

# Learning the Spectrum with Neural Quantum States

J. Rozalén Sarmiento, A. Rios



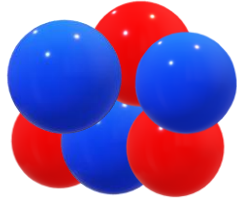
Institut de Ciències del Cosmos  
UNIVERSITAT DE BARCELONA



UNIVERSITAT DE  
BARCELONA

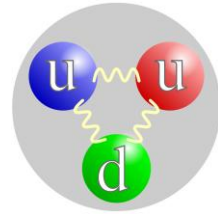
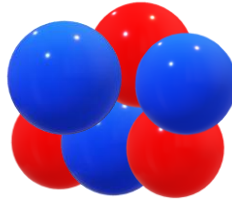
# Ab Initio Nuclear structure in a (nut)shell

1. Select your nucleus



# Ab Initio Nuclear structure in a (nut)shell

1. Select your nucleus



2. Select your Hamiltonian

$\mathcal{L}_{\text{QCD}}$

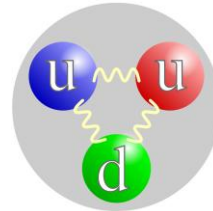
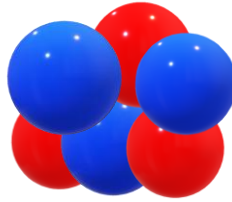


$$\hat{H}_{\text{nuc}} = \hat{T} + \hat{V}$$



# Ab Initio Nuclear structure in a (nut)shell

1. Select your nucleus



2. Select your Hamiltonian

$\mathcal{L}_{\text{QCD}}$



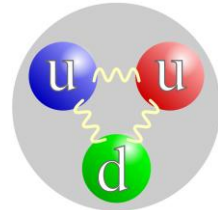
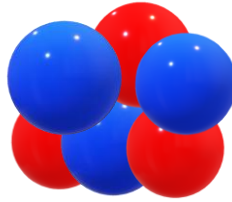
$$\hat{H}_{\text{nuc}} = \hat{T} + \hat{V}$$



3. Select your method

# Ab Initio Nuclear structure in a (nut)shell

1. Select your nucleus



2. Select your Hamiltonian

$\mathcal{L}_{\text{QCD}}$



$$\hat{H}_{\text{nuc}} = \hat{T} + \hat{V}$$



3. Select your method

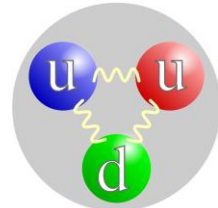
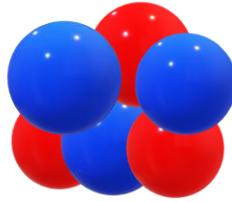
4. Predict your wave function



$$\psi_{\text{nucleus}}(\mathbf{x}; \sigma, \tau)$$

# Ab Initio Nuclear structure in a (nut)shell

1. Select your nucleus



2. Select your Hamiltonian

$\mathcal{L}_{\text{QCD}}$



$$\hat{H}_{\text{nuc}} = \hat{T} + \hat{V}$$



3. Select your method

shape?



4. Predict your wave function



$$\psi_{\text{nucleus}}(\mathbf{x}; \sigma, \tau)$$

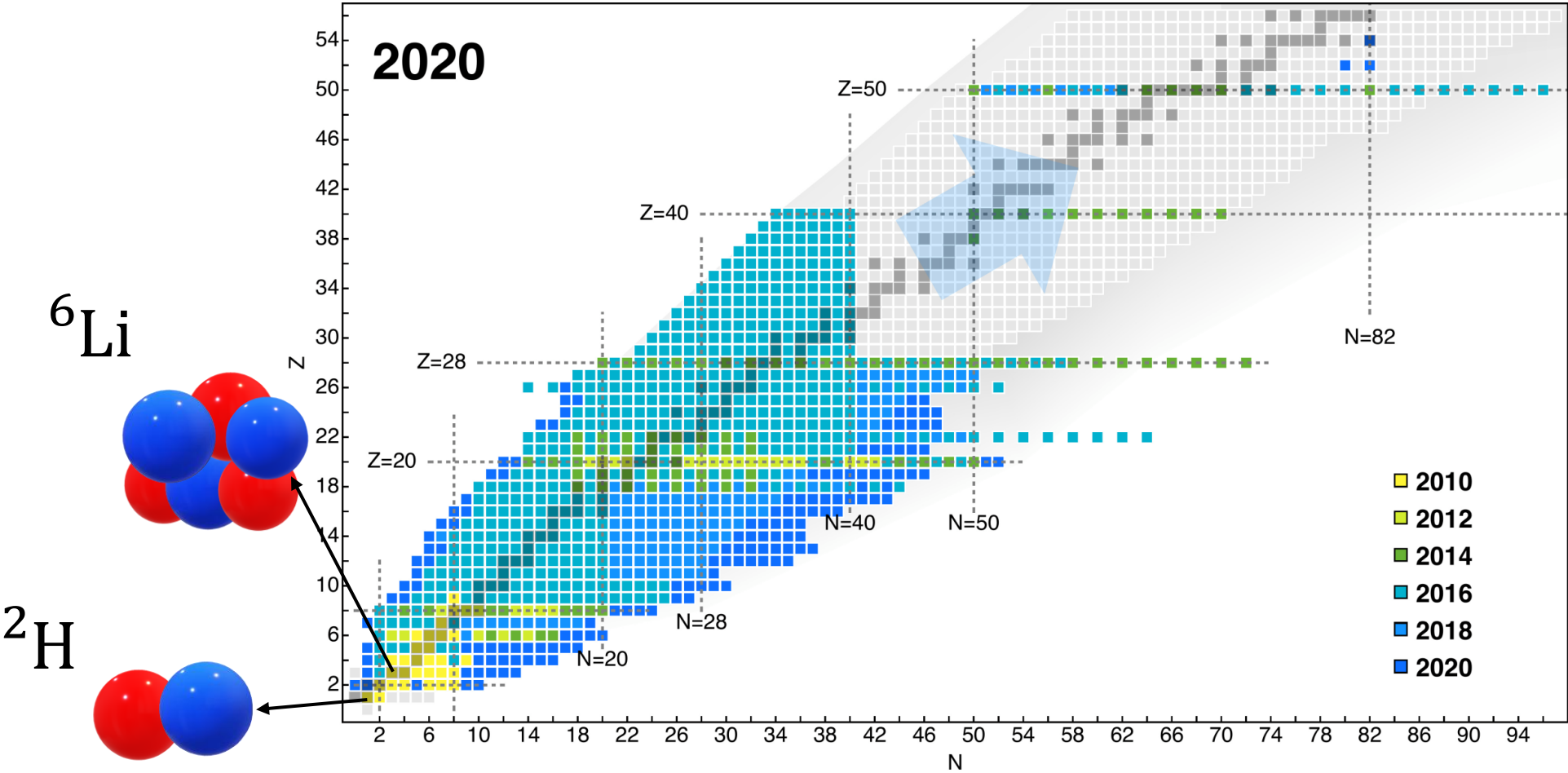
Energy



Excited states

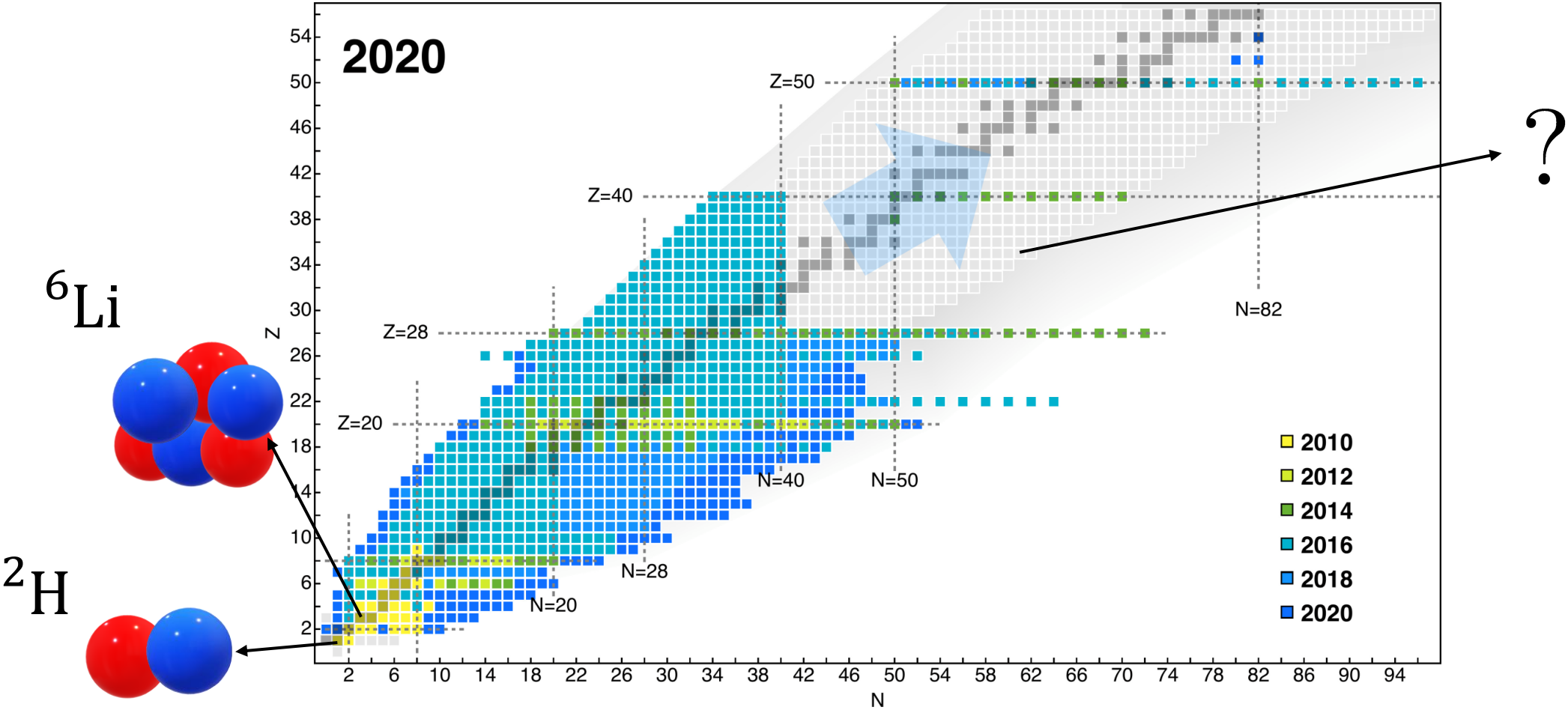
Ground state

# Ab Initio Nuclear Structure



H. Hergert, Front. Phys. **07** (2020)

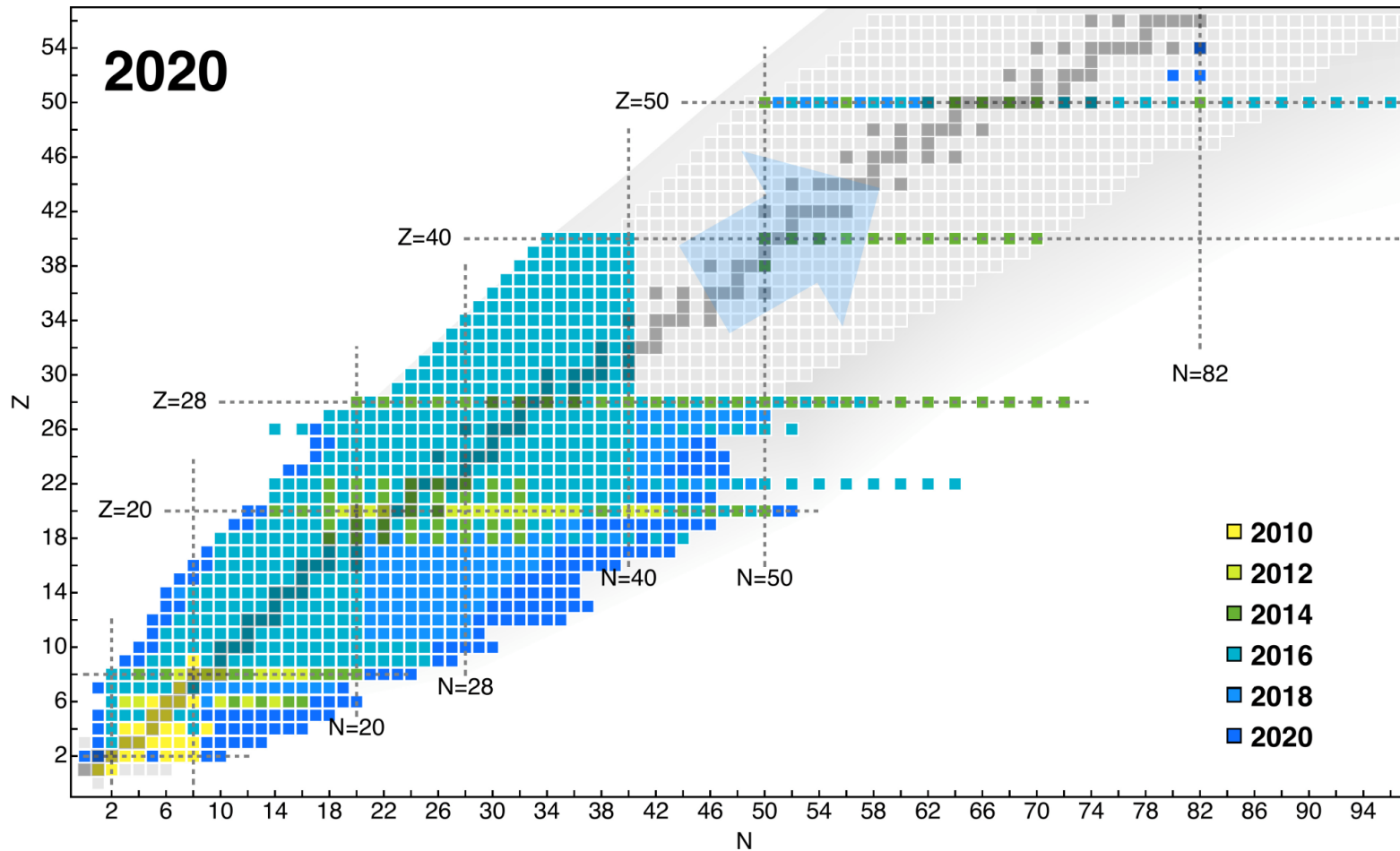
# Ab Initio Nuclear Structure



H. Hergert, Front. Phys. **07** (2020)

# Ab Initio Nuclear Structure

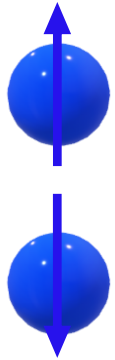
- IT-NCSM
- MR-IMSRG
- VS-IMSRG
- ADC



- Lattice EFT
- CCSD
- $\Lambda$ -CCSD

# Exponential Scaling Problem

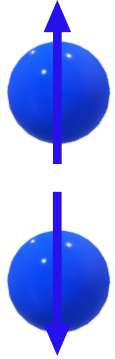
$$N=1$$



$$2^N=2$$

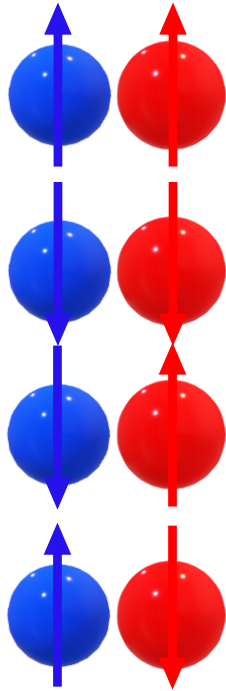
# Exponential Scaling Problem

$N=1$



$$2^N=2$$

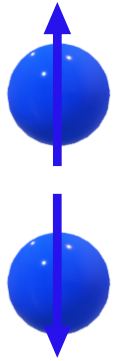
$N=2$



$$2^N=4$$

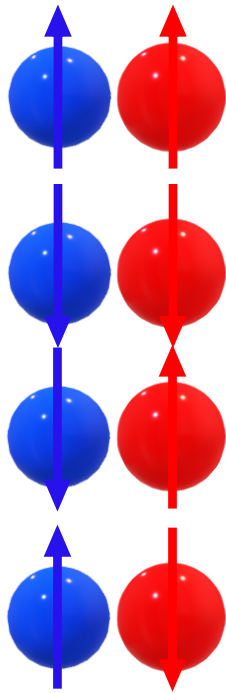
# Exponential Scaling Problem

$N=1$



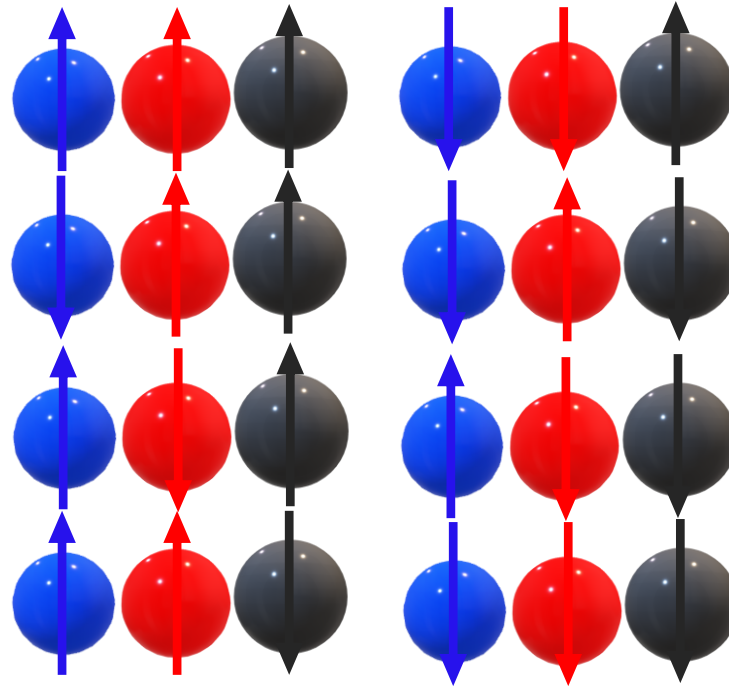
$$2^N=2$$

$N=2$



$$2^N=4$$

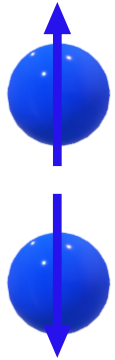
$N=3$



$$2^N=8$$

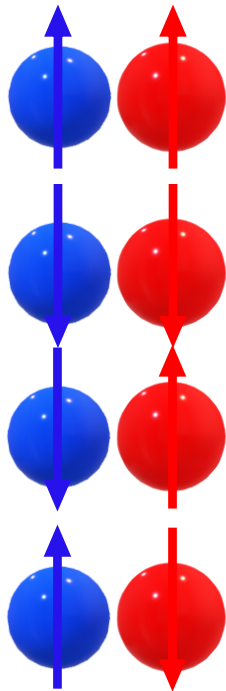
# Exponential Scaling Problem

N=1



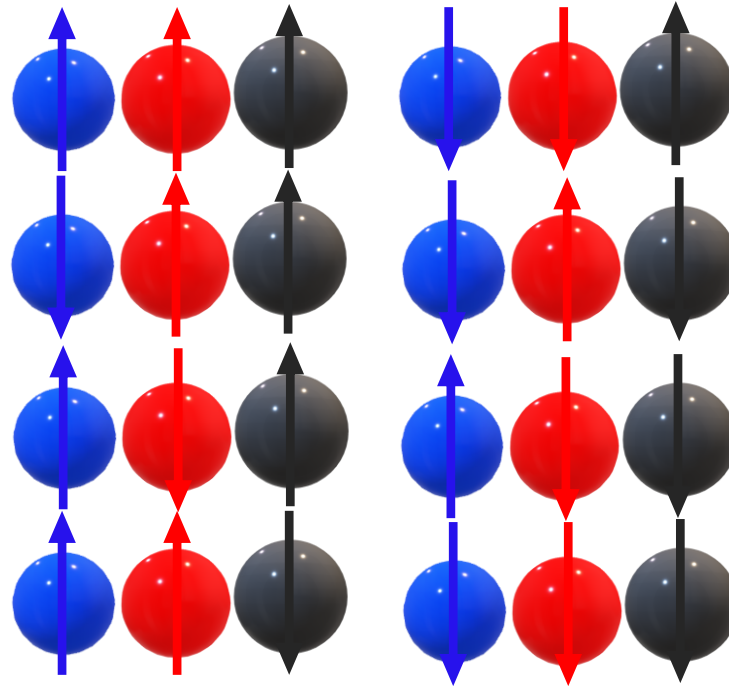
$$2^N = 2$$

N=2



$$2^N = 4$$

N = 3



$$2^N = 8$$

N = 80 (Zr)

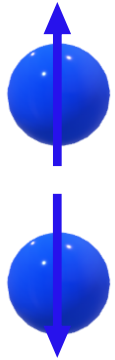


$$2^N = 1,208,925,819,614,629,174,706,176$$

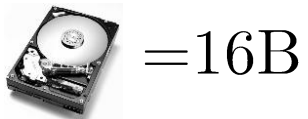
$$(1,2 \times 10^{24})$$

# Exponential Scaling Problem

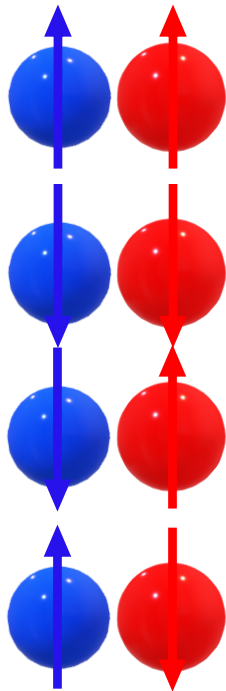
N=1



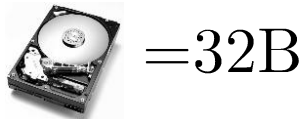
$$2^N = 2$$



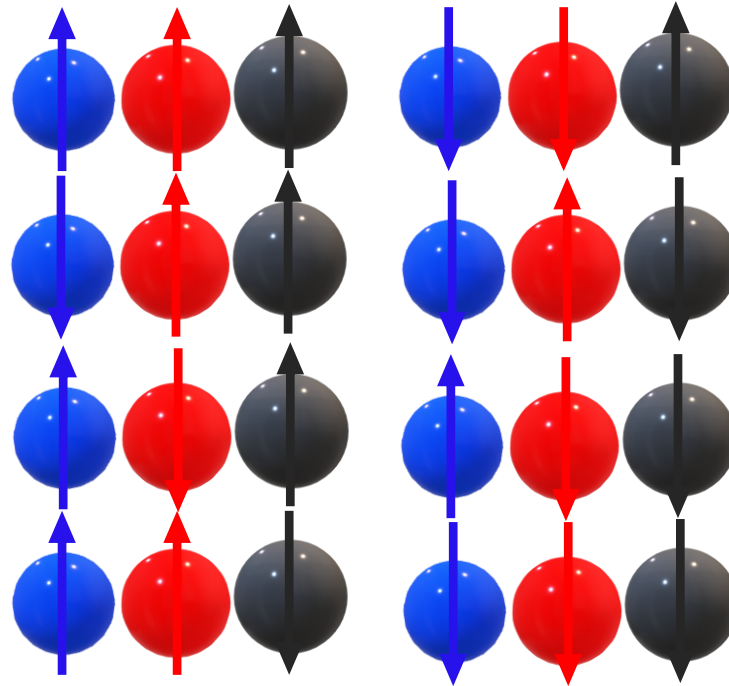
N=2



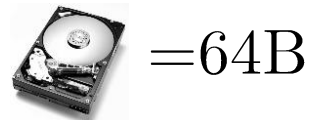
$$2^N = 4$$



N = 3



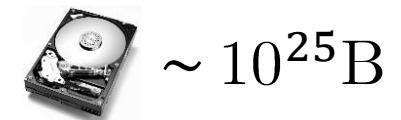
$$2^N = 8$$



N = 80 (Zr)

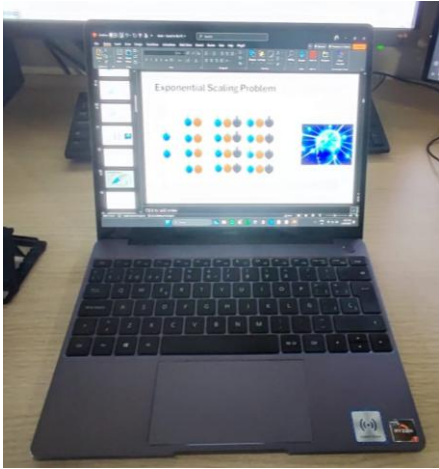


$$2^N = 1,208,925,819,614,629,174,706,176$$



# Exponential Scaling Problem

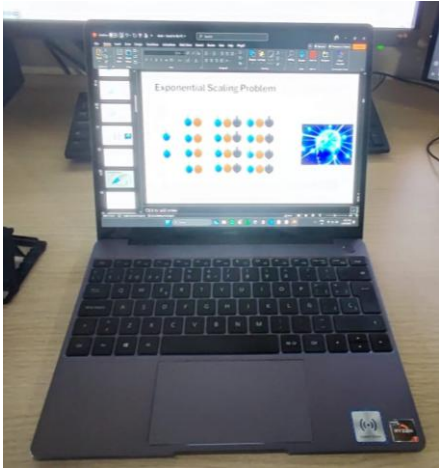
My laptop



$\sim 512 \text{ GB} \sim 10^{11} \text{ B}$

# Exponential Scaling Problem

My laptop



$\sim 512 \text{ GB} \sim 10^{11} \text{ B}$

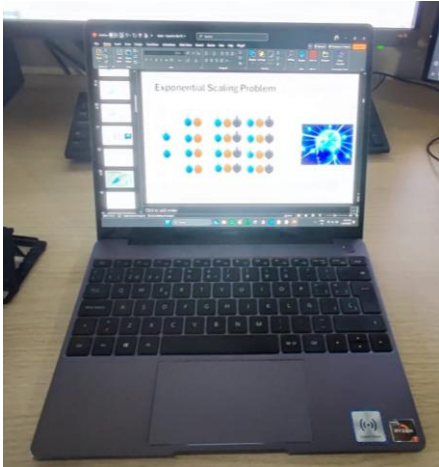
My rich friend's laptop



$\sim 8 \text{ TB} \sim 10^{12} \text{ B}$

# Exponential Scaling Problem

My laptop



$\sim 512 \text{ GB} \sim 10^{11} \text{ B}$

My rich friend's laptop



$\sim 8 \text{ TB} \sim 10^{12} \text{ B}$

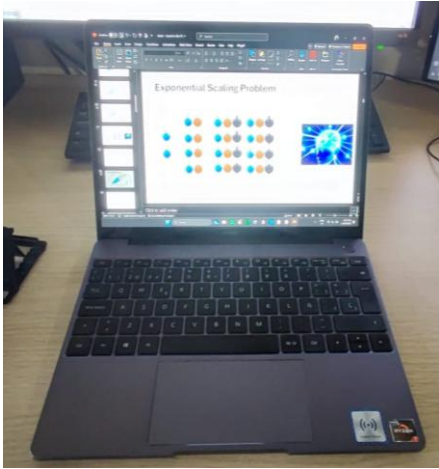
El Capitán



$\sim 5 \text{ PB} \approx 10^{15} \text{ B}$

# Exponential Scaling Problem

My laptop



$\sim 512 \text{ GB} \sim 10^{11} \text{ B}$

My rich friend's laptop



$\sim 8 \text{ TB} \sim 10^{12} \text{ B}$

El Capitán



$\sim 5 \text{ PB} \approx 10^{15} \text{ B}$

What we need:   $\sim 10^{25} \text{ B}$

What we have:   $\sim 10^{15} \text{ B}$

# The Solution: AI



**How can AI help us?**

# Neural Quantum States

- **Variational principle**  $\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \geq E_{\text{GS}}$   $\longrightarrow$  Find  $\theta$  that minimises  $\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}$

# Neural Quantum States

- **Variational principle**  $\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \geq E_{\text{GS}}$   $\longrightarrow$  Find  $\theta$  that minimises  $\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}$

- **Variational Monte Carlo**

$$|\psi_\theta\rangle = \int \psi_\theta(X) dX = \int \psi_\theta(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_N$$

$$\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \mathbb{E}_{X \sim p_\theta} \left[ \frac{\langle X | \hat{H} | \psi_\theta \rangle}{\langle X | \psi_\theta \rangle} \right] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\langle X_i | \hat{H} | \psi_\theta \rangle}{\langle X_i | \psi_\theta \rangle} \quad \text{“Local energy”}$$

# Neural Quantum States

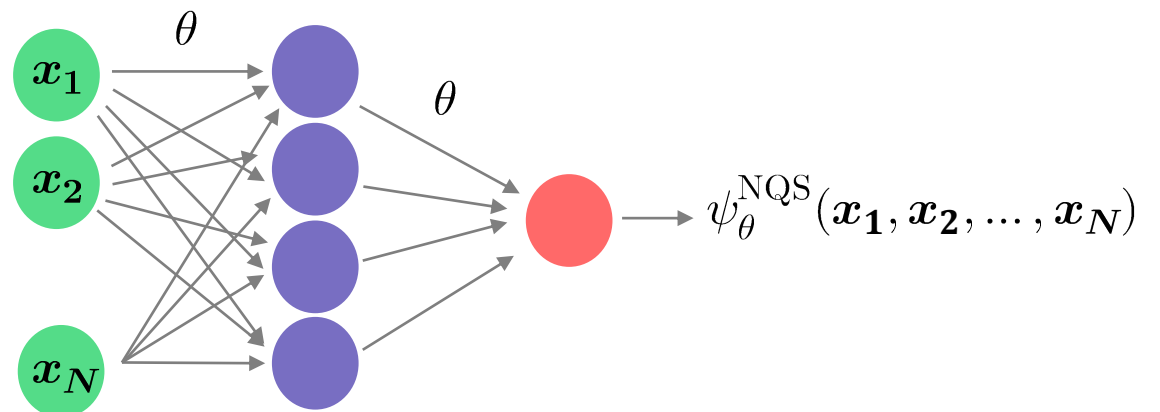
- **Variational principle**  $\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} \geq E_{\text{GS}}$   $\longrightarrow$  Find  $\theta$  that minimises  $\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle}$

- **Variational Monte Carlo**

$$|\psi_\theta\rangle = \int \psi_\theta(X) dX = \int \psi_\theta(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_N$$

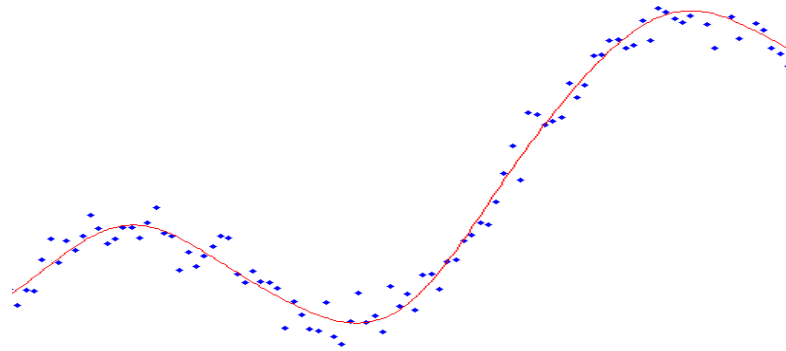
$$\frac{\langle \psi_\theta | \hat{H} | \psi_\theta \rangle}{\langle \psi_\theta | \psi_\theta \rangle} = \mathbb{E}_{X \sim p_\theta} \left[ \frac{\langle X | \hat{H} | \psi_\theta \rangle}{\langle X | \psi_\theta \rangle} \right] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\langle X_i | \hat{H} | \psi_\theta \rangle}{\langle X_i | \psi_\theta \rangle} \quad \text{“Local energy”}$$

- **NQS Ansatz:**  $\psi_{\text{NQS}}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N; \boldsymbol{\theta})$

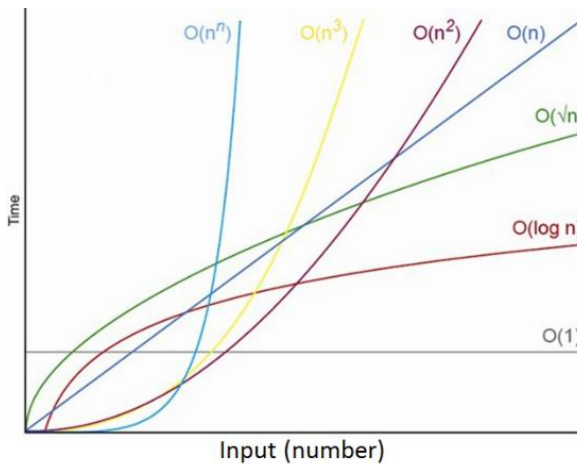


# Why Neural Networks?

- **NNs have “ $\infty$  power”**: a neural network can approximate any continuous function

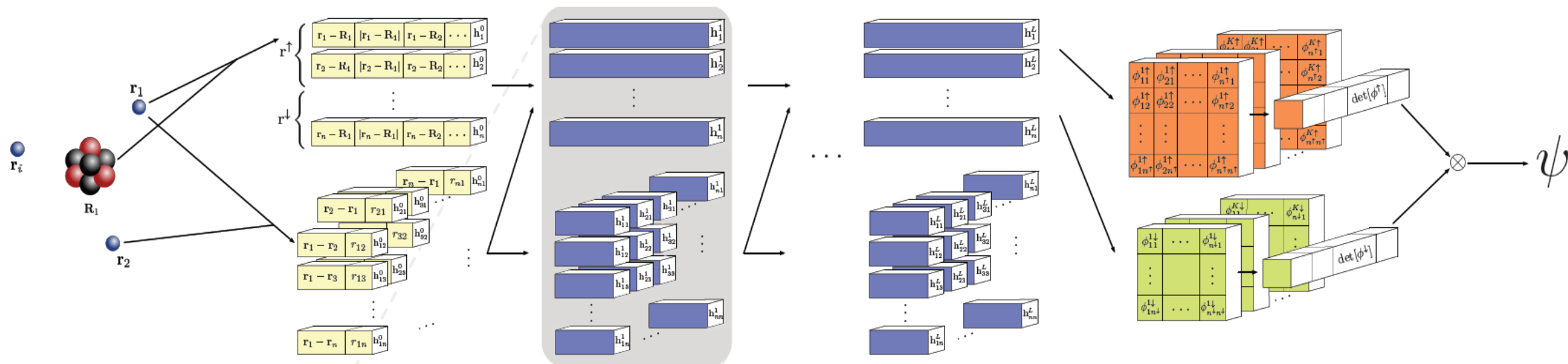


- **Space complexity**: polynomial scaling of memory resources... possibly!

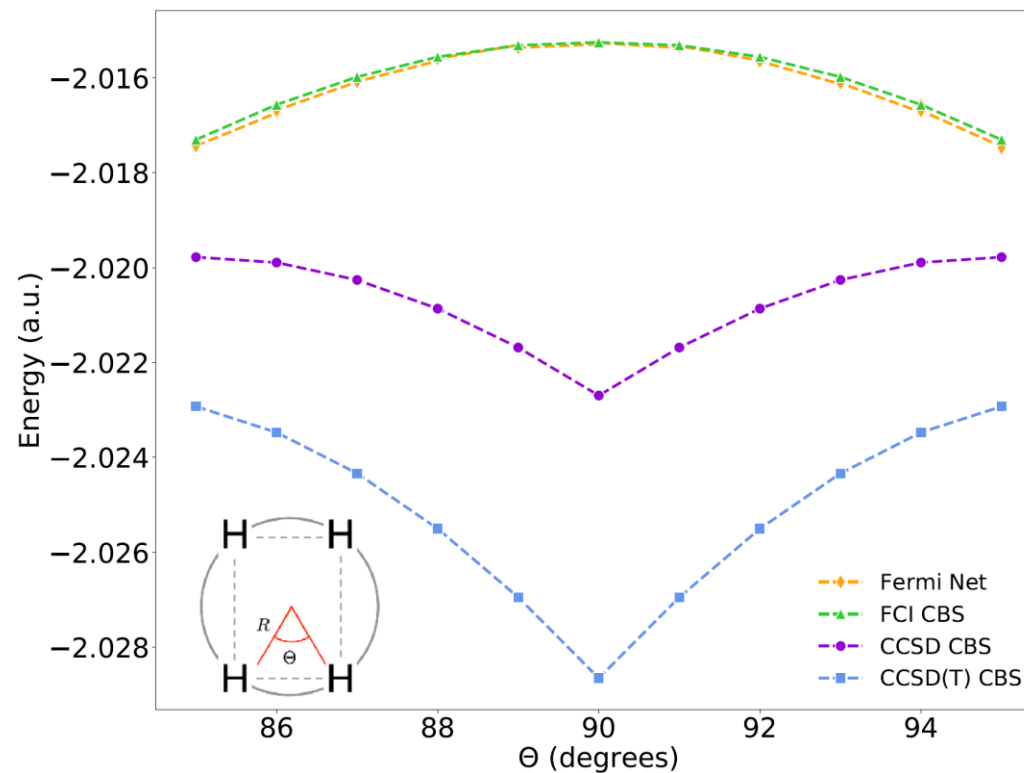
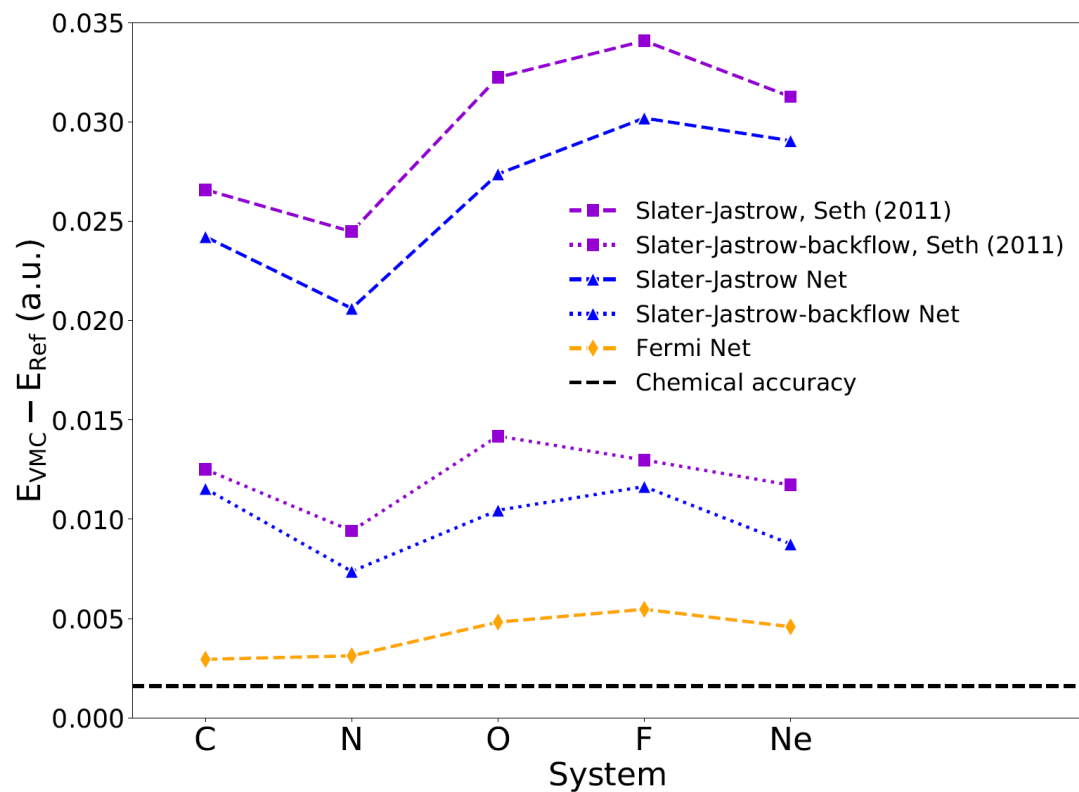


# **NQS showcases**

# FermiNet (Quantum Chemistry)



# FermiNet (Quantum Chemistry)

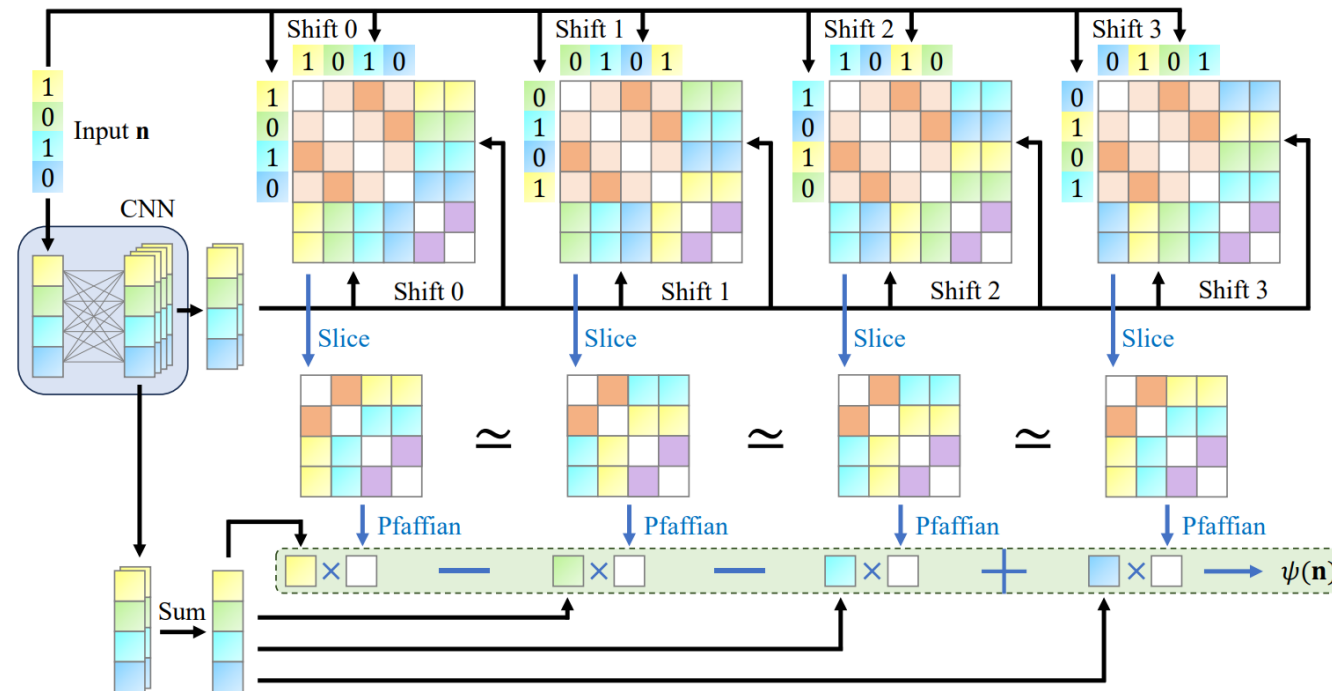


# Pfaffian-based Networks (Superconductivity)

- **System:** Hubbard model

$$\hat{\mathcal{H}} = -t \sum_{\langle ij \rangle, \sigma} \hat{c}_{i, \sigma}^\dagger \hat{c}_{j, \sigma} + U \sum_i \hat{n}_{i, \uparrow} \hat{n}_{i, \downarrow}$$

- **NQS Ansatz:**

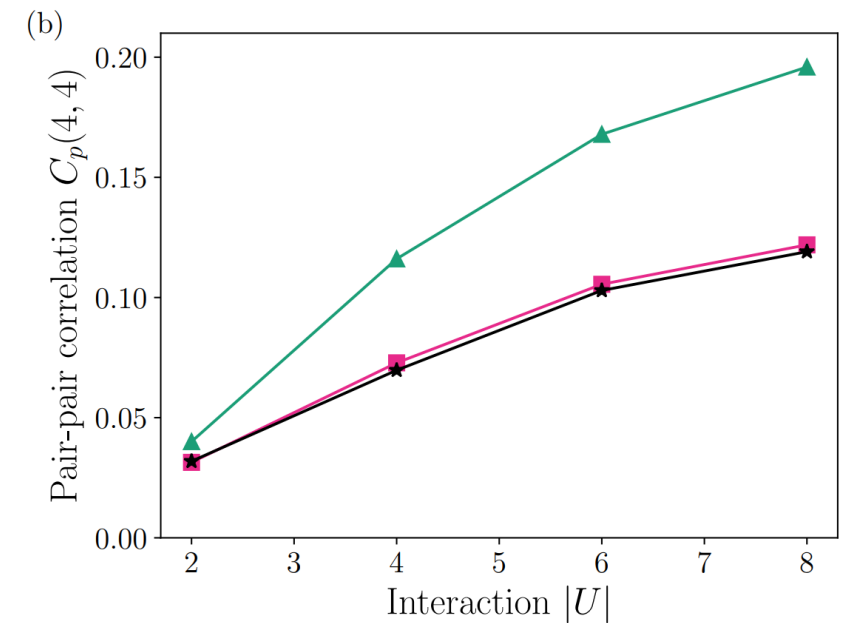
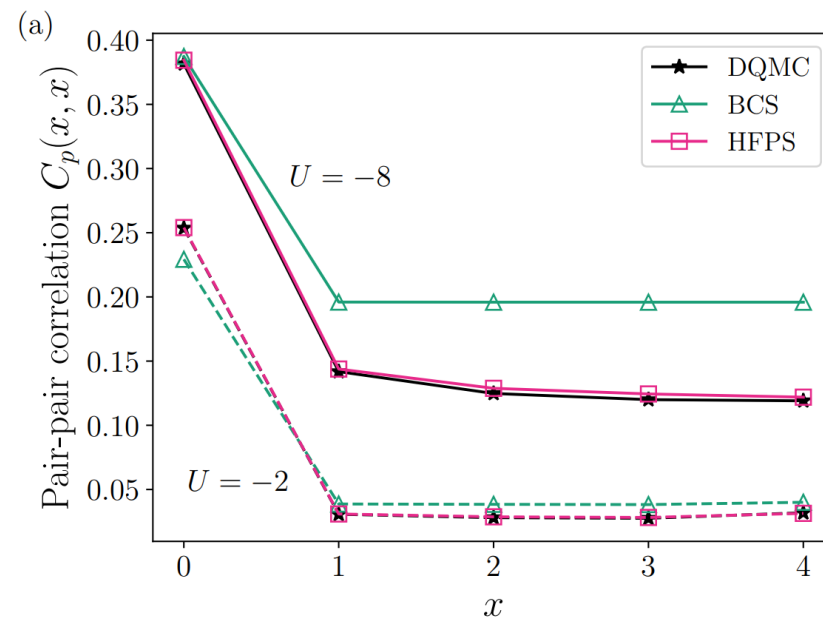


# Pfaffian-based Networks (Superconductivity)

- **System:** Hubbard model

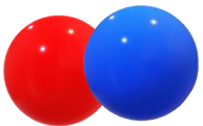
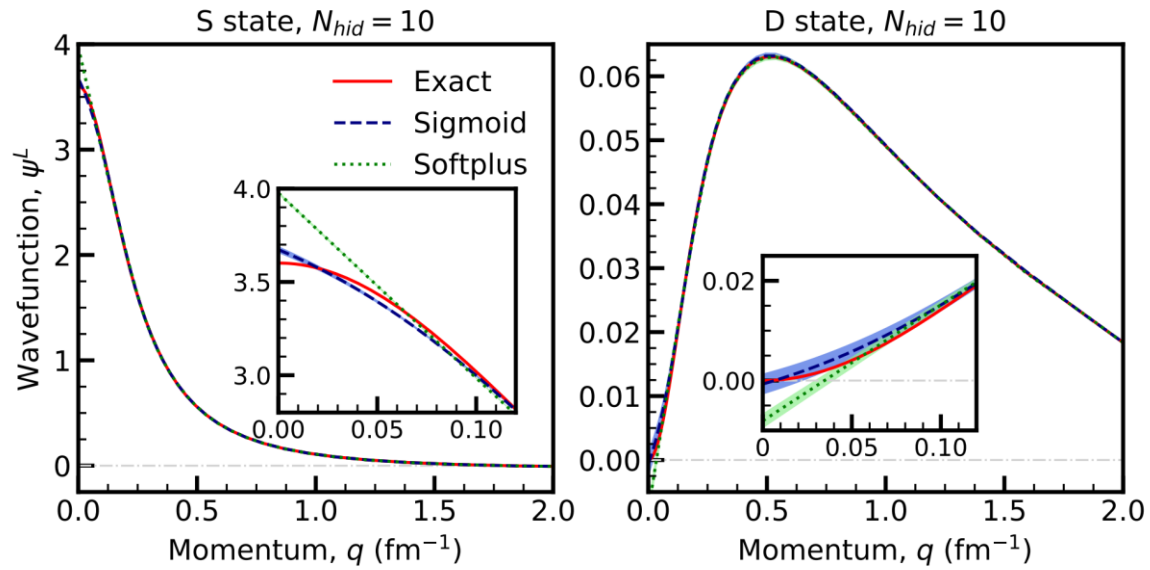
$$\hat{\mathcal{H}} = -t \sum_{\langle ij \rangle, \sigma} \hat{c}_{i, \sigma}^\dagger \hat{c}_{j, \sigma} + U \sum_i \hat{n}_{i, \uparrow} \hat{n}_{i, \downarrow}$$

- **Performance:**



# NQS for Nuclei

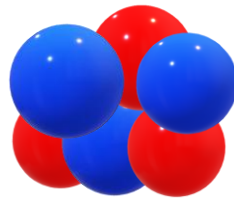
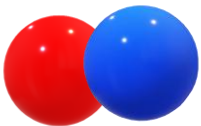
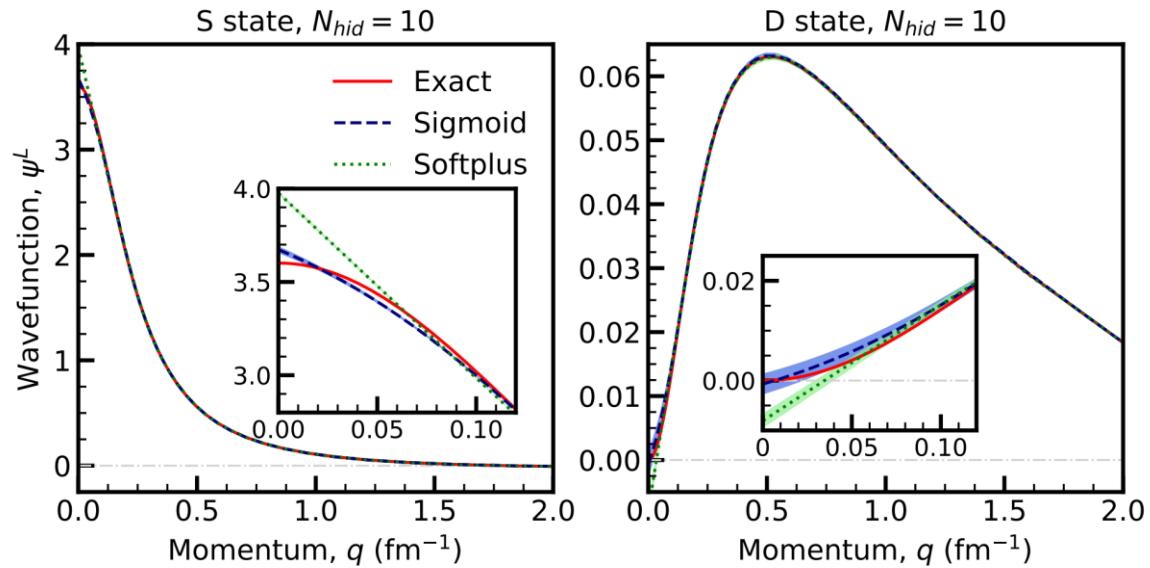
2020



J. Keeble & A. Rios, Phys. Lett. B **809**  
(2020)

# NQS for Nuclei

2020  ...



J. Keeble & A. Rios, Phys. Lett. B **809**  
(2020)

A. Gnech et al, Few-Body Syst. **63** (2022)

# NQS for Nuclei

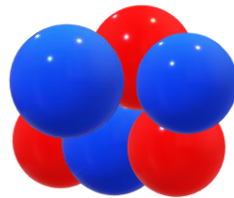
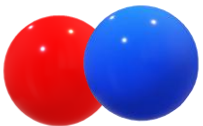
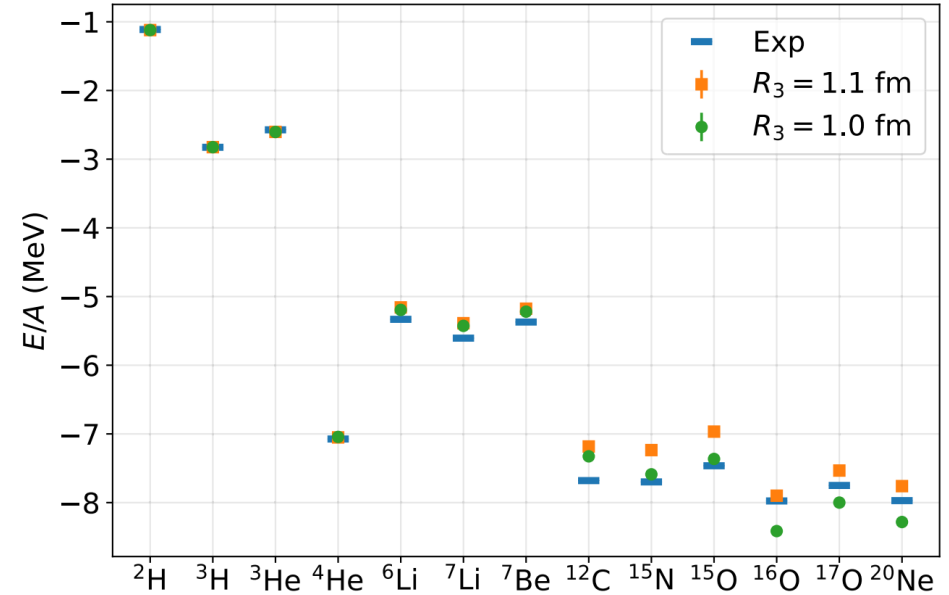
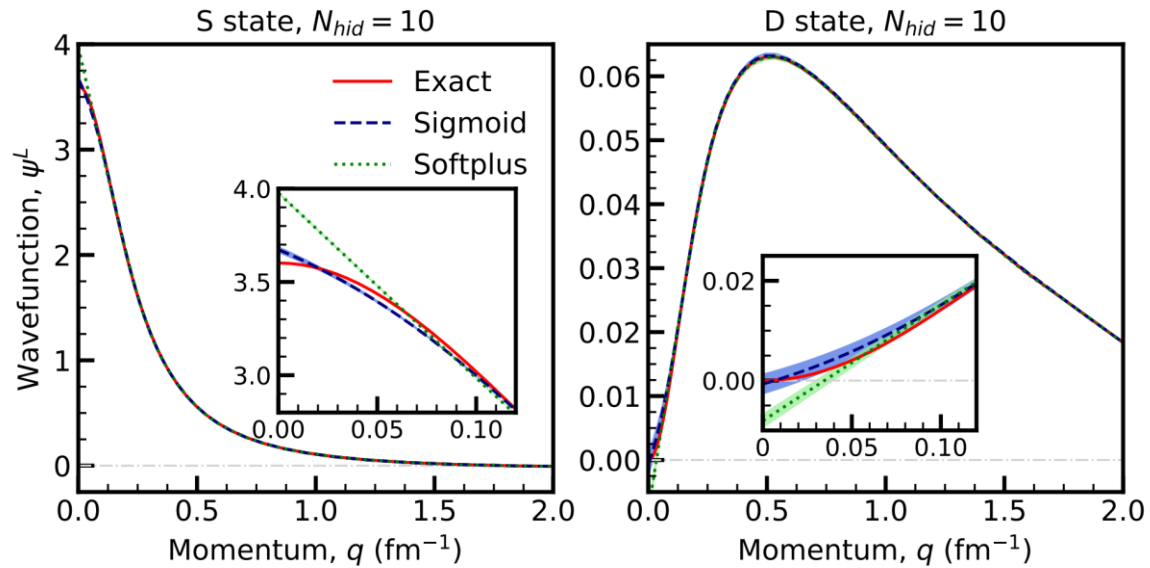
2020



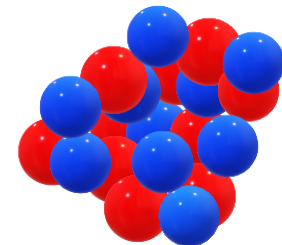
...



2024



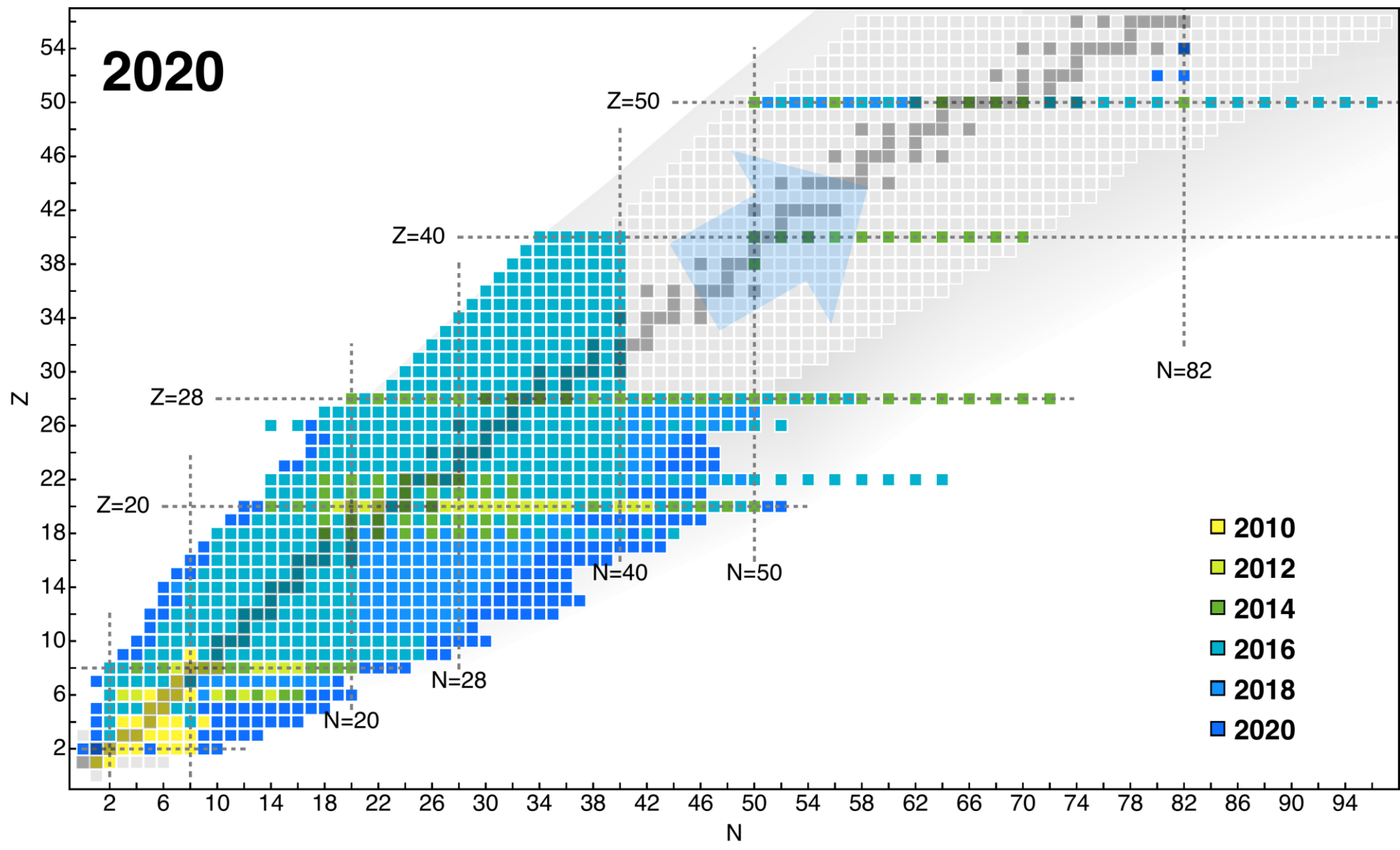
...

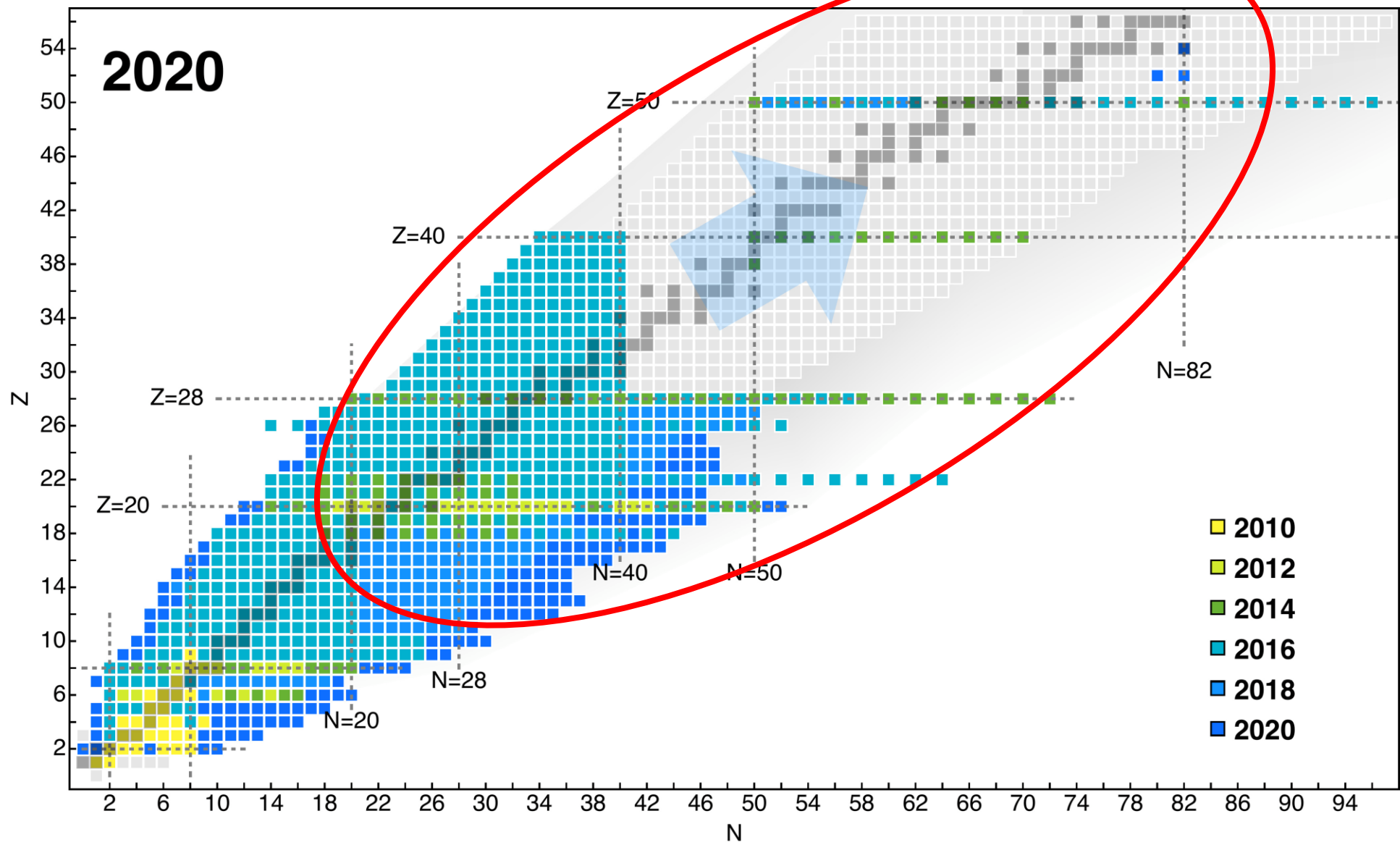


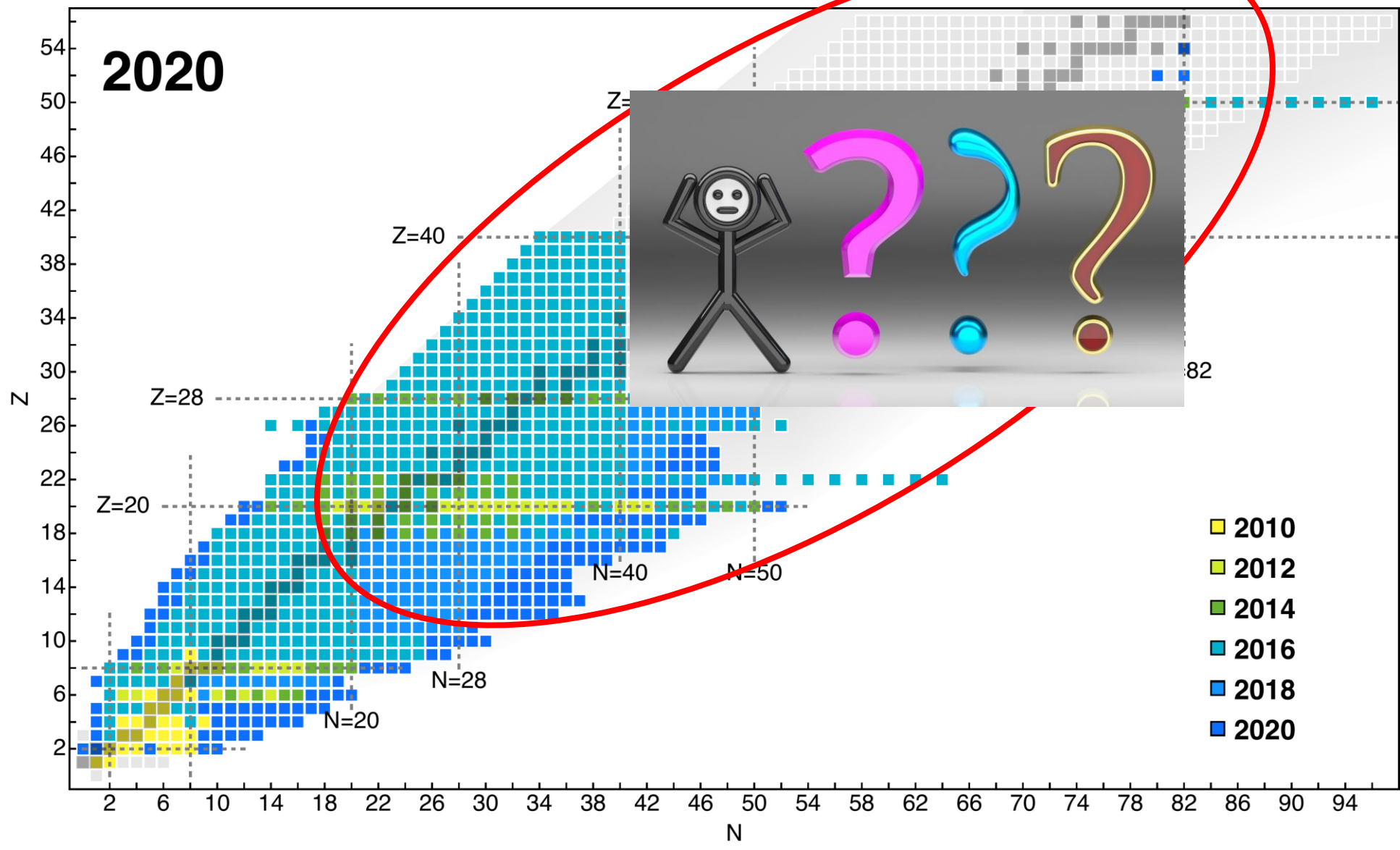
J. Keeble & A. Rios, Phys. Lett. B **809** (2020)

A. Gnech et al, Few-Body Syst. **63** (2022)

A. Gnech et al, PRL **133** (2024)

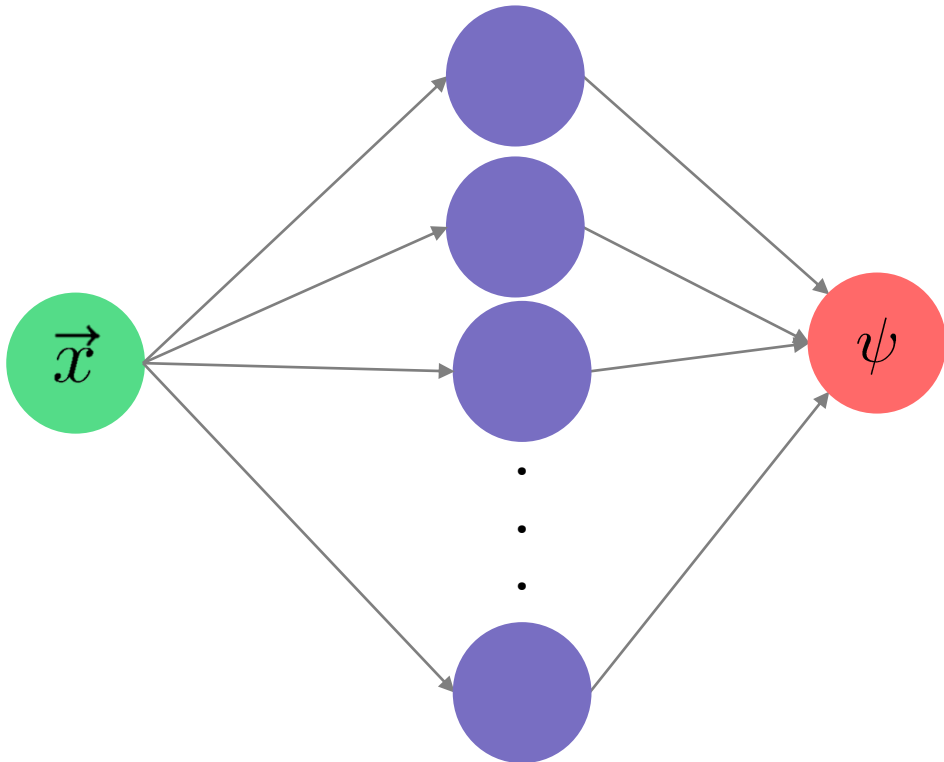






# **1. Neural Network Architecture**

# Physical properties in NNs



- **Particle exchange symmetry**

$$\psi(x_1, x_2, \dots, x_N) = \pm \psi(x_2, x_1, \dots, x_N)$$

- **Spherical symmetry**

$$\psi(\vec{x}) = \psi(R\vec{x})$$

- **Time-reversal symmetry**

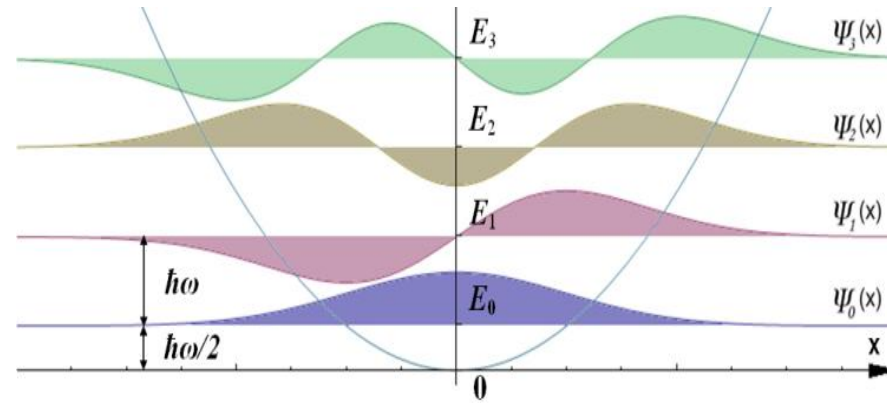
$$\psi(\vec{x}, \sigma) = \psi(T(\vec{x}, \sigma))$$

# Physical properties in NNs

- **1D Harmonic Oscillator**

$$\hat{H} = \frac{1}{2} \nabla^2 + \frac{1}{2} m \omega^2 x^2$$

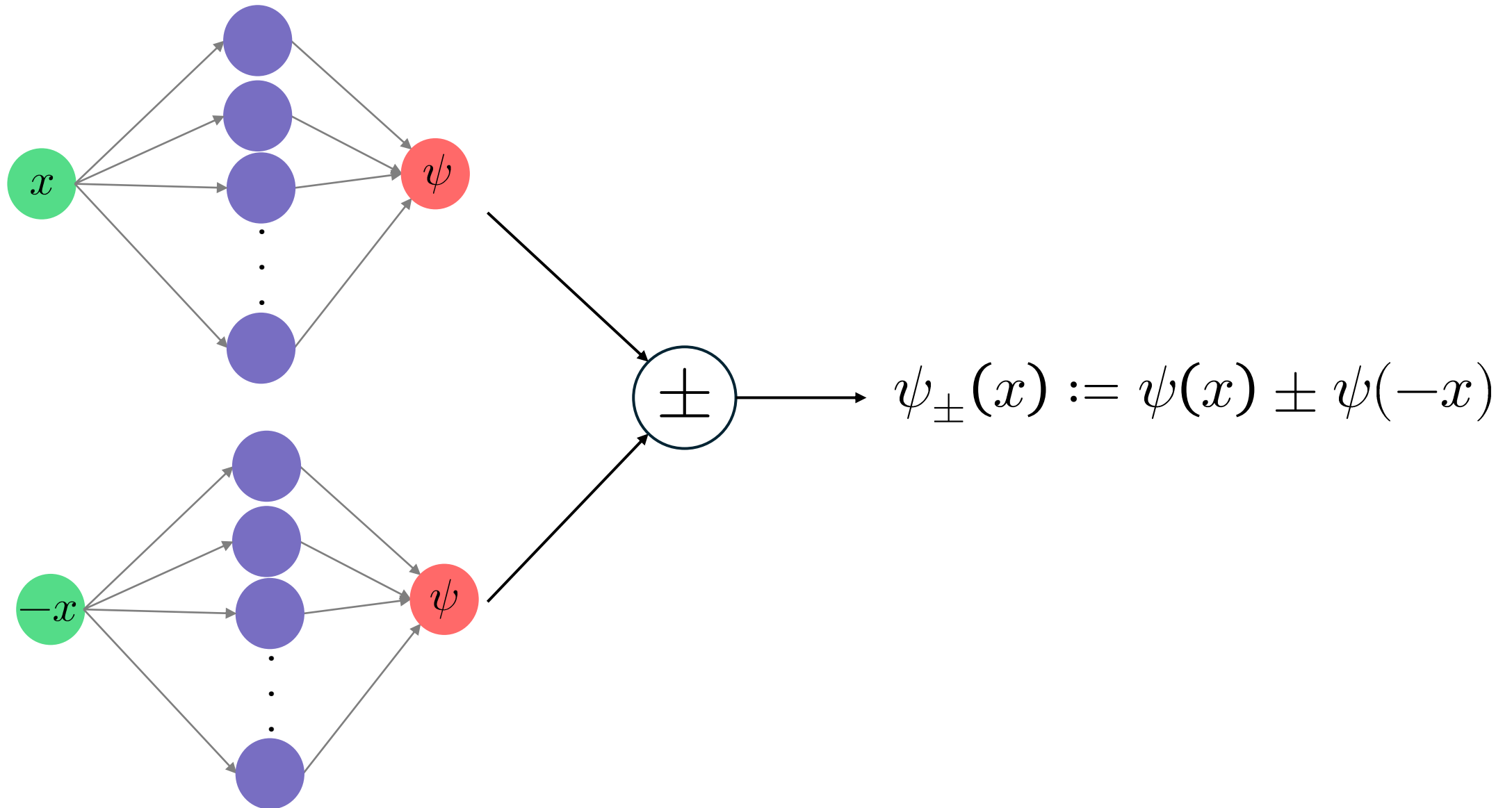
$$\Rightarrow [\hat{H}, \hat{P}] = 0, \hat{P} \text{ parity}$$



- **Reflection symmetry**

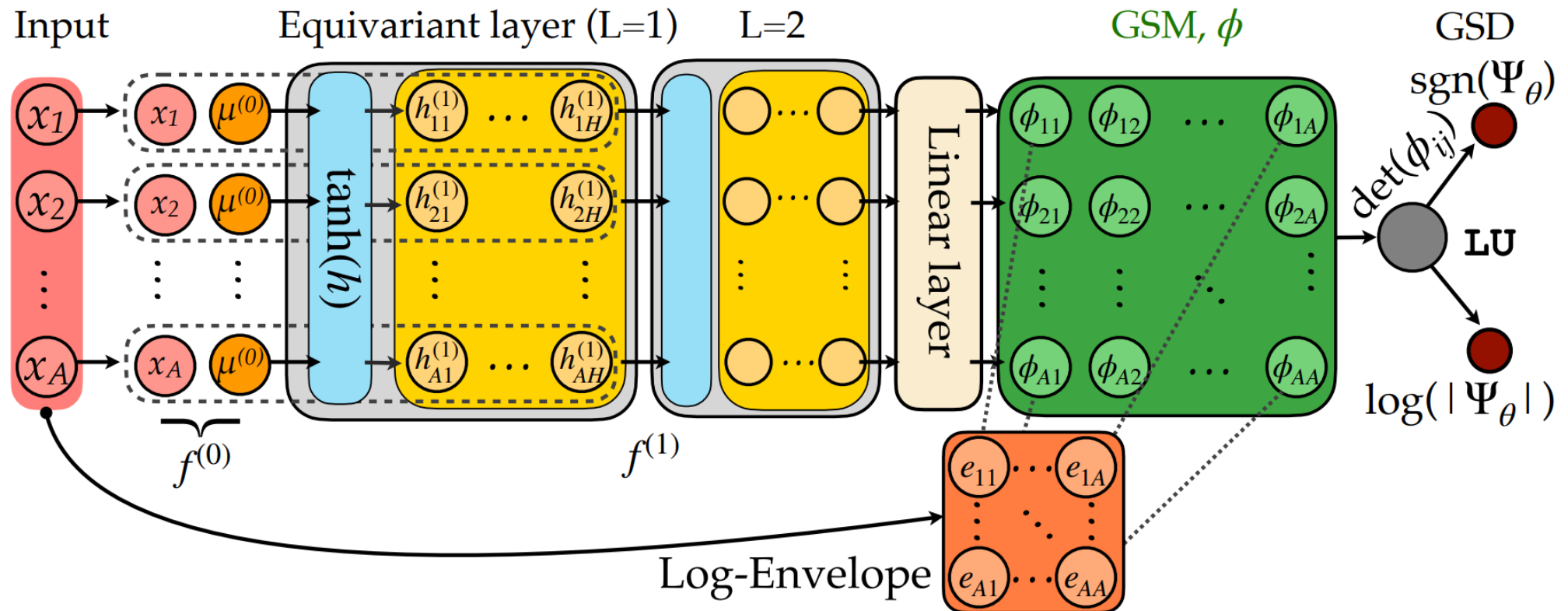
$$\begin{cases} \psi_+(x) := \psi(x) + \psi(-x) \\ \psi_-(x) := \psi(x) - \psi(-x) \end{cases}$$

# Reflection Symmetry



# Particle Exchange Symmetry

$$\psi_{\text{NQS}}(x_1, x_2, \dots, x_N) = \varphi_{\text{EQUIV}} \circ \det \phi_{\text{GSM}}(x_1, x_2, \dots, x_N)$$

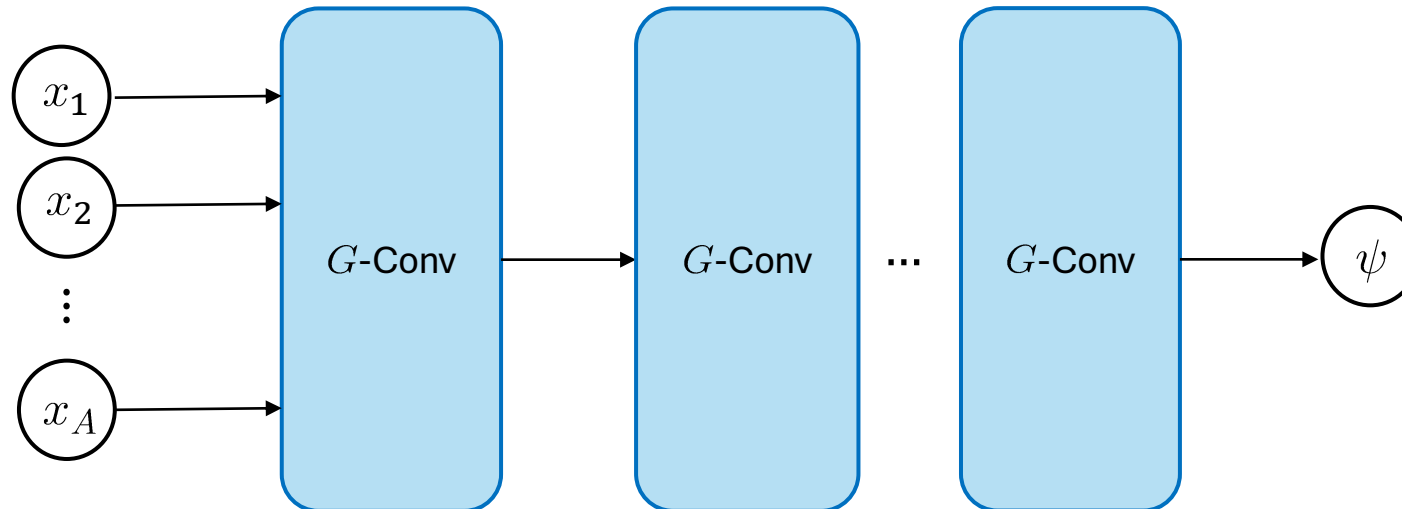


# Towards a universal recipe...

- **Continuous groups:** what about Lie groups?  $SU(2)$   $SO(3)$
- **Mixed groups:** what about product groups?  $SU(2) \times S_N$

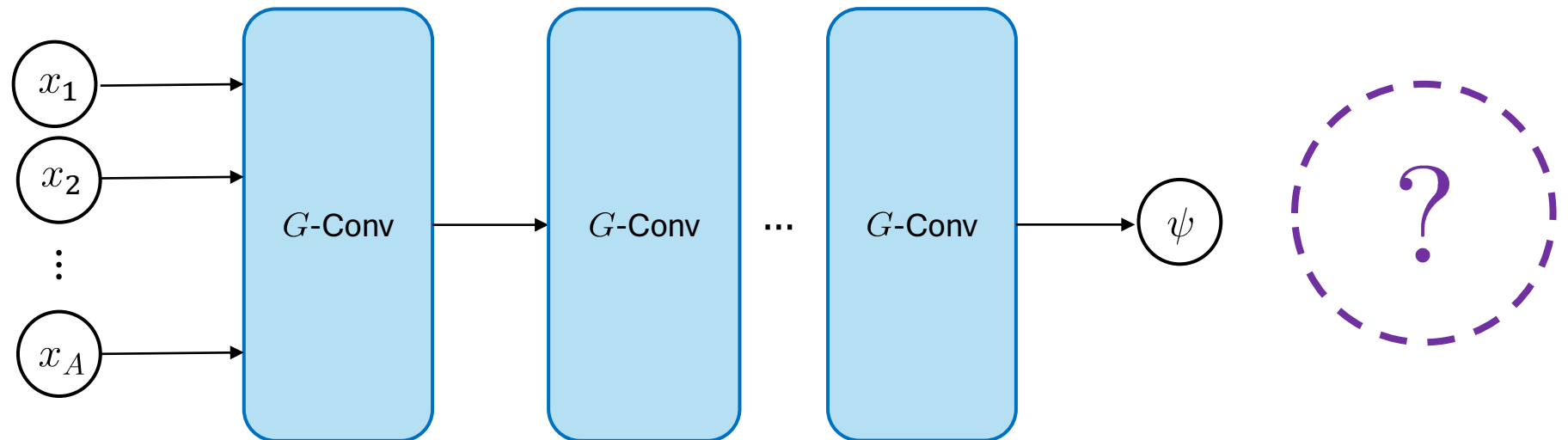
# Towards a universal recipe...

- **Continuous groups:** what about Lie groups?  $SU(2)$   $SO(3)$
- **Mixed groups:** what about product groups?  $SU(2) \times S_N$



# Towards a universal recipe...

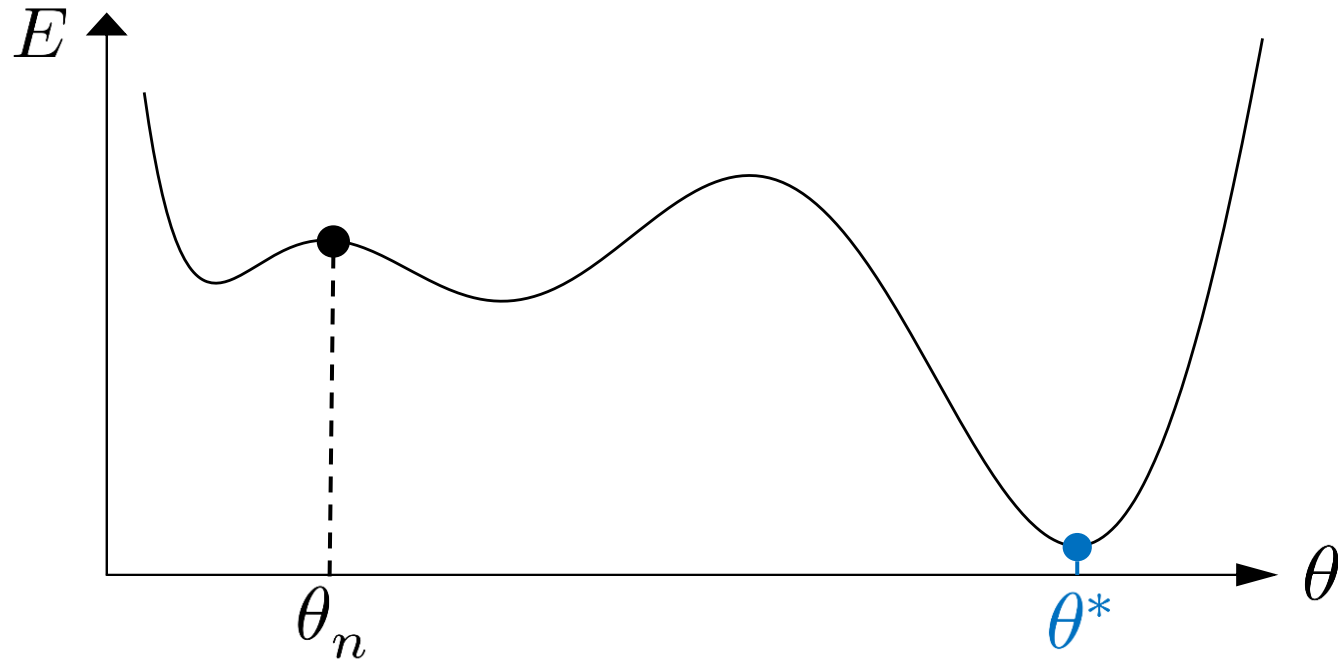
- **Continuous groups:** what about Lie groups?  $SU(2)$   $SO(3)$
- **Mixed groups:** what about product groups?  $SU(2) \times S_N$



## **2. Optimisation Strategy**

# Gradient Descent

- **Local Linear Model:**  $M_n(\delta) = L^T \delta + C$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$

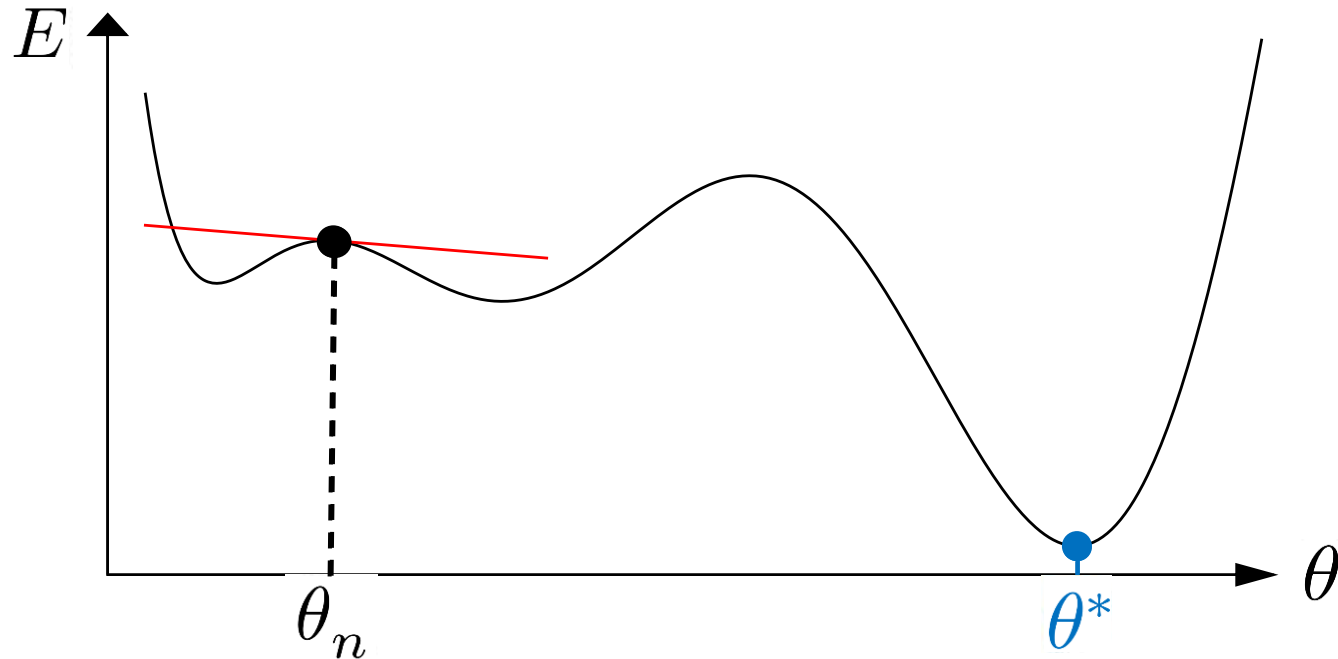


# Gradient Descent

- **Local Linear Model:**  $M_n(\delta) = L^T \delta + C$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$

$$\delta_n = -\alpha \nabla E(\theta_n)$$

$$\theta_{n+1} = \theta_n - \alpha \nabla E(\theta_n)$$



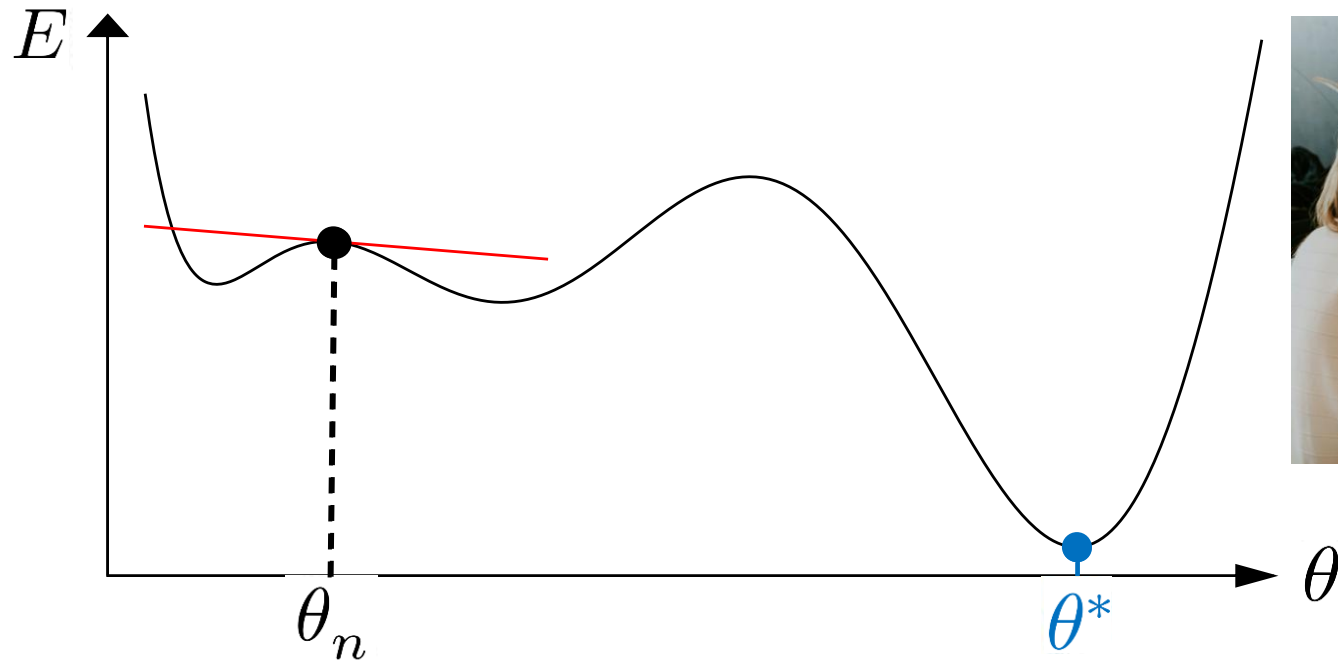
# Gradient Descent



- **Local Linear Model:**  $M_n(\delta) = L^T \delta + C$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$

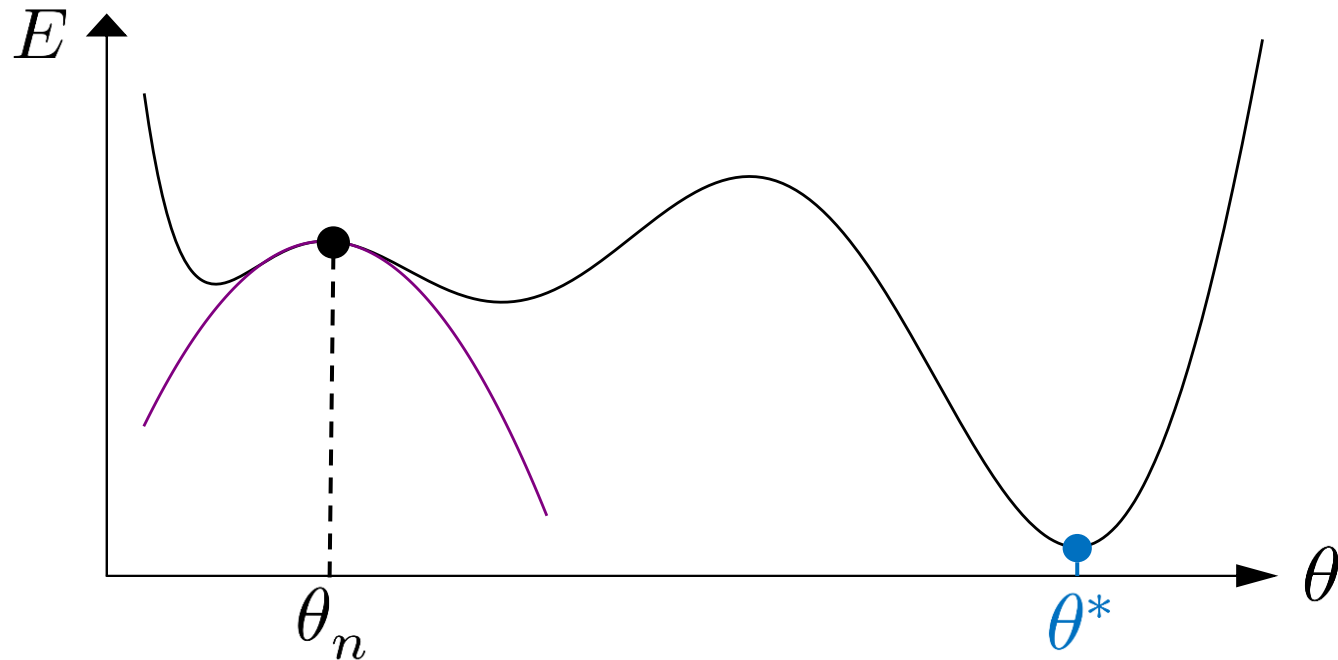
$$\delta_n = -\alpha \nabla E(\theta_n)$$

$$\theta_{n+1} = \theta_n - \alpha \nabla E(\theta_n)$$



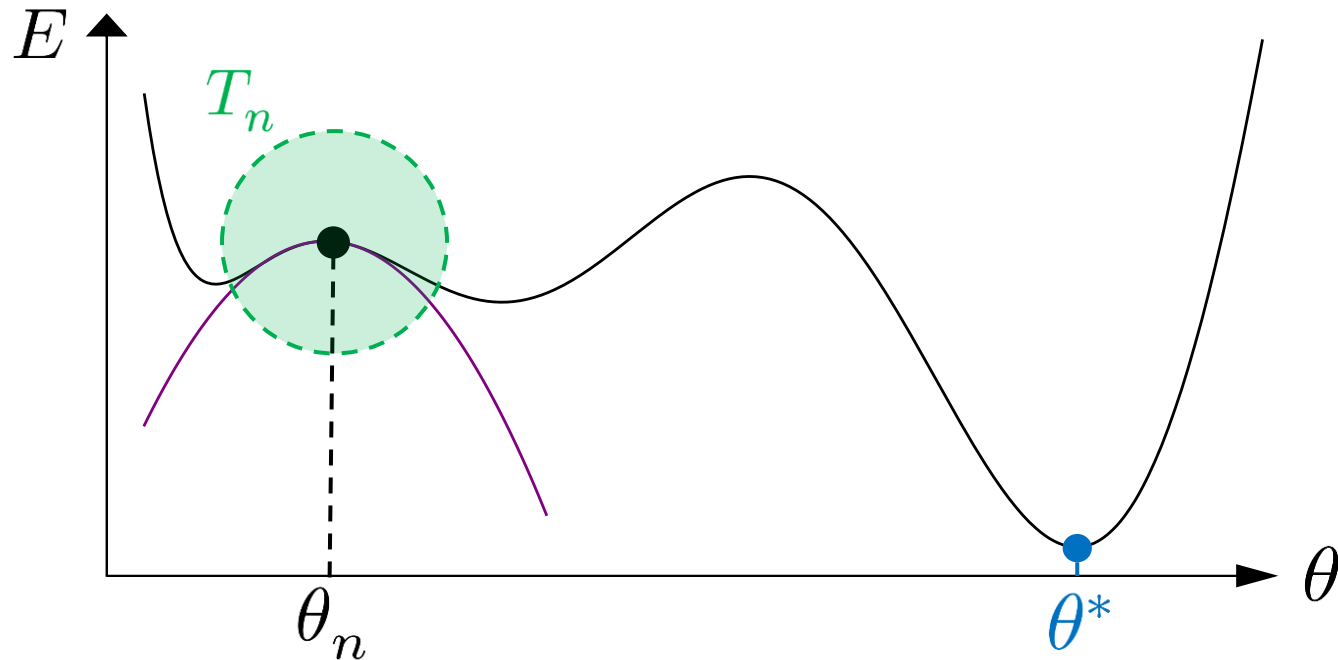
# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q\delta + L^T\delta + C$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n\delta \leq r^2\}$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$



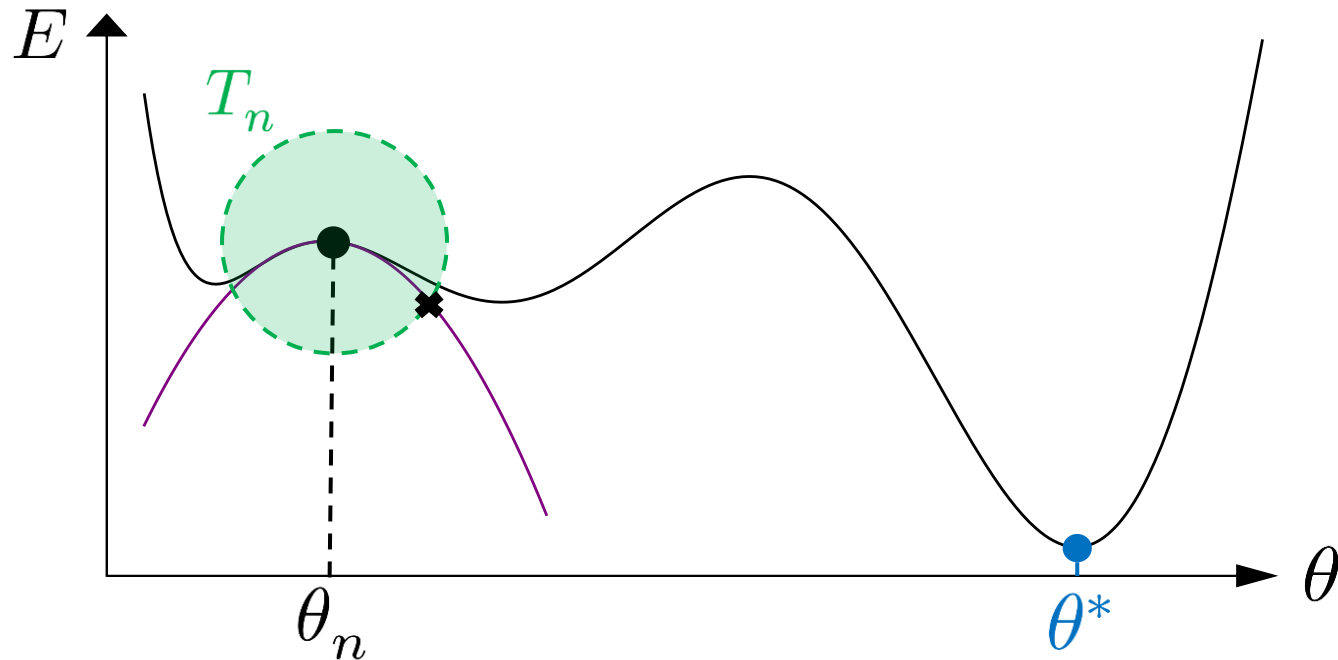
# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q\delta + L^T\delta + C$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n\delta \leq r^2\}$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$



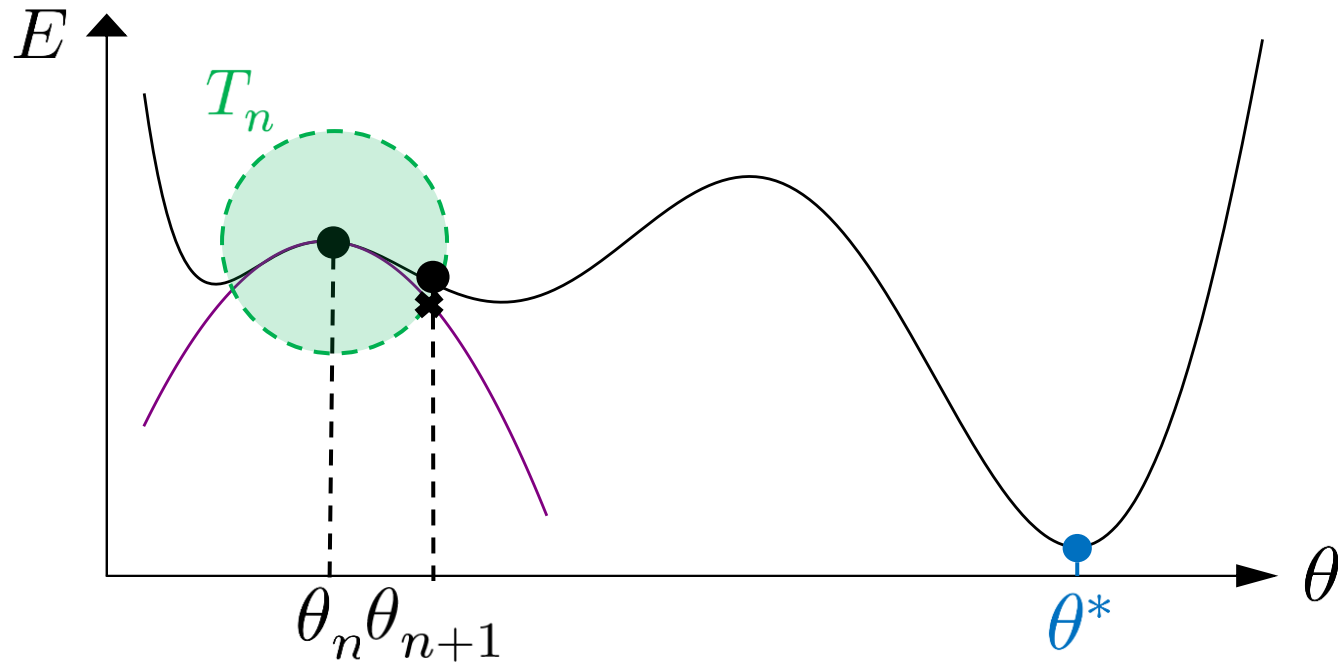
# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q \delta + L^T \delta + C$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n \delta \leq r^2\}$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$



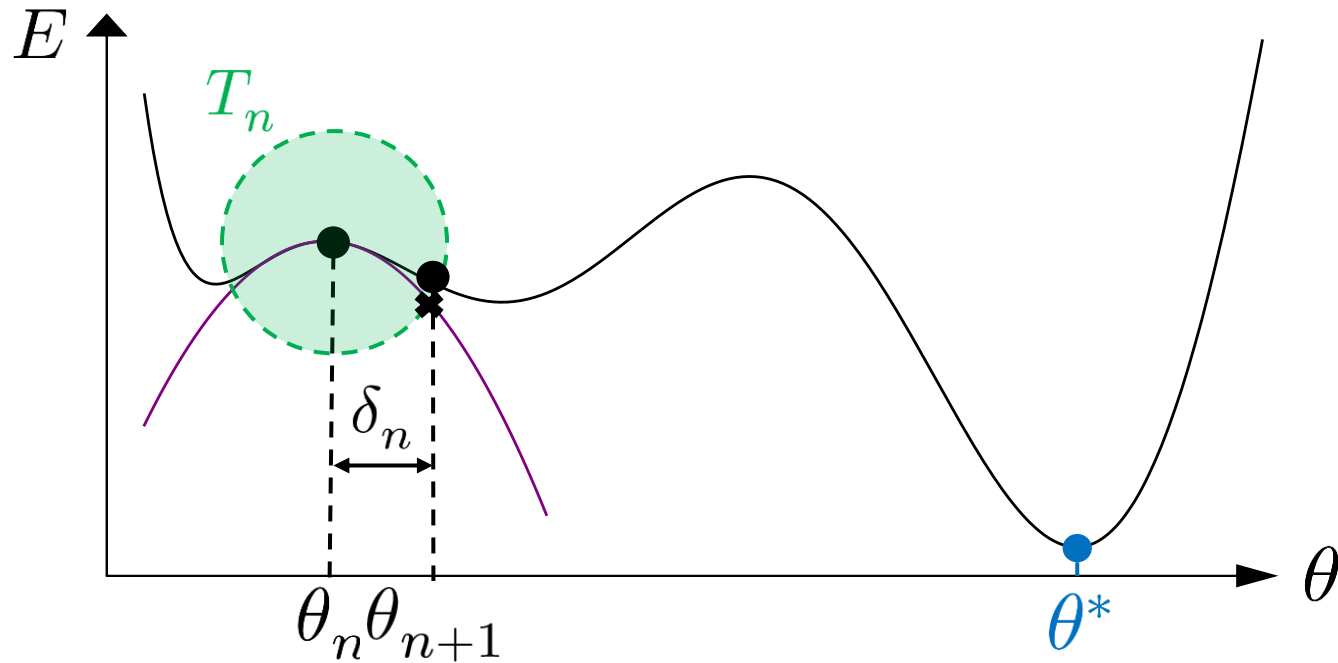
# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q \delta + L^T \delta + C$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n \delta \leq r^2\}$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$



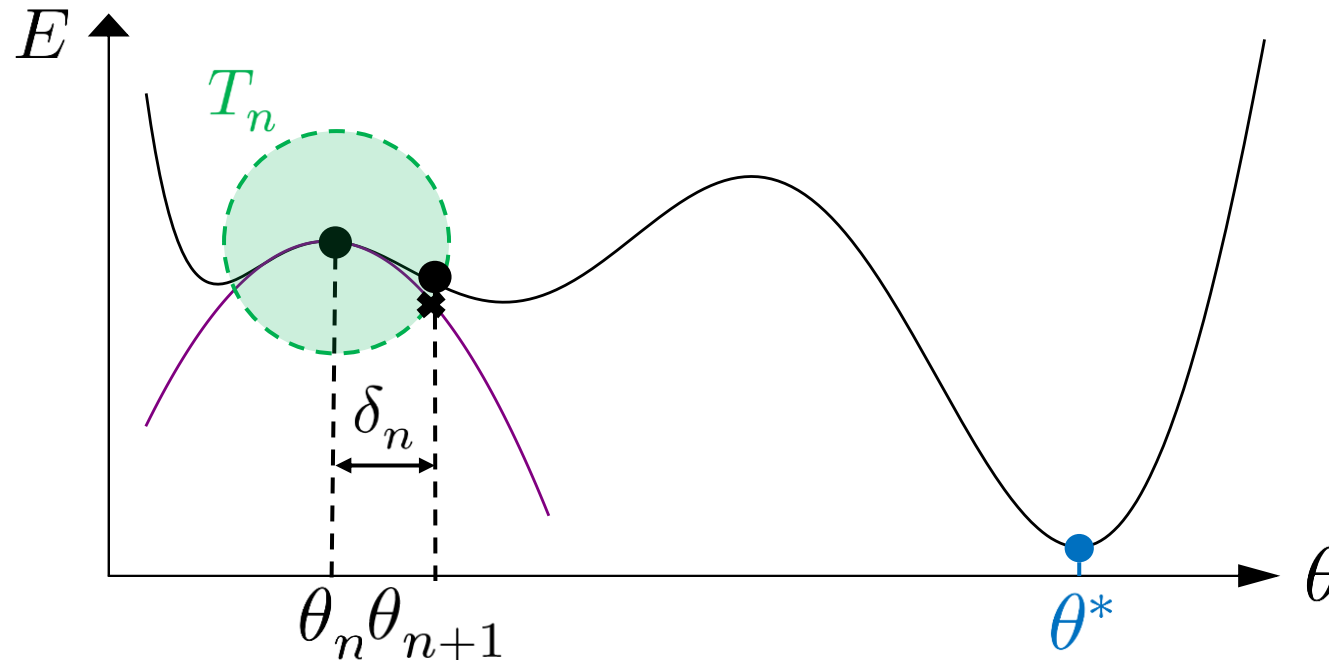
# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q\delta + L^T\delta + C$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n\delta \leq r^2\}$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n \equiv \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$



# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q\delta + L^T\delta + C$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n\delta \leq r^2\}$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n = - (Q_n + \lambda_n R_n)^{-1} \nabla E(\theta_n)$



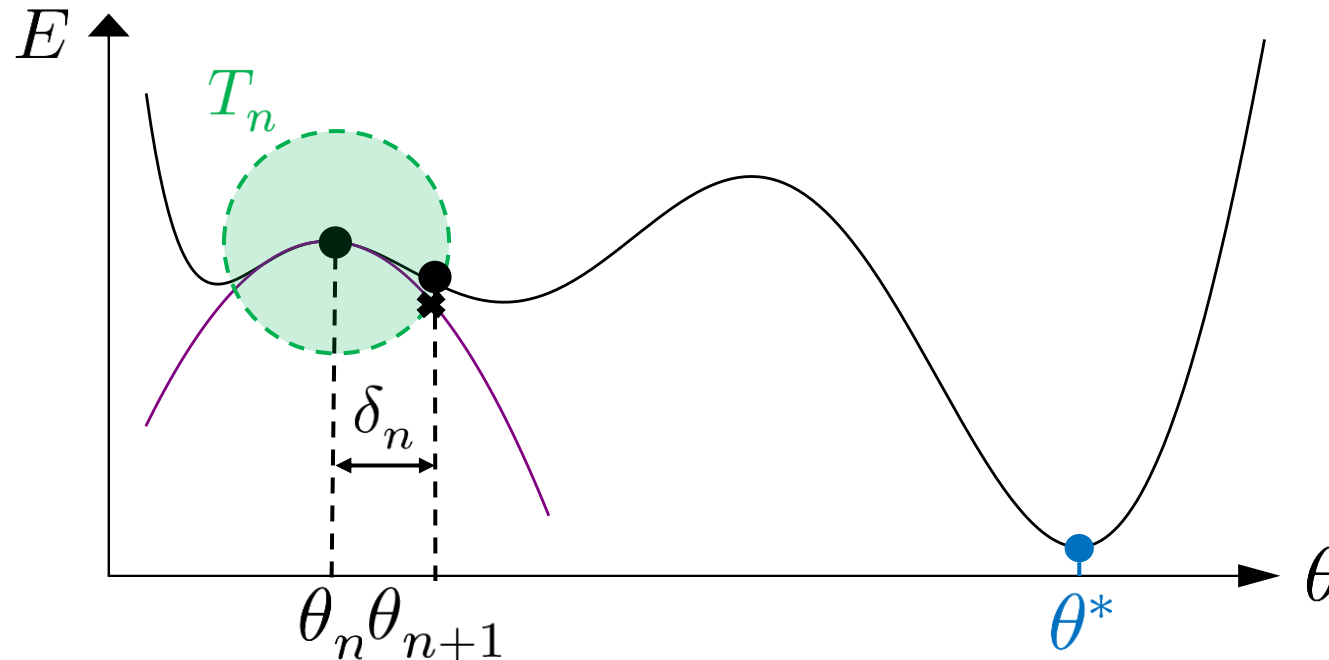
# Second-order optimisation

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q\delta + L^T\delta + C$

$$L = \nabla_{\theta} E$$

- **Trust region:**  $T_n = \{\delta: \delta^T R_n \delta \leq r^2\}$

- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n = - (Q_n + \lambda_n R_n)^{-1} \nabla E(\theta_n)$



# Second-order optimisation

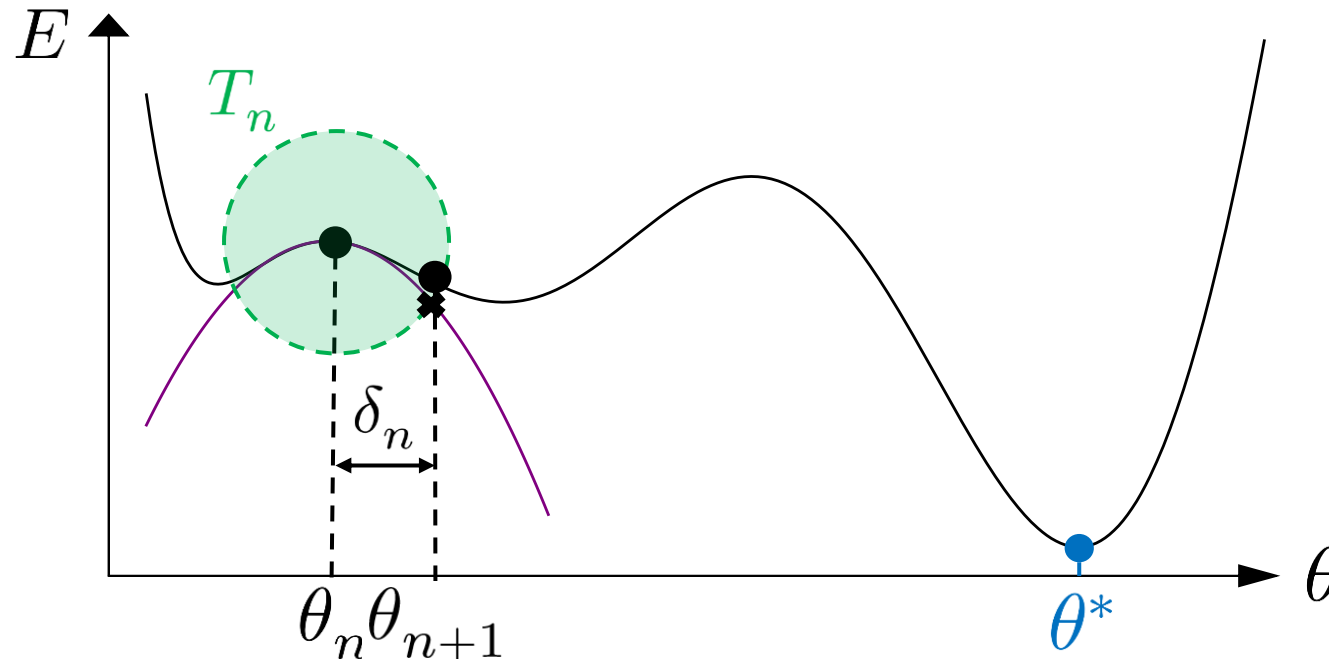
- Local Quadratic Model:  $M_n(\delta) = \frac{1}{2}\delta^T Q \delta + L^T \delta + C$

$$L = \nabla_{\theta} E$$

$Q?$

- Trust region:  $T_n = \{\delta: \delta^T R_n \delta \leq r^2\}$

- Update:  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n = - (Q_n + \lambda_n R_n)^{-1} \nabla E(\theta_n)$



# Stochastic Reconfiguration

“Quantum Geometric Tensor”

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2}\delta^T Q \delta + L^T \delta + C$ ,  $Q_{ij}(\theta) = \left\langle \frac{\partial \psi_\theta}{\partial \theta_i}, \frac{\partial \psi_\theta}{\partial \theta_j} \right\rangle - \left\langle \frac{\partial \psi_\theta}{\partial \theta_i}, \psi_\theta \right\rangle \left\langle \psi_\theta, \frac{\partial \psi_\theta}{\partial \theta_j} \right\rangle$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n \delta \leq r^2\}$ ,  $R_n = 1$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n = - (Q_n + \lambda_n R_n)^{-1} \nabla E(\theta_n)$

# Stochastic Reconfiguration

“Quantum Geometric Tensor”

- **Local Quadratic Model:**  $M_n(\delta) = \frac{1}{2} \delta^T Q \delta + L^T \delta + C$ ,  $Q_{ij}(\theta) = \left\langle \frac{\partial \psi_\theta}{\partial \theta_i}, \frac{\partial \psi_\theta}{\partial \theta_j} \right\rangle - \left\langle \frac{\partial \psi_\theta}{\partial \theta_i}, \psi_\theta \right\rangle \left\langle \psi_\theta, \frac{\partial \psi_\theta}{\partial \theta_j} \right\rangle$
- **Trust region:**  $T_n = \{\delta: \delta^T R_n \delta \leq r^2\}$ ,  $R_n = 1$
- **Update:**  $\theta_{n+1} = \theta_n + \delta_n$  where  $\delta_n = - (Q_n + \lambda_n R_n)^{-1} L_n$

Efficient optimization of deep neural quantum states toward machine precision

Ao Chen and Markus Heyl  
Center for Electronic Correlations and Magnetism,  
University of Augsburg, 86135 Augsburg, Germany

Neural quantum states (NQSs) have emerged as a novel promising numerical method to solve the quantum many-body problem. However, it has remained a central challenge to train modern large-scale deep network architectures to desired quantum state accuracy, which would be vital in utilizing the full power of NQSs and making them competitive or superior to conventional numerical approaches. Here, we propose a minimum-step stochastic reconfiguration (MinSR) method that reduces the optimization cost by orders of magnitude while keeping similar accuracy as compared to conventional stochastic reconfiguration. MinSR allows for accurate training on unprecedentedly deep NQS with up to 64 layers and more than  $10^5$  parameters in the spin-1/2 Heisenberg  $J_1$ - $J_2$  models on the square lattice. We find that this approach yields better variational energies as compared to existing numerical results and we further observe that the accuracy of our ground state calculations approaches different levels of machine precision on modern GPU and TPU hardware. The MinSR method opens up the potential to make NQS superior as compared to conventional computational methods with the capability to address yet inaccessible regimes for two-dimensional quantum matter in the future.

As a fundamental concept in quantum physics, the ground state wave function plays a central role in understanding the behavior of many-body quantum systems. The accurate numerical solution of ground states, however, becomes an extraordinary challenge for existing numerical methods, especially in complex and large two-dimensional systems. The respective challenges depend on the individual utilized method, such as the “curse of dimensionality” in exact diagonalization (ED) [1], the notorious sign problem [2] in quantum Monte Carlo (QMC) [3], or the entanglement growth and matrix contraction complexity in tensor network (TN) methods [4].

Recently, the neural quantum state (NQS) has been introduced as a promising alternative for the calculation of ground states of quantum matter by means of artificial intelligence. In particular, the variational quantum eigensolver [2]-perhaps the most promising quantum algorithms for first generation quantum computers-is based on the variational optimization of a cost function to be evaluated on a quantum device, providing a new playground for hybrid quantum-classical learning [3, 4]. However, arguably the most significant advances in quantum information science have seen a lot of progress. On the one hand, a number of promising results have been obtained suggesting the potential of quantum or classical machine learning to solve complex problems. On the other hand, the exponential growth of the Hilbert space of quantum systems makes the exact solution of the ground state wave function increasingly difficult. In this context, the development of efficient numerical methods to solve the ground state wave function is of great importance. In this work, we introduce a novel method, the minimum-step stochastic reconfiguration (MinSR), which allows for accurate training on unprecedentedly deep NQS with up to 64 layers and more than  $10^5$  parameters in the spin-1/2 Heisenberg  $J_1$ - $J_2$  models on the square lattice. We find that this approach yields better variational energies as compared to existing numerical results and we further observe that the accuracy of our ground state calculations approaches different levels of machine precision on modern GPU and TPU hardware. The MinSR method opens up the potential to make NQS superior as compared to conventional computational methods with the capability to address yet inaccessible regimes for two-dimensional quantum matter in the future.

networks, we find significantly lower variational energies outperforming conventional numerical approaches up to lattice sizes of  $16 \times 16$  spins. Most importantly, we observe that our ground state results reach different levels of machine precision. Thus, with MinSR we are able to reach the frontier where the sole limitation of applying the NQS approach to complex two-dimensional quantum matter is not anymore the expressive power of the neural network but rather the inherent numerical precision of the computing device. This is of key importance to exploit the full power of NQS for the calculation of ground states in the future opening up the potential to address yet inaccessible regimes of quantum many-body systems also in higher spatial dimensions.

## Quantum Natural Gradient

James Stokes<sup>1</sup>, Josh Izaac<sup>2</sup>, Nathan Killoran<sup>2</sup>, and Giuseppe Carleo<sup>3</sup>

<sup>1</sup>Center for Computational Quantum Physics and Center for Computational Mathematics, Flatiron Institute, 10010 USA

<sup>2</sup>Xanadu, 777 Bay Street, Toronto, Canada

<sup>3</sup>Center for Computational Quantum Physics, Flatiron Institute, New York, NY 10010 USA

A quantum generalization of Natural Gradient Descent is presented as part of a general-purpose optimization framework for variational quantum circuits. The optimization dynamics is interpreted as moving in the steepest descent direction with respect to the Quantum Information Geometry, corresponding to the real part of the Quantum Geometric Tensor (QGT), also known as the Fubini-Study metric tensor. An efficient algorithm is presented for computing a block-diagonal approximation to the Fubini-Study metric tensor for parametrized quantum circuits, which may be of independent interest.

### 1 Introduction

Variational optimization of parametrized quantum circuits is an integral component for many hybrid quantum-classical algorithms, which are arguably the most promising applications of quantum computing today. The success of these algorithms hinges on the careful choice of the parameter space and the optimization method. In this work, we present a quantum generalization of Natural Gradient Descent (NGD) [12] or SPSA [33]. Recent exploiting direct access to the gradient information has been exploited in design of circuits with minimal overheads with minimal overheads. One motivation for this work is theoretical: in the context of the Quantum Information Geometry, the Quantum Geometric Tensor (QGT) is the real part of the Quantum Geometric Tensor (QGT), also known as the Fubini-Study metric tensor. An efficient algorithm is presented for computing a block-diagonal approximation to the Fubini-Study metric tensor for parametrized quantum circuits, which may be of independent interest.

parameters, including derivative-free methods such as Nelder-Mead [12] or SPSA [33]. Recent exploiting direct access to the gradient information has been exploited in design of circuits with minimal overheads with minimal overheads. One motivation for this work is theoretical: in the context of the Quantum Information Geometry, the Quantum Geometric Tensor (QGT) is the real part of the Quantum Geometric Tensor (QGT), also known as the Fubini-Study metric tensor. An efficient algorithm is presented for computing a block-diagonal approximation to the Fubini-Study metric tensor for parametrized quantum circuits, which may be of independent interest.

One motivation for this work is theoretical: in the context of the Quantum Information Geometry, the Quantum Geometric Tensor (QGT) is the real part of the Quantum Geometric Tensor (QGT), also known as the Fubini-Study metric tensor. An efficient algorithm is presented for computing a block-diagonal approximation to the Fubini-Study metric tensor for parametrized quantum circuits, which may be of independent interest.

The application of Stochastic Gradient Descent (SGD), where the gradient is estimated from a randomly selected subset of data. Over the years, variations of this method to investigate unknown



163v2 [cond-mat.str-el] 2 Aug 2024

### A simple Large-Scale

Riccardo Rende,<sup>1,\*</sup> Luciano Loris V

<sup>1</sup>International School for Advanced Studies

<sup>2</sup>Dipartimento di Fisica,

Neural-network architecture functions. These networks require optimization using traditional methods with a limited number of parameters. Here, we leverage even in the deep learning seen a Deep Transformer architect energy in the  $J_1$ - $J_2$  Heisenberg benchmark in highly-frustrated scalability and efficiency of S method to investigate unknown

### I. INTRODUCTION

Deep learning has become crucial in fields, with neural networks being the key to many of the most impressive results. Well-known examples in computational Neural Networks for image recognition and Deep Transformers for language-related tasks. The success of deep networks comes from architectures with a large number of parameters (often in the billions), which allow for training these networks on large datasets. However, to successfully train these large models, one needs to navigate the complex non-convex landscape associated with the parameter space.

The most used methods rely on stochastic gradient descent (SGD), where the gradient is estimated from a randomly selected subset of data. Over the years, variations of this method to investigate unknown

163v2 [quant-ph] 13 Oct 2020

Geometric Tensor  
Institute for

Combining insights from the minimum-step stochastic reconfiguration method with neural network ground state estimation of quantum states in practice, little is known about hidden details of the algorithm of the quantum Fisher matrix initial dynamics, but the contrast to the spectral properties at convergence do not reveal a new measure of cost function that, generically, the value, suggesting that correlated stable representations of the ground state

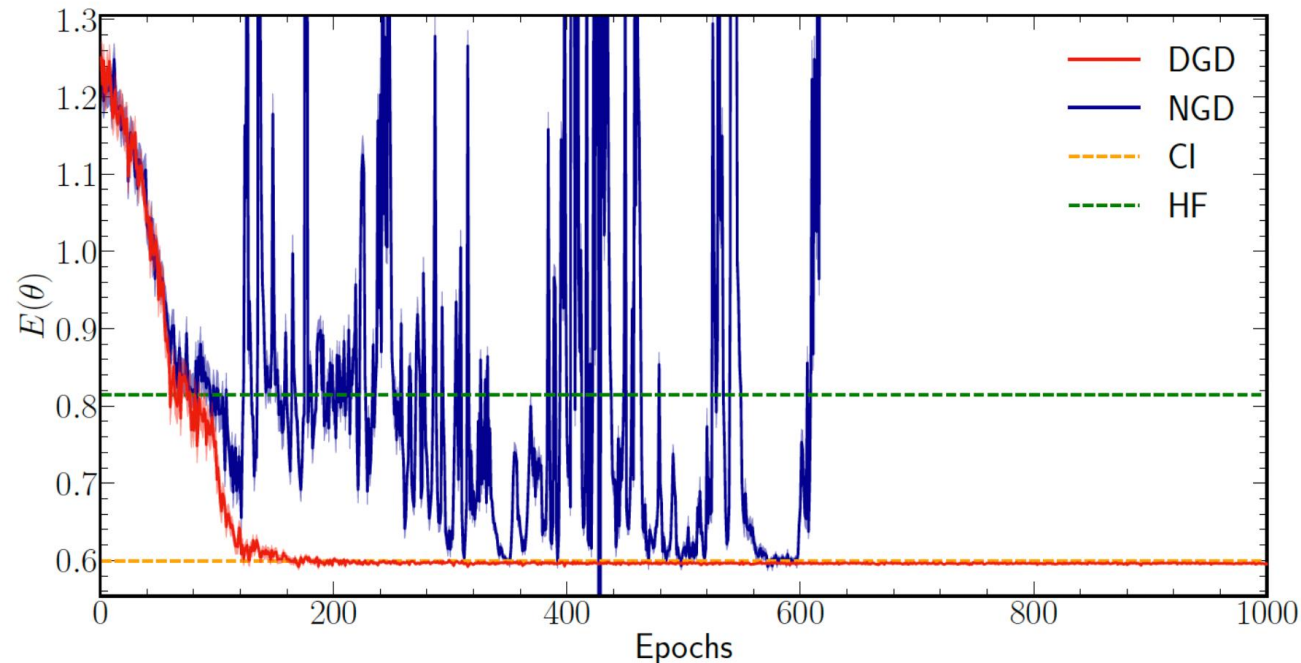
### I. INTRODUCTION

Recently the fields of machine learning and quantum information science have seen a lot of progress. On the one hand, a number of promising results have been obtained suggesting the potential of quantum or classical machine learning to solve complex problems. On the other hand, the exponential growth of the Hilbert space of quantum systems makes the exact solution of the ground state wave function increasingly difficult. In this context, the development of efficient numerical methods to solve the ground state wave function is of great importance. In this work, we introduce a novel method, the minimum-step stochastic reconfiguration (MinSR), which allows for accurate training on unprecedentedly deep NQS with up to 64 layers and more than  $10^5$  parameters in the spin-1/2 Heisenberg  $J_1$ - $J_2$  models on the square lattice. We find that this approach yields better variational energies as compared to existing numerical results and we further observe that the accuracy of our ground state calculations approaches different levels of machine precision on modern GPU and TPU hardware. The MinSR method opens up the potential to make NQS superior as compared to conventional computational methods with the capability to address yet inaccessible regimes for two-dimensional quantum matter in the future.

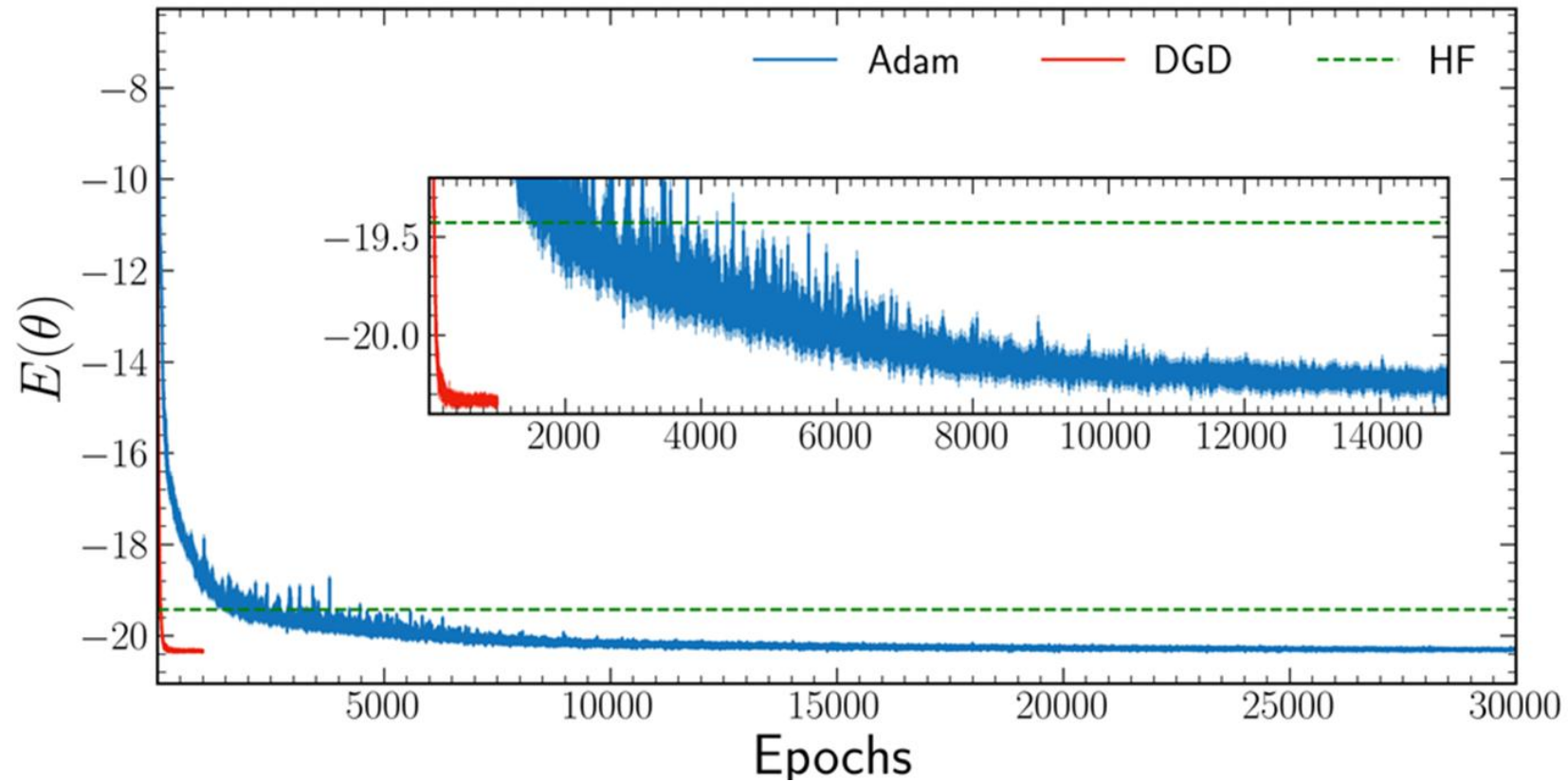
12108v3 [quant-ph] 14 May 2020

# Decisional Gradient Descent

- **System:** chain of **1D spin-polarized fermions**  $H = -\frac{1}{2} \sum_{i=1}^A \nabla_i^2 + \frac{1}{2} \sum_{i=1}^A x_i^2 + \frac{V_0}{\sqrt{2\pi}\sigma_0} \sum_{i<j} e^{-\frac{(x_i-x_j)^2}{2\sigma_0^2}}$
- **Quadratic term:**  $Q_{\text{VMC}}(\theta)_{\theta_i\theta_j} \equiv \frac{1}{4} \sum_{k=1}^A E_{X \sim p_\theta} [\partial_{\theta_i} \partial_{x_k} \ln p_\theta(X) \partial_{\theta_j} \partial_{x_k} \ln p_\theta(X)]$

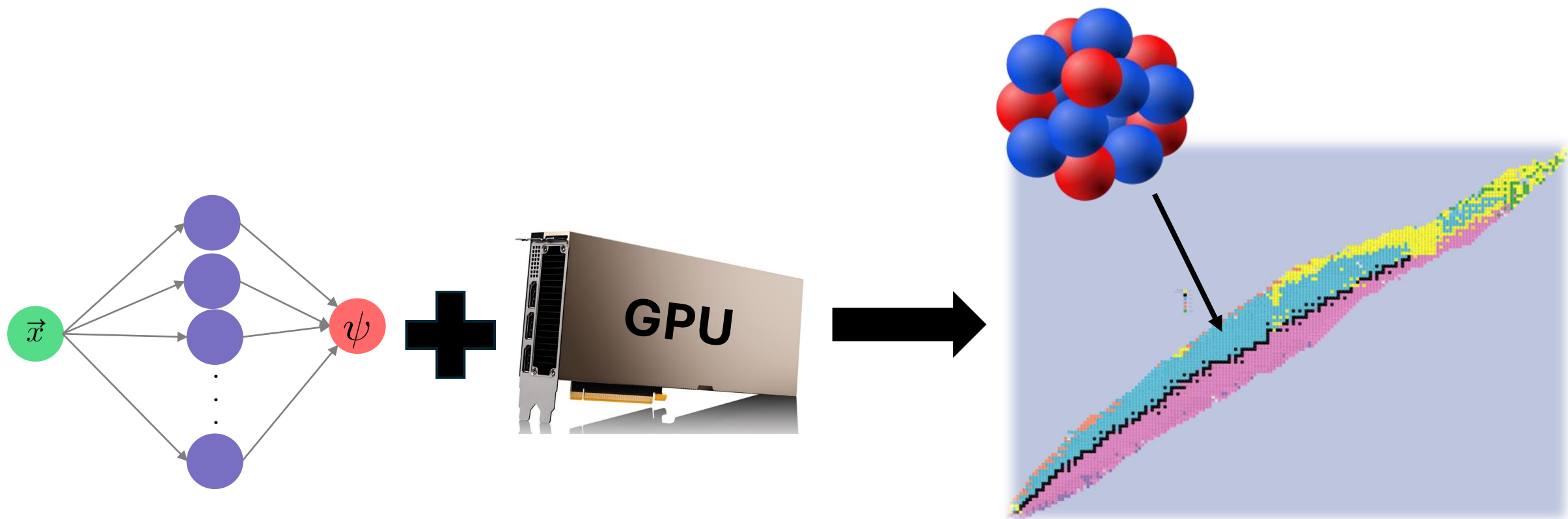


# DGD vs. ADAM



# Conclusions

# Takeaway



# Thank you



**J. Rozalén Sarmiento, A. Rios**



**javi.rozalen@icc.ub.edu**



**Institut de Ciències del Cosmos**  
UNIVERSITAT DE BARCELONA



**EXCELENCIA  
MARÍA  
DE MAEZTU**  
2020-2024



**UNIVERSITAT DE  
BARCELONA**