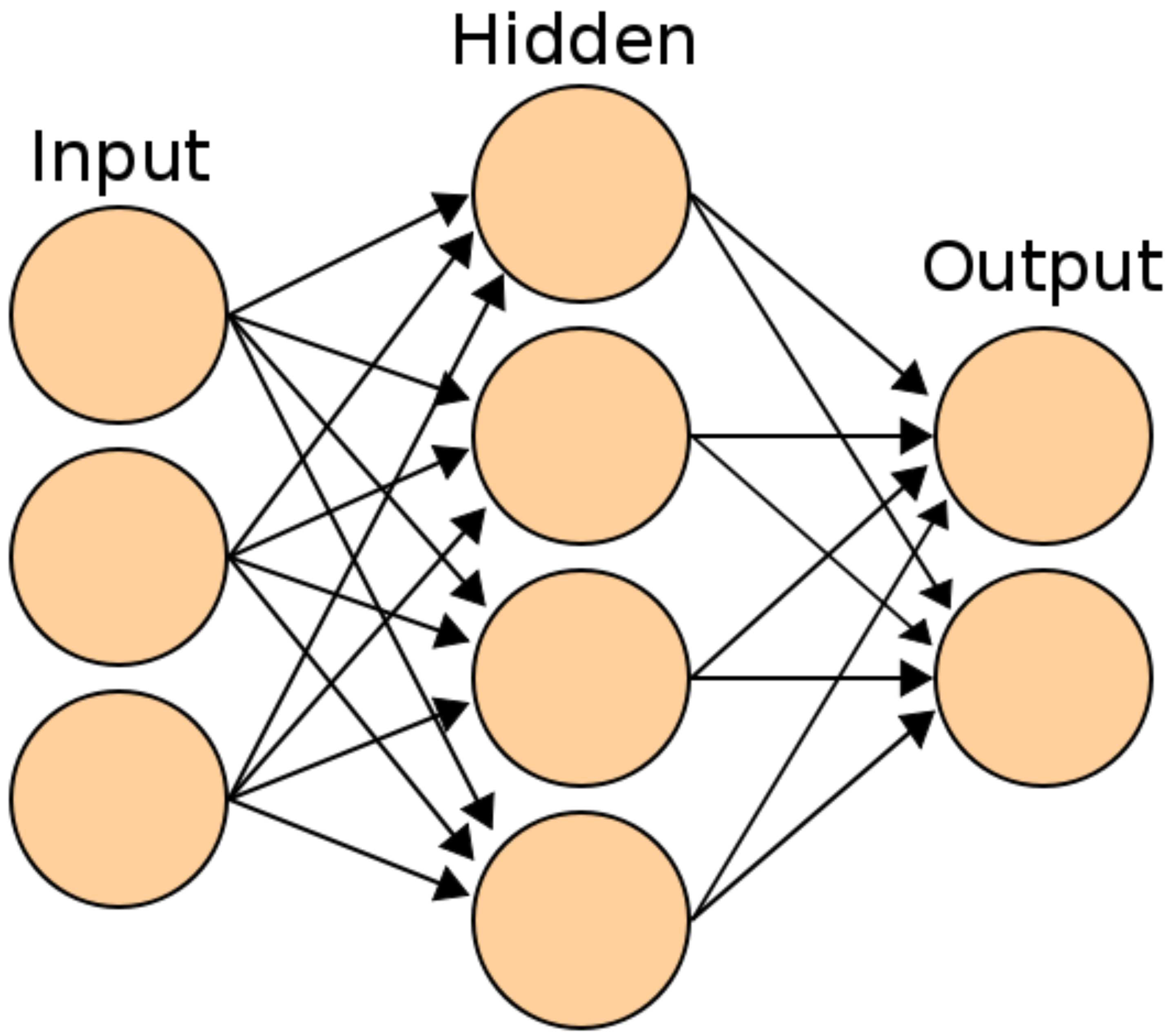
Figure 2: **Left**. Optimization of a quantum circuit with depth $P = 2$, trained to learn BNN architectures for the 8×8 MNIST (0, 1 binary classification). The solid black line shows the test set classification accuracy, averaged over the whole sample at each iteration, with symbols highlighting iterations with an accepted QN-SPSA parameter update. The dotted lines represent the same quantity, averaged over the $1, 5, 10\%$ best shots at each iteration. In the inset, we plot the average (whole-sample) binary cross-entropy loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta})$. **Right**. Sampling of the final state $|\psi(\boldsymbol{\theta}^*)\rangle$. Samples are divided depending on the outcome of the qubit encoding the hyperparameter, i.e. the activation function of the classifier. An histogram is constructed showing the test accuracy of the sampled BNNs.

Conclusions

- We have introduced HyperNetworks which train binary neural networks in quantum superposition
- One optimization loop trains parameters, hyperparameters, and architectural choices in binary neural networks
- Speculation: “quantum computers are currently reaching the ability to vastly outperform supercomputers' energy efficiency by many orders of magnitude over classical computers.”
- Horrible encoding. Z$->X, Y, Z, , X \otimes X, X \otimes Y, X \otimes Z, 1 \otimes Z,$ etc
- Binary neural networks save energy at inference time. We are suggesting is that we can potentially save energy in training, architectural design and hyperparameter search.
- Neural networks perform best when they are large— we are exploring better encoding of the problem.
- Vast arrays of problems in ML that can be recast as a black box optimization

Artificial neural networks

Artificial neural networks are a family of models
used to approximate functions that can depend on a large number of inputs.
Artificial neural networks are generally presented as systems of
interconnected "neurons" which exchange messages between each other

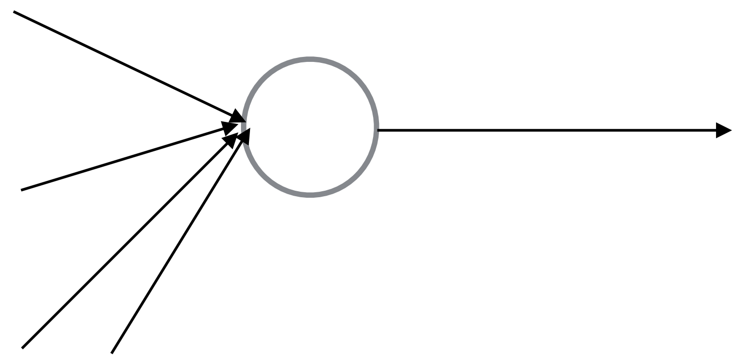


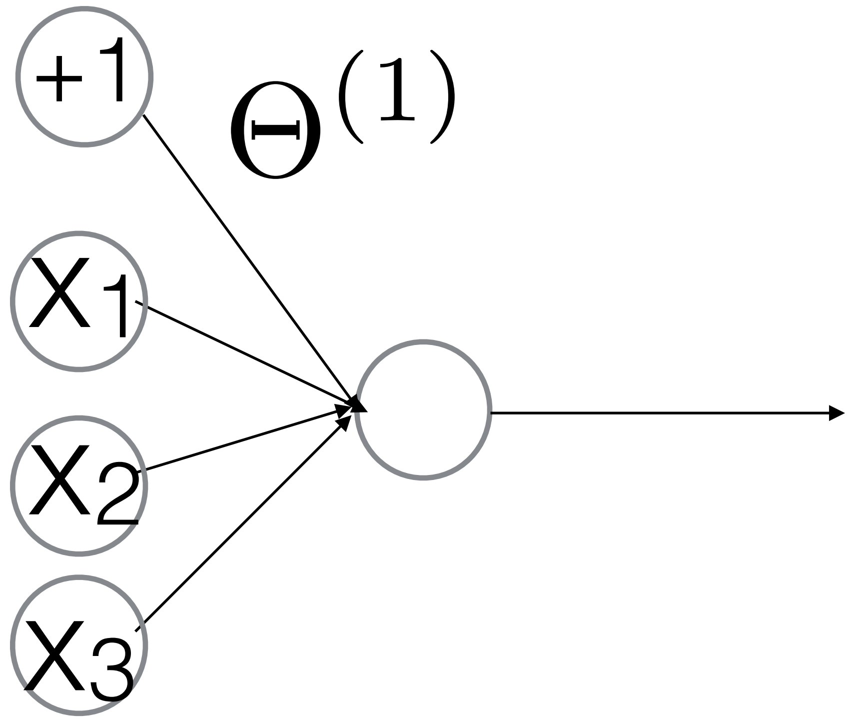
Connections= sets of adaptive weights, i.e. numerical parameters that are tuned by a learning algorithm

$$f : \mathbf{R}^n \to \mathbf{R}^m$$

Wikipedia

A neuron:





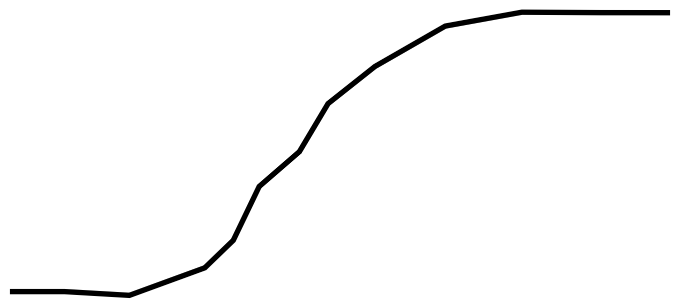
Sigmoid neuron

$$\beta = 1$$

$$h_\Theta(x) = \frac{1}{1 + e^{-\beta \Theta^T x}}$$

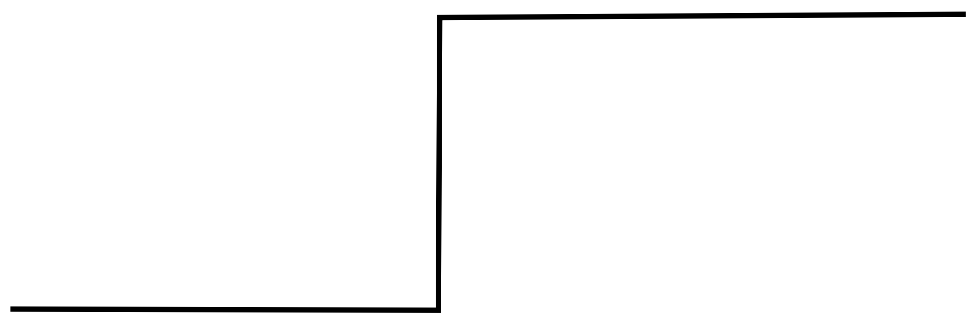
$$\Theta = (\theta_0\, \theta_1\, \theta_2\, \theta_3)$$

$$x = (1\ x_1\, x_2\, x_3)$$



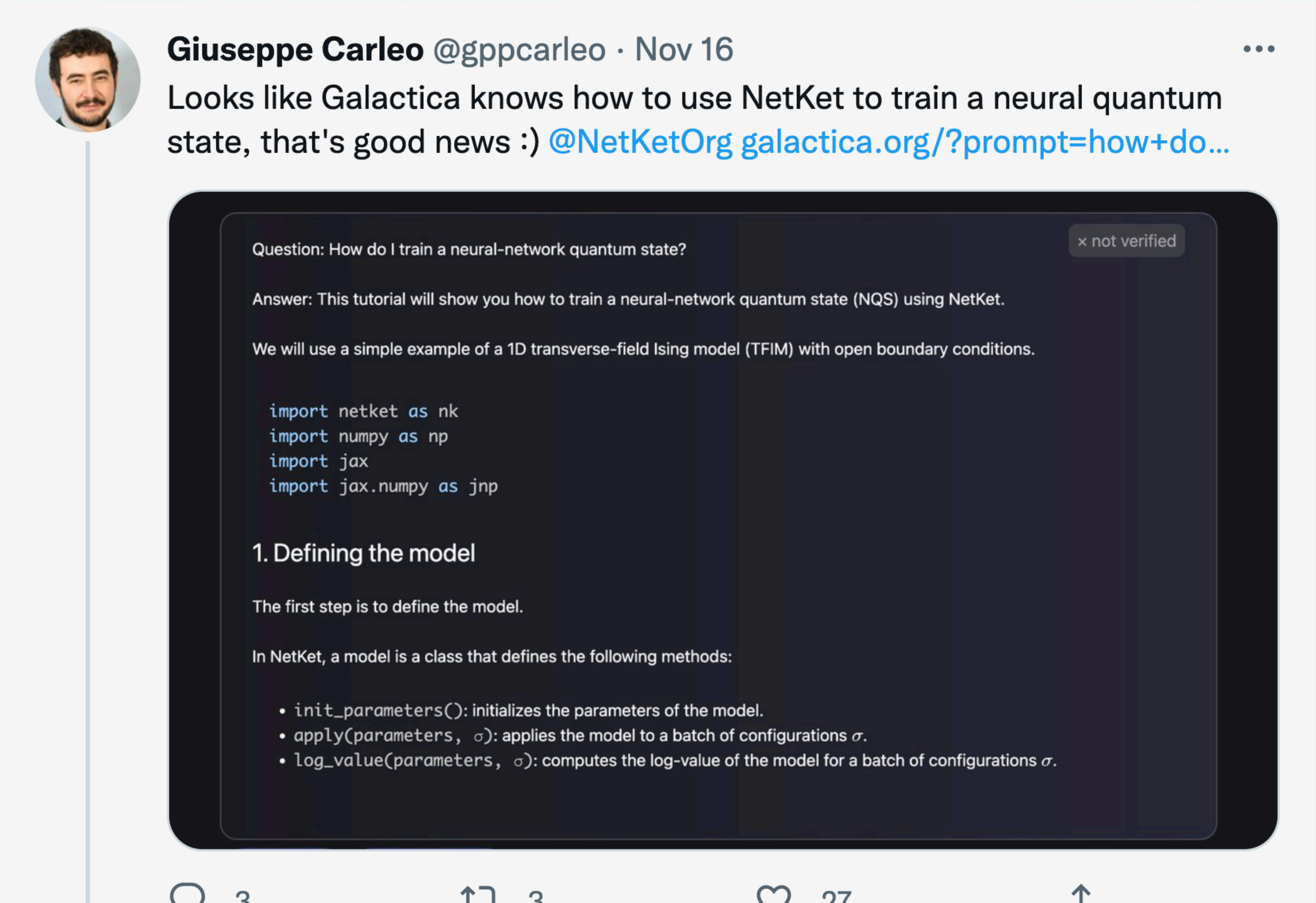
Perceptron:

$$\beta \to \infty$$



Where Θ 's are the parameters you fiddle with

Deep learning



Giuseppe Carleo @gppcarleo · Nov 16

Looks like Galactica knows how to use NetKet to train a neural quantum state, that's good news :) @NetKetOrg galactica.org/?prompt=how+do...

× not verified

Question: How do I train a neural-network quantum state?

Answer: This tutorial will show you how to train a neural-network quantum state (NQS) using NetKet.

We will use a simple example of a 1D transverse-field Ising model (TFIM) with open boundary conditions.

```
import netket as nk
import numpy as np
import jax
import jax.numpy as jnp
```

1. Defining the model

The first step is to define the model.

In NetKet, a model is a class that defines the following methods:

- `init_parameters()`: initializes the parameters of the model.
- `apply(parameters, σ)`: applies the model to a batch of configurations σ.
- `log_value(parameters, σ)`: computes the log-value of the model for a batch of configurations σ.

3 3 27