

# A decisional step for Variational Monte Carlo

Accelerating Variational Monte Carlo with decision geometry

**Collaborators:** Arnau Rios, James Keeble, Javier Rozalén Sarmiento

**Publications:** Physical Review A, **108** 063320 (2023)

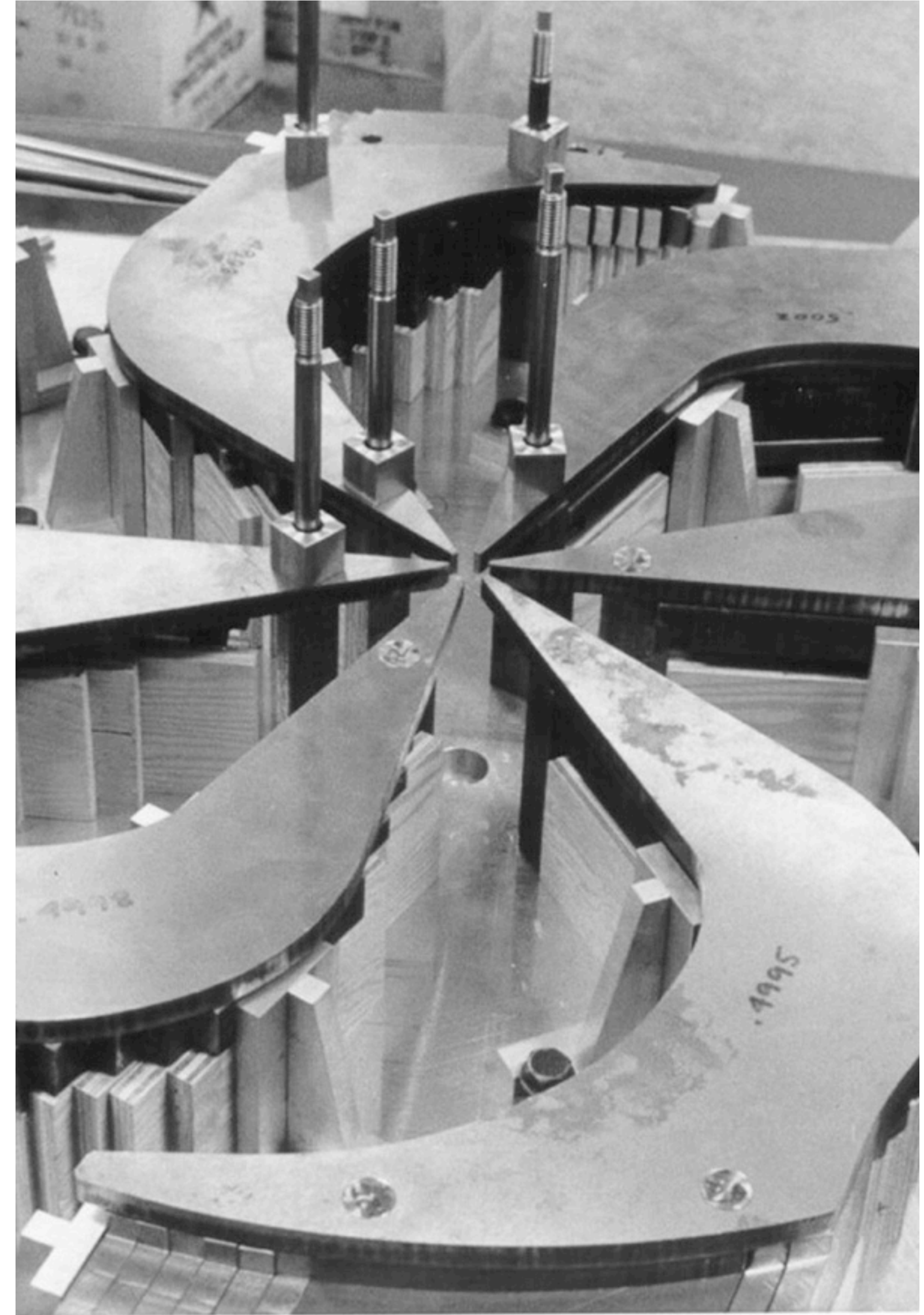
Arxiv: 2401.17550 [nucl-th]

Mehdi Drissi

TRIUMF - Theory department

CEA Saclay - DPhN

10th of April 2024



# Outline

- **Variational Monte Carlo with Neural Quantum States**
  - Overview of VMC with NQS
  - The Kronecker-Factored Approximate Curvature (KFAC)
- **Augmented KFAC for VMC problems**
  - Scaling improvement from a Quasi-Newton approach
  - Direction improvement from MINRES
- **Decision geometry for VMC**
  - Game theory reformulation of VMC
  - Testing decisional gradient descent

# Variational Monte-Carlo in a nutshell

## General Many-body problem

- Many-body system of interacting particles
  - Input Hamiltonian:  $H$
- Here focus on:
  - Many-body system of  $A$  fermions
  - Canonical ensemble at  $T = 0$
- Goal:
  - Finding  $\{E_{gs}, |\Psi_{gs}\rangle\}$  s.t.  $H|\Psi_{gs}\rangle = E_{gs}|\Psi_{gs}\rangle$

## Variational approach

- Rayleigh-Ritz variational principle

$$\forall |\Psi\rangle \in \mathcal{H}_A, \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq \frac{\langle \Psi_{gs} | H | \Psi_{gs} \rangle}{\langle \Psi_{gs} | \Psi_{gs} \rangle}$$

Variational  
reformulation

$$E_{gs} = \min_{|\Psi\rangle} \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$
$$|\Psi_{gs}\rangle = \operatorname{argmin}_{|\Psi\rangle} \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

# Variational Monte-Carlo in a nutshell

## General Many-body problem

- Many-body system of interacting particles
  - Input Hamiltonian:  $H$
- Here focus on:
  - Many-body system of  $A$  fermions
  - Canonical ensemble at  $T = 0$
- Goal:
  - Finding  $\{E_{gs}, |\Psi_{gs}\rangle\}$  s.t.  $H|\Psi_{gs}\rangle = E_{gs}|\Psi_{gs}\rangle$

## Technical challenges and solutions of VMC

- |   |   |                   |
|---|---|-------------------|
| ● <u>Infinite dimensional variational space</u>                             | → | <b>Tradeoffs</b>  |
| ○ Ansatz based wave-functions   |   | Biased estimation |
| ● <u>High-dimension integrals</u> <small>[Metropolis et al. (1953)]</small> | → | Statistical noise |
| ○ Markov Chain Monte-Carlo sampling   |   |                   |
| ● <u>Non-linear global optimization problem</u>                             | → | Local minima      |
| ○ Iterative linear/quadratic local optimization                             |   |                   |

## Variational approach

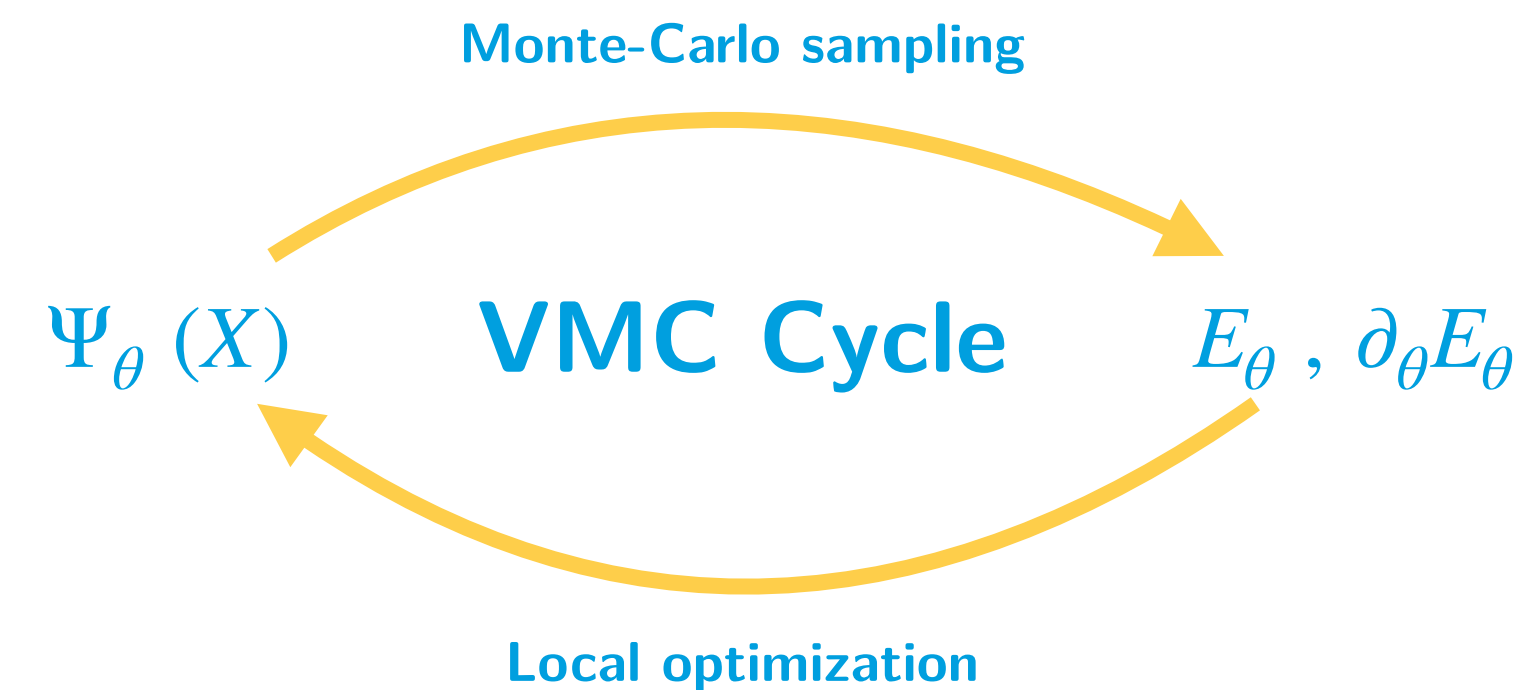
- Rayleigh-Ritz variational principle

$$\forall |\Psi\rangle \in \mathcal{H}_A, \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq \frac{\langle \Psi_{gs} | H | \Psi_{gs} \rangle}{\langle \Psi_{gs} | \Psi_{gs} \rangle}$$

Variational reformulation

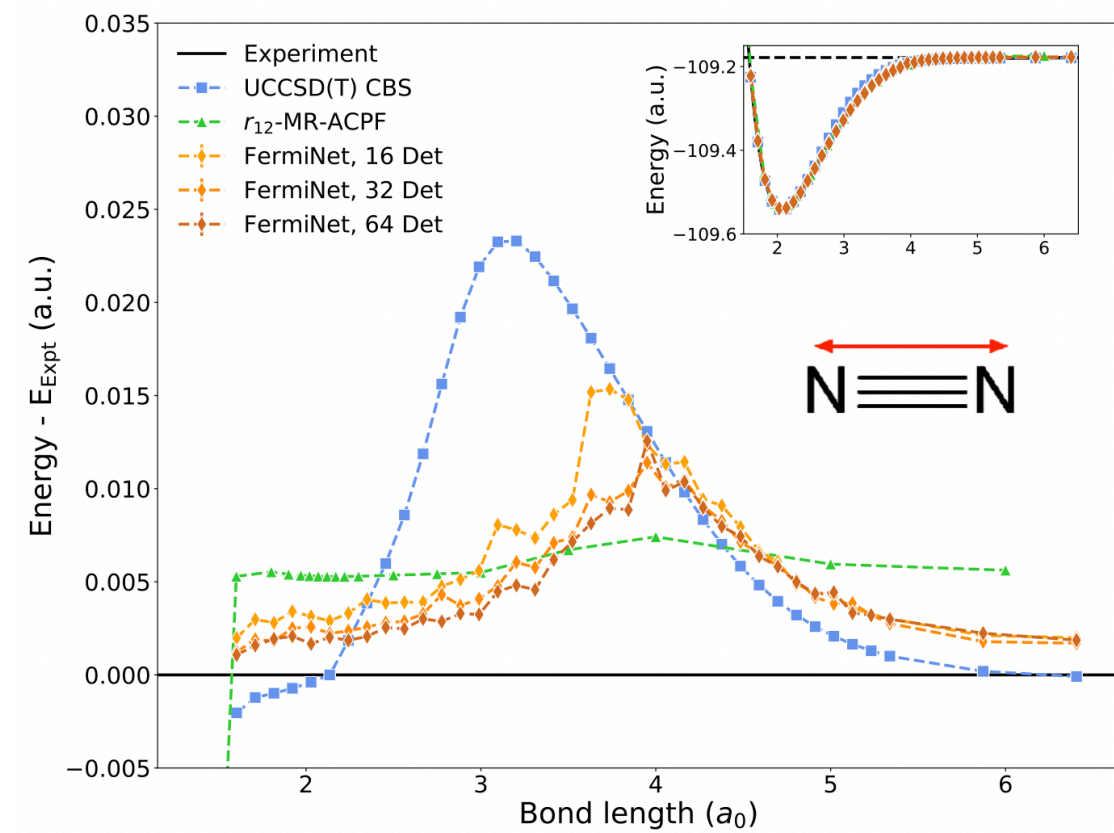
$$E_{gs} = \min_{|\Psi\rangle} \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

$$|\Psi_{gs}\rangle = \operatorname{argmin}_{|\Psi\rangle} \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$



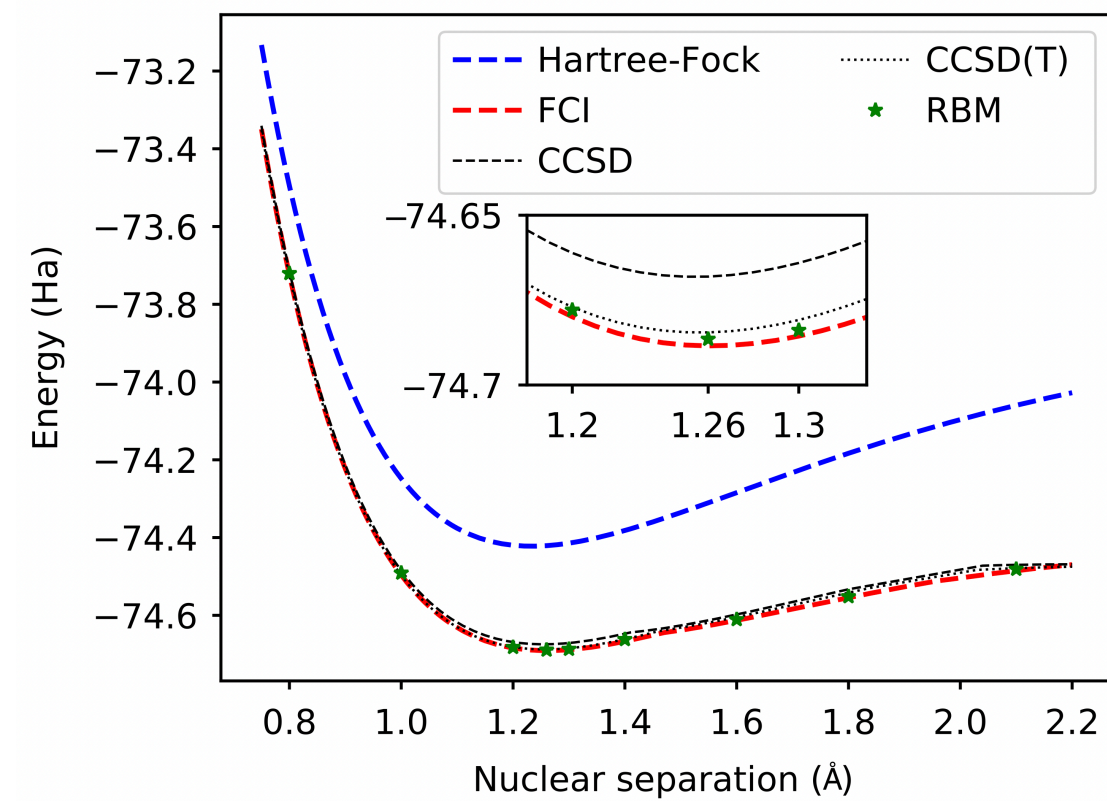
# A promising approach for many-body theory

## FermiNet (DeepMind & Imperial)



[Pfau, Spencer, Matthews and Foulkes (2020)]

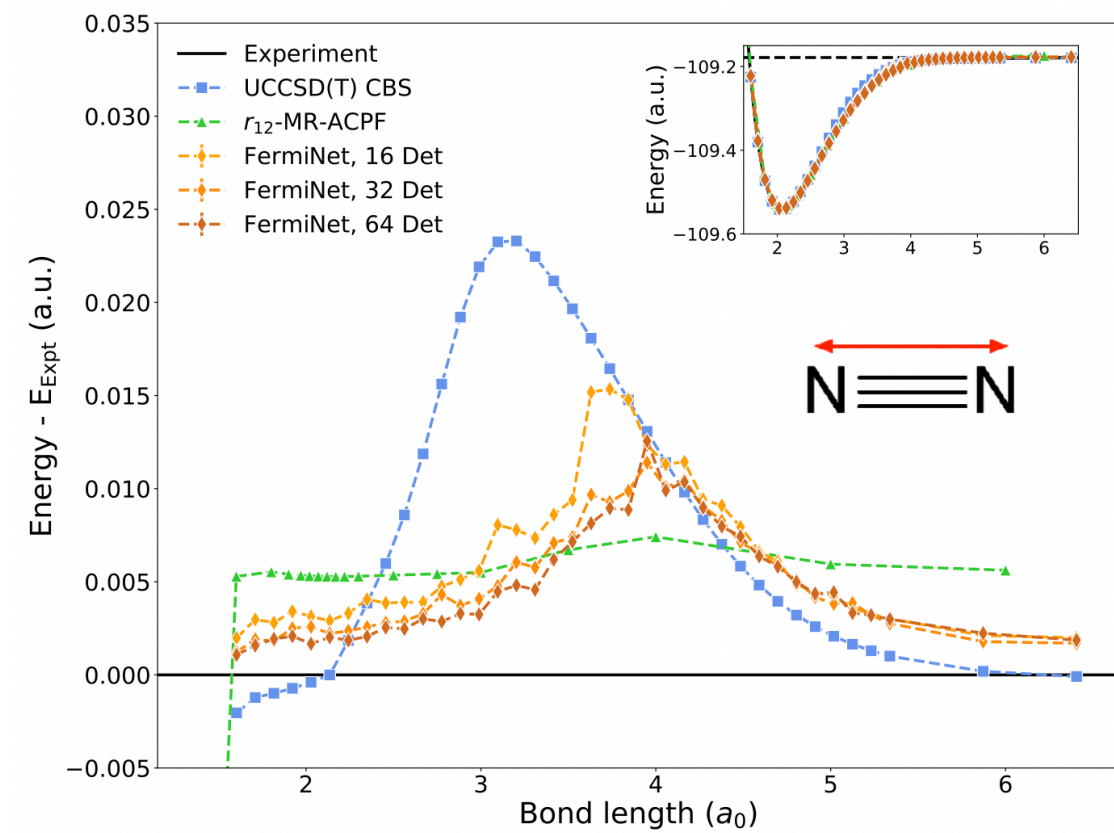
## NetKet (EPFL & Flatiron)



[Choo, Mezzacapo and Carleo (2020)]

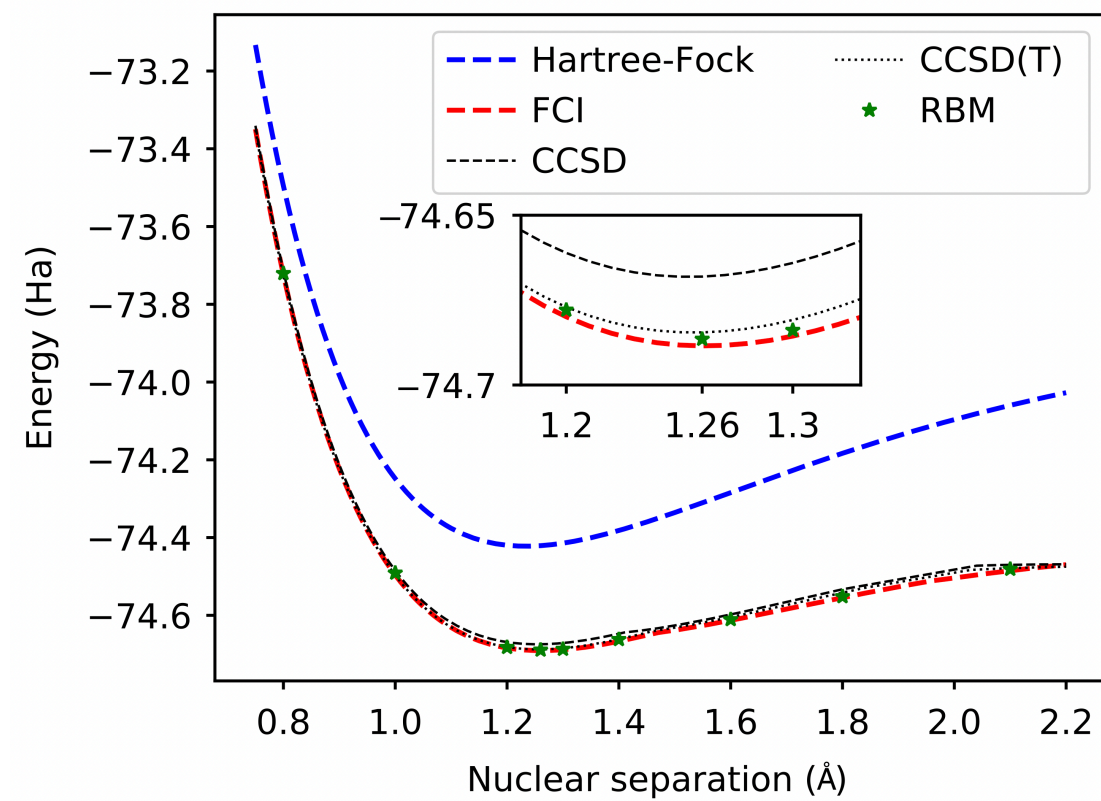
# A promising approach for many-body theory

## FermiNet (DeepMind & Imperial)



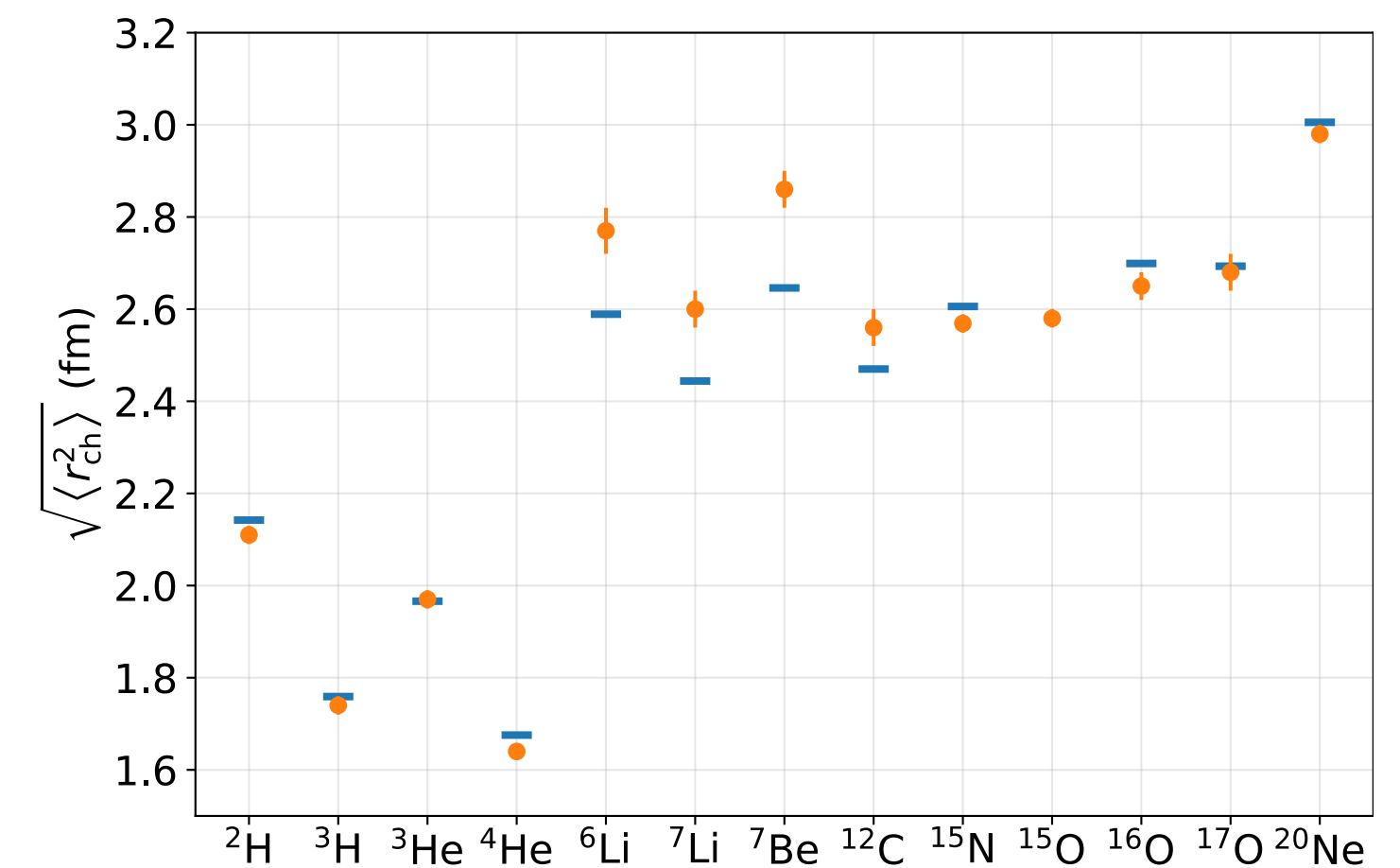
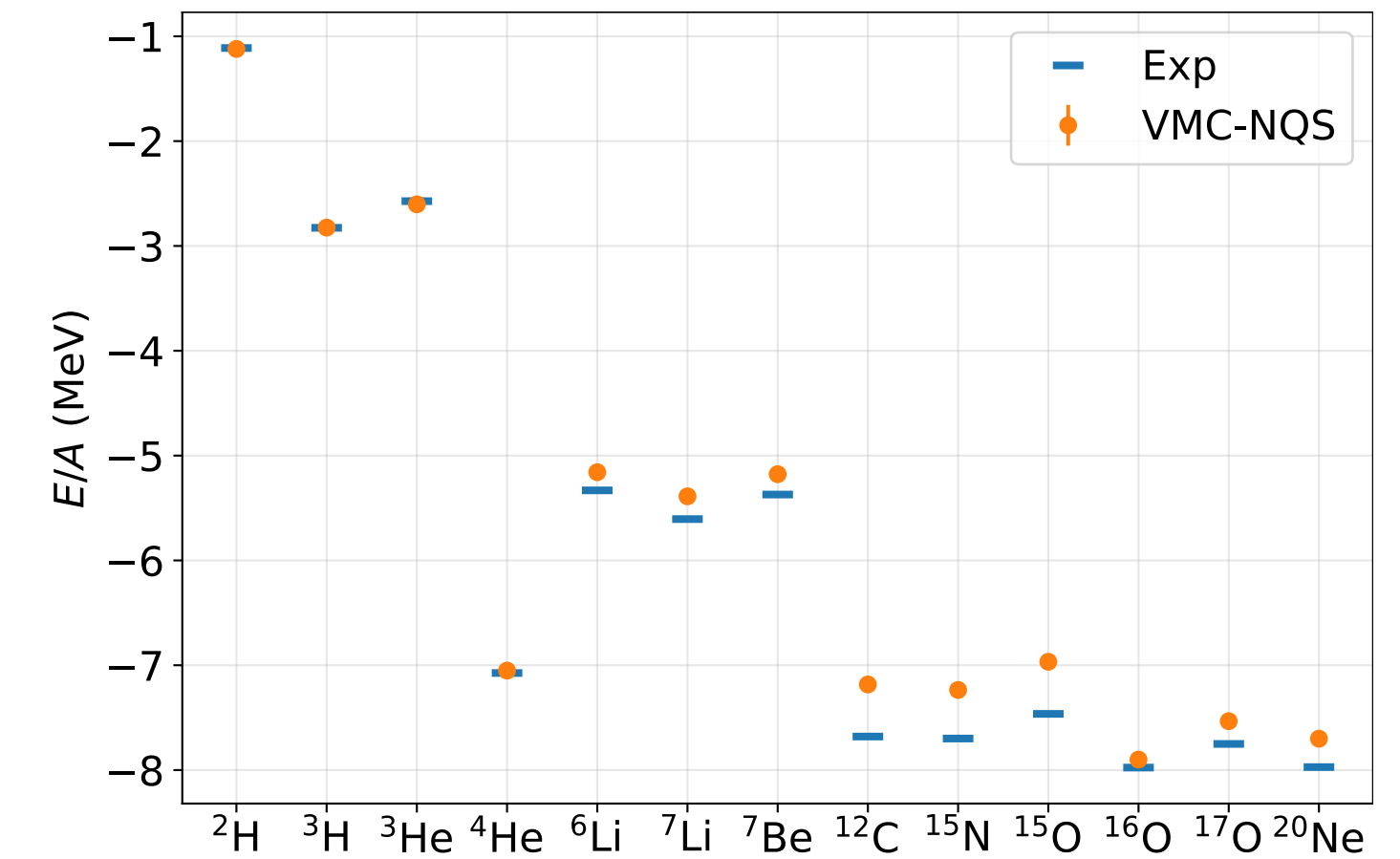
[Pfau, Spencer, Matthews and Foulkes (2020)]

## NetKet (EPFL & Flatiron)



[Choo, Mezzacapo and Carleo (2020)]

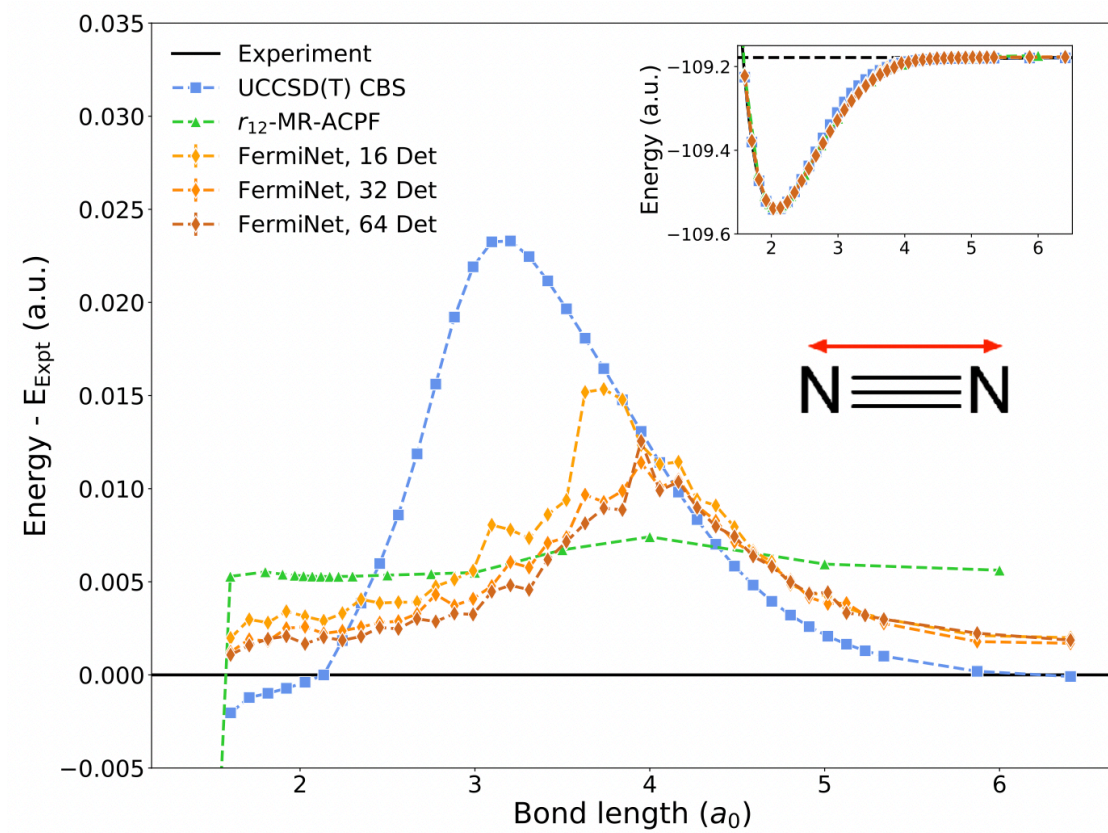
## MPNN (Argonne National Lab)



[A. Gnech, B. Fore, A. Lovato, 2308.16266 [nucl-th]]

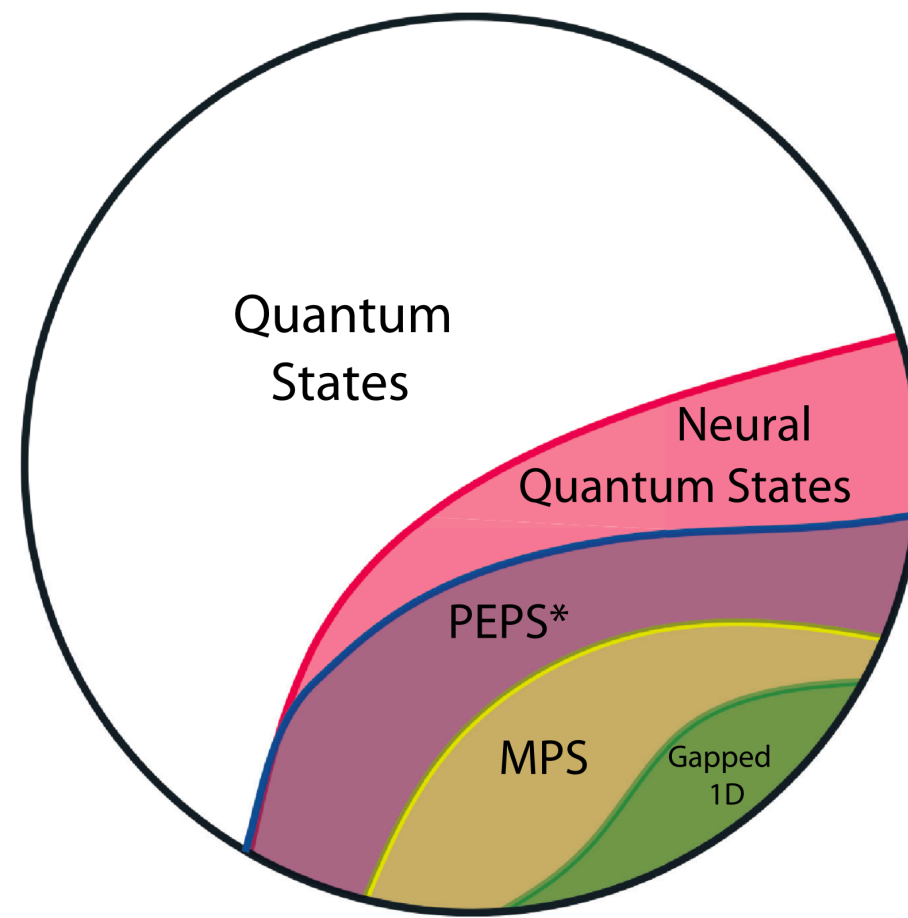
# A promising approach for many-body theory

## FermiNet (DeepMind & Imperial)



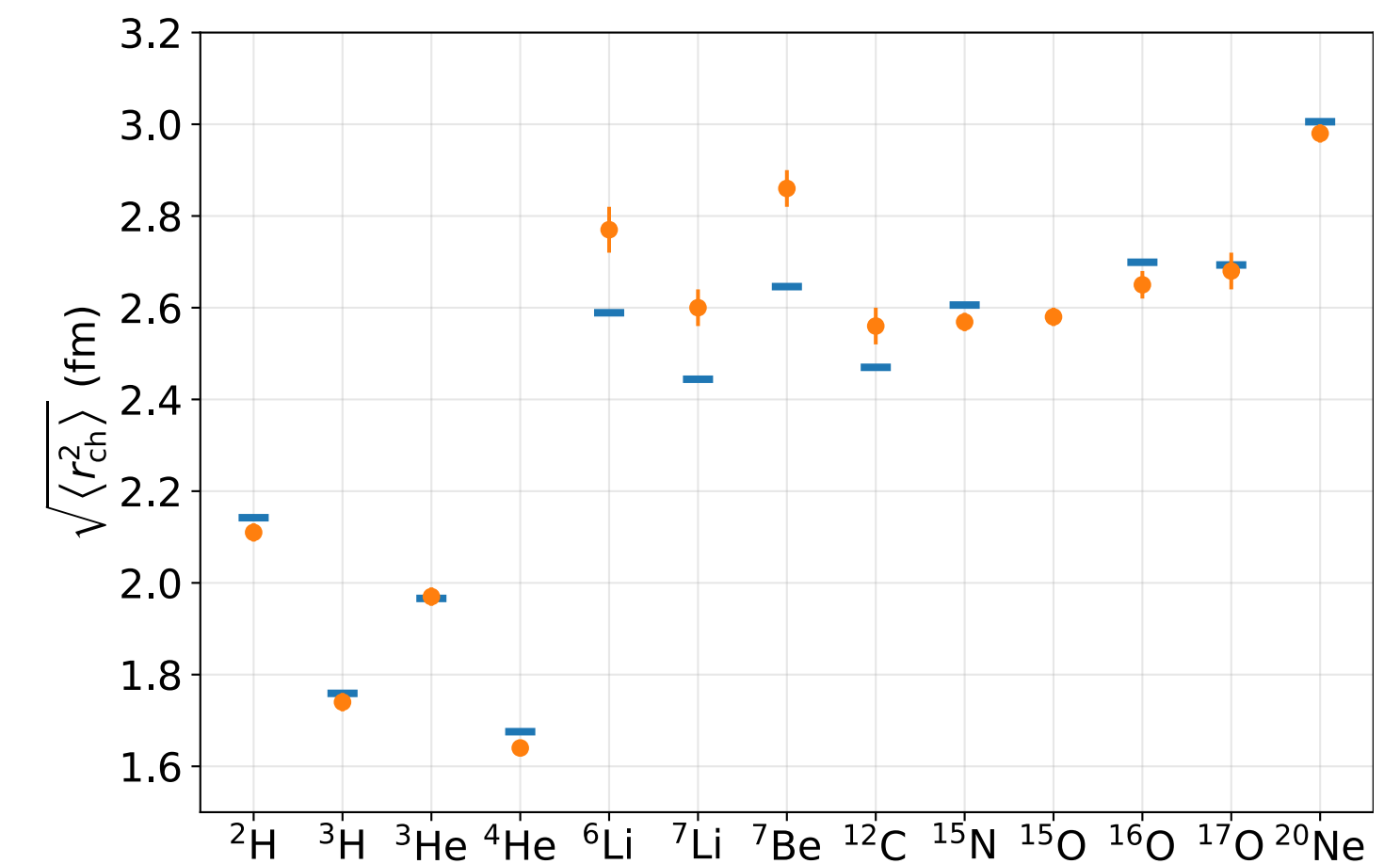
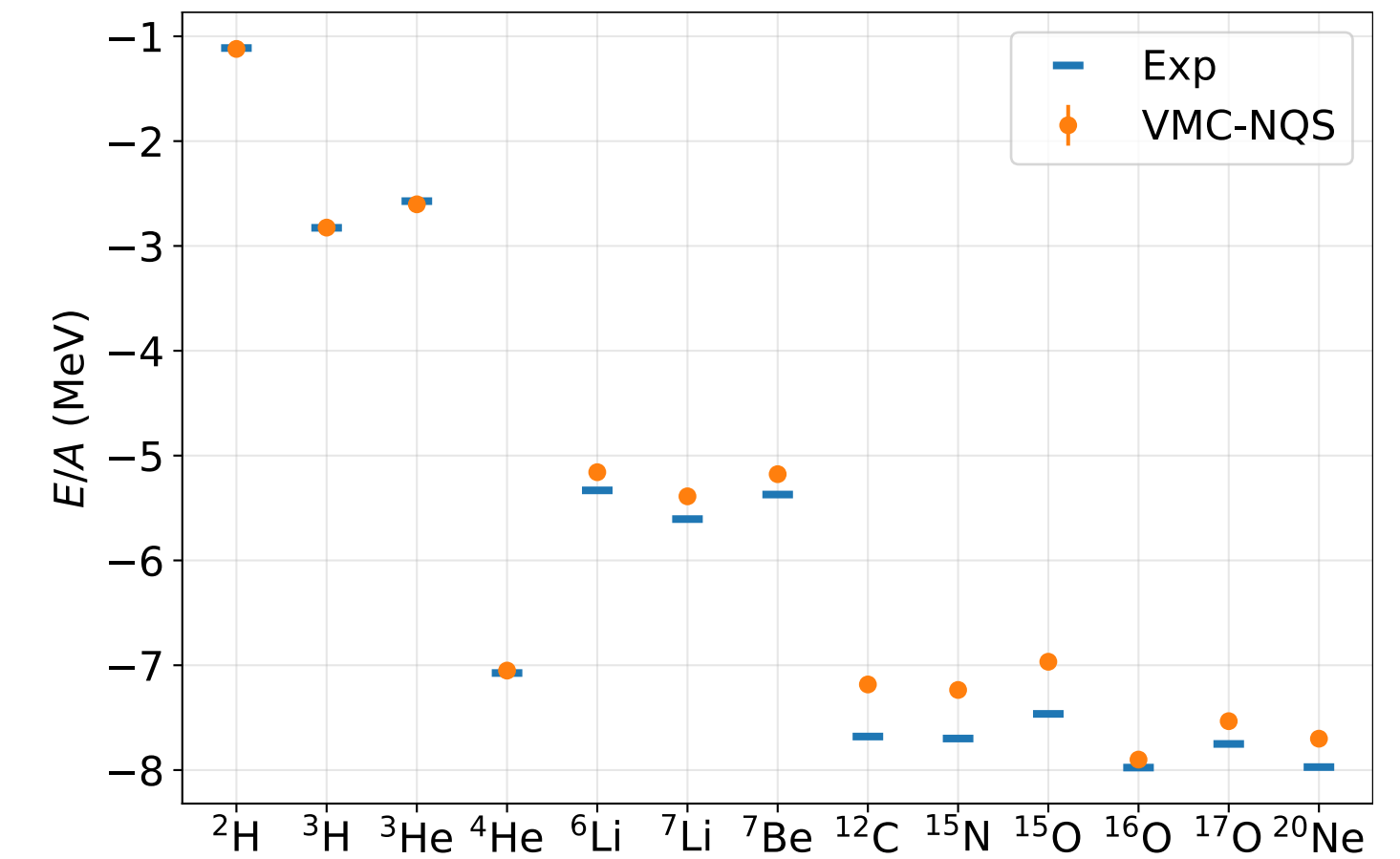
[Pfau, Spencer, Matthews and Foulkes (2020)]

## A flexible tool for Variational Monte Carlo (VMC)



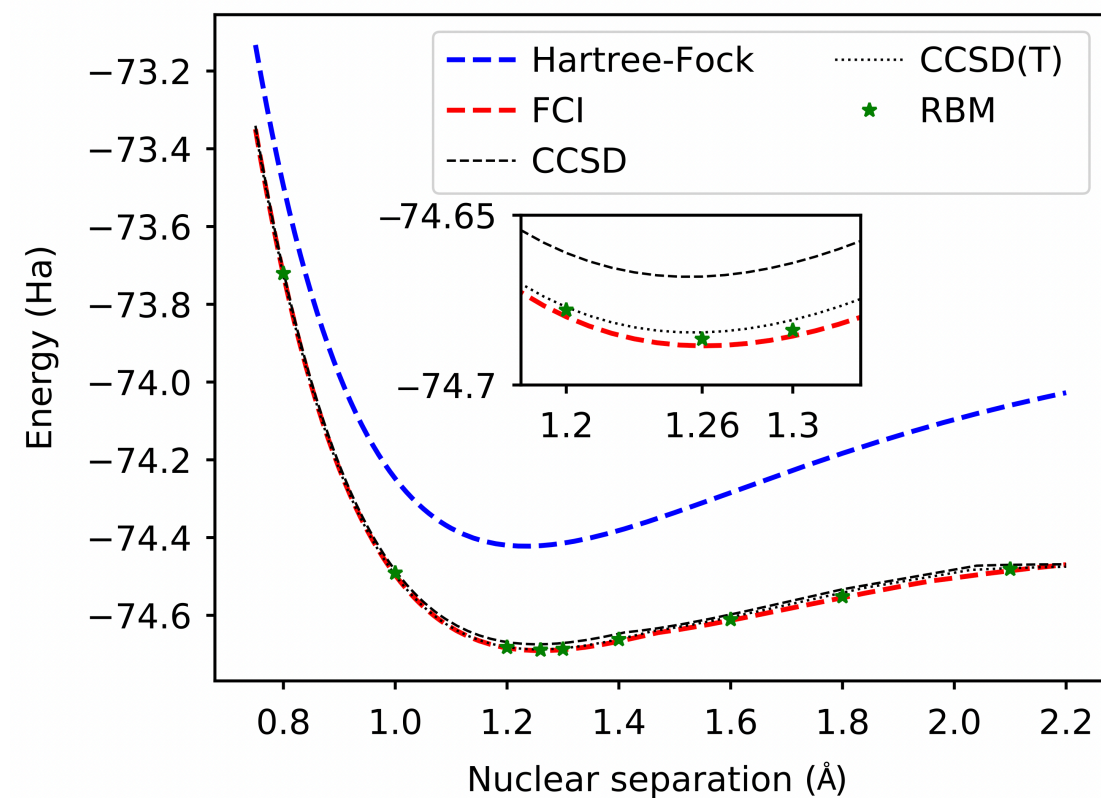
[O. Sharir, A. Shashua, G. Carleo (2022)]

## MPNN (Argonne National Lab)



[A. Gnech, B. Fore, A. Lovato, 2308.16266 [nucl-th] ]

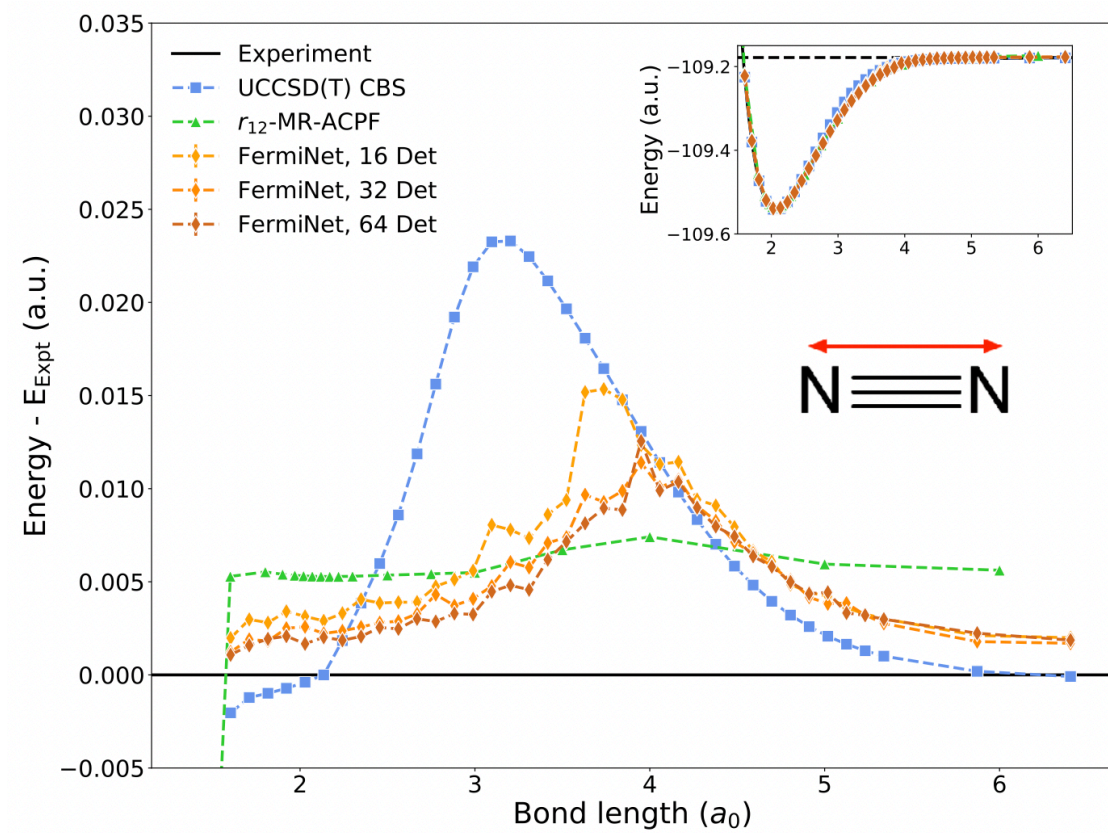
## NetKet (EPFL & Flatiron)



[Choo, Mezzacapo and Carleo (2020)]

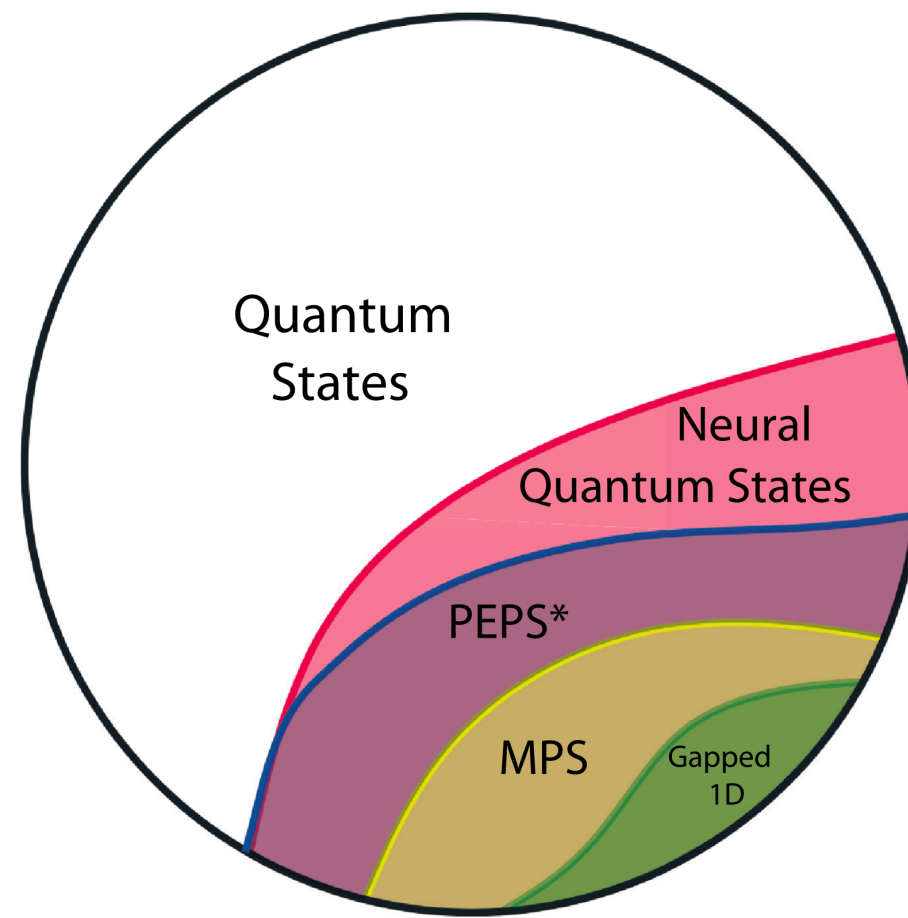
# A promising approach for many-body theory

## FermiNet (DeepMind & Imperial)



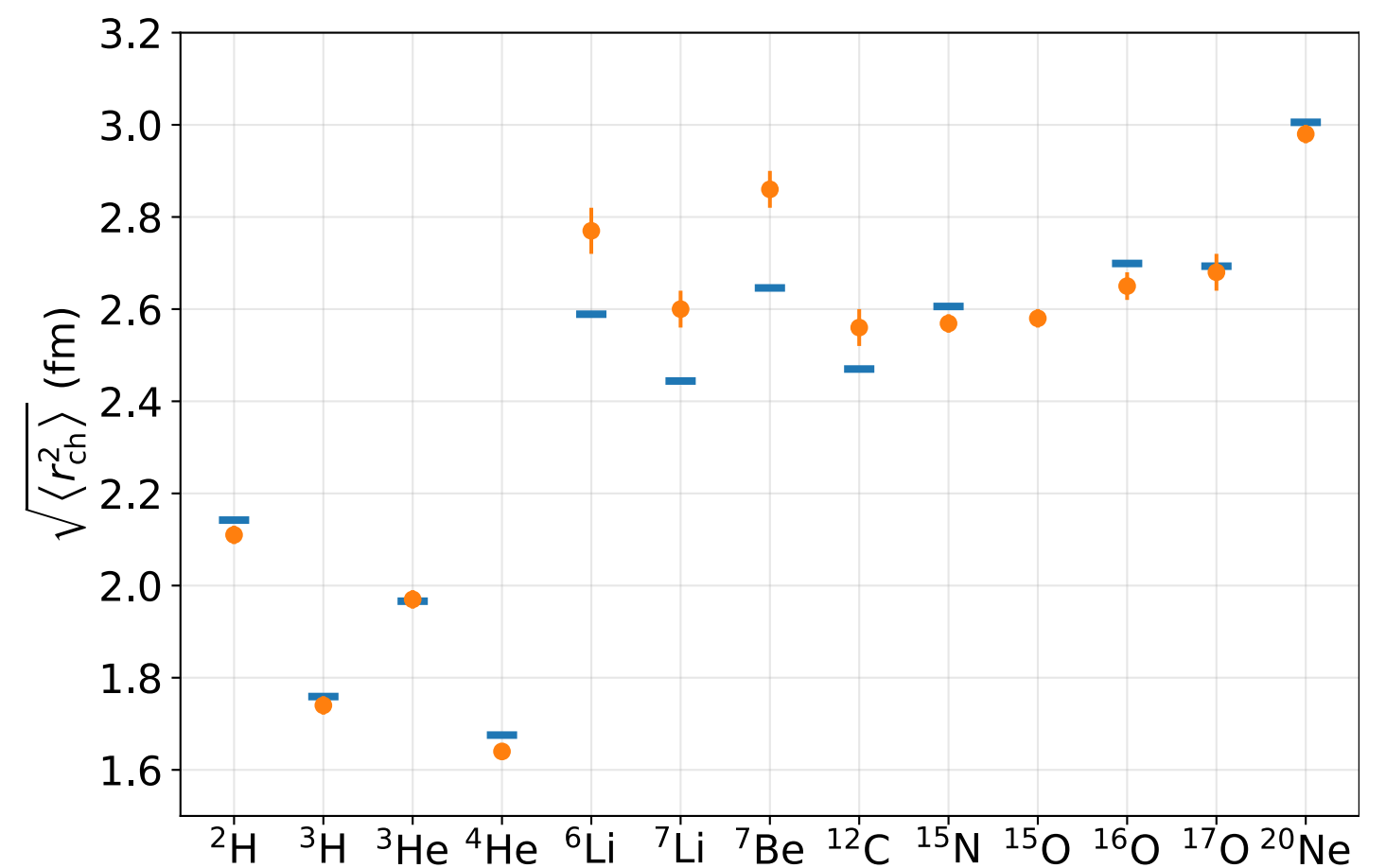
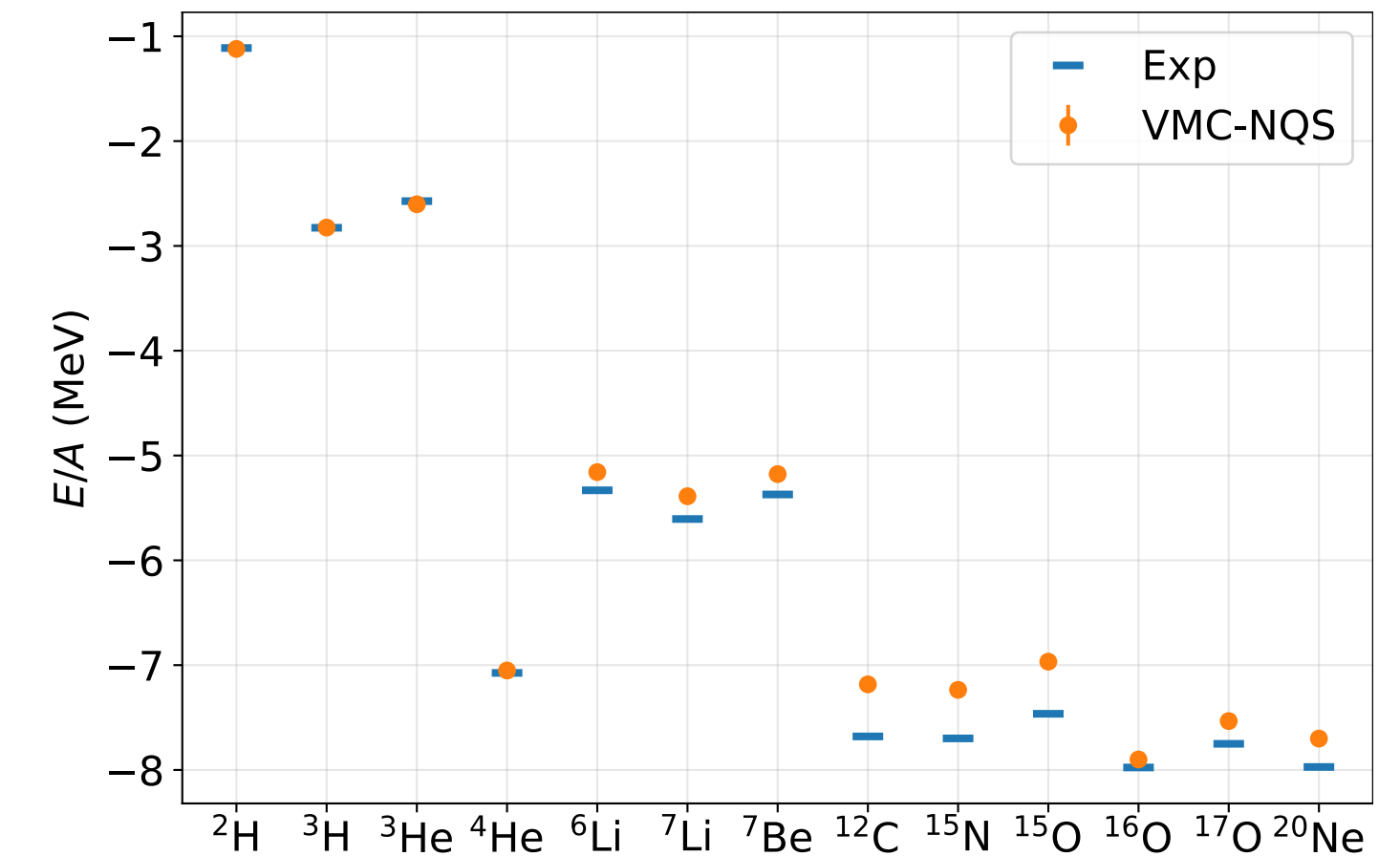
[Pfau, Spencer, Matthews and Foulkes (2020)]

## A flexible tool for Variational Monte Carlo (VMC)



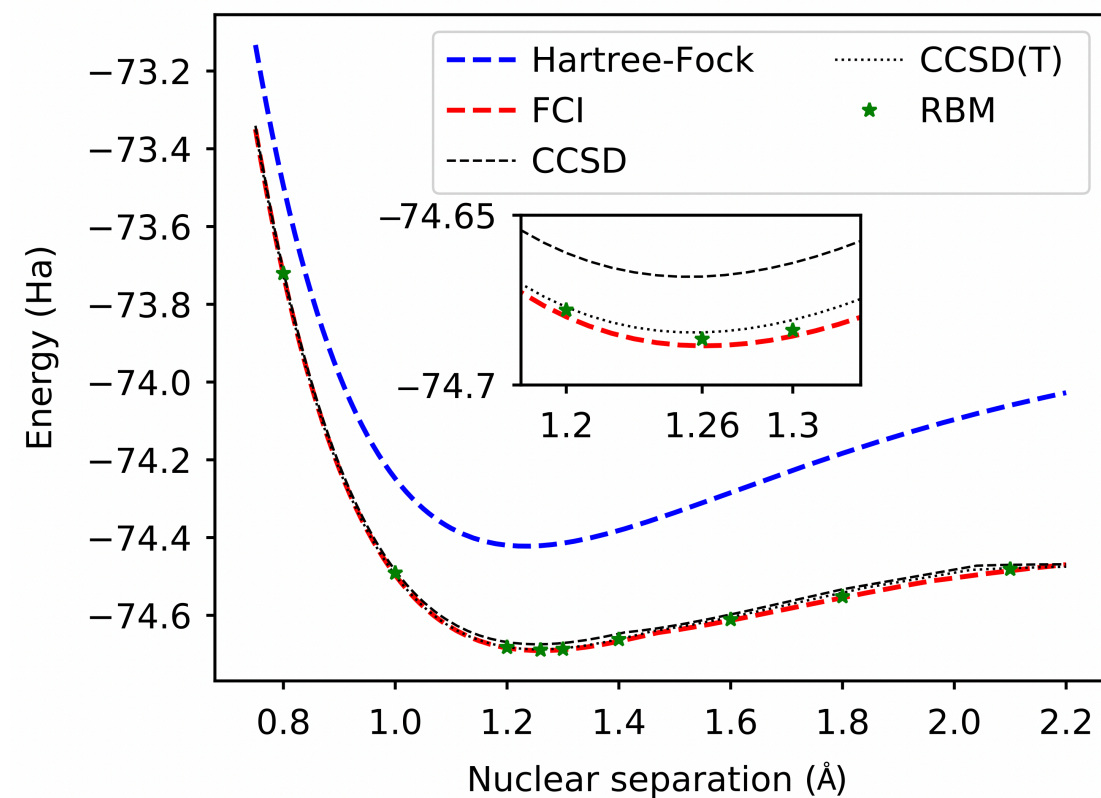
[O. Sharir, A. Shashua, G. Carleo (2022)]

## MPNN (Argonne National Lab)



[A. Gnech, B. Fore, A. Lovato, 2308.16266 [nucl-th]]

## NetKet (EPFL & Flatiron)



[Choo, Mezzacapo and Carleo (2020)]

## Current work on NQS

- Development of a simple NQS to test new ideas

[J. Keeble, M. Drissi, A. Rojo-Francàs, B. Juliá-Díaz, A. Rios, PRA **108**, 063320 (2023)]

- Recent development of a new optimizer tailored to VMC

[M. Drissi, J. Keeble, J. Rozalén Sarmiento, A. Rios, 2401.17550 [nucl-th]]



# A simple yet insightful many-body problem

## Many-body system

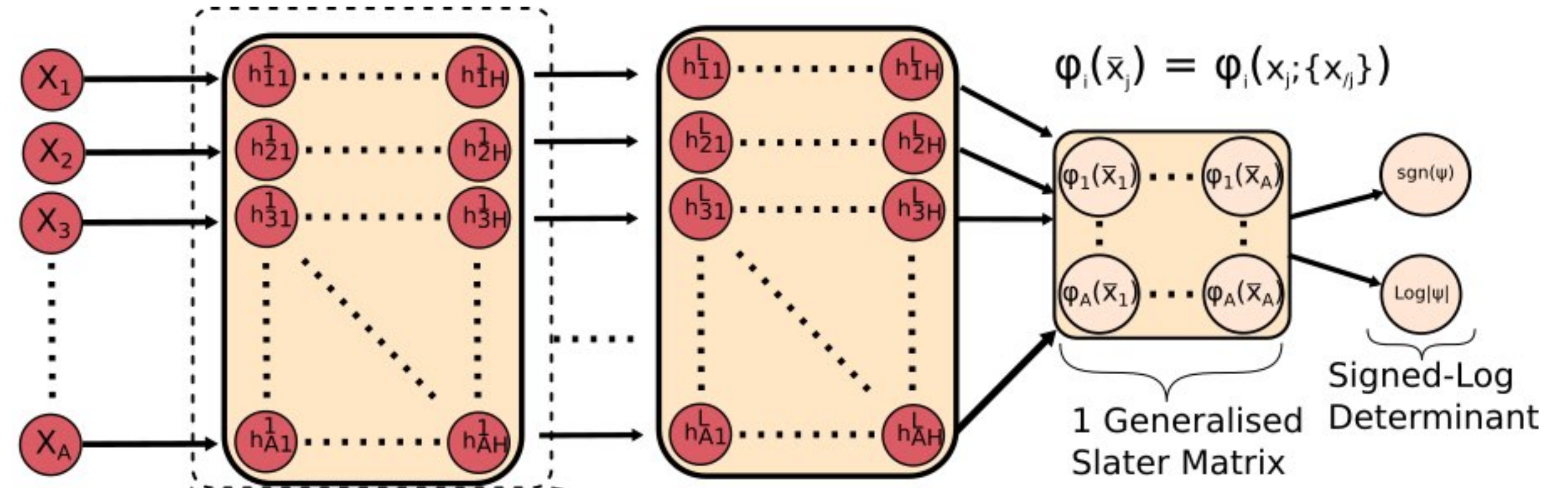
- Hamiltonian in 1D

$$H = - \sum_i \frac{1}{2} \partial_{x_i}^2 + \sum_i \frac{1}{2} x_i^2 + \sum_{i < j} \frac{V_0}{\sqrt{2\pi\sigma_0}} \exp\left(-\frac{(x_i - x_j)^2}{2\sigma_0^2}\right)$$

Harmonic trap
Gaussian interaction

- Constraints:

- Fixed particle number  $A$
- Fixed temperature  $T = 0$



# A simple yet insightful many-body problem

## Many-body system

- Hamiltonian in 1D

$$H = - \sum_i \frac{1}{2} \partial_{x_i}^2 + \sum_i \frac{1}{2} x_i^2 + \sum_{i < j} \frac{V_0}{\sqrt{2\pi\sigma_0}} \exp\left(-\frac{(x_i - x_j)^2}{2\sigma_0^2}\right)$$

Harmonic trap

Gaussian interaction

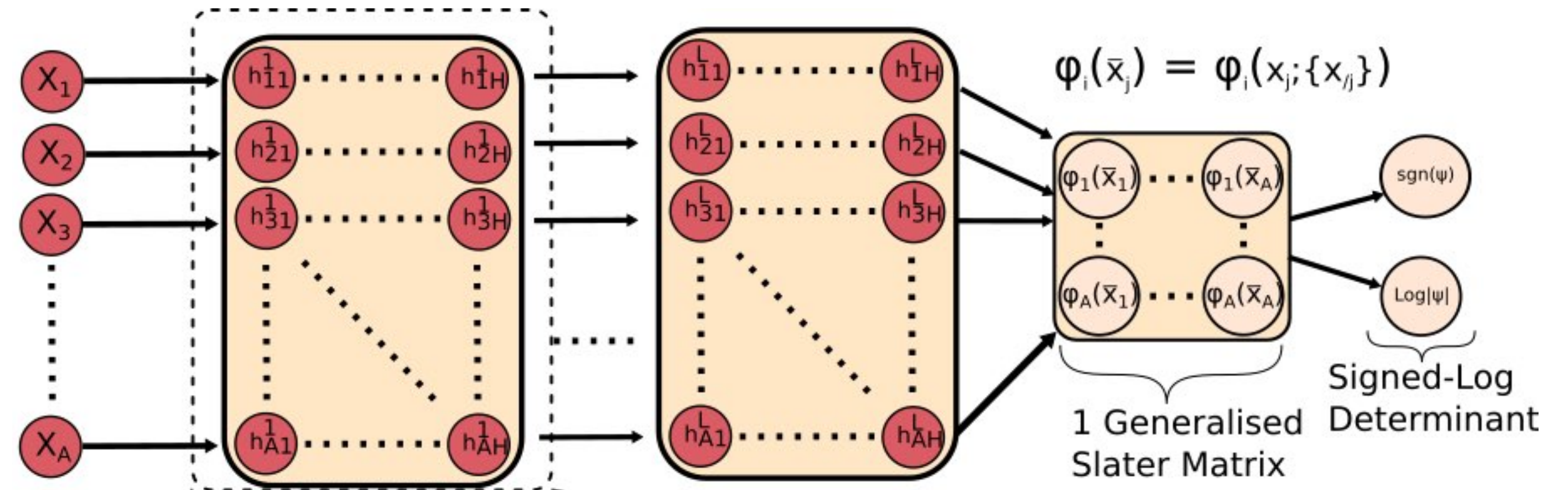
- Constraints:

- Fixed particle number  $A$
- Fixed temperature  $T = 0$

## NQS architecture

- Default architectural hyperparameters

- Number of layers:  $L = 2$
- Width of each layer:  $H = 64$
- Number of determinants:  $D = 1$
- Total number of parameters  $\sim 10\,000$



# A simple yet insightful many-body problem

## Many-body system

- Hamiltonian in 1D

$$H = - \sum_i \frac{1}{2} \partial_{x_i}^2 + \sum_i \frac{1}{2} x_i^2 + \sum_{i < j} \frac{V_0}{\sqrt{2\pi\sigma_0}} \exp\left(-\frac{(x_i - x_j)^2}{2\sigma_0^2}\right)$$

Harmonic trap

Gaussian interaction

- Constraints:

- Fixed particle number  $A$
- Fixed temperature  $T = 0$

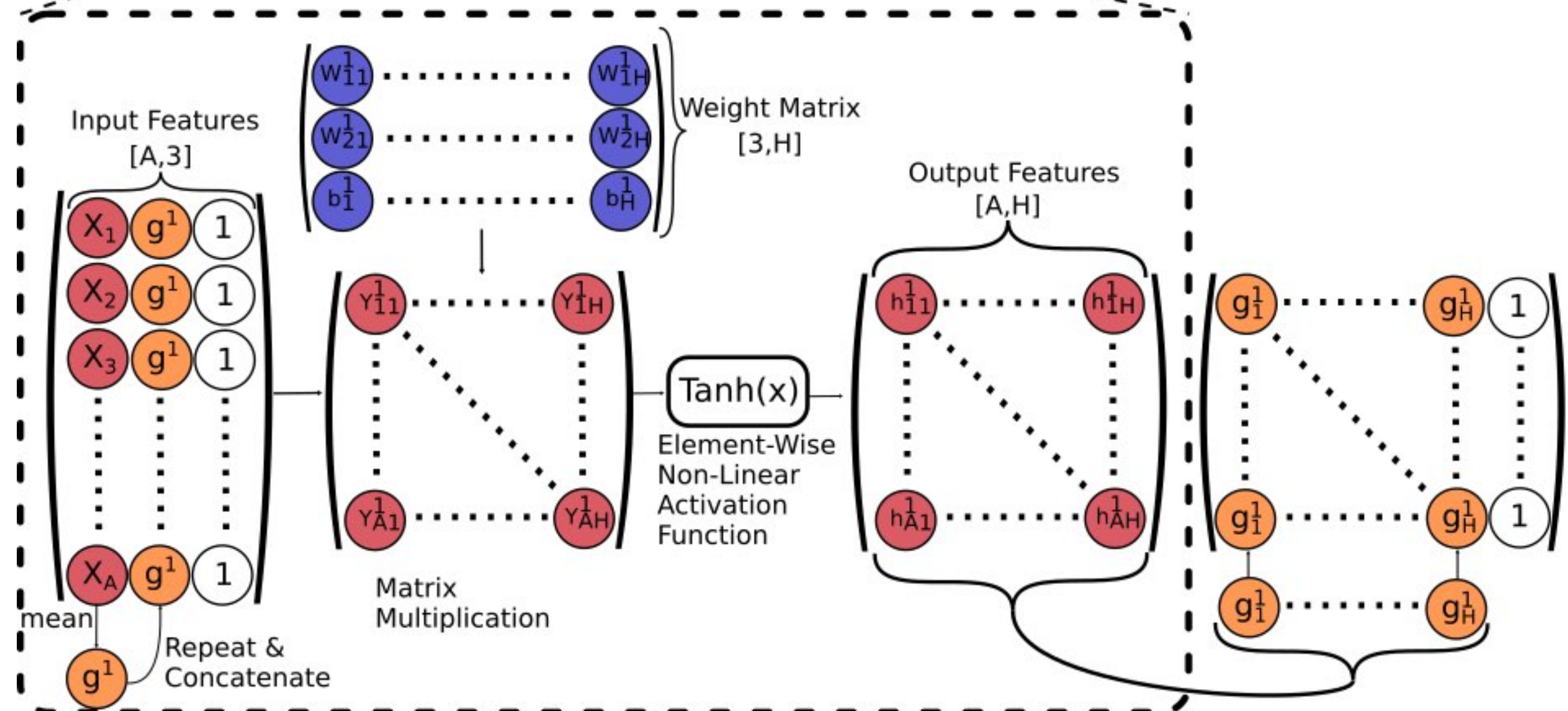
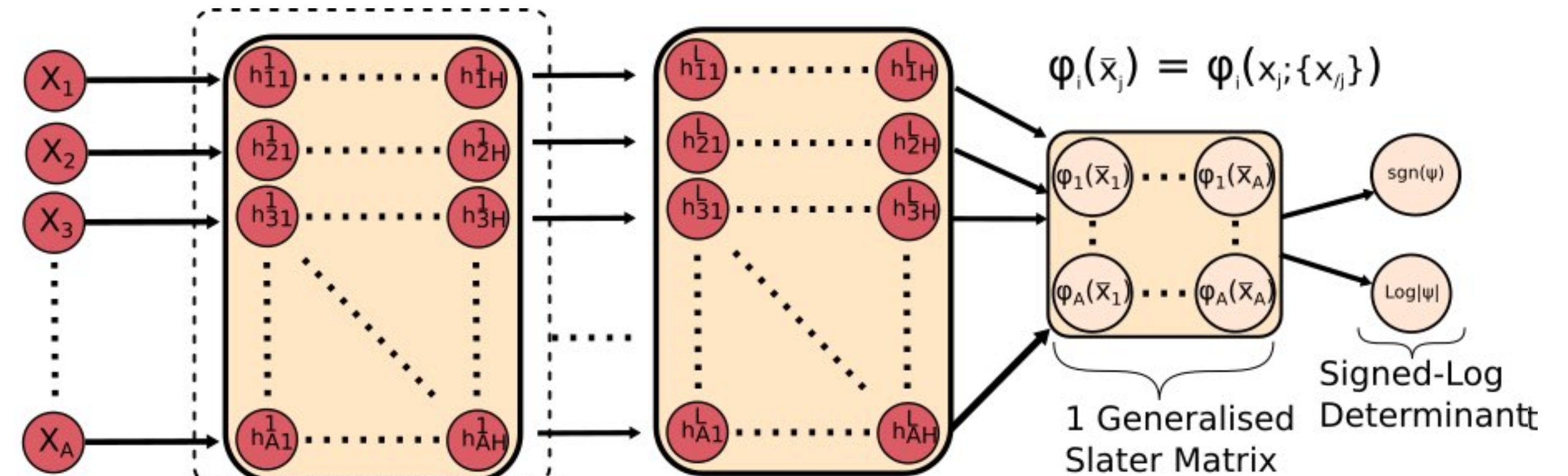
## NQS architecture

- Default architectural hyperparameters

- Number of layers:  $L = 2$
- Width of each layer:  $H = 64$
- Number of determinants:  $D = 1$
- Total number of parameters  $\sim 10\,000$

- Permutation equivariant layers

- Permutation of input rows
- Permutation of output rows
- Propagates all the way to the orbitals
- Final layer with determinant: equivariance  $\Rightarrow$  antisymmetry



# Outline

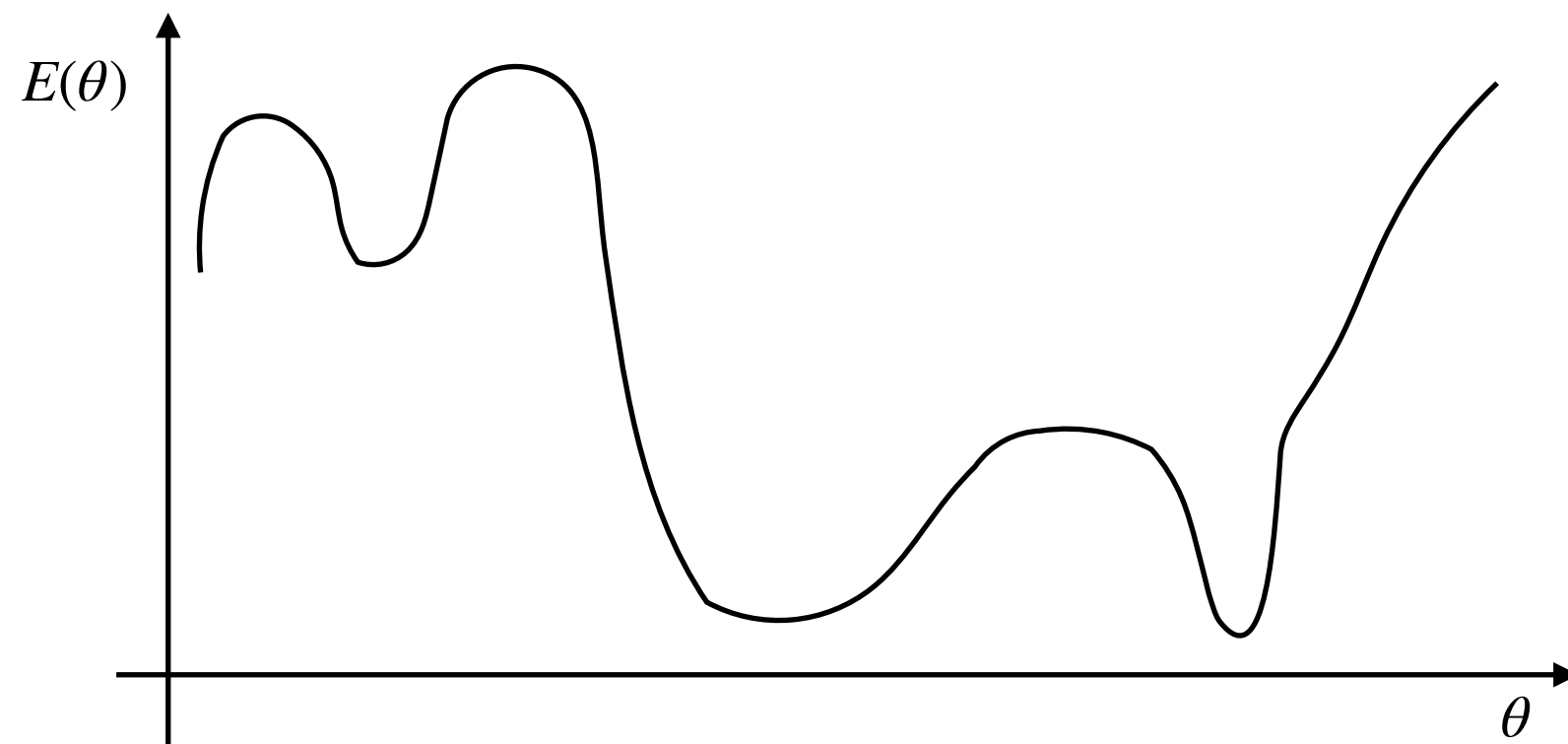
- **Variational Monte Carlo with Neural Quantum States**
  - Overview of VMC with NQS
  - The Kronecker-Factored Approximate Curvature (KFAC)
- **Augmented KFAC for VMC problems**
  - Scaling improvement from a Quasi-Newton approach
  - Direction improvement from MINRES
- **Decision geometry for VMC**
  - Game theory reformulation of VMC
  - Testing decisional gradient descent

# Global non-linear optimization for NQS

## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## 1D example





# Global non-linear optimization for NQS

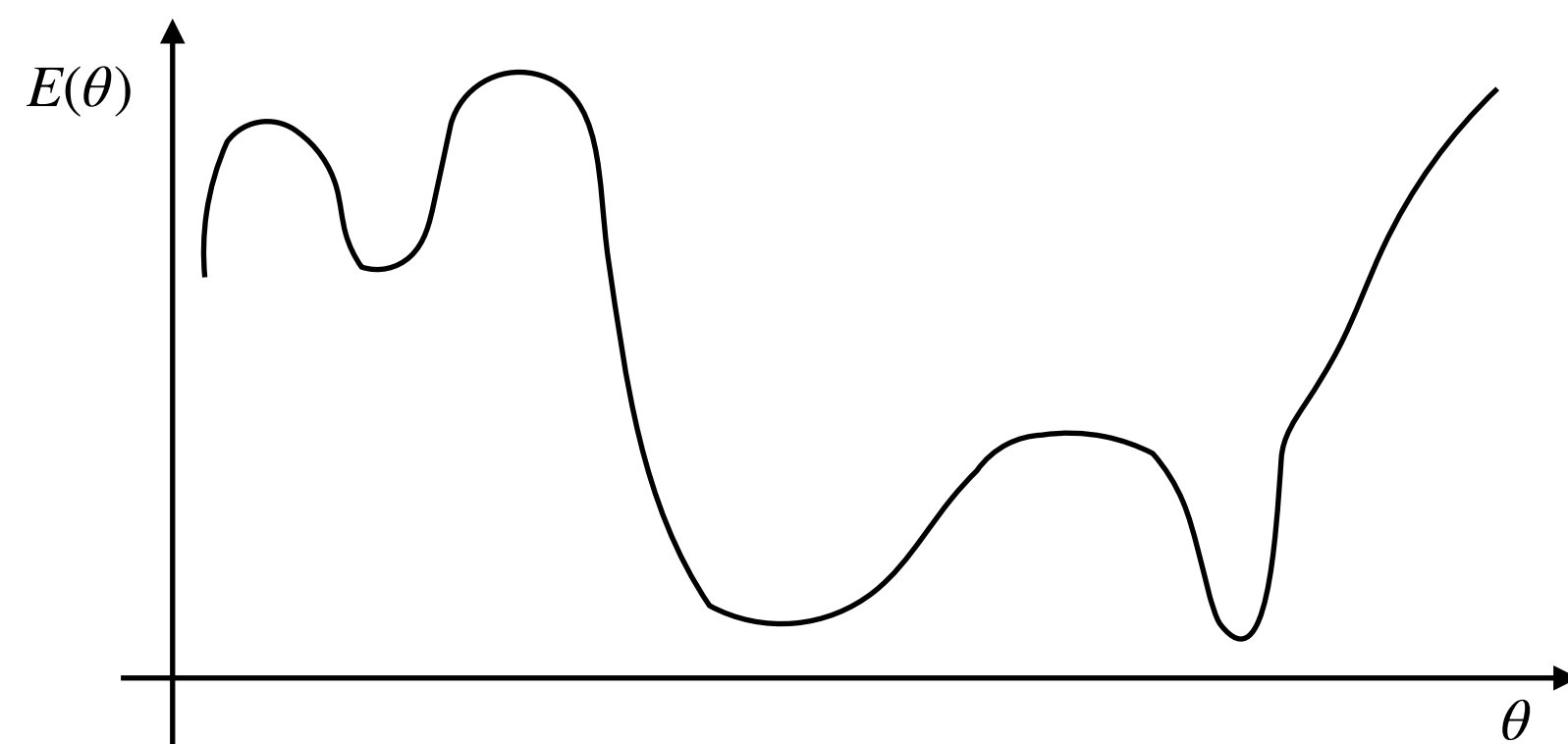
## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## General strategy

- Complicated problem   
Many trivial problems 
- Sequence of linear/quadratic optimizations

## 1D example

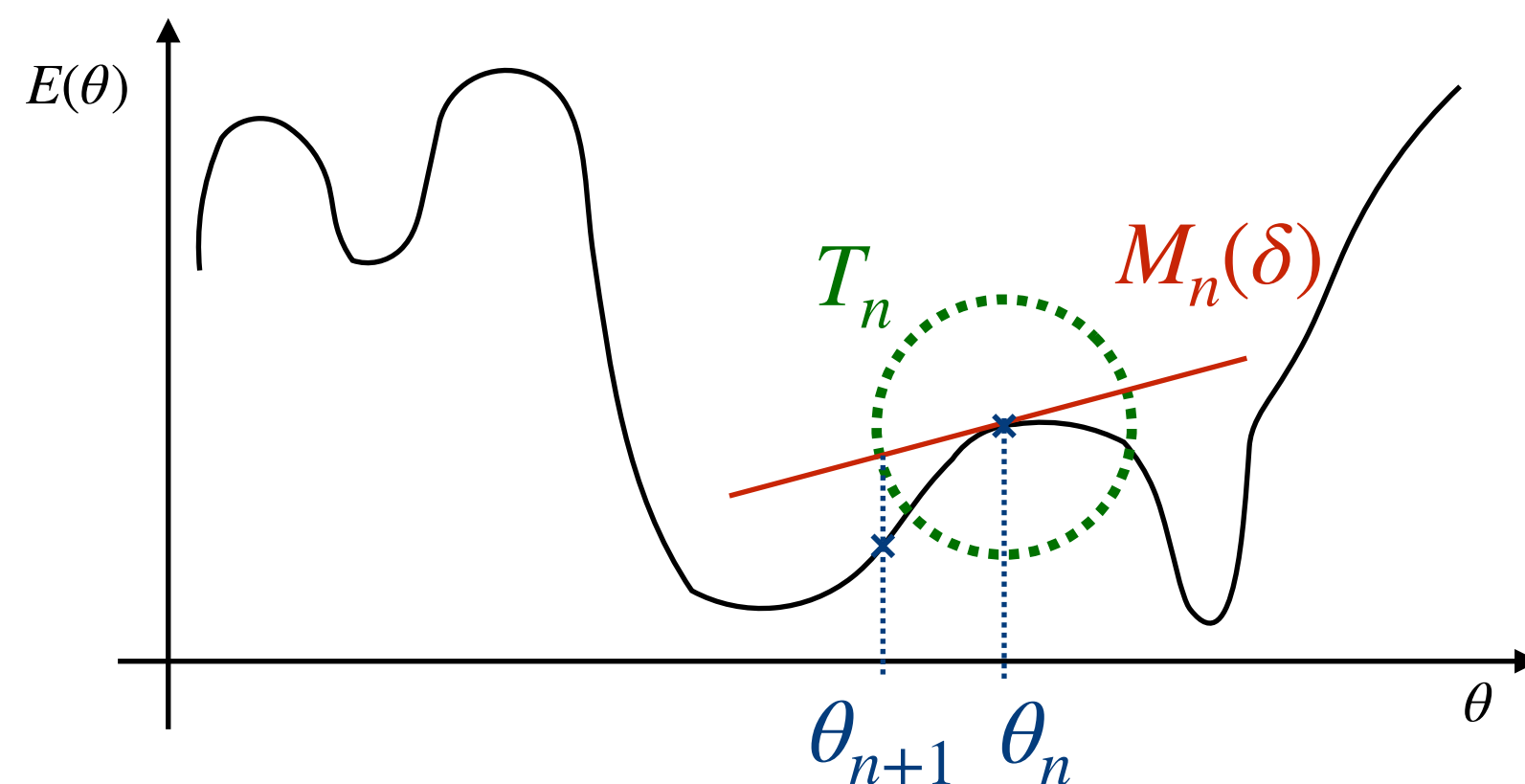


# Global non-linear optimization for NQS

## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## 1D example



## General strategy

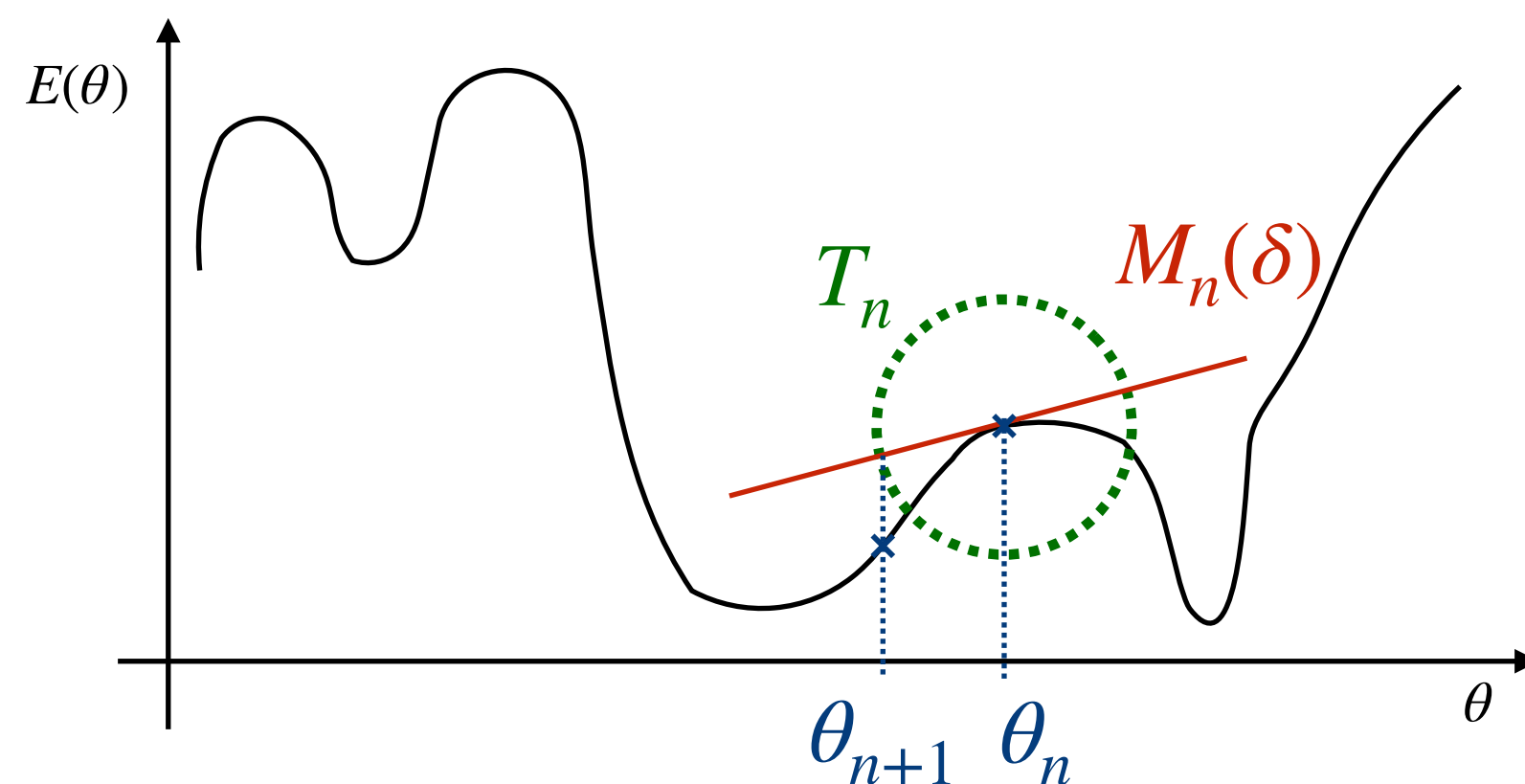
- Complicated problem  $\leftrightarrow$   
Many trivial problems  $\leftrightarrow$
- Sequence of linear/quadratic optimizations
- Iterative algorithm
  - $E_{n+1} = E(\theta_{n+1})$   
 $\theta_{n+1} = \theta_n + \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$
  - where,  
 $M_n(\delta)$  = local model  
 $T_n$  = region where  $M_n(\delta)$  is trusted

# Global non-linear optimization for NQS

## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## 1D example



## General strategy

- Complicated problem  $\rightarrow$   
Many trivial problems  $\rightarrow$
- Sequence of linear/quadratic optimizations
- Iterative algorithm
  - $E_{n+1} = E(\theta_{n+1})$   
 $\theta_{n+1} = \theta_n + \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$
  - where,  
 $M_n(\delta)$  = local model  
 $T_n$  = region where  $M_n(\delta)$  is trusted
- In practice: regularized quadratic model
  - $M_n(\delta) = \frac{1}{2} \delta^T Q \delta + L^T \delta + C$
  - $M_n(\delta) \leftarrow M_n(\delta) + \frac{1}{2} \delta^T R_n \delta$ , with  $R_n \geq 0$

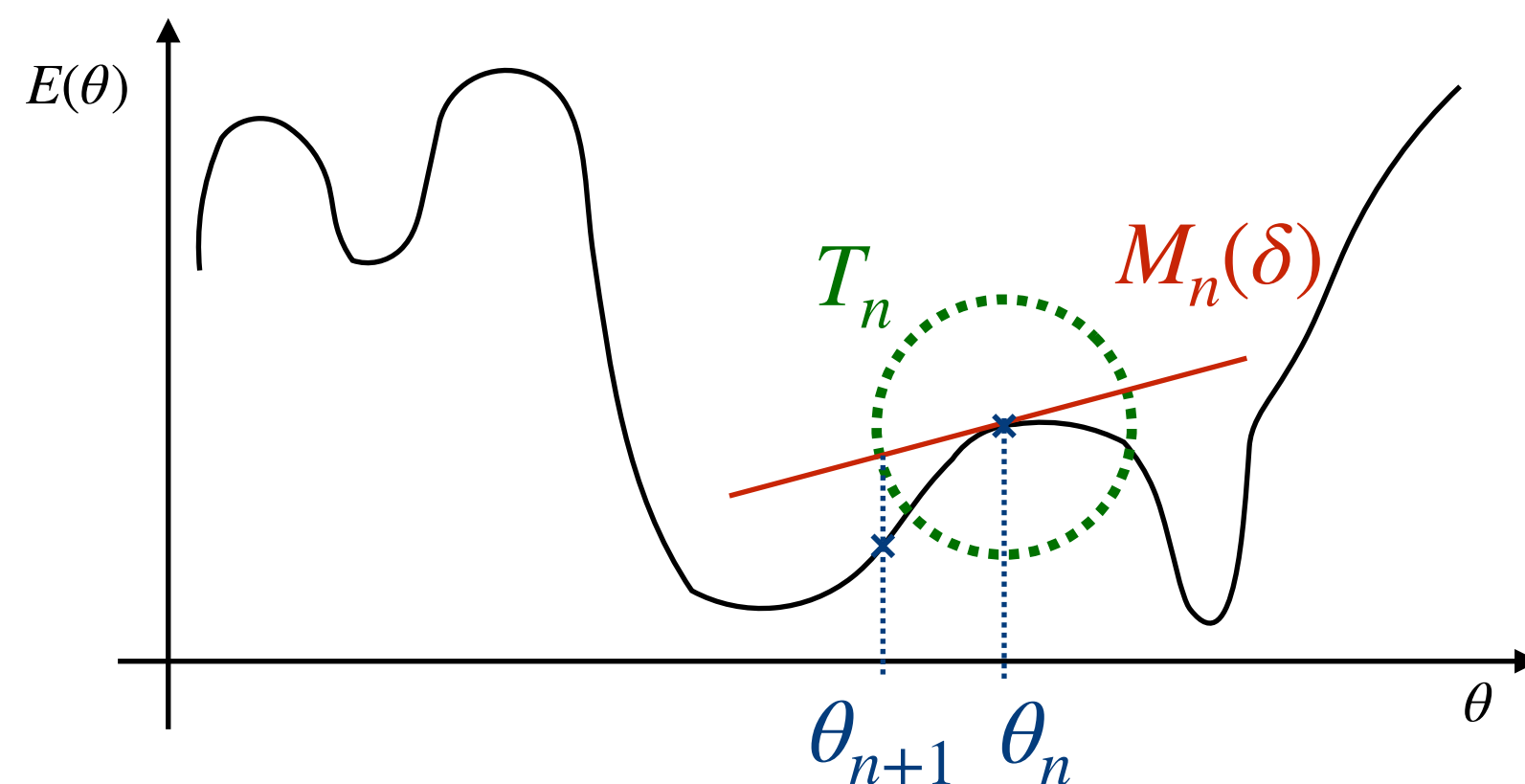


# Global non-linear optimization for NQS

## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## 1D example



## General strategy

- Complicated problem  $\leftrightarrow$   
Many trivial problems  $\leftrightarrow$
- Sequence of linear/quadratic optimizations
- Iterative algorithm
  - $E_{n+1} = E(\theta_{n+1})$   
 $\theta_{n+1} = \theta_n + \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$
  - where,  
 $M_n(\delta)$  = local model  
 $T_n$  = region where  $M_n(\delta)$  is trusted
- In practice: regularized quadratic model
  - $M_n(\delta) = \frac{1}{2} \delta^T Q \delta + L^T \delta + C$
  - $M_n(\delta) \leftarrow M_n(\delta) + \frac{1}{2} \delta^T R_n \delta$ , with  $R_n \geq 0$

## Optimizers discussed here

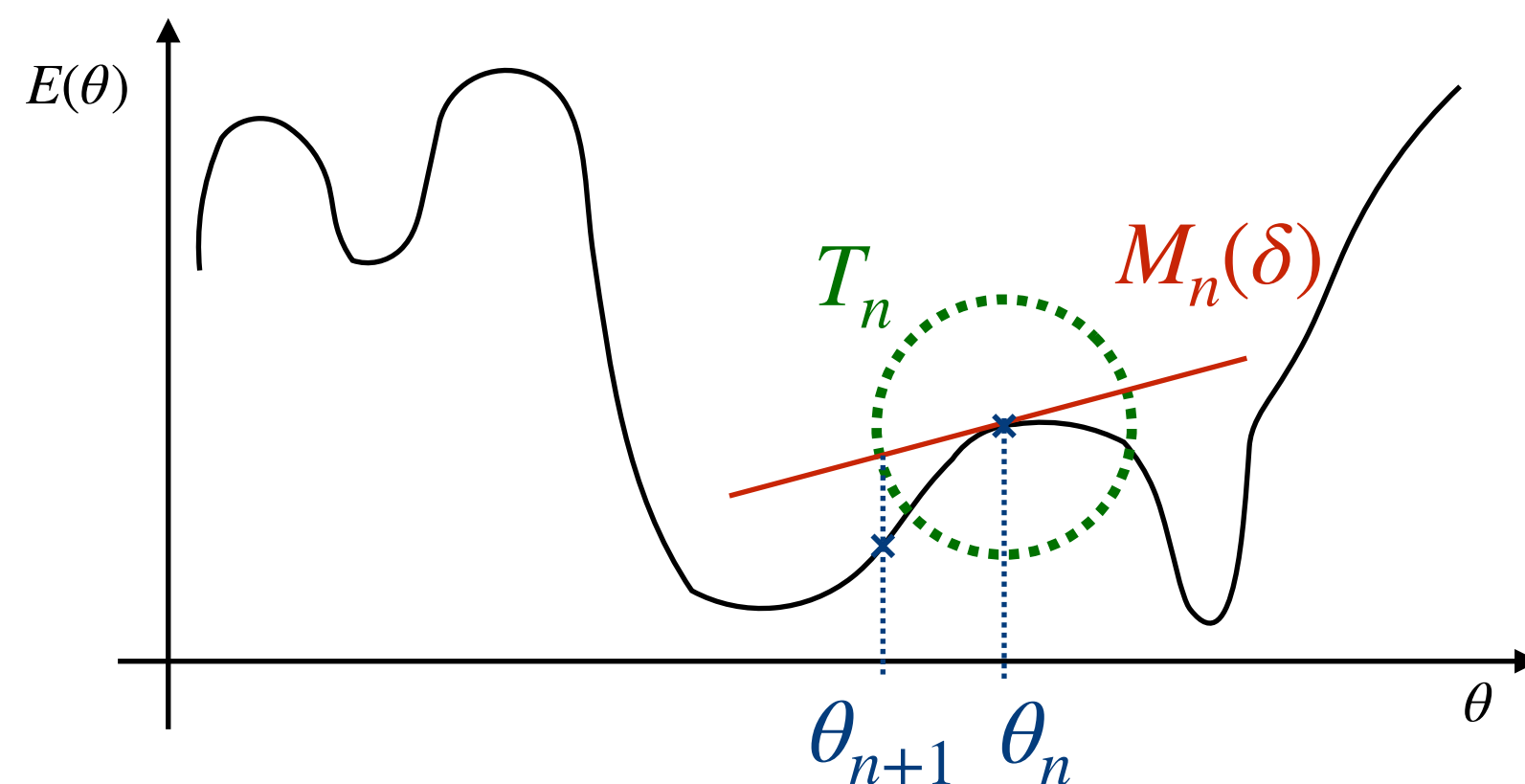
- Gradient descent ( $\sim$  Adam)
  - $M_n(\delta) \equiv \nabla E(\theta_n)^T \delta + E(\theta_n)$
  - $T_n \equiv \{\delta : \|\delta\|_2 \leq \alpha \|\nabla E(\theta_n)\|_2\}$
  - $\alpha \equiv$  learning rate
  - $\rightarrow \delta_n = -\alpha \nabla E(\theta_n)$

# Global non-linear optimization for NQS

## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## 1D example



## General strategy

- Complicated problem  $\leftrightarrow$   
Many trivial problems  $\leftrightarrow$
- Sequence of linear/quadratic optimizations
- Iterative algorithm
  - $E_{n+1} = E(\theta_{n+1})$
  - $\theta_{n+1} = \theta_n + \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$
  - where,
    - $M_n(\delta)$  = local model
    - $T_n$  = region where  $M_n(\delta)$  is trusted
- In practice: regularized quadratic model
  - $M_n(\delta) = \frac{1}{2} \delta^T Q \delta + L^T \delta + C$
  - $M_n(\delta) \leftarrow M_n(\delta) + \frac{1}{2} \delta^T R_n \delta$ , with  $R_n \geq 0$

## Optimizers discussed here

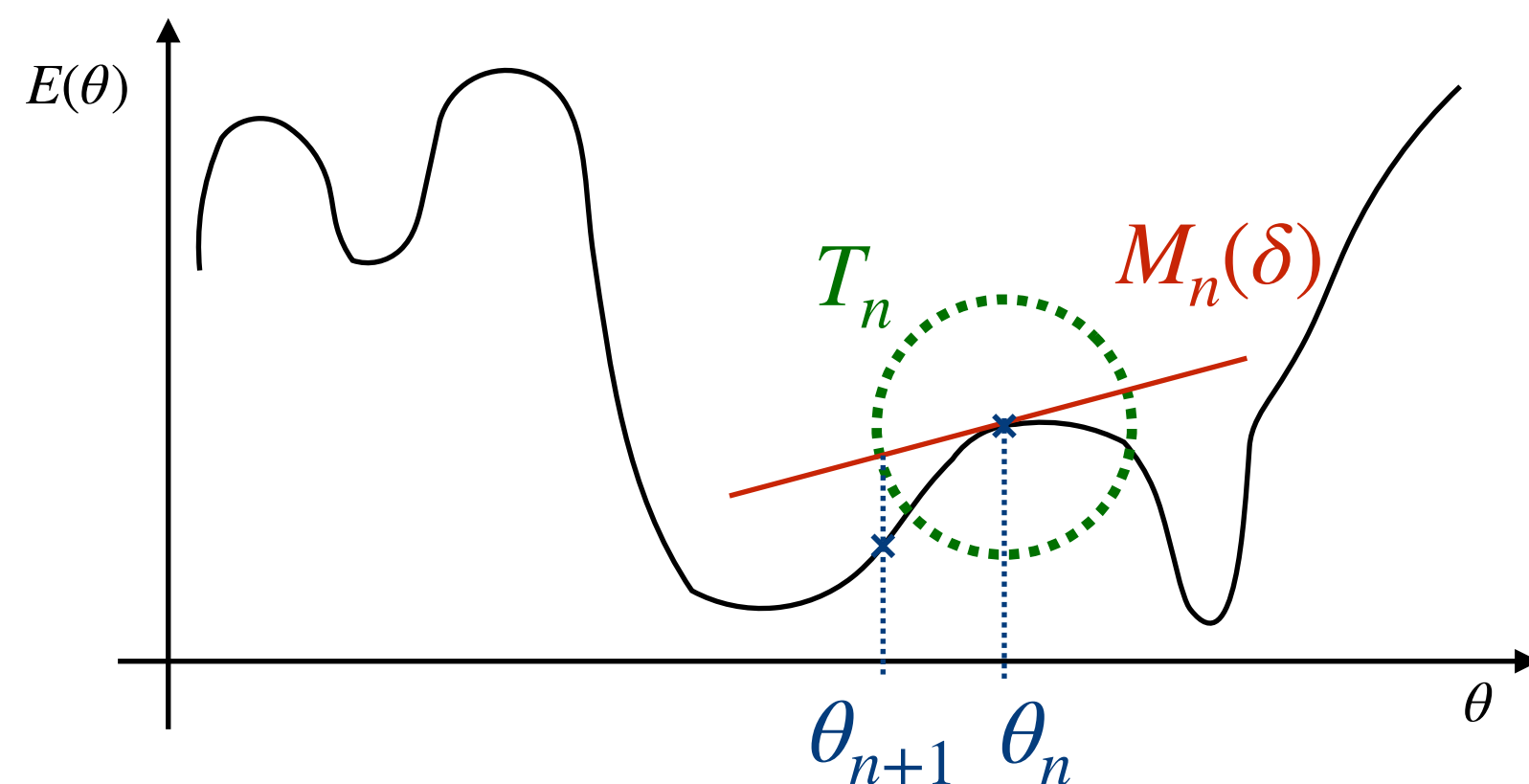
- Gradient descent ( $\sim$  Adam)
  - $M_n(\delta) \equiv \nabla E(\theta_n)^T \delta + E(\theta_n)$
  - $T_n \equiv \{\delta : \|\delta\|_2 \leq \alpha \|\nabla E(\theta_n)\|_2\}$
  - $\alpha \equiv$  learning rate
  - $\rightarrow \delta_n = -\alpha \nabla E(\theta_n)$
- Natural gradient descent
  - Fisher information metric
    - $F_{ij}(p) \equiv \mathbb{E}_{X \sim p} \left[ \partial_{\theta_i} \ln p(X) \partial_{\theta_j} \ln p(X) \right]$
  - $T_n(r) = \{\delta : \delta^T F(\theta_n) \delta \leq r^2\}$
  - $\Leftrightarrow M_n(\delta) = \frac{1}{2} \delta^T F \delta + \nabla E(\theta_n)^T \delta + E(\theta_n)$
  - $\rightarrow \delta_n = -F^{-1}(\theta_n) \nabla E(\theta_n)$

# Global non-linear optimization for NQS

## Definition of the problem

- Let  $E(\theta)$  be our cost function
- Goal
  - $E^* = \min_{\theta \in \mathbb{R}^D} E(\theta)$
  - $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^D} E(\theta)$
- Problem
  - $D > 10\,000$
  - $E(\theta)$  highly non-linear

## 1D example



## General strategy

- Complicated problem  $\leftrightarrow$   
Many trivial problems  $\leftrightarrow$
- Sequence of linear/quadratic optimizations
- Iterative algorithm
  - $E_{n+1} = E(\theta_{n+1})$
  - $\theta_{n+1} = \theta_n + \operatorname{argmin}_{\delta \in T_n} M_n(\delta)$
  - where,
    - $M_n(\delta) = \text{local model}$
    - $T_n = \text{region where } M_n(\delta) \text{ is trusted}$
- In practice: regularized quadratic model
  - $M_n(\delta) = \frac{1}{2} \delta^T Q \delta + L^T \delta + C$
  - $M_n(\delta) \leftarrow M_n(\delta) + \frac{1}{2} \delta^T R_n \delta$ , with  $R_n \geq 0$

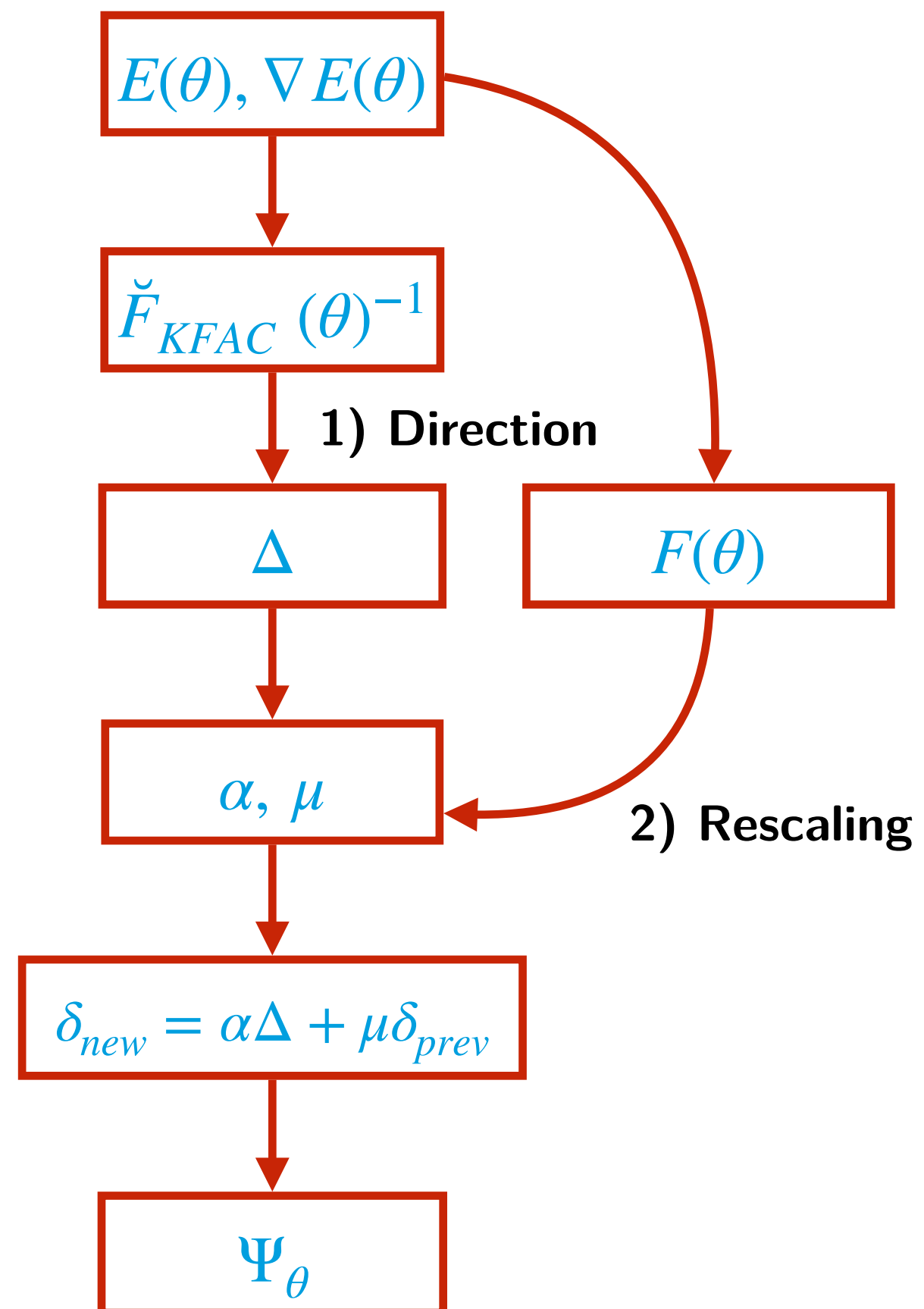
## Optimizers discussed here

- Gradient descent ( $\sim$  Adam)
  - $M_n(\delta) \equiv \nabla E(\theta_n)^T \delta + E(\theta_n)$
  - $T_n \equiv \{ \delta : \|\delta\|_2 \leq \alpha \|\nabla E(\theta_n)\|_2 \}$
  - $\alpha \equiv \text{learning rate}$
  - $\rightarrow \delta_n = -\alpha \nabla E(\theta_n)$
- Natural gradient descent
  - Fisher information metric
 
$$F_{ij}(p) \equiv \mathbb{E}_{X \sim p} \left[ \partial_{\theta_i} \ln p(X) \partial_{\theta_j} \ln p(X) \right]$$
  - $T_n(r) = \{ \delta : \delta^T F(\theta_n) \delta \leq r^2 \}$
  - $\Leftrightarrow M_n(\delta) = \frac{1}{2} \delta^T F \delta + \nabla E(\theta_n)^T \delta + E(\theta_n)$
  - $\rightarrow \delta_n = -F^{-1}(\theta_n) \nabla E(\theta_n)$
- KFAC (Kronecker-Factored Approximate Curvature)
  - NQS  $\Rightarrow D > 10\,000$
  - $F^{-1}(\theta_n) \nabla E(\theta_n) \Rightarrow O(D^2)$
  - KFAC  $\sim$  crude approx of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher

# Direct application of KFAC

## KFAC optimizer

[Martens, Grosse (2015)]

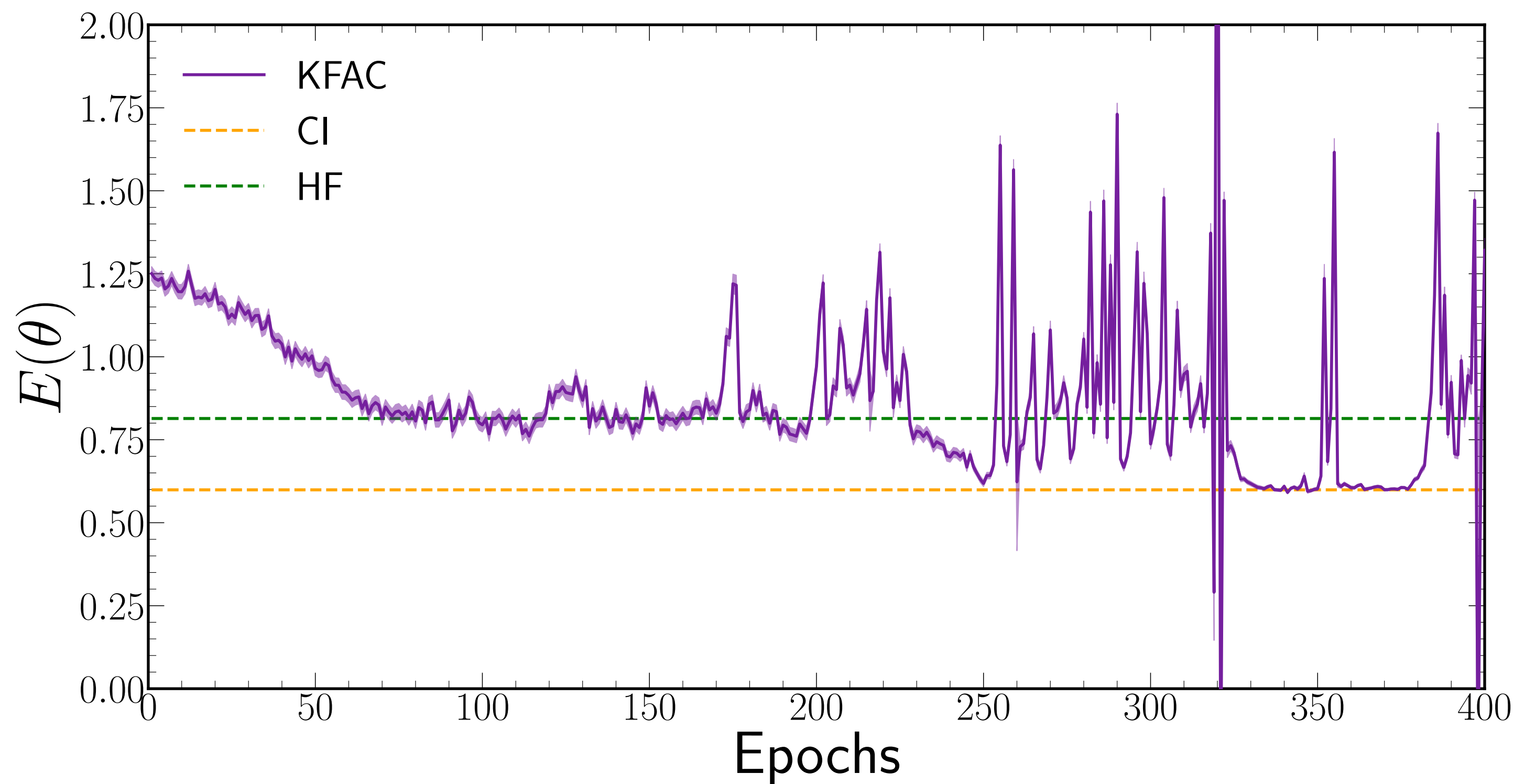
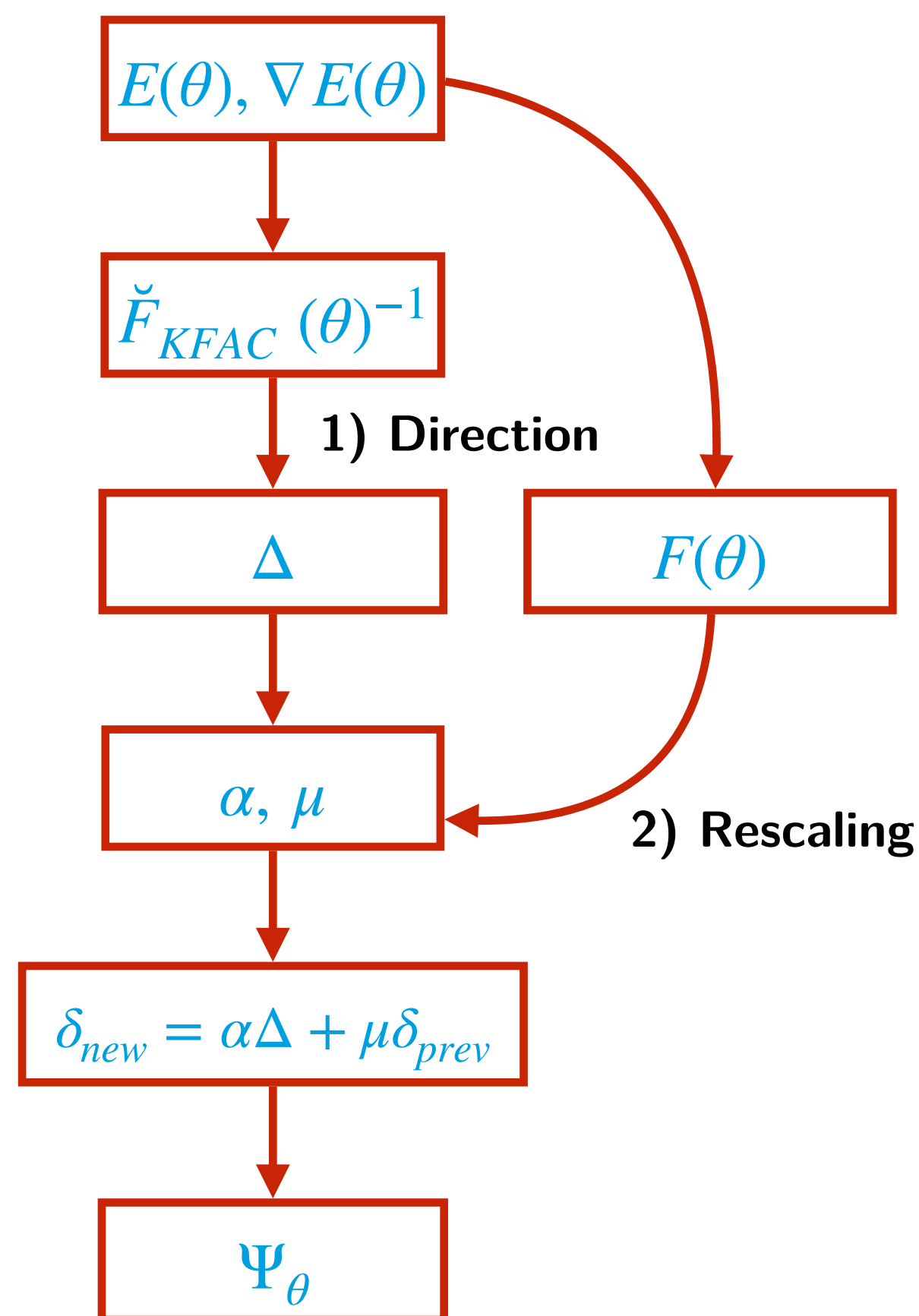


# Direct application of KFAC

KFAC:  $A = 2, V_0 = -10$

## KFAC optimizer

[Martens, Grosse (2015)]

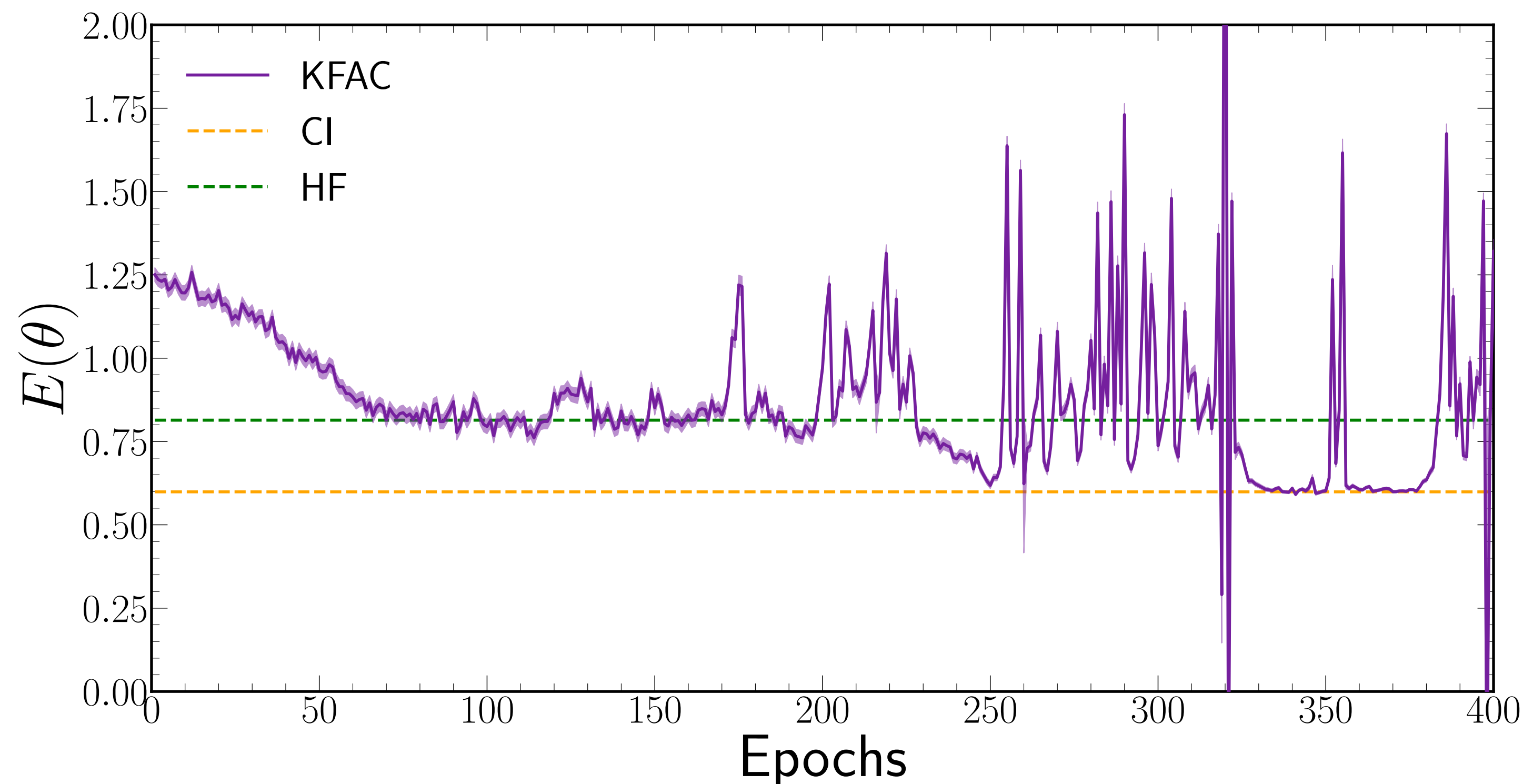
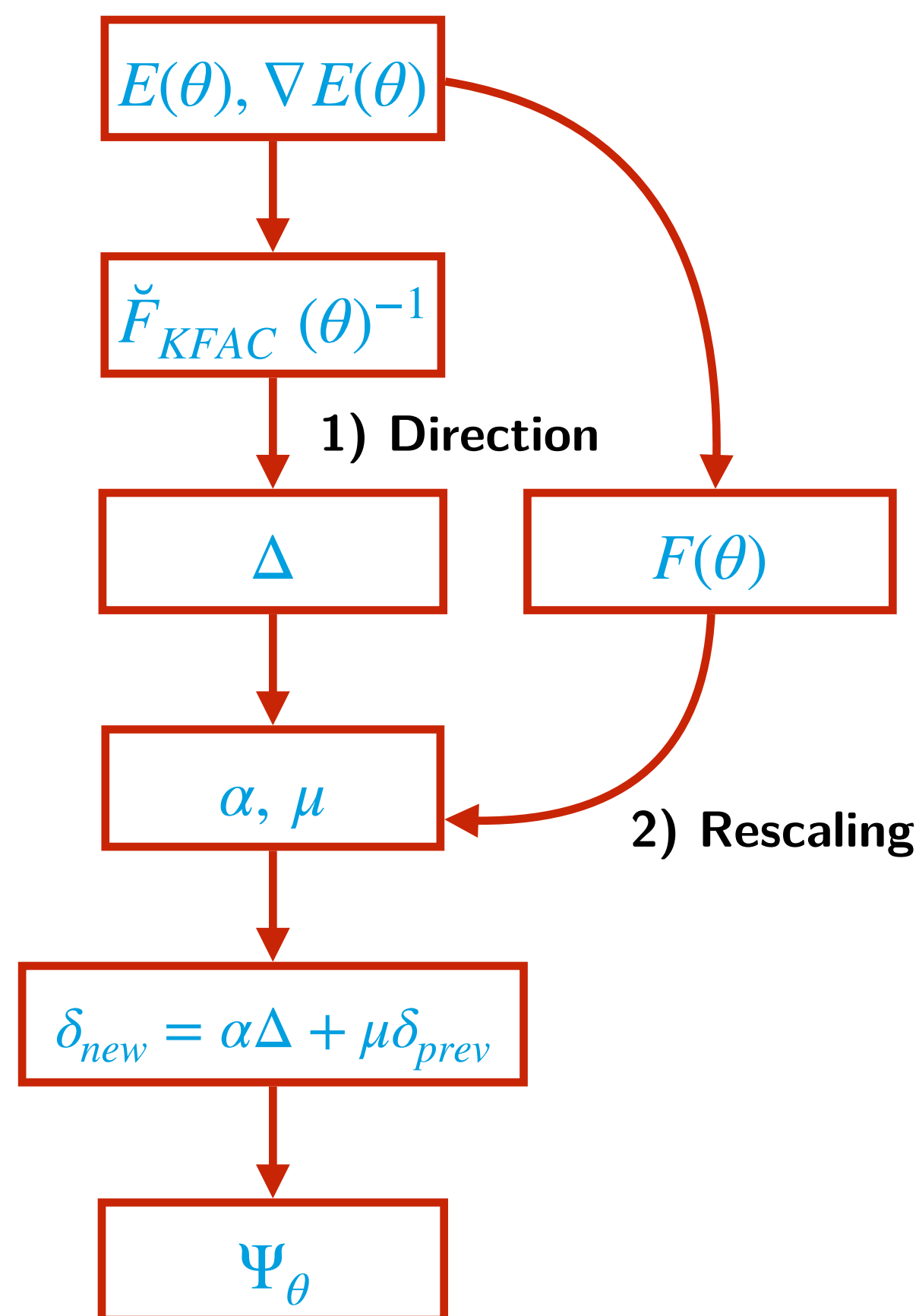


# Direct application of KFAC

KFAC:  $A = 2, V_0 = -10$

## KFAC optimizer

[Martens, Grosse (2015)]



## Extensive testing

- ⦿ Sometimes works nicely, sometimes unstable, sometimes fake convergence
- ➔ Difficult to predict performance
- ➔ **Not reliable optimization ⇒ How to improve it ?**

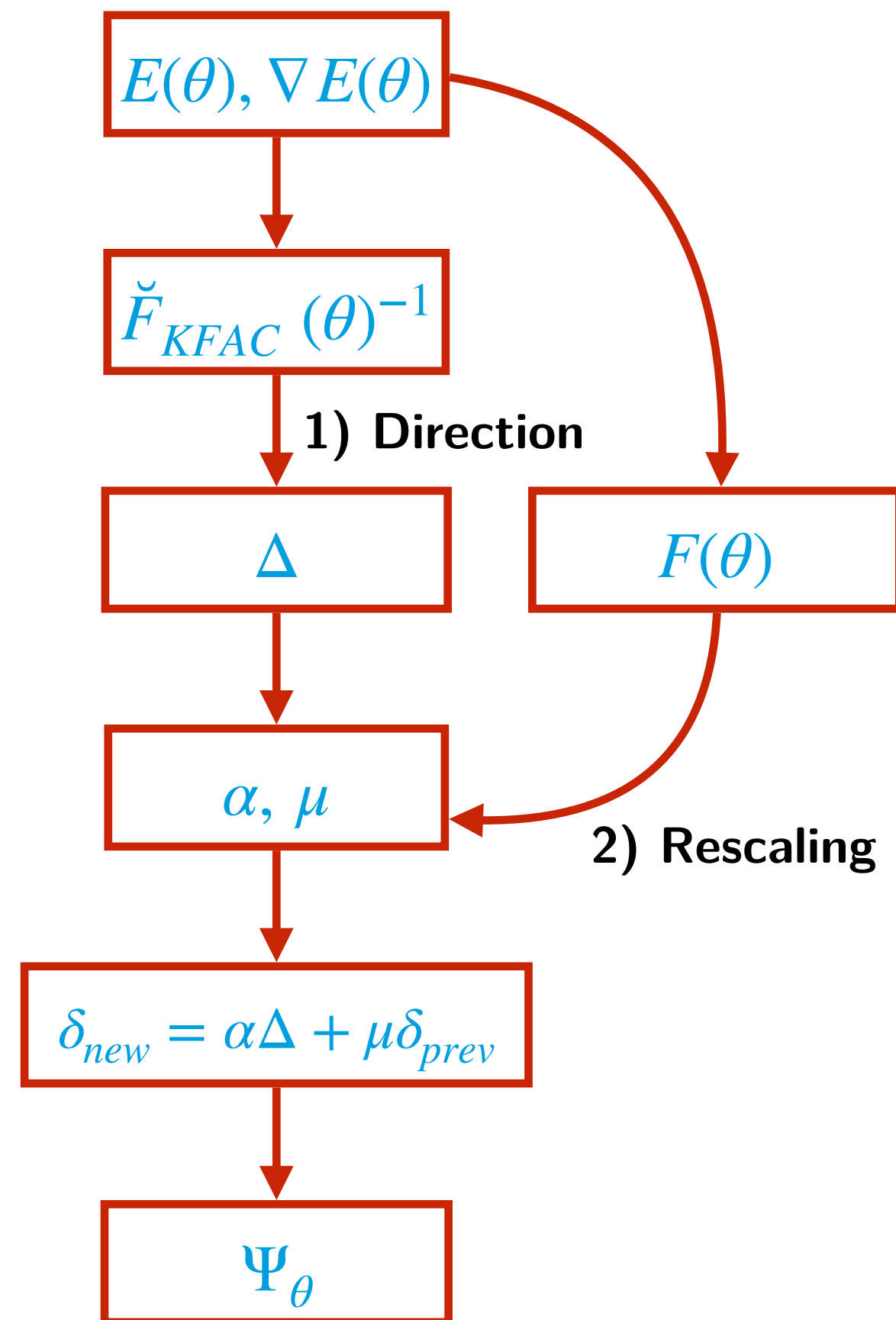
# Outline

- **Variational Monte Carlo with Neural Quantum States**
  - Overview of VMC with NQS
  - The Kronecker-Factored Approximate Curvature (KFAC)
- **Augmented KFAC for VMC problems**
  - Scaling improvement from a Quasi-Newton approach
  - Direction improvement from MINRES
- **Decision geometry for VMC**
  - Game theory reformulation of VMC
  - Testing decisional gradient descent

# Alternative approach to KFAC shortcomings

## Recap: KFAC optimizer

[Martens, Grosse (2015)]

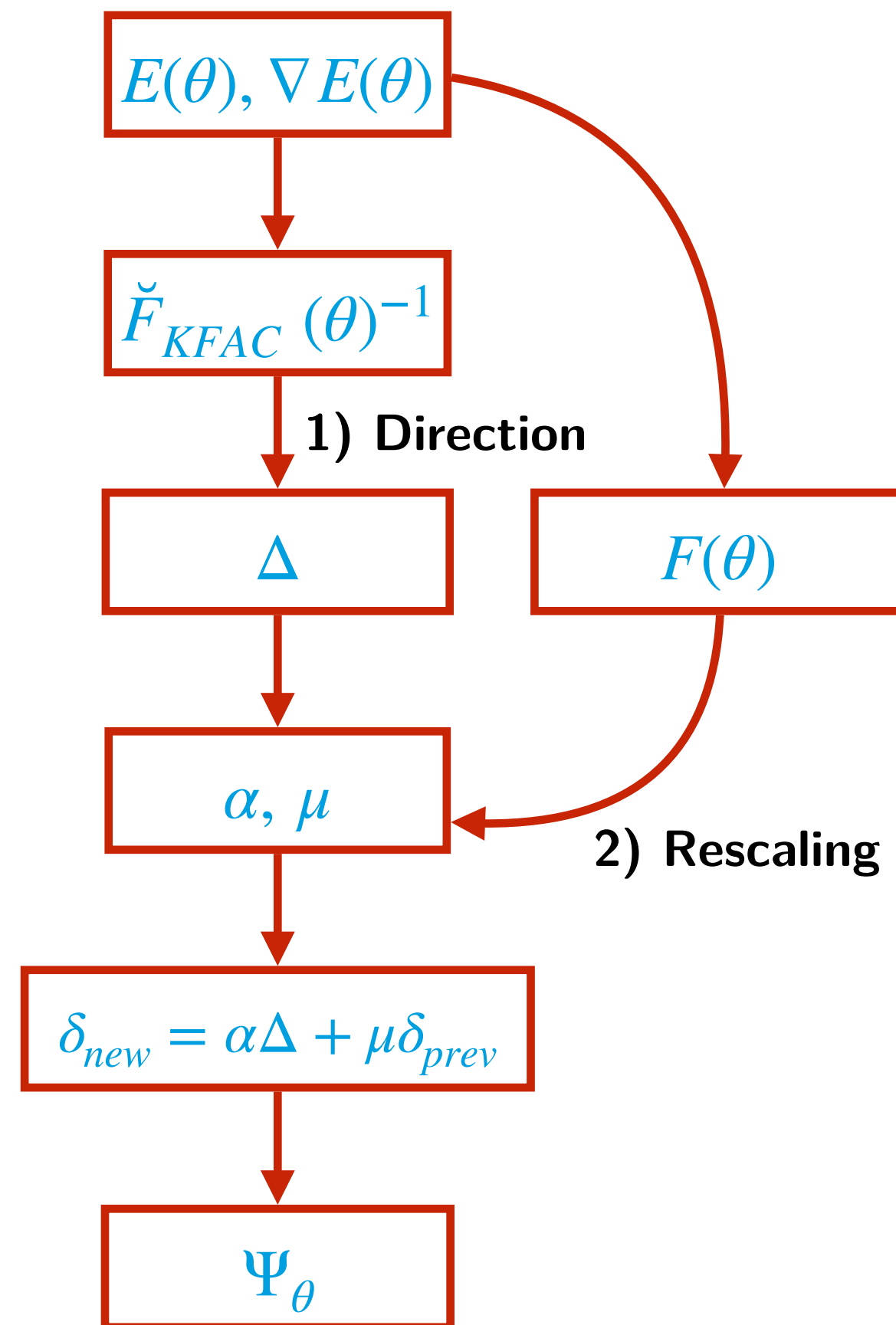




# Alternative approach to KFAC shortcomings

## Recap: KFAC optimizer

[Martens, Grosse (2015)]



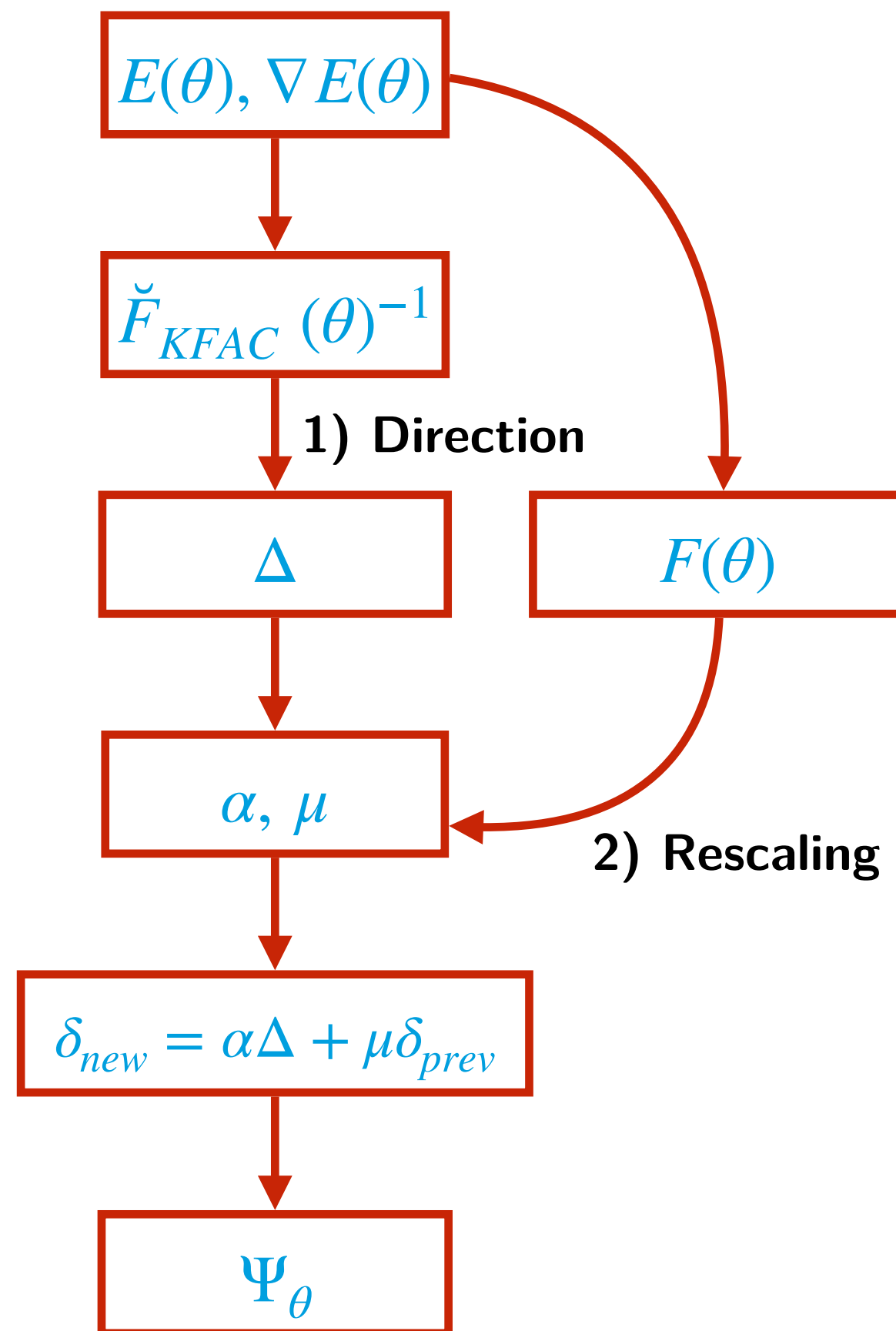
## Improving scaling of the update

- Analysis
  - Original argument for KFAC:  $F \sim$  Hessian
  - Only valid for supervised learning problems
  - **VMC  $\neq$  supervised learning**
- Proposed solution
  - Just use a better quadratic model !

# Alternative approach to KFAC shortcomings

## Recap: KFAC optimizer

[Martens, Grosse (2015)]



### Improving scaling of the update

- Analysis
  - Original argument for KFAC:  $F \sim \text{Hessian}$
  - Only valid for supervised learning problems
  - **VMC  $\neq$  supervised learning**
- Proposed solution
  - Just use a better quadratic model !

### Quasi-Newton KFAC

- Supervised learning: [Martens (2020), Amari (2016)]
  - $F(\theta) \sim \text{Cost function's Hessian} + \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)| = 0$
- In our case: cost function =  $E(\theta)$

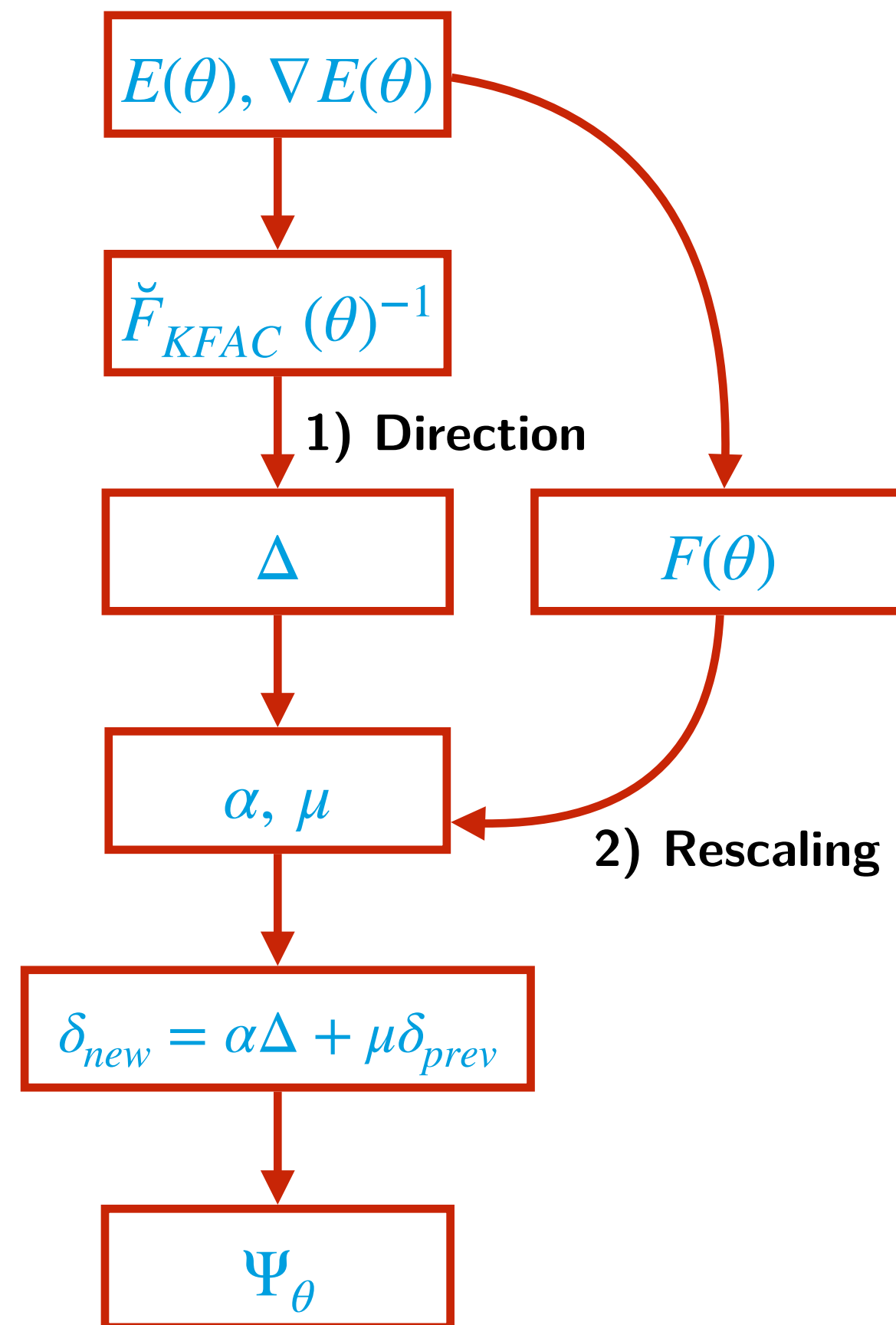
#### ● Hessian:

$$\begin{aligned}
 \partial_{\theta_1} \partial_{\theta_2} E(\theta) = & 2\mathbb{E} [(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)|] \\
 & + 4\mathbb{E} [(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \ln |\Psi_{\theta}(X)| \partial_{\theta_2} \ln |\Psi_{\theta}(X)|] \\
 & + 2\mathbb{E} [\partial_{\theta_1} E_{L,\theta}(X) \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]
 \end{aligned}$$

# Alternative approach to KFAC shortcomings

## Recap: KFAC optimizer

[Martens, Grosse (2015)]



### Improving scaling of the update

- Analysis
  - Original argument for KFAC:  $F \sim \text{Hessian}$
  - Only valid for supervised learning problems
  - **VMC  $\neq$  supervised learning**
- Proposed solution
  - Just use a better quadratic model !

### Quasi-Newton KFAC

- Supervised learning: [Martens (2020), Amari (2016)]
  - $F(\theta) \sim \text{Cost function's Hessian} + \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)| = 0$
- In our case: cost function =  $E(\theta)$

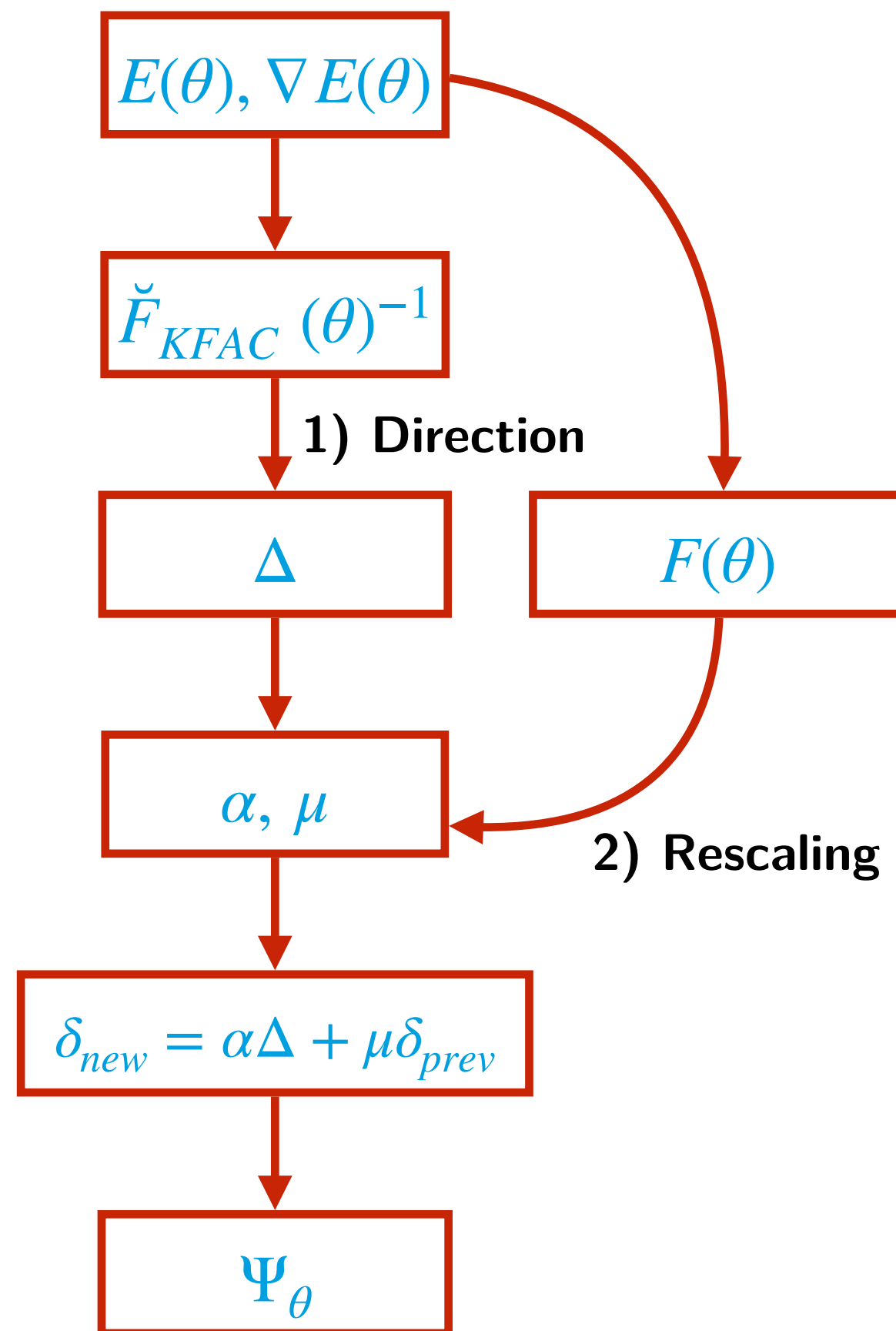
#### ● Hessian:

$$\begin{aligned}
 \partial_{\theta_1} \partial_{\theta_2} E(\theta) = & \cancel{2\mathbb{E} [(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]} \\
 & + 4\mathbb{E} [(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \ln |\Psi_{\theta}(X)| \partial_{\theta_2} \ln |\Psi_{\theta}(X)|] \\
 & + 2\mathbb{E} [\partial_{\theta_1} E_{L,\theta}(X) \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]
 \end{aligned}$$

# Alternative approach to KFAC shortcomings

## Recap: KFAC optimizer

[Martens, Grosse (2015)]



### Improving scaling of the update

- Analysis
  - Original argument for KFAC:  $F \sim \text{Hessian}$
  - Only valid for supervised learning problems
  - **VMC  $\neq$  supervised learning**
- Proposed solution
  - Just use a better quadratic model !

### Quasi-Newton KFAC

- Supervised learning: [Martens (2020), Amari (2016)]
  - $F(\theta) \sim \text{Cost function's Hessian} + \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)| = 0$
- In our case: cost function =  $E(\theta)$

#### • Hessian:

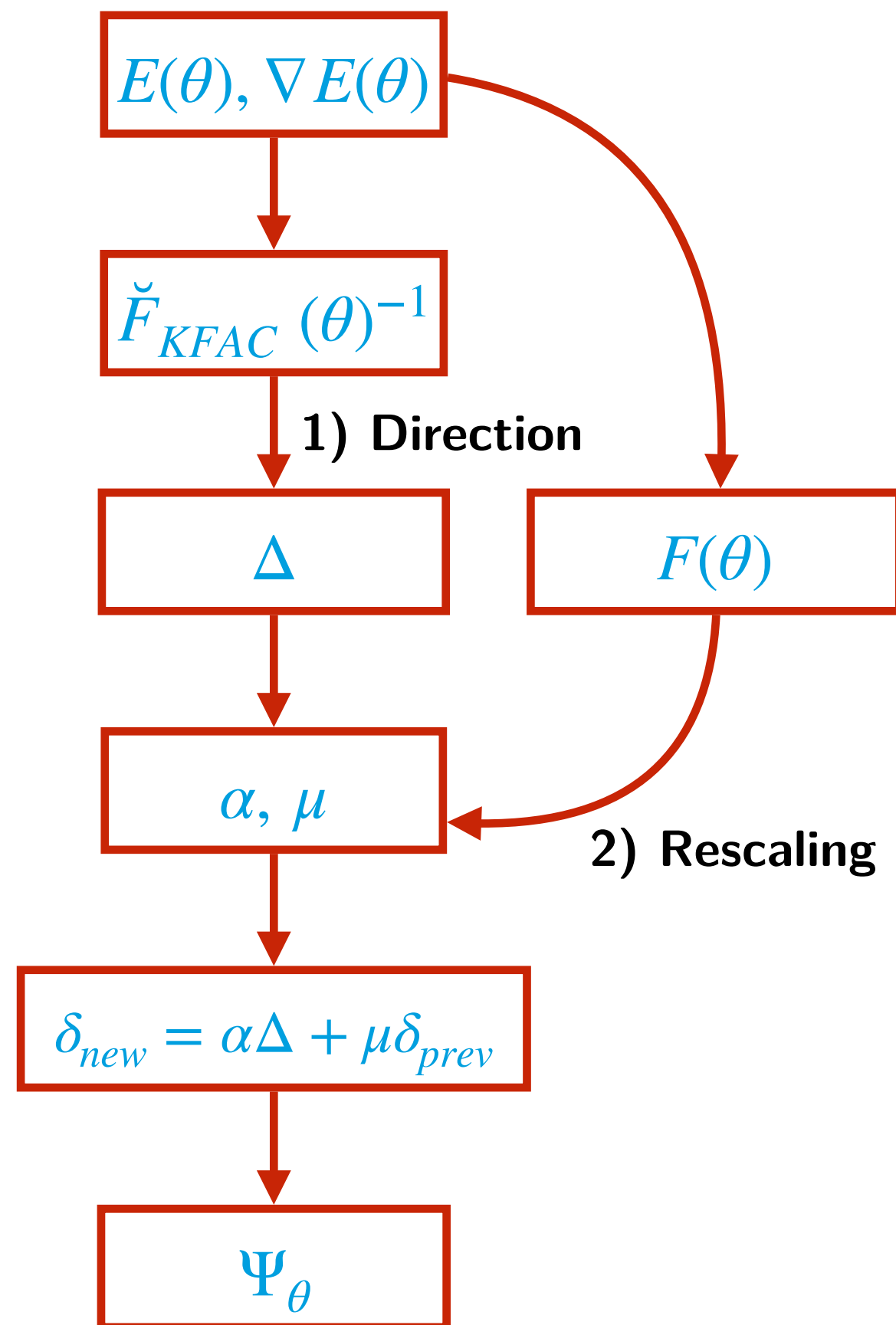
$$\partial_{\theta_1} \partial_{\theta_2} E(\theta) = \cancel{2\mathbb{E}[(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]} + 4\mathbb{E}[(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \ln |\Psi_{\theta}(X)| \partial_{\theta_2} \ln |\Psi_{\theta}(X)|] + 2\mathbb{E}[\partial_{\theta_1} E_{L,\theta}(X) \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]$$

Defines our quasi-Hessian  $H_Q(\theta)$

# Alternative approach to KFAC shortcomings

## Recap: KFAC optimizer

[Martens, Grosse (2015)]



## Improving scaling of the update

- Analysis
  - Original argument for KFAC:  $F \sim \text{Hessian}$
  - Only valid for supervised learning problems
  - **VMC  $\neq$  supervised learning**
- Proposed solution
  - Just use a better quadratic model !

## Quasi-Newton KFAC

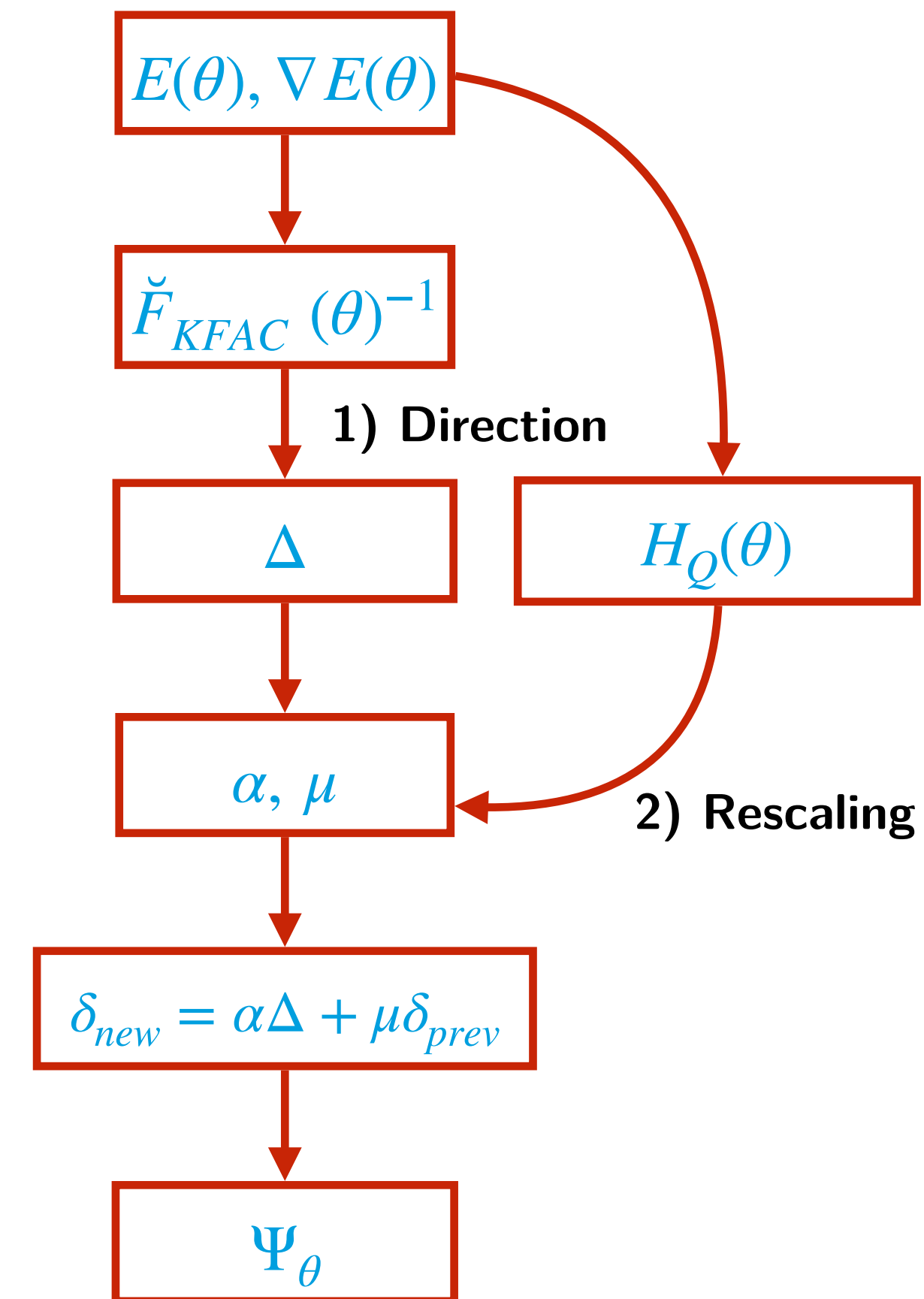
- Supervised learning: [Martens (2020), Amari (2016)]
  - $F(\theta) \sim \text{Cost function's Hessian} + \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)| = 0$
- In our case: cost function =  $E(\theta)$

### Hessian:

$$\partial_{\theta_1} \partial_{\theta_2} E(\theta) = \cancel{2\mathbb{E}[(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]} + 4\mathbb{E}[(E_{L,\theta} - E(\theta)) \partial_{\theta_1} \ln |\Psi_{\theta}(X)| \partial_{\theta_2} \ln |\Psi_{\theta}(X)|] + 2\mathbb{E}[\partial_{\theta_1} E_{L,\theta}(X) \partial_{\theta_2} \ln |\Psi_{\theta}(X)|]$$

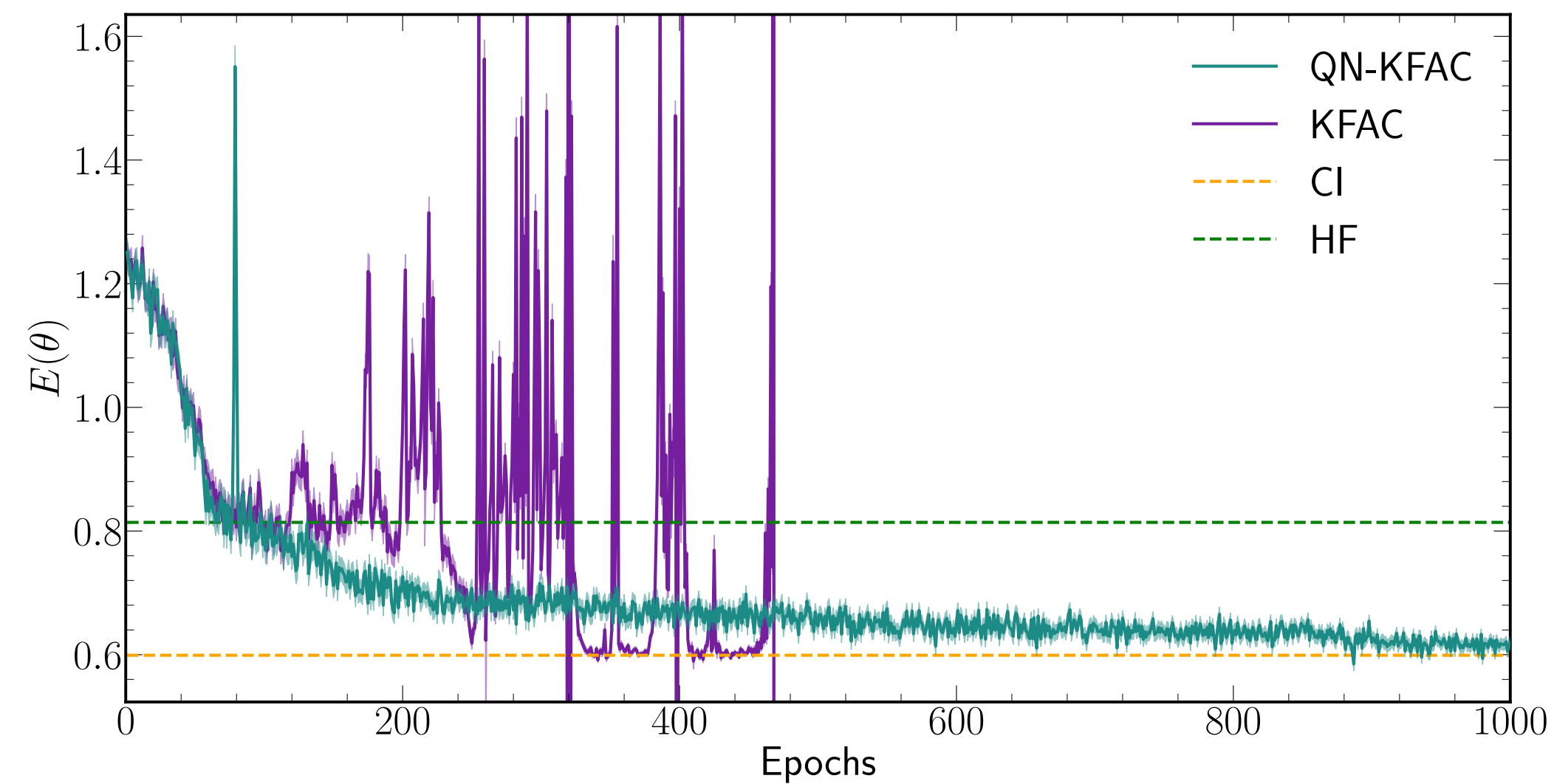
Defines our quasi-Hessian  $H_Q(\theta)$

## QN-KFAC optimizer

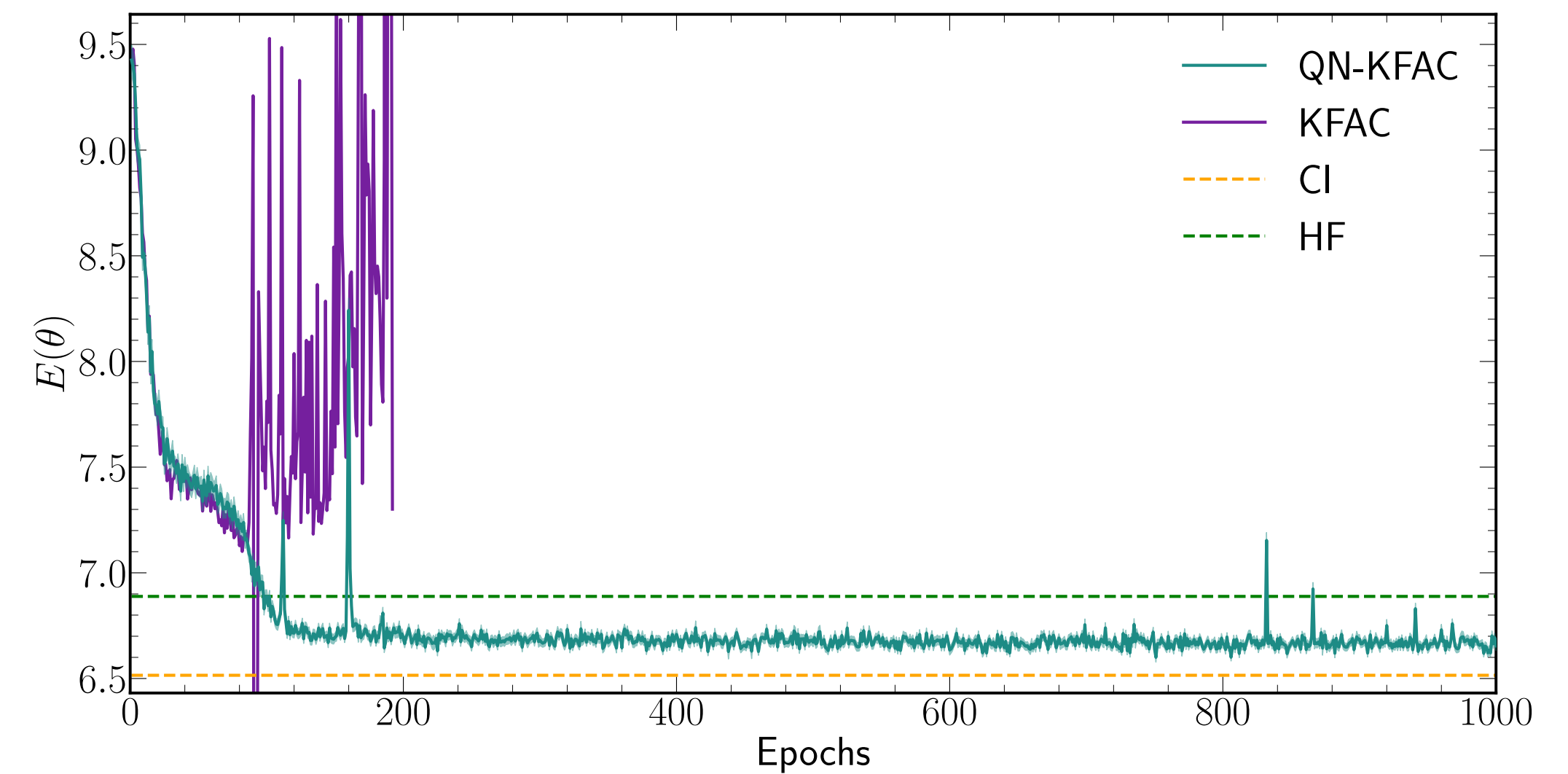


# Impact of new re-scaling on convergence

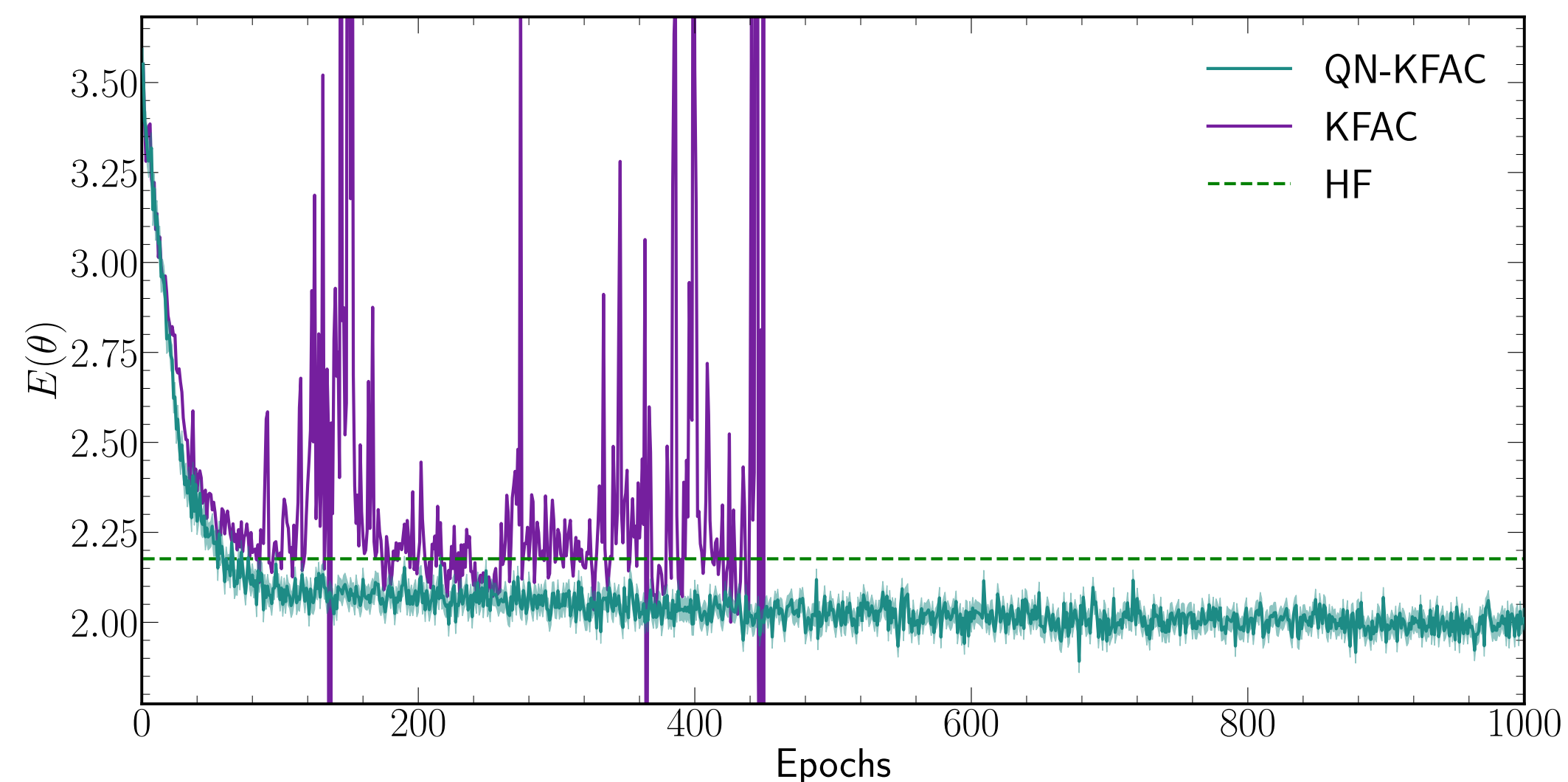
KFAC vs QN-KFAC:  $A = 2, V_0 = -10$



KFAC vs QN-KFAC:  $A = 3, V_0 = 20$



KFAC vs QN-KFAC:  $A = 5, V_0 = -10$



## QN-KFAC vs KFAC

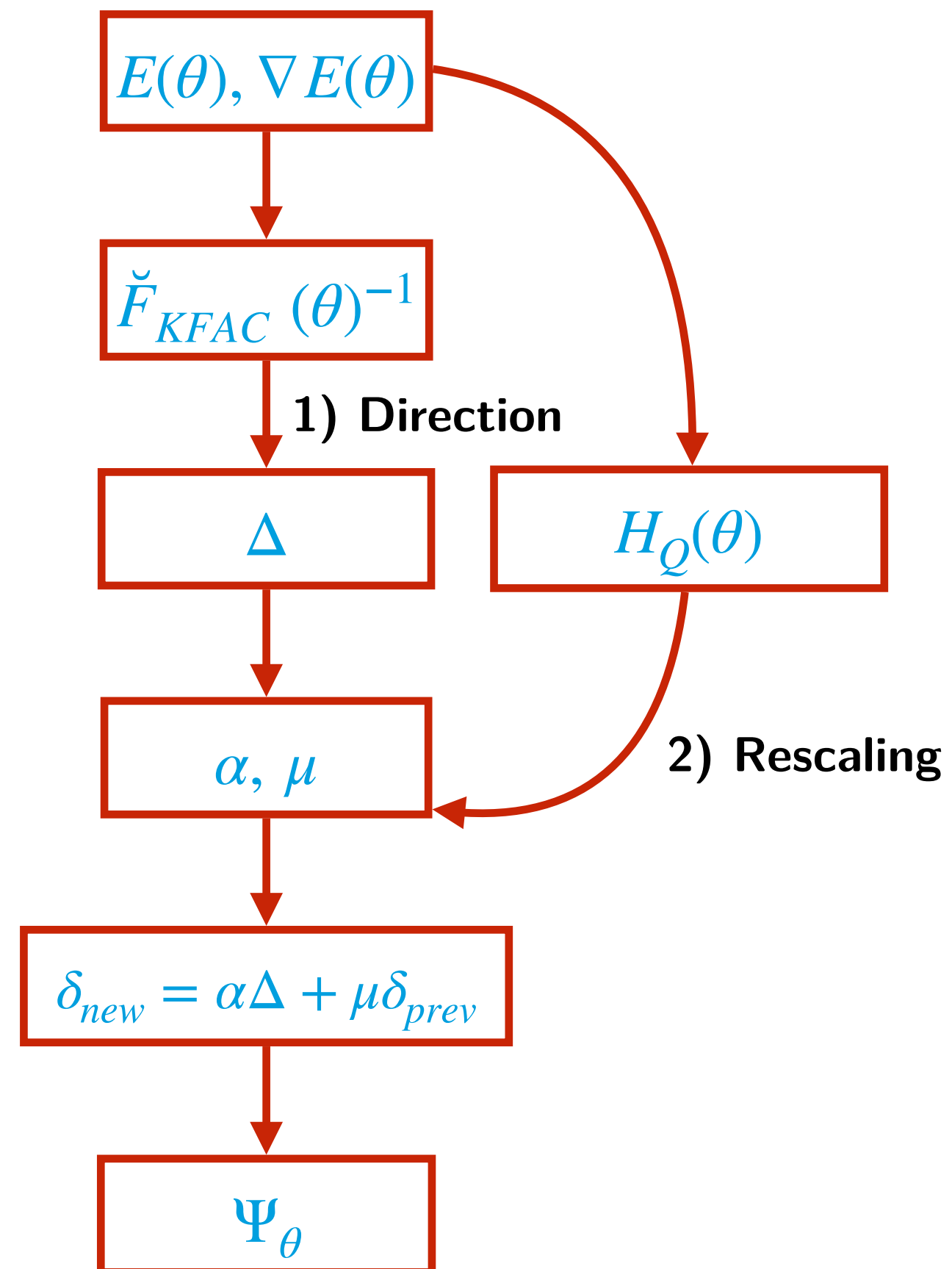
- Overall Improvements
  - Energy fluctuations much reduced
  - Reduction of cases where it get stuck in local minima
- But not perfect
  - Still some instabilities (not shown here because large  $\lambda_{init}$ )
  - Can take time to get out of local minima
  - Slow final convergence

# Outline

- **Variational Monte Carlo with Neural Quantum States**
  - Overview of VMC with NQS
  - The Kronecker-Factored Approximate Curvature (KFAC)
- **Augmented KFAC for VMC problems**
  - Scaling improvement from a Quasi-Newton approach
  - Direction improvement from MINRES
- **Decision geometry for VMC**
  - Game theory reformulation of VMC
  - Testing decisional gradient descent

# Testing direction improvement with MINRES

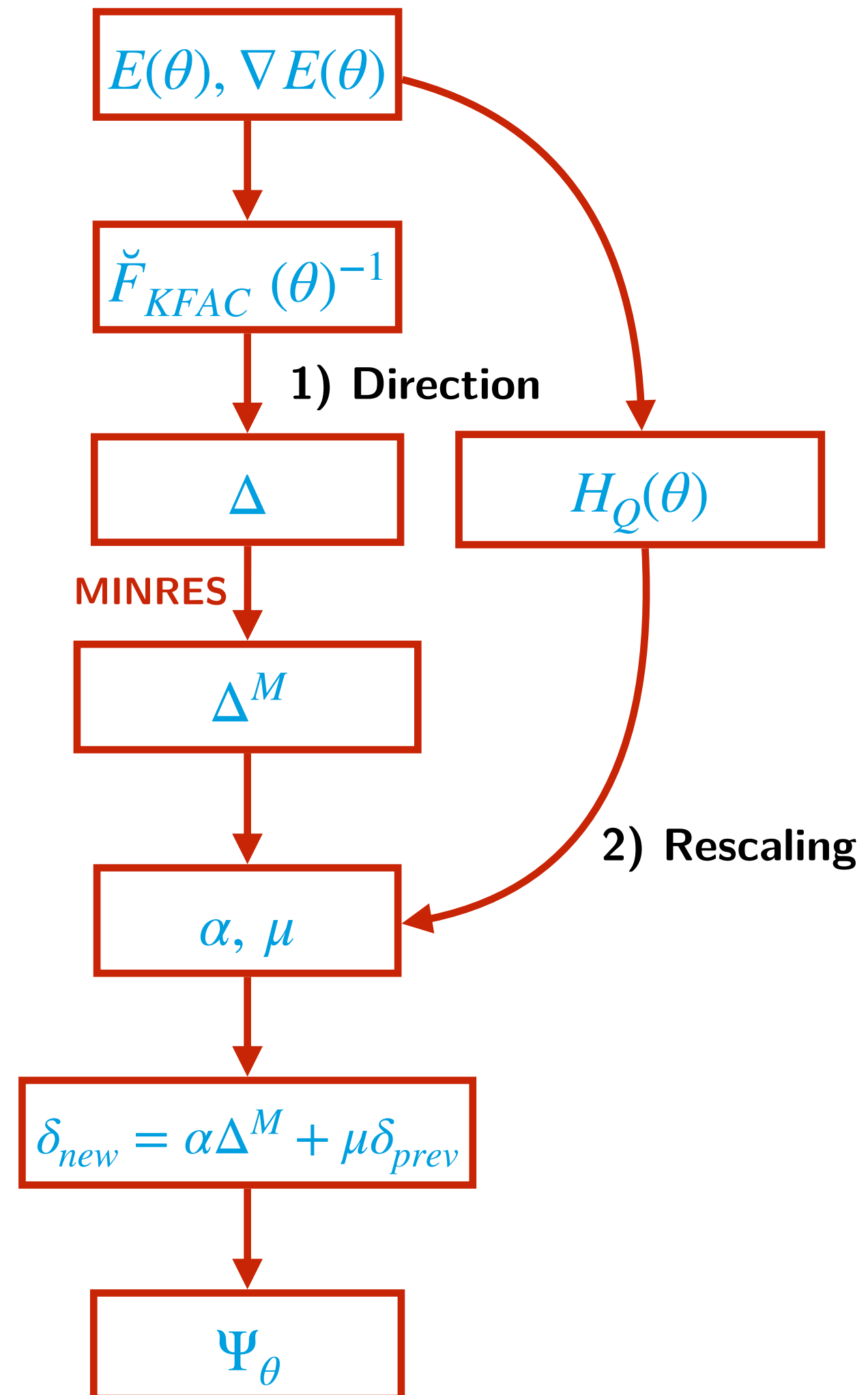
QN-KFAC optimizer





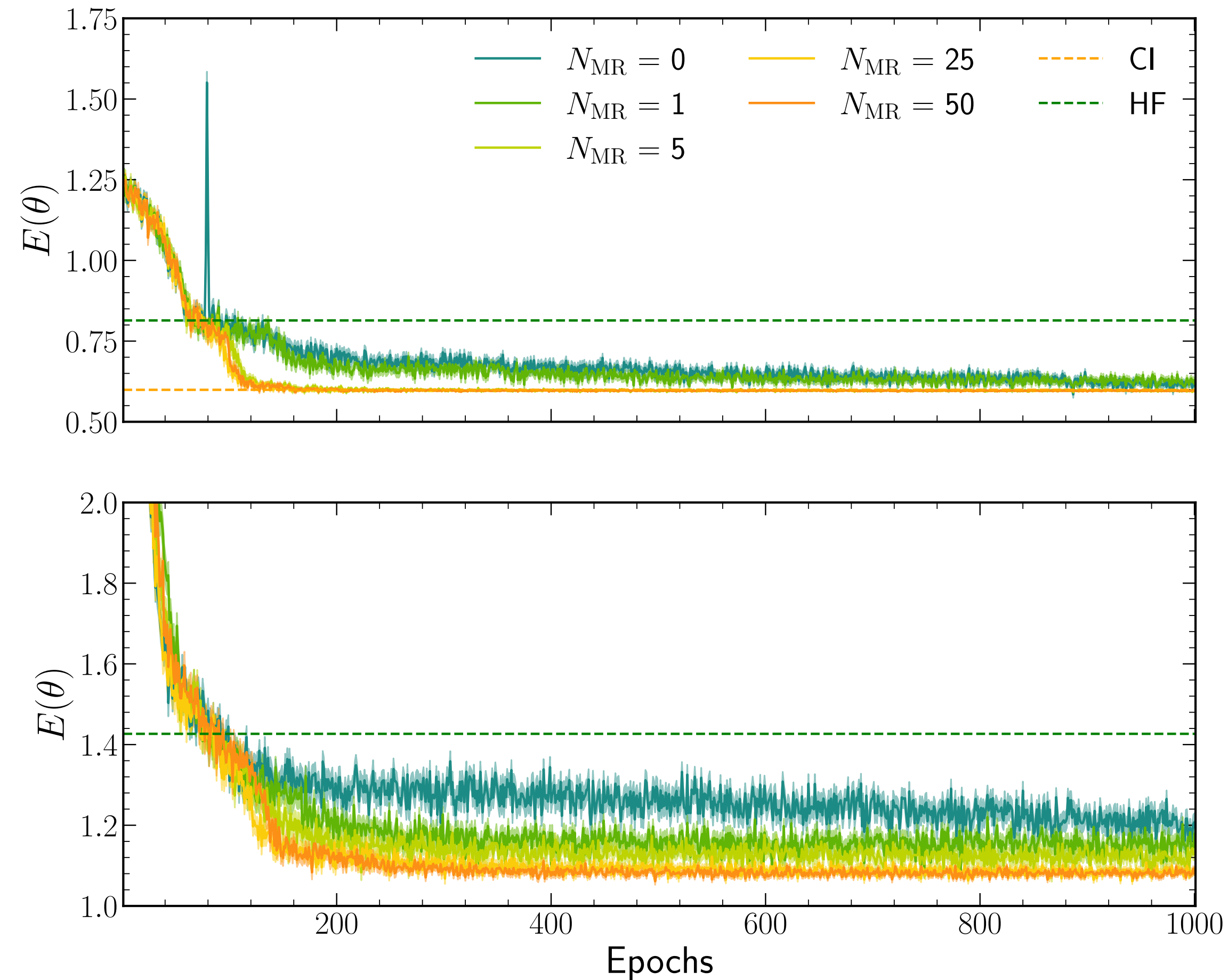
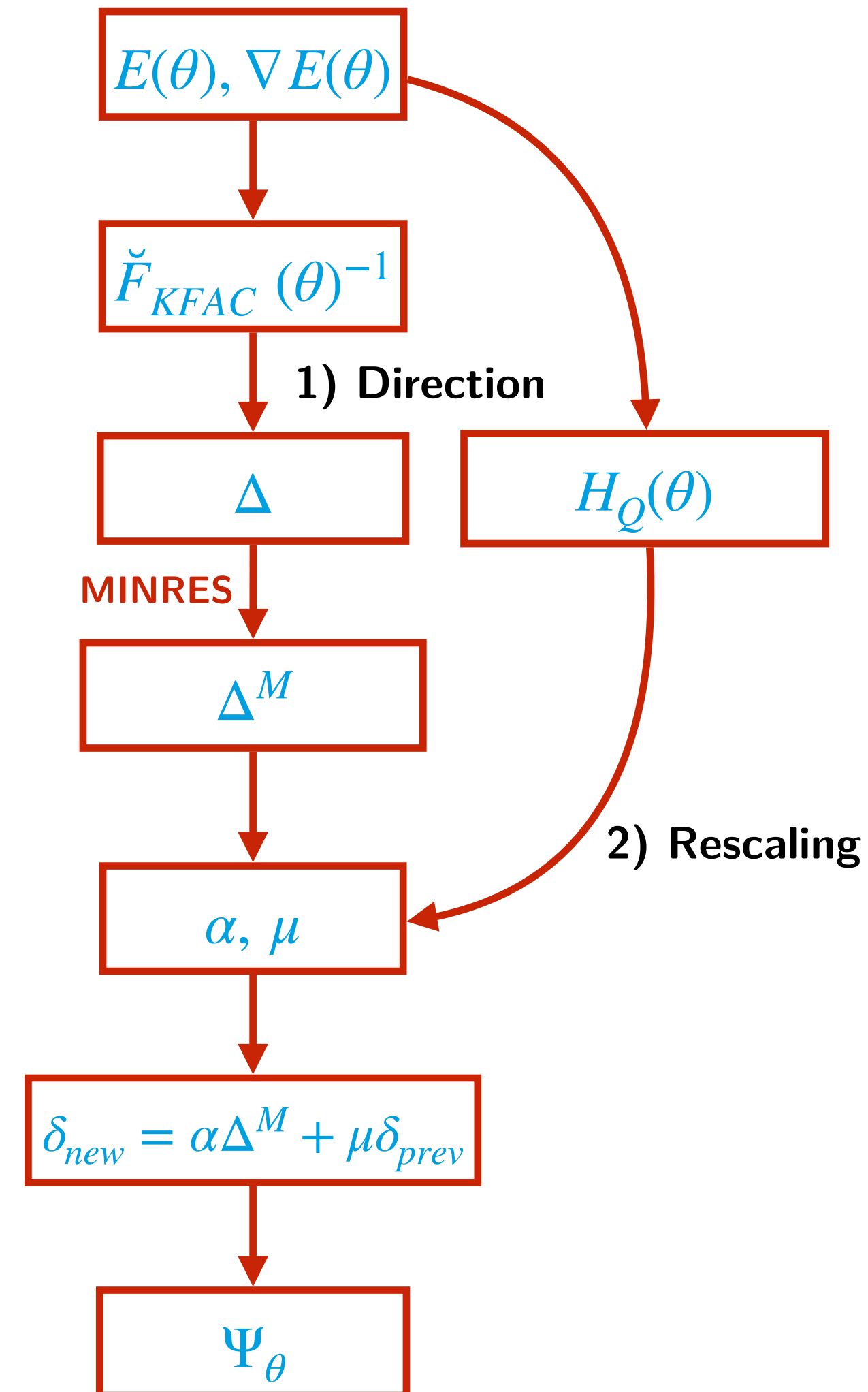
# Testing direction improvement with MINRES

QN-MR-KFAC optimizer



# Testing direction improvement with MINRES

## QN-MR-KFAC optimizer

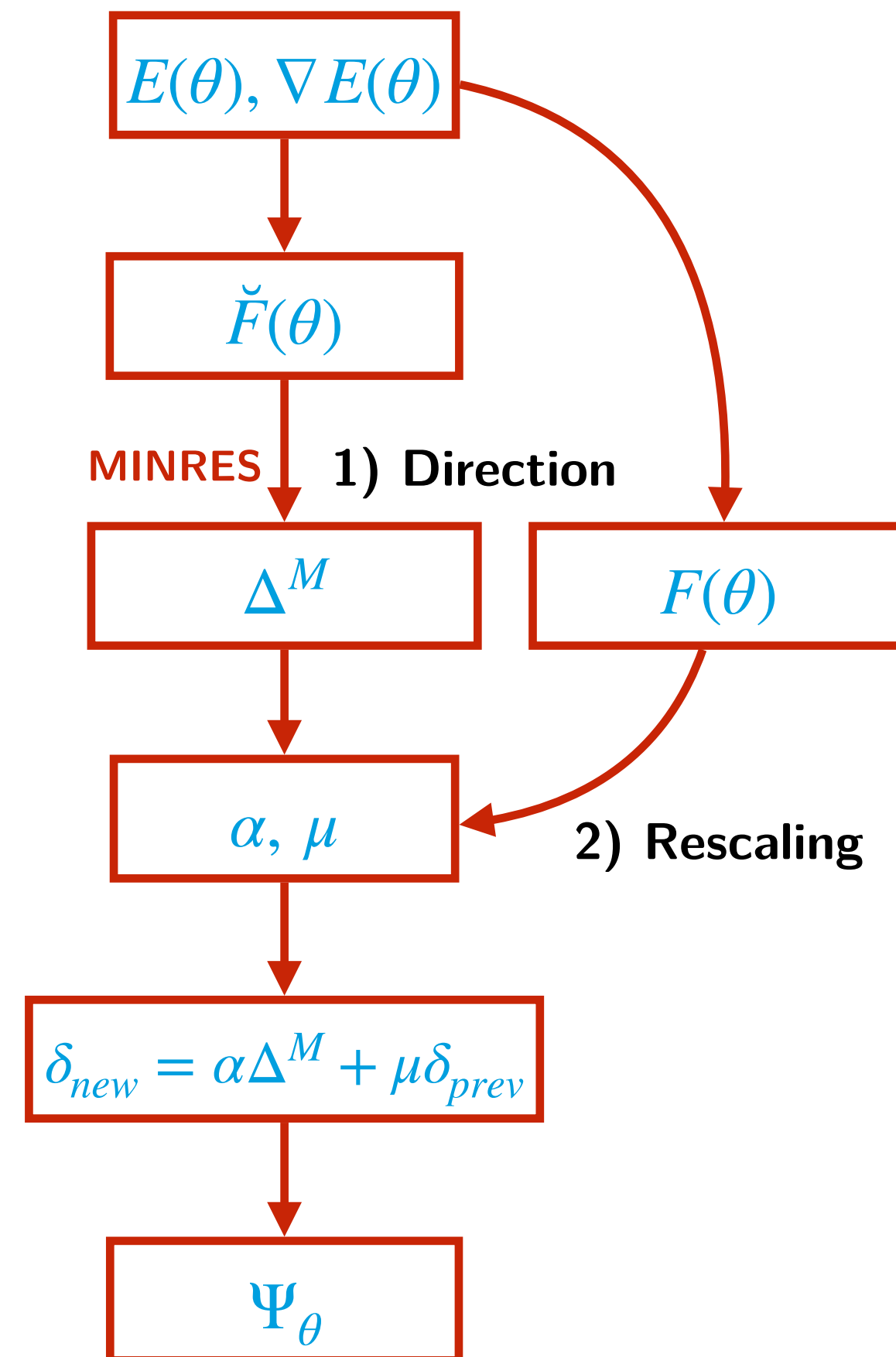


### Improving KFAC estimation of direction update

- Test: take  $\Delta$  as initial guess for MINRES on  $\check{F}(\theta) \cdot x = \nabla E(\theta) \Rightarrow$  **Improved direction  $\Delta^M$**
- Observation: **MINRES  $\Rightarrow$  Better accuracy!** (and in general more stable)

# Testing pure NGD with MINRES

## Natural Gradient Descent (NGD)

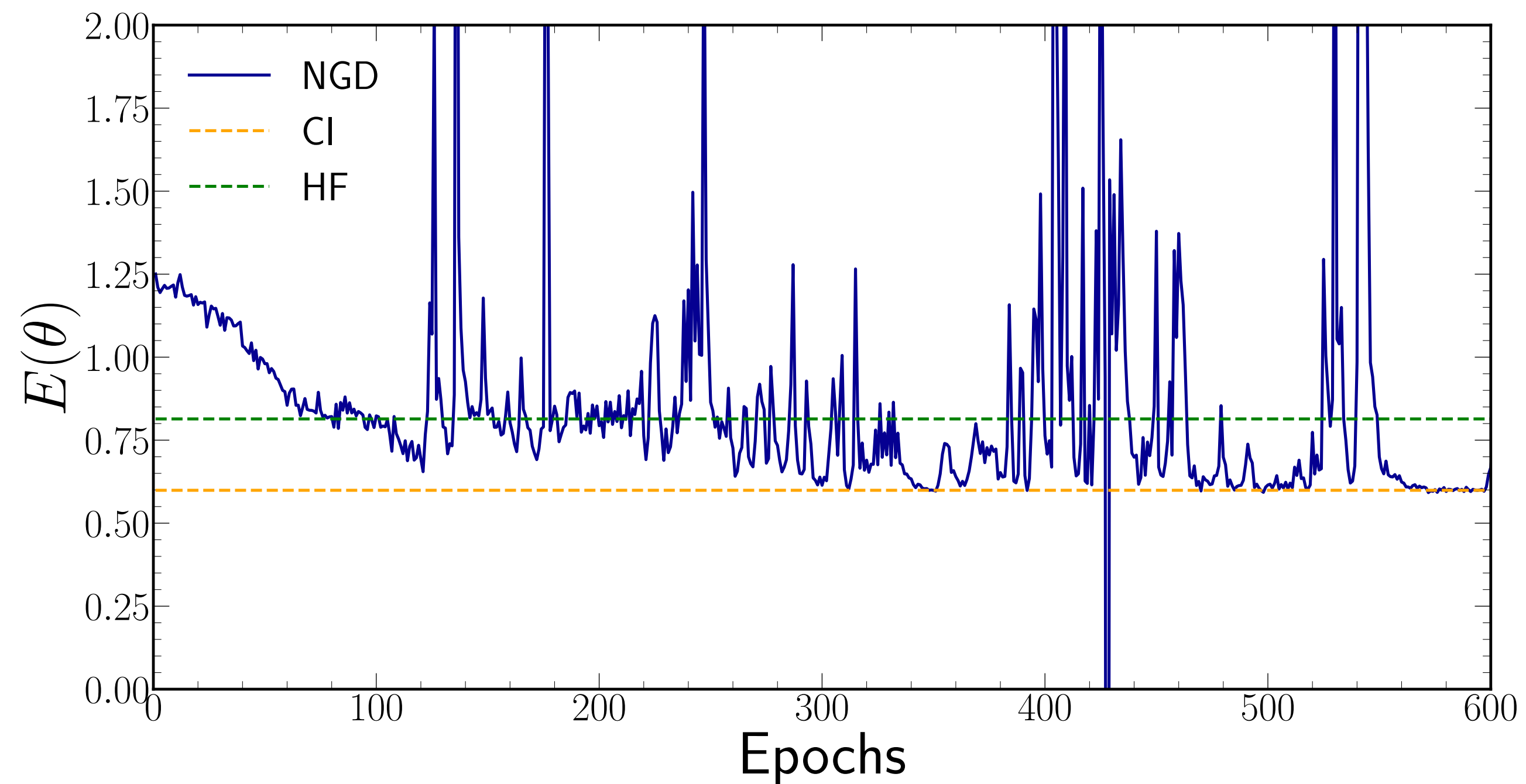
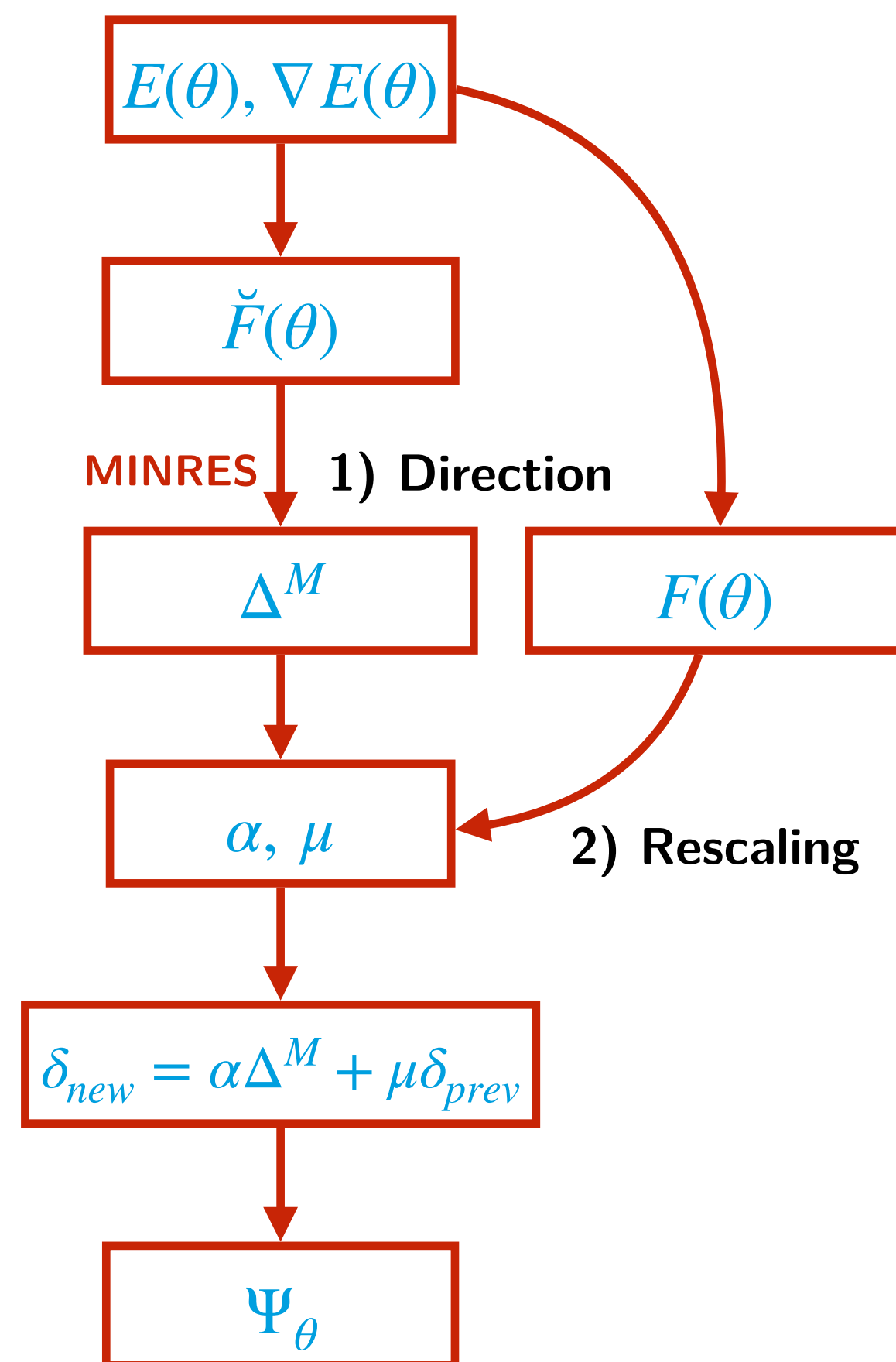


# Testing pure NGD with MINRES

Natural gradient Descent (NGD):  $A = 2, V_0 = -10$

15

## Natural Gradient Descent (NGD)

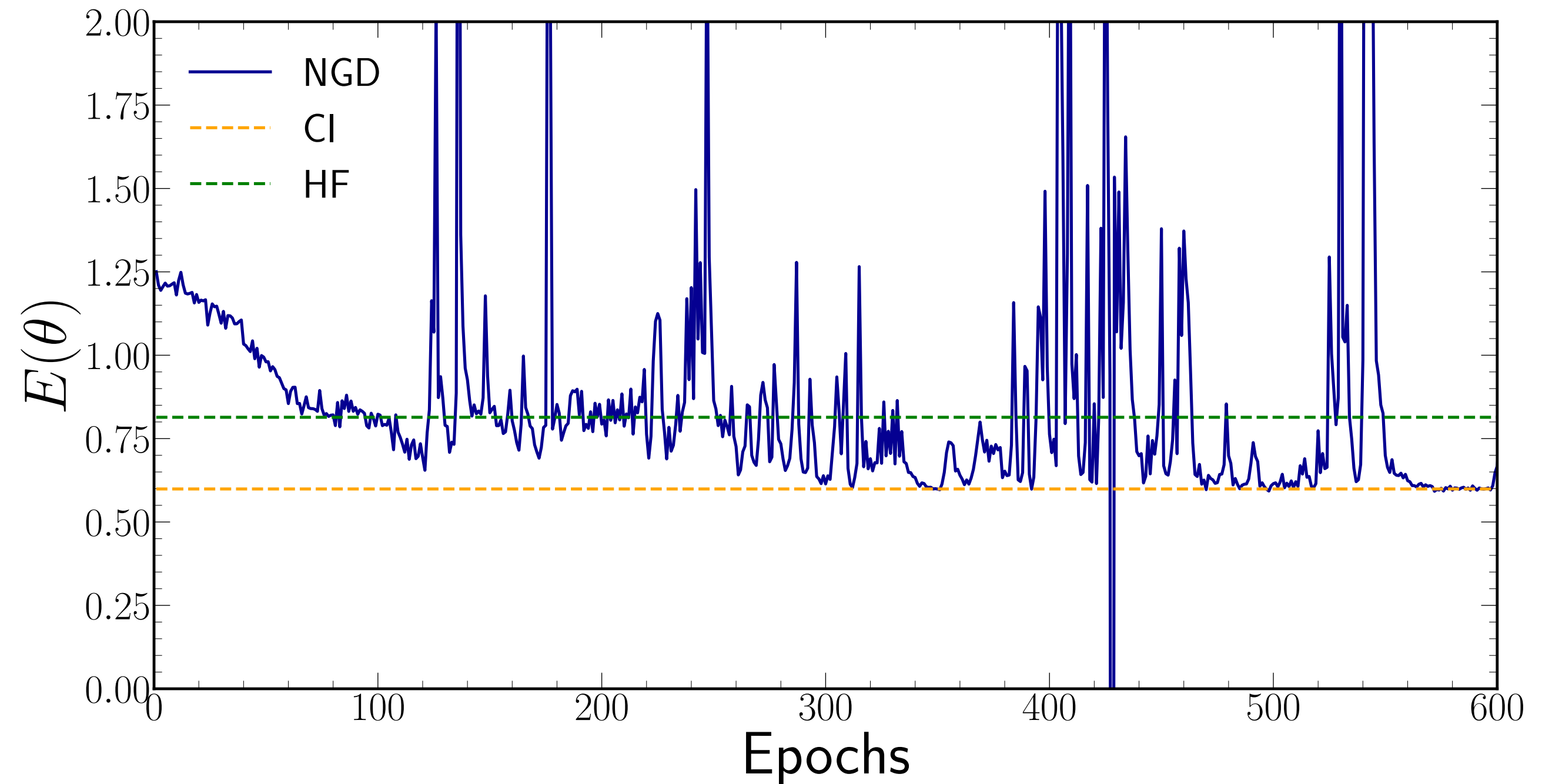
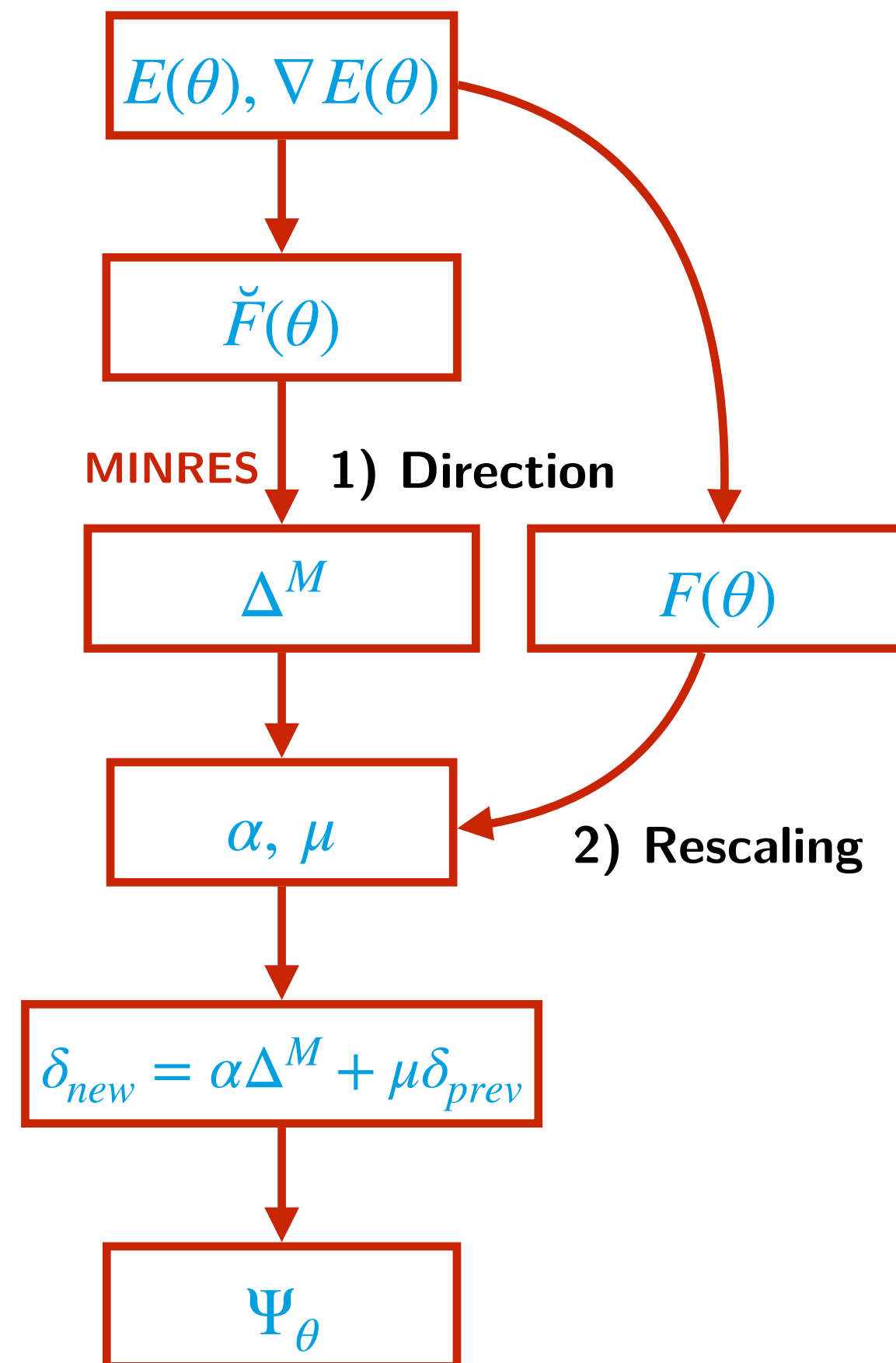


# Testing pure NGD with MINRES

Natural gradient Descent (NGD):  $A = 2, V_0 = -10$

15

## Natural Gradient Descent (NGD)



## Failure of Natural Gradient Descent (NGD)

- Testing information geometry with MINRES:
  - Observation: even when using exact Fisher  $F(\theta) \rightarrow$  **huge instabilities**
  - Confirms relevance of  $H_Q$  and suggests that information geometry is sub-optimal for VMC
- Can we find better than the Fisher metric?
  - Quasi-Hessian  $H_Q \neq$  PSD  $\Rightarrow$  lead to **instabilities** as well
  - ➔ Better geometry for VMC?**

# Outline

- **Variational Monte Carlo with Neural Quantum States**
  - Overview of VMC with NQS
  - The Kronecker-Factored Approximate Curvature (KFAC)
- **Augmented KFAC for VMC problems**
  - Scaling improvement from a Quasi-Newton approach
  - Direction improvement from MINRES
- **Decision geometry for VMC**
  - Game theory reformulation of VMC
  - Testing decisional gradient descent

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- ➔ Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} [\partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X)]$
- ➔  $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$



# From information to decision geometry

17

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- ➔ Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} [\partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X)]$
- ➔  $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Efficient implementation

[Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- ➔ **Fast and reliable convergence**

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Non supervised learning problem

- Minimize  $h(\theta) = -\mathbb{E}_{X \sim p_\theta} [S(X, p_\theta)] \equiv -S(p_\theta, p_\theta)$ 
  - $S \equiv$  scoring rule,  $p_\theta \equiv$  model to optimize
  - Very general problem

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- ➔ Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} [\partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X)]$
- ➔  $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Efficient implementation

[Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- ➔ **Fast and reliable convergence**

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Non supervised learning problem

- Minimize  $h(\theta) = -\mathbb{E}_{X \sim p_\theta} [S(X, p_\theta)] \equiv -S(p_\theta, p_\theta)$ 
  - $S \equiv$  scoring rule,  $p_\theta \equiv$  model to optimize
  - Very general problem

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- ➔ Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} [\partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X)]$
- ➔  $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Decision geometry [Dawid (2006)]

- Necessary condition
  - $\forall p, q, S(p, p) \leq S(p, q) \Rightarrow$  proper scoring rule [Gneiting, Raftery (2007)]
- Game-theory generalizations
  - Entropy:  $H(p) \equiv S(p, p)$
  - Cross-entropy:  $H(p, q) \equiv S(p, q)$
  - Divergence:  $D_S(p, q) \equiv S(p, q) - S(p, p)$
  - $D_S(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T G_S(\theta) \delta + O(\delta^3)$  **Decision geometry**
- ➔  $\delta_{\text{DGD}} = -G_S^{-1}(\theta) \nabla h(\theta)$
- Recovers information geometry:  $S(p, x) = -\ln p(x)$

## Efficient implementation

[Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- ➔ **Fast and reliable convergence**

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Non supervised learning problem

- Minimize  $h(\theta) = -\mathbb{E}_{X \sim p_\theta} [S(X, p_\theta)] \equiv -S(p_\theta, p_\theta)$ 
  - $S \equiv$  scoring rule,  $p_\theta \equiv$  model to optimize
  - Very general problem

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- ➔ Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} [\partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X)]$
- ➔  $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Decision geometry [Dawid (2006)]

- Necessary condition
  - $\forall p, q, S(p, p) \leq S(p, q) \Rightarrow$  proper scoring rule [Gneiting, Raftery (2007)]
- Game-theory generalizations
  - Entropy:  $H(p) \equiv S(p, p)$
  - Cross-entropy:  $H(p, q) \equiv S(p, q)$
  - Divergence:  $D_S(p, q) \equiv S(p, q) - S(p, p)$
  - $D_S(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T G_S(\theta) \delta + O(\delta^3)$  **Decision geometry**
- ➔  $\delta_{\text{DGD}} = -G_S^{-1}(\theta) \nabla h(\theta)$
- Recovers information geometry:  $S(p, x) = -\ln p(x)$

## Efficient implementation

[Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- ➔ **Fast and reliable convergence**

## Open questions

- When is a scoring rule leading to efficient DGD?
- Efficient implementation?
- ...

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Non supervised learning problem

- Minimize  $h(\theta) = -\mathbb{E}_{X \sim p_\theta} [S(X, p_\theta)] \equiv -S(p_\theta, p_\theta)$ 
  - $S \equiv$  scoring rule,  $p_\theta \equiv$  model to optimize
  - Very general problem

## Variational Monte Carlo problem

- Minimize  $E(\theta) = \mathbb{E}_{X \sim |\Psi_\theta|^2} [E_{L,\theta}(X)]$ 
  - $E_{L,\theta}(X) \equiv -\frac{1}{2} \sum_i \left[ \partial_{x_i}^2 \ln |\Psi_\theta(X)| + \left( \partial_{x_i} \ln |\Psi_\theta(X)| \right)^2 \right] + V(X)$
  - Quantum many-body problem

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- ➔ Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} \left[ \partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X) \right]$
- ➔  $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Decision geometry [Dawid (2006)]

- Necessary condition
  - $\forall p, q, S(p, p) \leq S(p, q) \Rightarrow$  proper scoring rule [Gneiting, Raftery (2007)]
- Game-theory generalizations
  - Entropy:  $H(p) \equiv S(p, p)$
  - Cross-entropy:  $H(p, q) \equiv S(p, q)$
  - Divergence:  $D_S(p, q) \equiv S(p, q) - S(p, p)$
  - $D_S(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T G_S(\theta) \delta + O(\delta^3)$  **Decision geometry**
- ➔  $\delta_{\text{DGD}} = -G_S^{-1}(\theta) \nabla h(\theta)$
- Recovers information geometry:  $S(p, x) = -\ln p(x)$

## Efficient implementation

[Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- ➔ **Fast and reliable convergence**

## Open questions

- When is a scoring rule leading to efficient DGD?
- Efficient implementation?
- ...

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Non supervised learning problem

- Minimize  $h(\theta) = -\mathbb{E}_{X \sim p_\theta} [S(X, p_\theta)] \equiv -S(p_\theta, p_\theta)$ 
  - $S \equiv$  scoring rule,  $p_\theta \equiv$  model to optimize
  - Very general problem

## Variational Monte Carlo problem

- Minimize  $E(\theta) = \mathbb{E}_{X \sim |\Psi_\theta|^2} [E_{L,\theta}(X)]$ 
  - $E_{L,\theta}(X) \equiv -\frac{1}{2} \sum_i \left[ \partial_{x_i}^2 \ln |\Psi_\theta(X)| + \left( \partial_{x_i} \ln |\Psi_\theta(X)| \right)^2 \right] + V(X)$
  - Quantum many-body problem

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} [\partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X)]$
- $\Rightarrow \delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Decision geometry [Dawid (2006)]

- Necessary condition
  - $\forall p, q, S(p, p) \leq S(p, q) \Rightarrow$  proper scoring rule [Gneiting, Raftery (2007)]
- Game-theory generalizations
  - Entropy:  $H(p) \equiv S(p, p)$
  - Cross-entropy:  $H(p, q) \equiv S(p, q)$
  - Divergence:  $D_S(p, q) \equiv S(p, q) - S(p, p)$
  - $D_S(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T G_S(\theta) \delta + O(\delta^3)$  **Decision geometry**
- $\Rightarrow \delta_{\text{DGD}} = -G_S^{-1}(\theta) \nabla h(\theta)$

## Game-theory reformulation of VMC

- Natural scoring rule
  - $\forall p_\theta, x, S_{\text{VMC}}(x, p_\theta) \equiv -E_{L,\theta}(x) \rightarrow$  **Proper scoring rule**
- Induced geometry (Divergence)
  - $D_{\text{VMC}}(p_\theta, p_{\theta'}) = \frac{1}{8} \sum_i \mathbb{E} \left[ \left( \partial_{x_i} \ln p_\theta(X) - \partial_{x_i} \ln p_{\theta'}(X) \right)^2 \right]$
  - $G_{\text{VMC}}(\theta)_{\theta_1 \theta_2} = \frac{1}{4} \sum_i \mathbb{E} \left[ \left( \partial_{\theta_1} \partial_{x_i} \ln p_\theta(X) \right) \left( \partial_{\theta_2} \partial_{x_i} \ln p_\theta(X) \right) \right]$
- $\Rightarrow \delta_{\text{VMC}} = -G_{\text{VMC}}^{-1}(\theta) \nabla E(\theta)$
- Physically motivated geometry

$$D_{\text{VMC}}(p_\theta, p_{\theta'}) = \mathbb{E}_{X \sim p_\theta} [E_{L,\theta}(X) - E_{L,\theta'}(X)] \simeq E(\theta) - E(\theta') \text{ (up to re-weighting)}$$

## Efficient implementation [Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- $\Rightarrow$  **Fast and reliable convergence**

## Open questions

- When is a scoring rule leading to efficient DGD?
- Efficient implementation?
- ...

# From information to decision geometry

## Supervised learning problem

- Minimize  $L(\theta) = \mathbb{E}_{X \sim q} [-\ln p_\theta(X)]$  (cross-entropy loss)
  - $q \equiv$  target distribution,  $p_\theta \equiv$  model to optimize
  - Equivalent to “fitting data points” problems

## Non supervised learning problem

- Minimize  $h(\theta) = -\mathbb{E}_{X \sim p_\theta} [S(X, p_\theta)] \equiv -S(p_\theta, p_\theta)$ 
  - $S \equiv$  scoring rule,  $p_\theta \equiv$  model to optimize
  - Very general problem

## Variational Monte Carlo problem

- Minimize  $E(\theta) = \mathbb{E}_{X \sim |\Psi_\theta|^2} [E_{L,\theta}(X)]$ 
  - $E_{L,\theta}(X) \equiv -\frac{1}{2} \sum_i \left[ \partial_{x_i}^2 \ln |\Psi_\theta(X)| + \left( \partial_{x_i} \ln |\Psi_\theta(X)| \right)^2 \right] + V(X)$
  - Quantum many-body problem

## Natural gradient descent [Amari (1997)]

- Local problem: solve for  $\delta$  such that  $\|\delta\|_F = \text{cst}$
- Kullback-Leibler divergence and the Fisher matrix
  - $D_{\text{KL}}(p_1, p_2) \equiv \mathbb{E}_{X \sim p_1} [-\ln p_2(X) - (-\ln p_1(X))]$
  - $D_{\text{KL}}(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T F(\theta) \delta + O(\delta^3)$  **Information geometry**
- Fisher metric:  $F(\theta)_{\theta_1 \theta_2} \equiv \mathbb{E}_{X \sim p_\theta} \left[ \partial_{\theta_1} \ln p_\theta(X) \partial_{\theta_2} \ln p_\theta(X) \right]$
- $\delta_{\text{NGD}} = -F^{-1}(\theta) \nabla L(\theta)$

## Decision geometry [Dawid (2006)]

- Necessary condition
  - $\forall p, q, S(p, p) \leq S(p, q) \Rightarrow$  proper scoring rule [Gneiting, Raftery (2007)]
- Game-theory generalizations
  - Entropy:  $H(p) \equiv S(p, p)$
  - Cross-entropy:  $H(p, q) \equiv S(p, q)$
  - Divergence:  $D_S(p, q) \equiv S(p, q) - S(p, p)$
  - $D_S(p_\theta, p_{\theta+\delta}) = \frac{1}{2} \delta^T G_S(\theta) \delta + O(\delta^3)$  **Decision geometry**
- $\delta_{\text{DGD}} = -G_S^{-1}(\theta) \nabla h(\theta)$

## Game-theory reformulation of VMC

- Natural scoring rule
  - $\forall p_\theta, x, S_{\text{VMC}}(x, p_\theta) \equiv -E_{L,\theta}(x) \rightarrow$  **Proper scoring rule**
- Induced geometry (Divergence)
  - $D_{\text{VMC}}(p_\theta, p_{\theta'}) = \frac{1}{8} \sum_i \mathbb{E} \left[ \left( \partial_{x_i} \ln p_\theta(X) - \partial_{x_i} \ln p_{\theta'}(X) \right)^2 \right]$
  - $G_{\text{VMC}}(\theta)_{\theta_1 \theta_2} = \frac{1}{4} \sum_i \mathbb{E} \left[ \left( \partial_{\theta_1} \partial_{x_i} \ln p_\theta(X) \right) \left( \partial_{\theta_2} \partial_{x_i} \ln p_\theta(X) \right) \right]$
- $\delta_{\text{VMC}} = -G_{\text{VMC}}^{-1}(\theta) \nabla E(\theta)$
- Physically motivated geometry
  - $D_{\text{VMC}}(p_\theta, p_{\theta'}) = \mathbb{E}_{X \sim p_\theta} [E_{L,\theta}(X) - E_{L,\theta'}(X)]$
  - $\simeq E(\theta) - E(\theta')$  (up to re-weighting)

## Efficient implementation [Martens, Grosse (2015)]

- KFAC (Kronecker-Factored Approximate Curvature)
  - KFAC  $\sim$  crude approximation of the Fisher metric
  - Direction update using KFAC Fisher
  - Scaling update using exact Fisher
- Fast and reliable convergence**

## Open questions

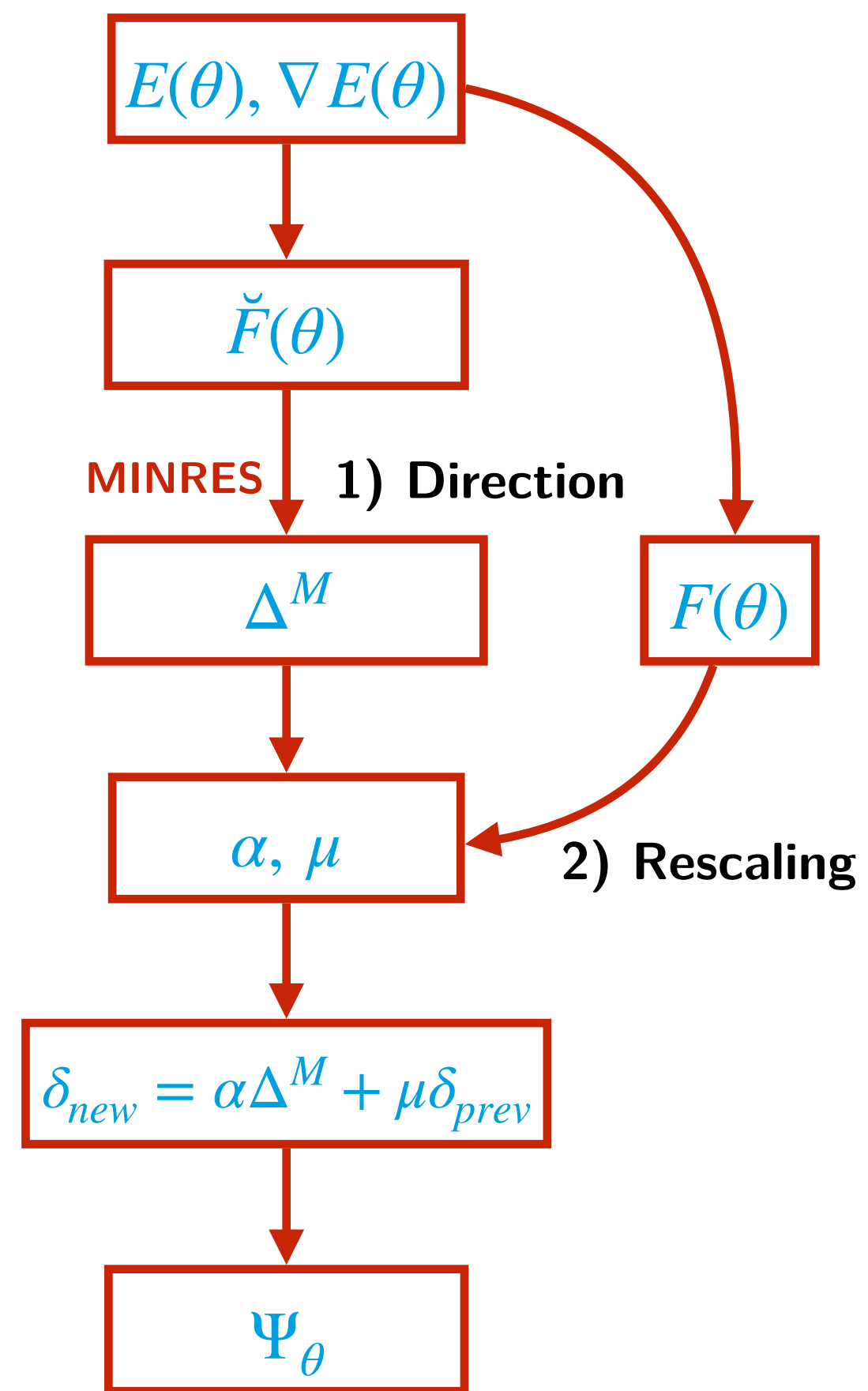
- When is a scoring rule leading to efficient DGD?
- Efficient implementation?
- ...

## Practicable optimizer?

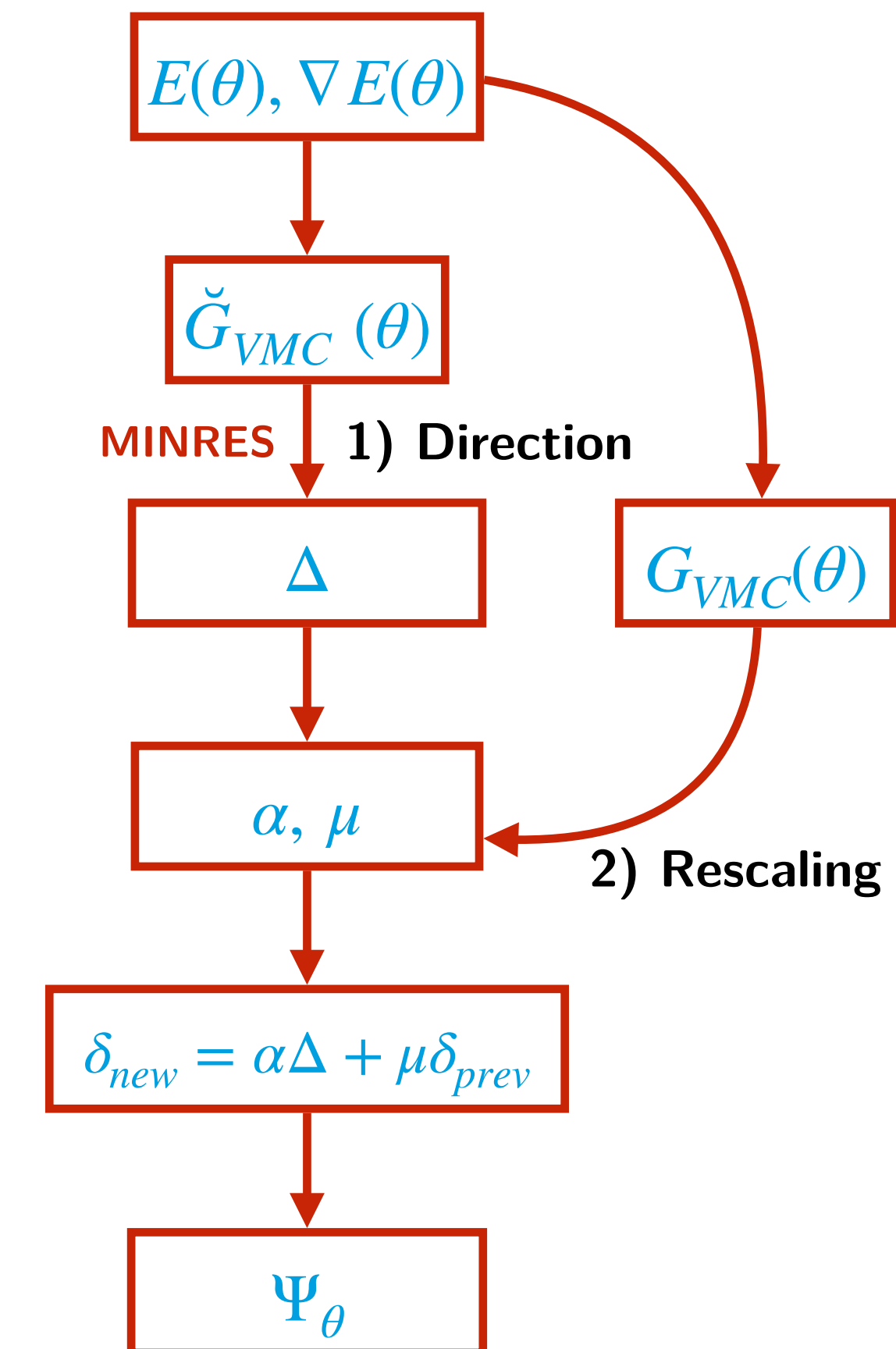
- How good is it strategically? (Convergence in epochs)
- Can it be performant? (Overall wall-time and stability)

# Decision vs information geometry

## Natural Gradient Descent



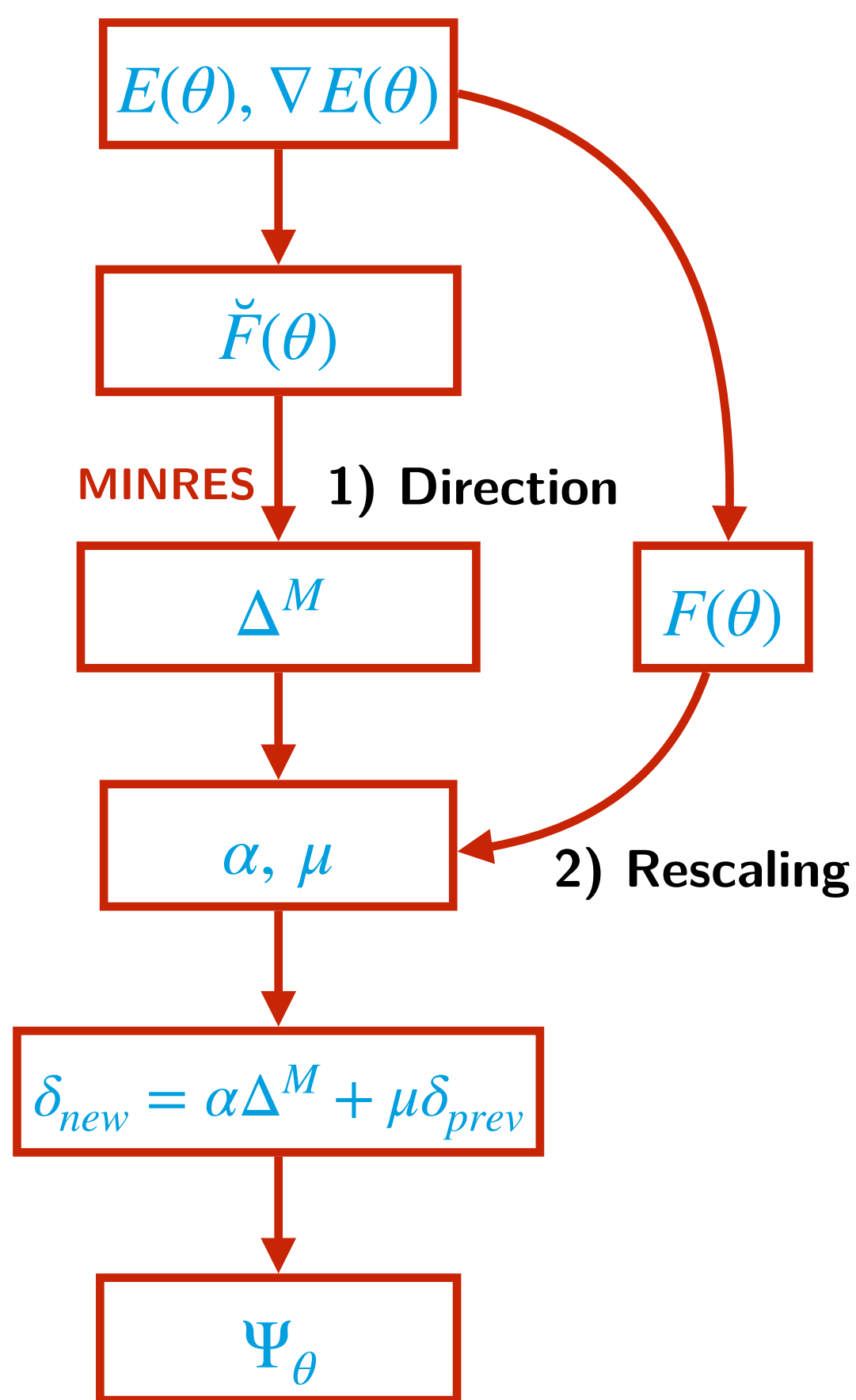
## Decisional Gradient Descent



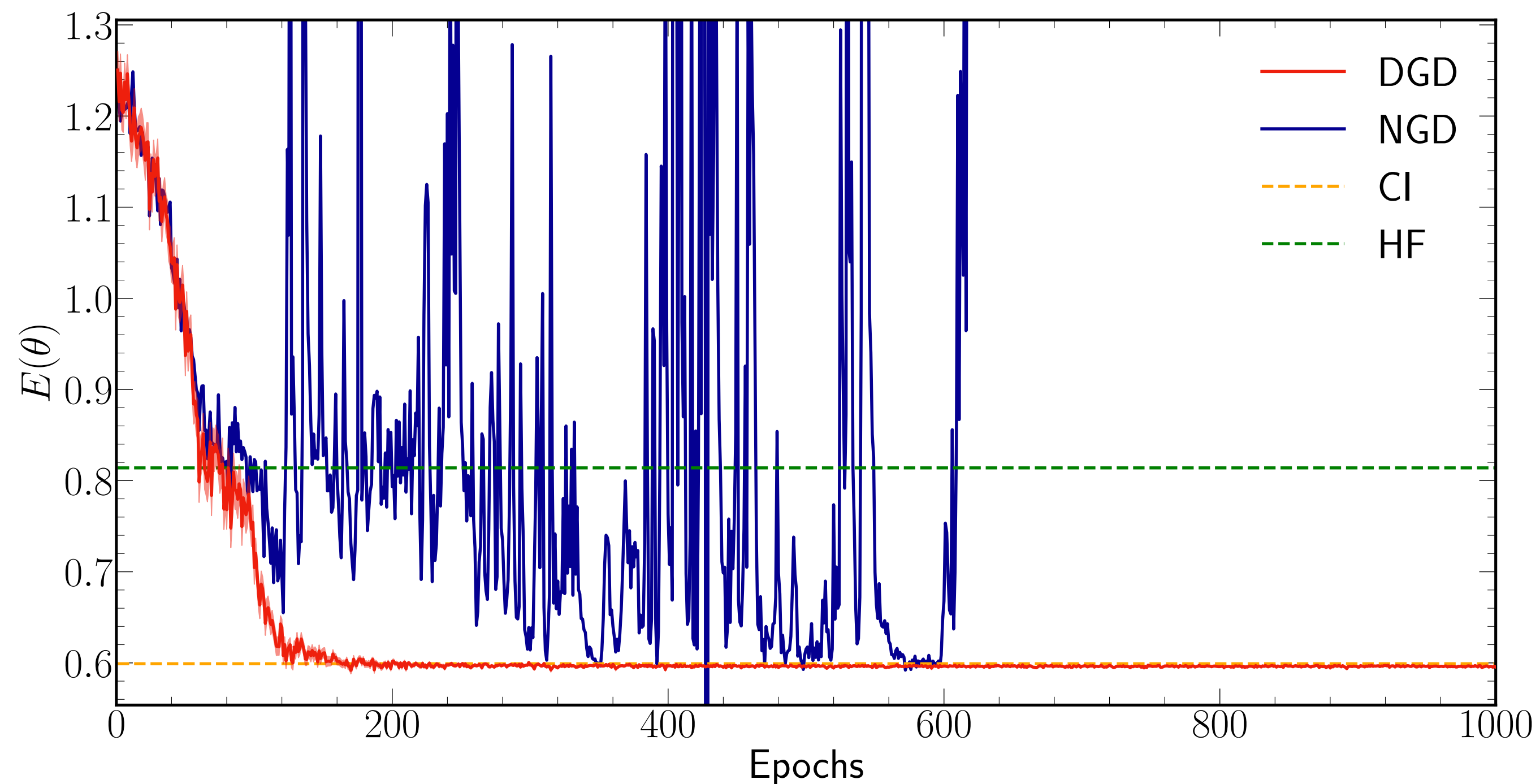


# Decision vs information geometry

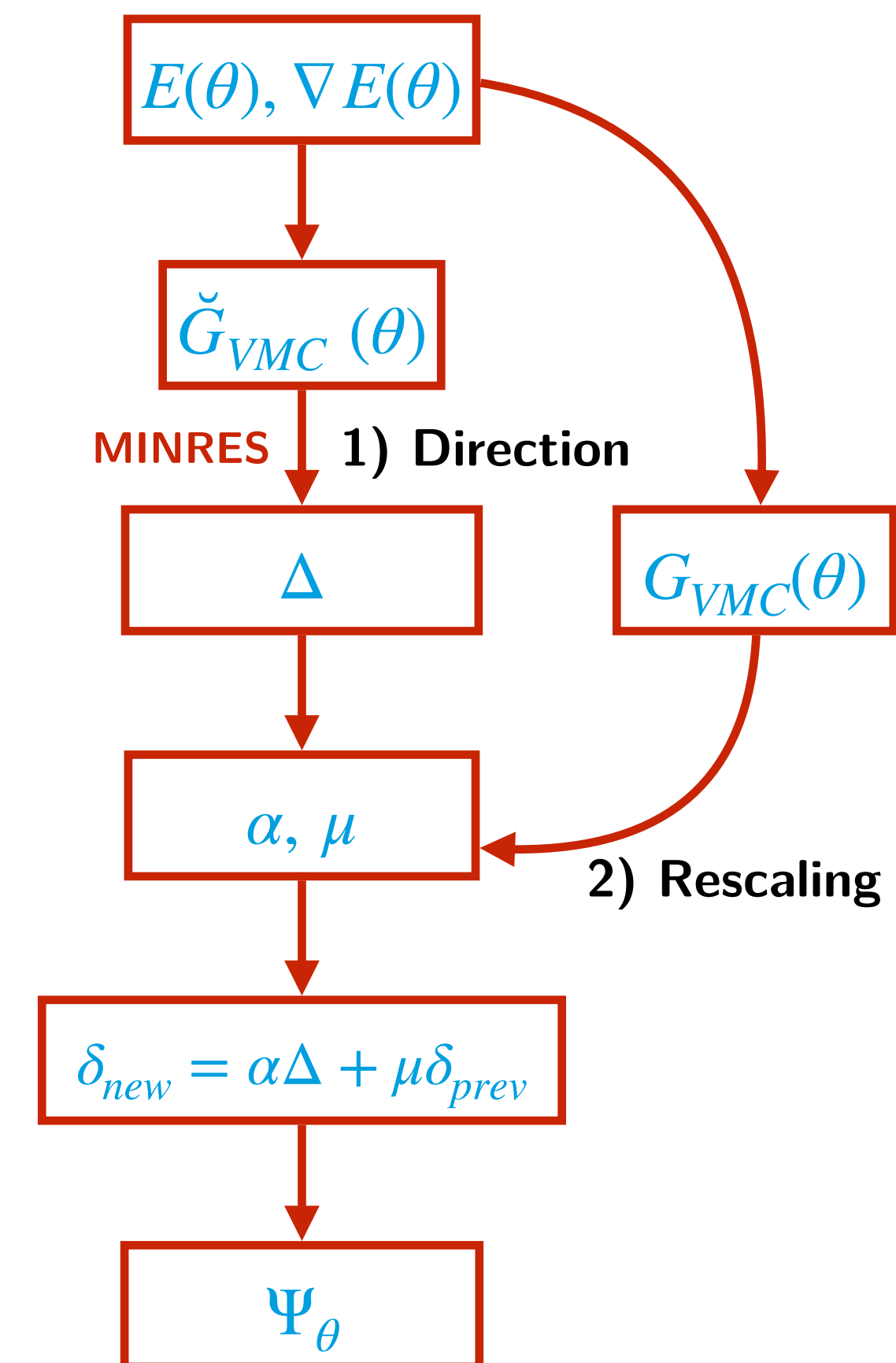
## Natural Gradient Descent



NGD vs DGD:  $A = 2, V_0 = -10$



## Decisional Gradient Descent

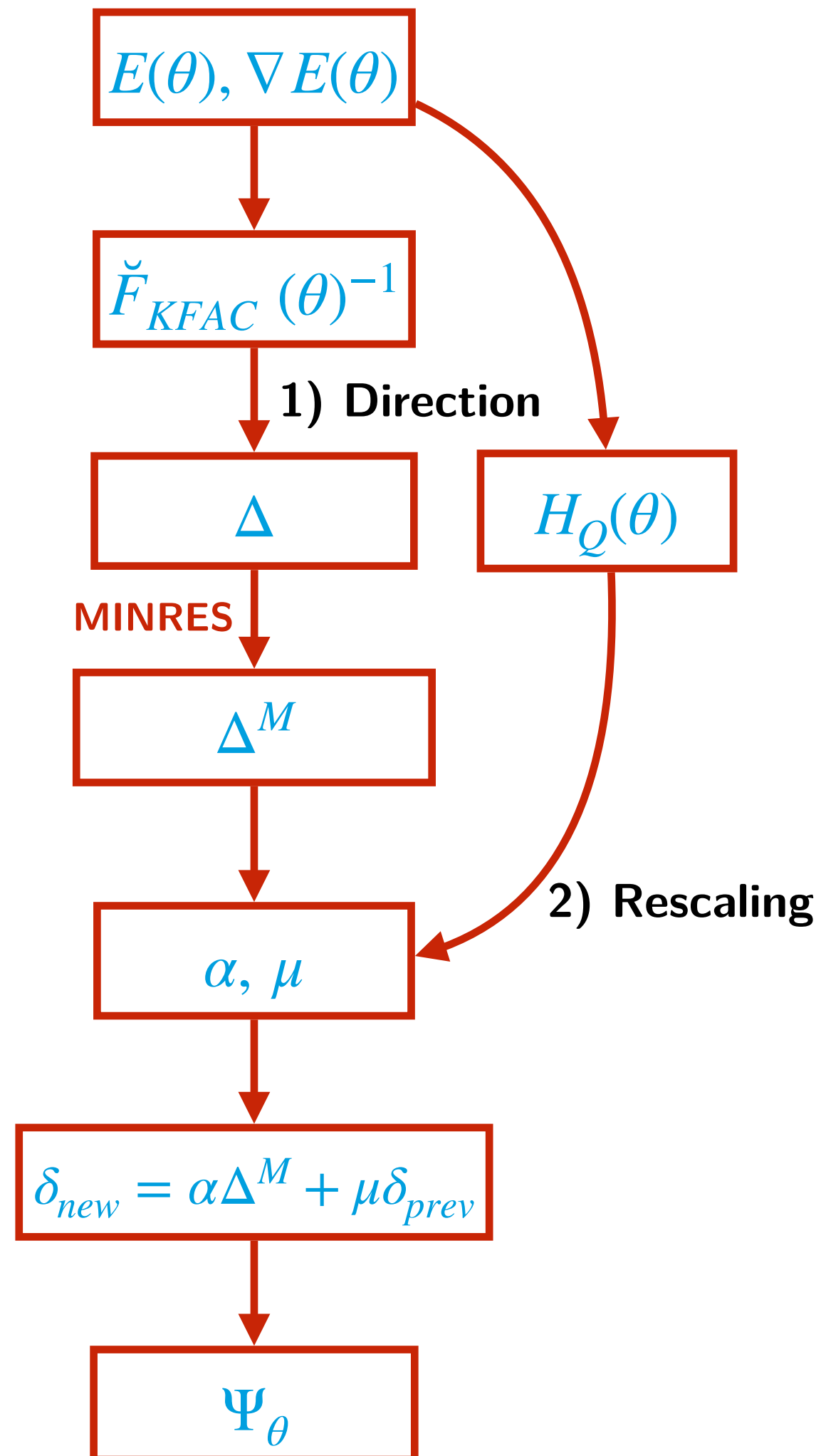


### Results

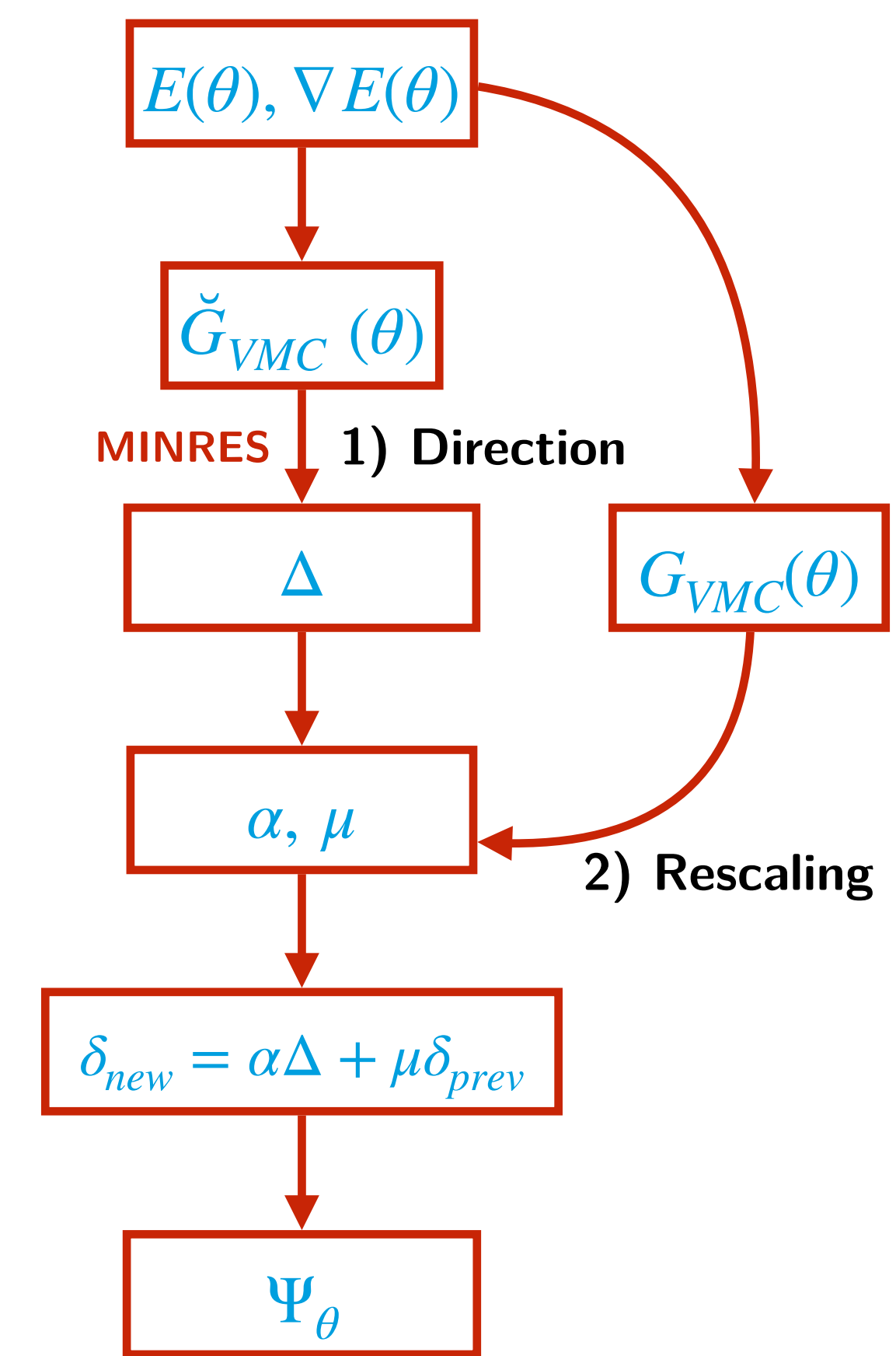
- Stability: huge improvement from decision geometry in all cases
- ➔ Much better starting point for designing optimizers for VMC

# Comparing with our previous best optimizer

## QN-MR-KFAC

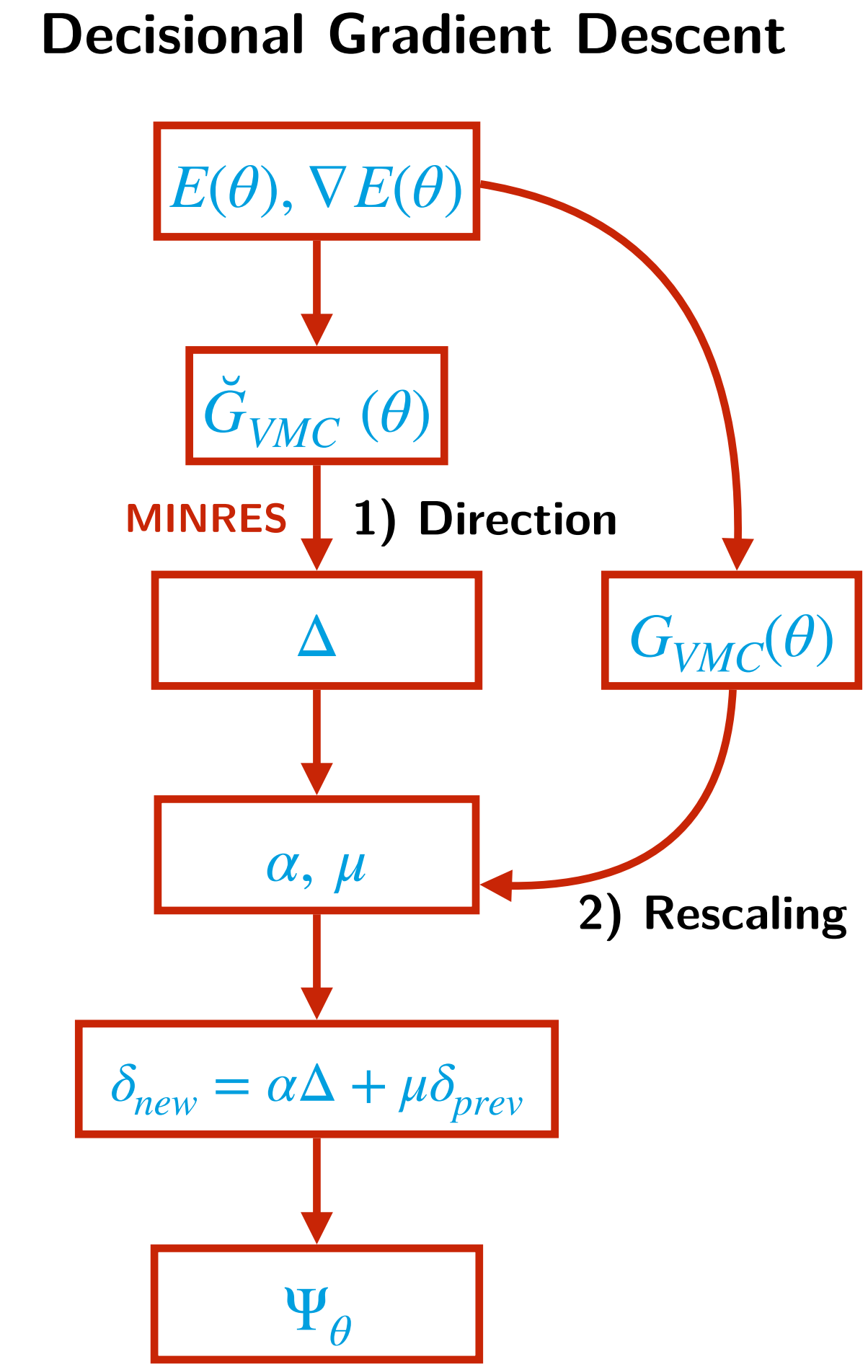
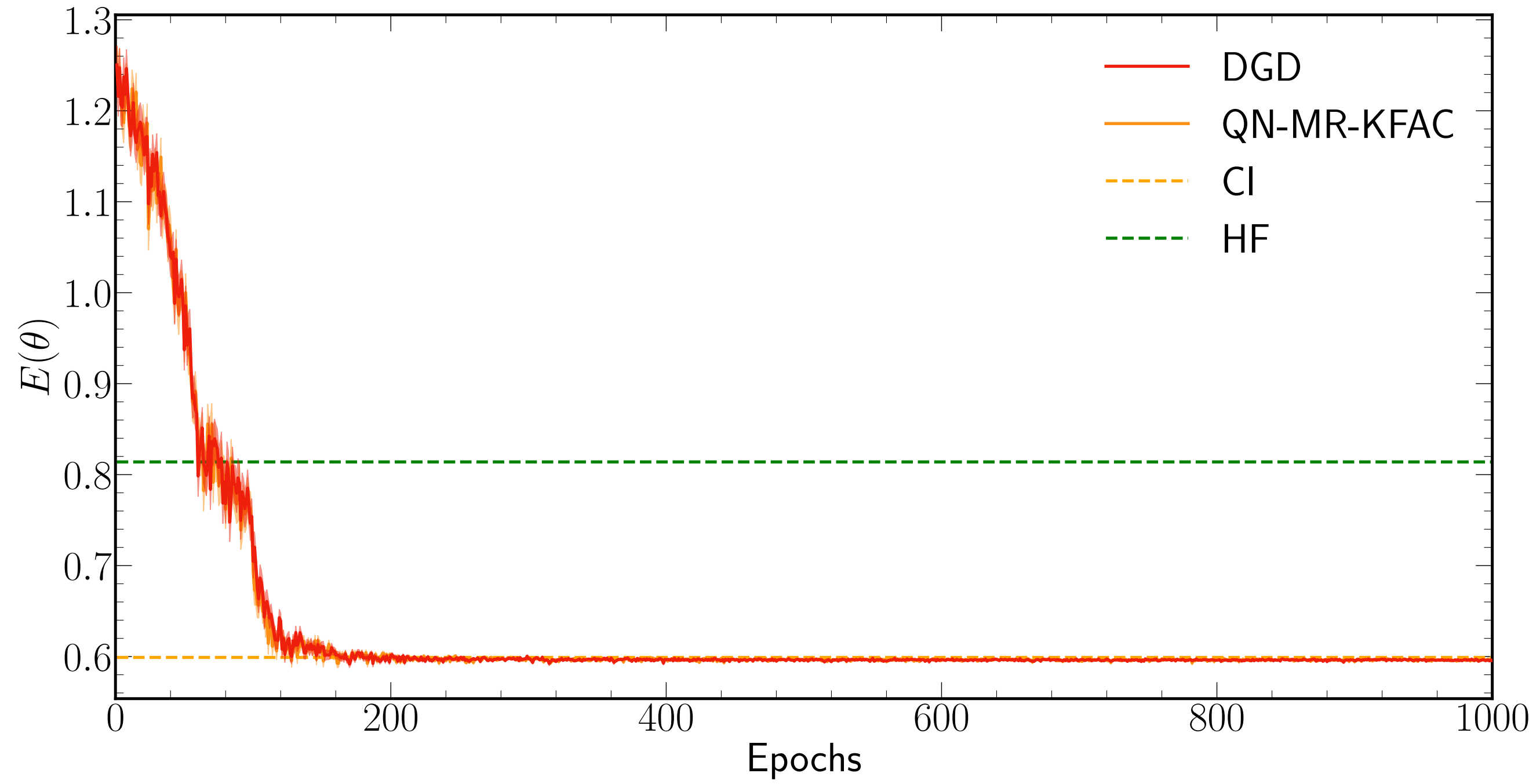
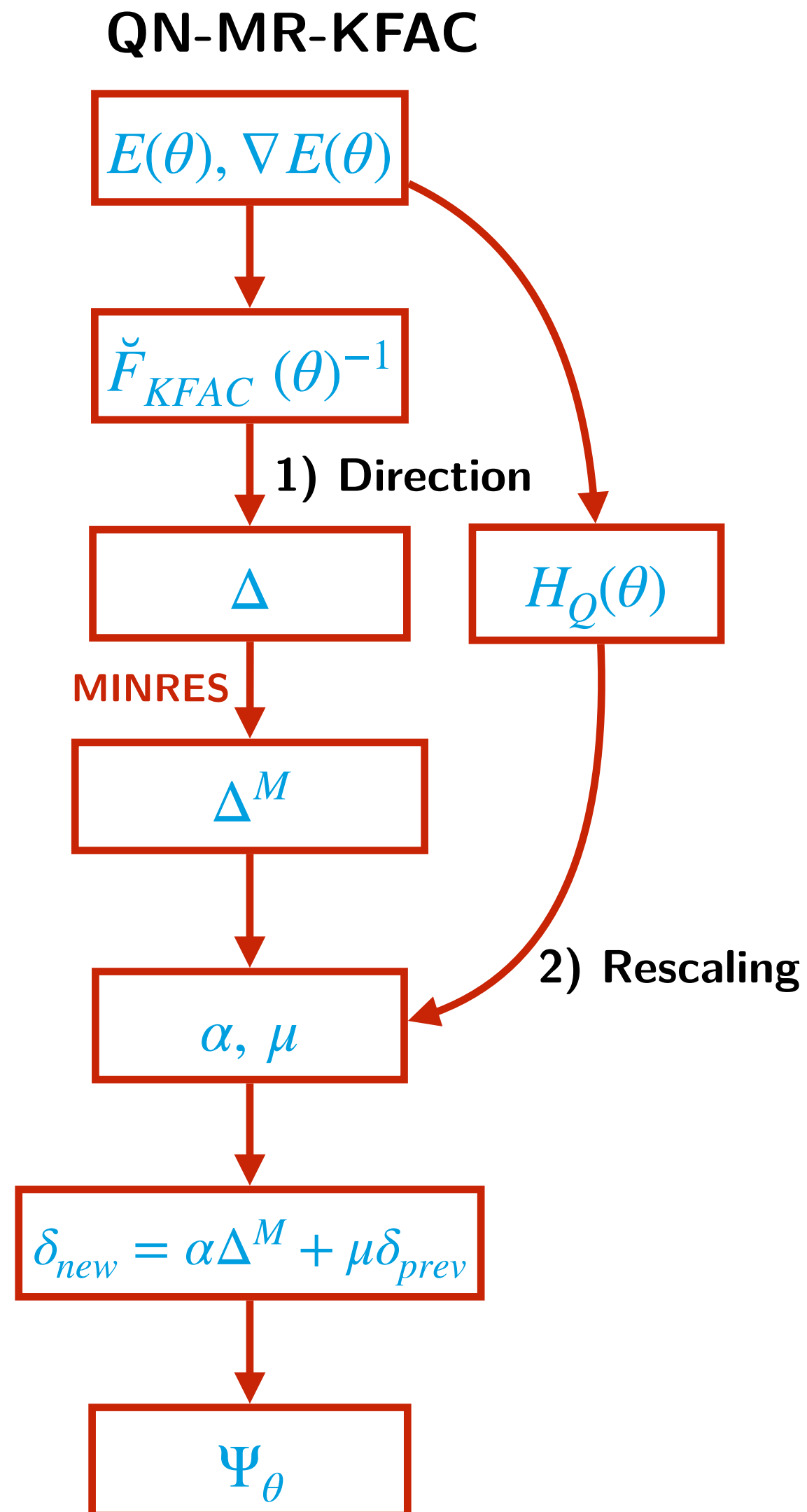


## Decisional Gradient Descent



# Comparing with our previous best optimizer

QN-MR-KFAC vs DGD:  $A = 2, V_0 = -10$



### Results

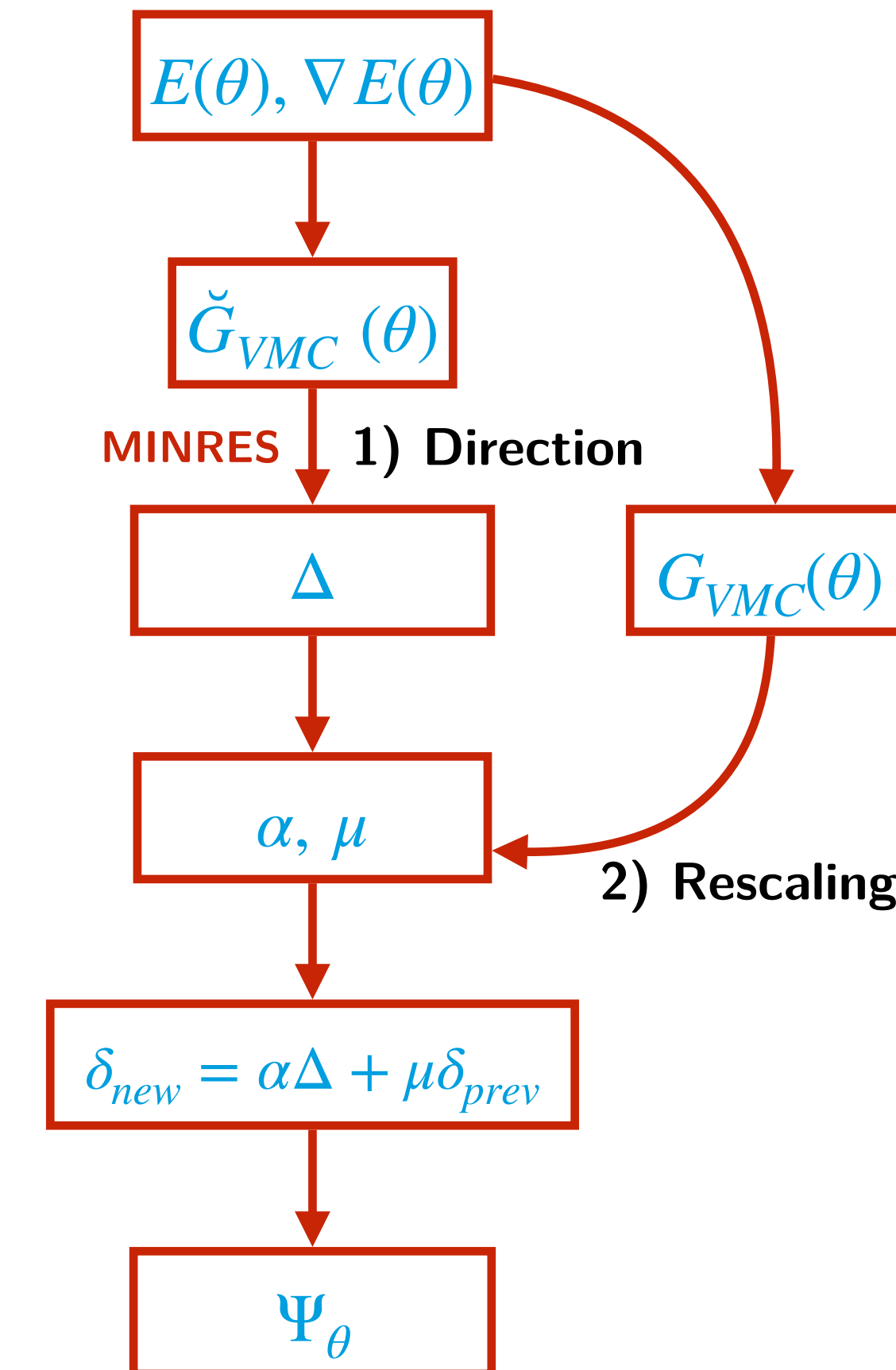
- Stability: DGD more stable than any other refinement of KFAC (not shown here)
- Accuracy and speed: DGD on par with QN-MR-KFAC

# Comparing with Adam

## Decisional Gradient Descent

**Adam**

$$m_n = \frac{\beta_1 m_{n-1} + (1 - \beta_1) \nabla E(\theta_n)}{(1 - \beta_1^n)}$$
$$s_n = \frac{\beta_2 s_{n-1} + (1 - \beta_2) (\nabla E(\theta_n))^2}{(1 - \beta_2^n)}$$
$$\theta_{n+1} = \theta_n - \alpha \frac{m_n}{\sqrt{s_n} + \epsilon}$$



# Comparing with Adam

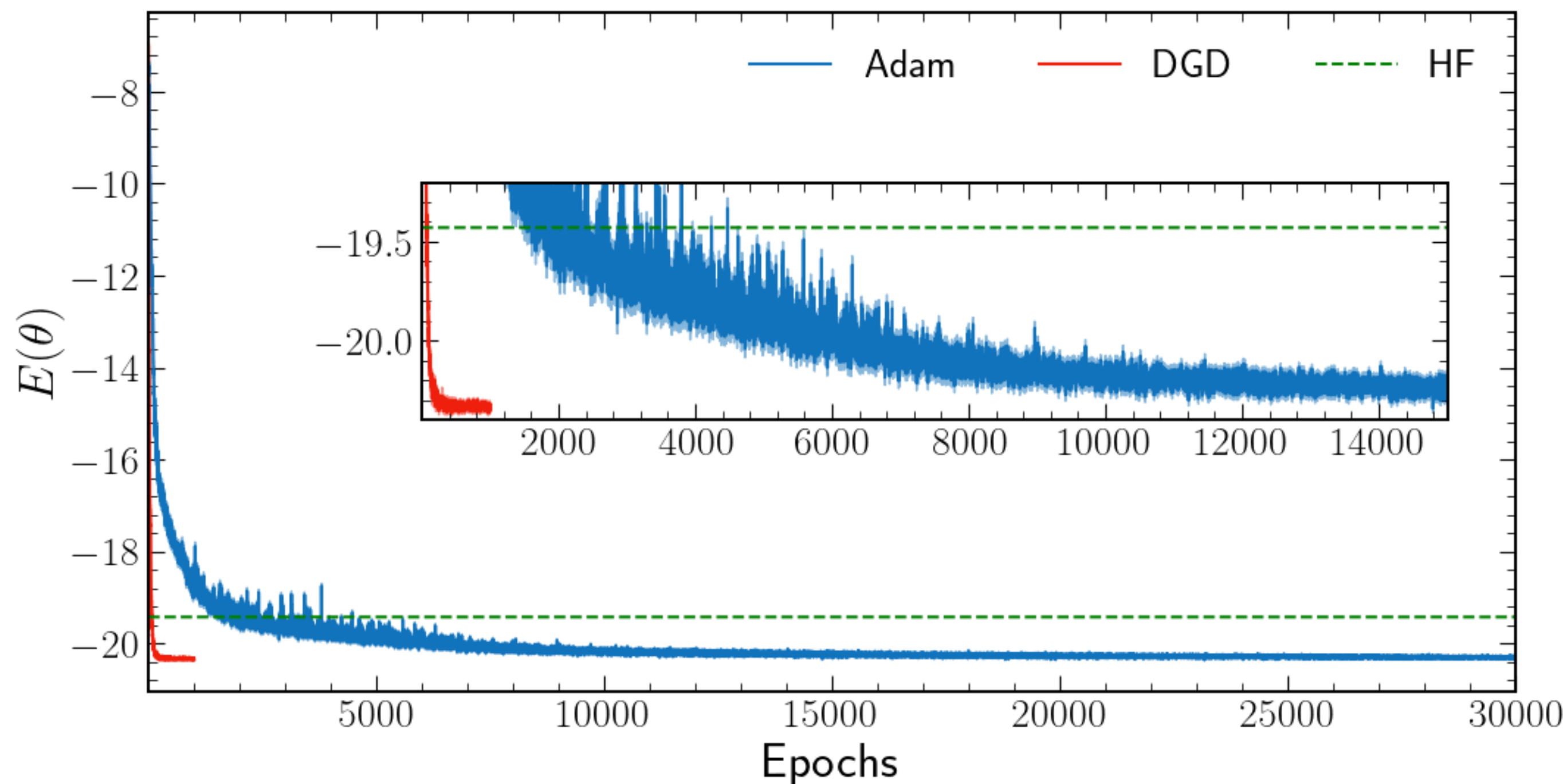
Adam vs DGD:  $A = 5, V_0 = -20$

**Adam**

$$m_n = \frac{\beta_1 m_{n-1} + (1 - \beta_1) \nabla E(\theta_n)}{(1 - \beta_1^n)}$$

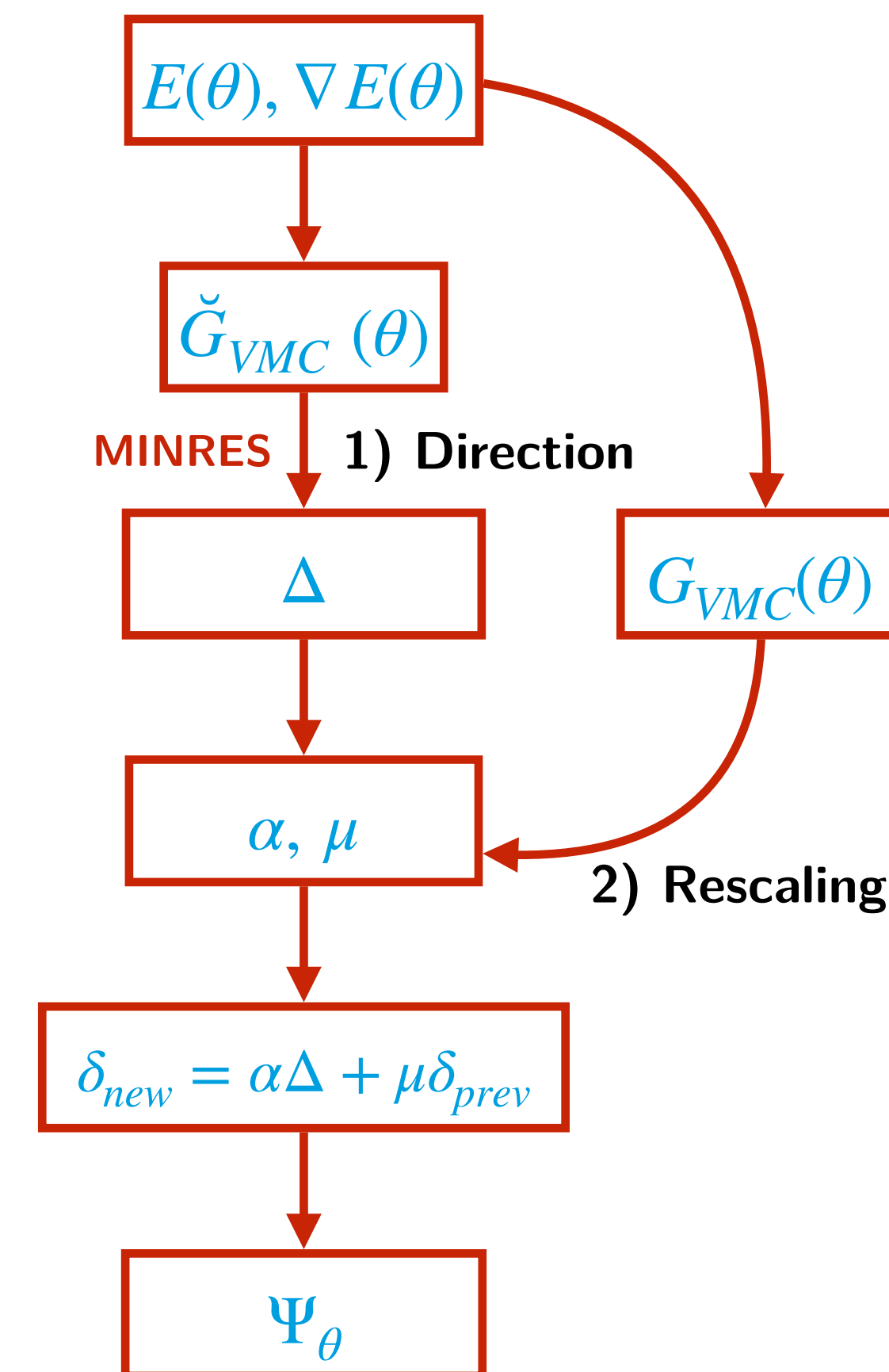
$$s_n = \frac{\beta_2 s_{n-1} + (1 - \beta_2) (\nabla E(\theta_n))^2}{(1 - \beta_2^n)}$$

$$\theta_{n+1} = \theta_n - \alpha \frac{m_n}{\sqrt{s_n} + \epsilon}$$

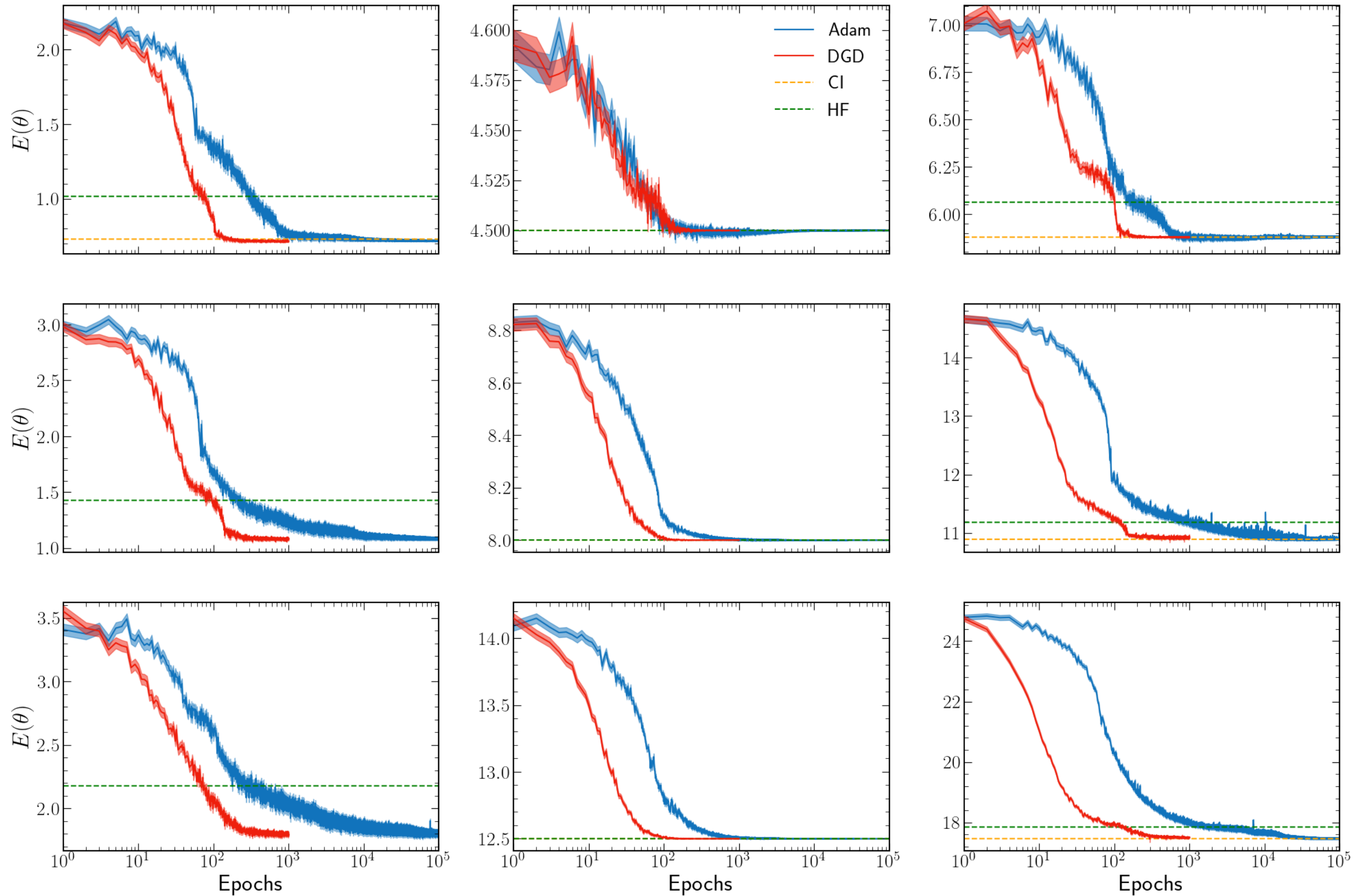


- Results**
- ✓ Accuracy: DGD systematically over perform Adam
  - ✓ Speed: DGD converges 10-100x faster than Adam in #epochs
  - ✗ Wall-time: Very naïve implementation of DGD  $\Rightarrow$  too early to quantify

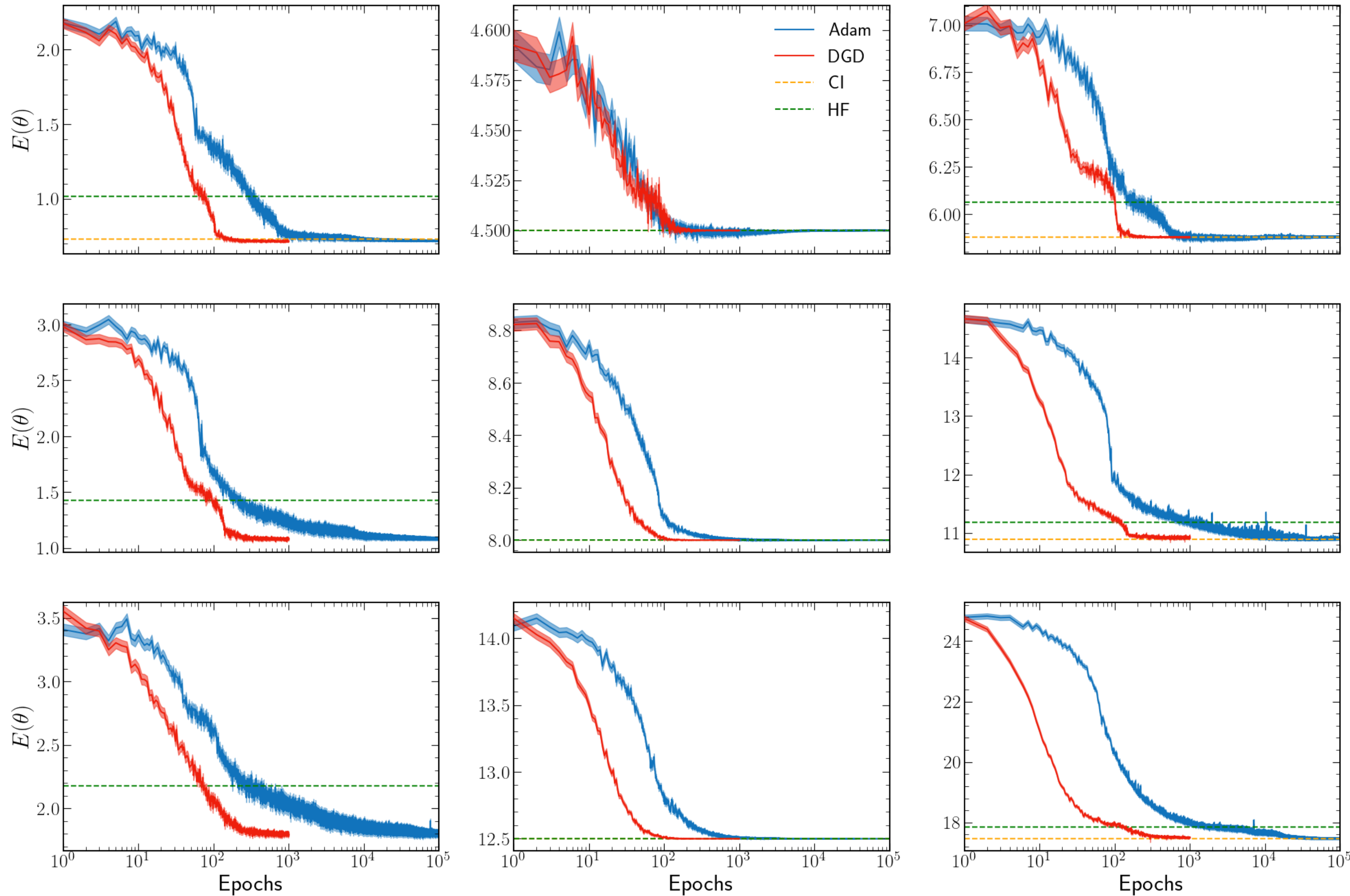
## Decisional Gradient Descent



# Testing across phenomenologies

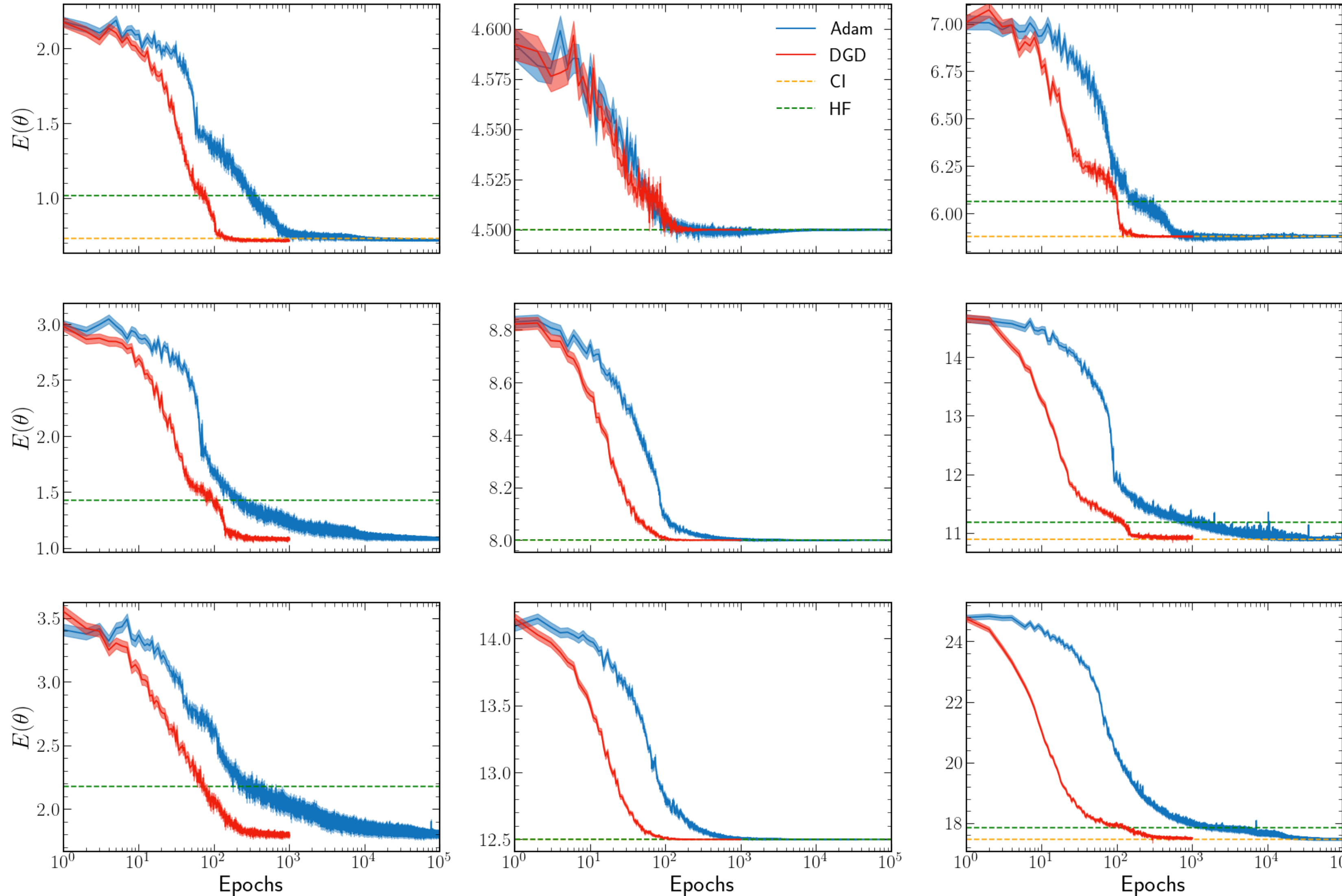


# Testing across phenomenologies

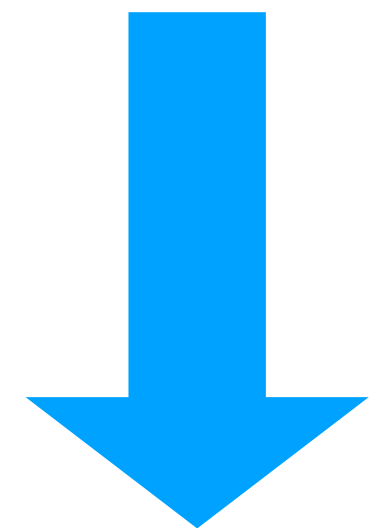


**Convergence of DGD:  
22 out of the 25 cases**

# Testing across phenomenologies



**Convergence of DGD:  
22 out of the 25 cases**



**Confirms the great potential  
of DGD for future optimizers!**



# Conclusions

# Conclusions

## VMC with neural networks

- Rapidly evolving field!
- Competitive with CCSD(T) in quantum chemistry
- **More systematic studies to be performed**
  - Numerical implementation to be optimized
  - Optimal architecture for nuclear systems?
  - Numerical complexity (time/memory)
- **Realistic nuclear systems now being investigated**
  - ➔ On-going work to reach  $A \sim 100$  nuclei (ANL)

# Conclusions

## VMC with neural networks

- Rapidly evolving field!
- Competitive with CCSD(T) in quantum chemistry
- **More systematic studies to be performed**
  - Numerical implementation to be optimized
  - Optimal architecture for nuclear systems?
  - Numerical complexity (time/memory)
- **Realistic nuclear systems now being investigated**
  - ➔ On-going work to reach  $A \sim 100$  nuclei (ANL)

## The optimizer: a critical part

- Simple many-body systems  $\Rightarrow$  easy to test new ideas
- **A promising novel optimizer based on decision geometry!**
  - Motivated by deficiencies of KFAC for VMC
  - Game theory re-formulation of VMC  $\Rightarrow$  **Decisional gradient descent**
  - Accurate, stable and fast
  - Simplest implementation  $\Rightarrow$  solid foundation for future improvements
- **With many potential refinements!**
  - Hessian-free-like  $\Rightarrow$  Inspiration for many potential algo improvements
  - KFAC-like approximation on decision metric?
  - Adapting the geometry for different many-body problems?
  - Other ML problems? **Can it be made as versatile as Adam?**

# Thank you Merci

[www.triumf.ca](http://www.triumf.ca)

Follow us **@TRIUMFLab**



The author(s) gratefully acknowledges the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana as well as the technical support provided by the Instituto de Física Corpuscular, IFIC (CSIC-UV).

